

**CERCETĂRI APLICÂND
SPECTROMETRIA
ȘI TERMOGRAFIA
PENTRU DETERMINAREA
CONCENTRAȚIILOR DE SO₂ EMISE
DE CENTRALELE FUNCȚIONÂND
PE CĂRBUNE**

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea "Politehnica" din Timișoara
în domeniul INGINERIE MECANICĂ
de către

Ing. Gheorghe Cătălin NIȘULESCU

Conducător științific:
Referenți științifici:

prof.univ.dr.ing.habil Ioana Ionel
prof.univ.dr. Sabina Ștefan
prof.univ.dr.ing. Petru Negrea
conf.univ.dr.ing. Marius Paulescu

Ziua susținerii tezei: 29.09.2012

Seriile Teze de doctorat ale UPT sunt:

- | | |
|------------------------|---|
| 1. Automatică | 7. Inginerie Electronică și Telecomunicații |
| 2. Chimie | 8. Inginerie Industrială |
| 3. Energetică | 9. Inginerie Mecanică |
| 4. Ingineria Chimică | 10. Știința Calculatoarelor |
| 5. Inginerie Civilă | 11. Știința și Ingineria Materialelor |
| 6. Inginerie Electrică | |

Universitatea „Politehnica” din Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2012

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității „Politehnica” din Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată în urma cercetărilor desfășurate pe parcursul activității mele ca doctorand, în cadrul Departamentului de Mașini Mecanice, Utilaje și Transporturi (MMUT) al Facultății de Mecanică a Universității "Politehnica" din Timișoara, pe perioada 2009-2012.

Activitatea s-a desfășurat în cadrul Laboratorului de Analize de Combustibili, Investigații Ecologice și Dispersia Noxelor din Centrul de Cercetare pentru Mașini și Echipamente Termice, Transporturi și Combaterea Poluării.

Calitatea aerului este foarte importantă pentru existența omenirii pe acest Pământ. Pentru a-și satisface cerințele vieții, omul participă la activități diverse, în diferite domenii de activitate: industrie, agricultură, transporturi, etc. În urma acestor activități apar fenomene nedorite cum ar fi poluarea aerului, apei, solului, care au efecte negative asupra sănătății omului, a degradării mediului înconjurător în general. În această lucrare, am acordat o atenție deosebită fenomenului de poluare cu dioxid de sulf.

Mă preocupă calitatea aerului și de aceea prin tema mea de doctorat îmi aduc aportul la îmbunătățirea și monitorizarea acesteia.

Mulțumesc doamnei prof.dr.ing.habil Ioana IONEL, care în calitate de conducător de doctorat, mi-a oferit sprijin moral și profesional, sprijin pe care l-am simțit în toate etapele parcurse până acum pentru a finaliza lucrarea.

Mulțumesc personalului din laborator cu care am colaborat foarte bine în elaborarea studiului experimental. Pe această cale îmi exprim recunoștința și îi asigur că oricând va fi nevoie, îi voi sprijini. Aduc mulțumire și distinsului colectiv al Departamentului MMUT care m-a sprijinit.

Mulțumiri deosebite se cuvin și doamnei prof.dr. Sabina ȘTEFAN, dar și prof.dr.ing. Petru NEGREA și conf.dr. Marius PAULESCU în calitate de referenți științifici.

Cu deosebit respect aduc mulțumiri celor care m-au sprijinit pe parcursul elaborării acestei lucrări: Conducerii Laboratorului de Analize de Combustibili, Investigații Ecologice și Dispersia Noxelor al Facultății de Mecanică Timișoara, pentru baza materială, informațională și experimentală, tuturor colaboratorilor direcți pentru sprijinul acordat, familiei și prietenilor mei.

Timișoara, septembrie 2012

ing. Gheorghe Cătălin Nișulescu

Nișulescu, Gheorghe Cătălin

Cercetări aplicând spectrometria și termografia pentru determinarea concentrațiilor de SO₂ emise de centralele funcționând pe cărbune

Teze de doctorat ale UPT, Seria 9, Nr. 121, Editura Politehnica, 2012, 248 pagini, 85 figuri, 14 tabele.

ISSN:1842-4937

ISBN:978-606-554-534-2

Cuvinte cheie:

poluare, camera UV, camera IR, dioxid de sulf, calitatea aerului, concentrație, emisie, centrale, cărbune

Rezumat,

Cercetările în UV și IR urmăresc conceperea unei metode originale (aplicând principii de spectrometrie și termografie) referitoare la monitorizarea emisiei de SO₂ printr-un procedeu neclasic, de la distanță. S-a realizat o aplicație numerică (program de calcul) care permite identificarea concentrațiilor acestei specii de poluant emise de surse antropice. Aplicația s-a conceput pe bază de experimentări comparative realizate la coșul de fum al unor centrale termo-electrice din România (în localitățile Halânga, Arad și Rovinari). S-au efectuat măsurători continue cu metoda standard SR EN 50379-1:2004 și simultan s-au cules date cu ajutorul camerelor UV și IR. Datele s-au prelucrat ulterior folosind programul Camera UVIR dezvoltat. Concluziile lucrării evidențiază: (1) importanța cunoașterii gradului de poluare cu SO₂ și necesitatea monitorizării pentru a putea proteja mediul înconjurător, posibilitatea de identificare a acestuia de la distanță;(2) limitele și avantajele metodei propuse;(3) condițiile ce trebuie respectate pentru implementarea metodei de determinare a emisiei de SO₂, scoțând în evidență posibilitatea de control de la distanță. Teza este interdisciplinară, subiectul dezvoltat bazându-se pe cunoștințe din domeniul mecanicii, tehnologiei informației, fizicii și chimiei.

CUPRINS

LISTA FIGURILOR	9
LISTA TABELELOR	12
NOTAȚII	13
1 INTRODUCERE ÎN TEMATICA LUCRĂRII	14
1.1 Generalități	14
1.2 Scopul general al lucrării	19
1.3 Rezumatul capitolului Introducere	19
2 STADIUL ACTUAL DE CUNOAȘTERE AL POLUĂRII ATMOSFERICE CU SO₂	20
2.1 Poluarea atmosferică	20
2.1.1 Atmosfera	20
2.1.2 Clasificarea poluanților aerului	22
2.1.2.1 Natura poluanților	23
2.1.2.2 Sursele poluanților	24
2.1.2.3 Răspândirea poluanților	26
2.1.3 Efectul de seră	27
2.1.4 Calitatea aerului	27
2.1.5 Efectele aerului poluat asupra sănătății populației	29
2.1.6 Poluarea și echilibrul biologic	30
2.2 Dioxidul de sulf	32
2.2.1 Proprietăți ale dioxidului de sulf	34
2.2.2 Proprietăți corozive	35
2.3 Efectele nocive ale poluanților	36
2.3.1 Efectele nocive ale oxizilor de sulf	37
2.3.2 Emisii anuale de dioxid de sulf (SO ₂)	38
2.4 Rezumatul capitolului Stadiul actual de cunoaștere al poluării atmosferice cu SO ₂	40
3 METODE DE MĂSURARE A POLUĂRII ATMOSFERICE CU SO₂	41
3.1 Determinarea concentrației procentuale a dioxidului de sulf în gaze	41
3.1.1 Legea Lambert – Beer	42
3.1.2 Aspecte teoretice legate de fenomenul de fluorescență.	42
3.1.3 Determinări prin analiza Gaussiană	43

6 Cuprins

3.2	Metode standard de măsurare a poluării atmosferice cu SO₂	46
3.2.1	Analizorul de gaze TESTO 300 XXL	46
3.3	Metode experimentale de măsurare a poluării atmosferice cu SO₂	53
3.3.1	Camera IR	55
3.3.1.1	Camera de formare a imaginilor infraroșii termice	59
3.3.1.2	Filtrare	60
3.3.1.3	Teoria radiației și modelarea	63
3.3.1.4	Detectarea IR la sol	64
3.3.1.5	Alarma pe bază de histogramă	66
3.3.1.6	Fitarea gaussiană	70
3.3.1.7	Analizele microfizice de cenușă	71
3.3.2	Camera UV	89
3.3.2.1	Introducere în folosirea camerelor	91
3.3.2.2	Testarea pe teren	94
3.3.2.3	Traectoria de lumină idealizată	97
3.3.2.4	Efectul de „diluare a luminii”	98
3.3.2.5	Dispersia multiplă	98
3.3.2.6	Modelarea transferului radiativ	99
3.3.2.7	Descrierea modelului	99
3.3.2.8	Sistem optic avansat	105
3.4	Rezumat al capitolului Metode de măsurare a poluării atmosferice cu SO₂	115
4	CAMPANII DE MĂSURĂTORI	117
4.1	Descrierea centralelor termice	117
4.1.1	CET Arad	117
4.1.2	Romag Termo Halânga	117
4.2	Măsurători cu camera UV	119
4.2.1	Prezentarea camerei UV și a filtrelor folosite	119
4.2.2	Metode de calibrare	120
4.2.3	Prezentarea tehnicilor de măsurare	121
4.2.4	Campanii de măsurători folosind camera UV	123
4.3	Măsurători cu camera IR	124
4.3.1	Prezentarea camerei IR și a filtrelor folosite	124
4.3.2	Metode de calibrare	125
4.3.3	Prezentarea tehnicilor de măsurare	127
4.3.4	Campanii de măsurători folosind camera IR	127
4.4	Comparații între metodele de măsurare a poluării atmosferice cu SO₂	129
4.4.1	Comparatie între metodele standard și cele experimentale	129

4.5	Model matematic de calcul al fluxului și a concentrației de SO₂	138
4.5.1	Metoda de procesare a datelor experimentale	138
4.6	Rezumat al capitolului Campanii de măsurători	141
5	APLICAȚIE DE PROCESARE A DATELOR EXPERIMENTALE - CAMERA UVIR V1.0	142
5.1	Prezentarea generală	142
5.2	Prezentarea meniului.	143
5.2.1	Meniul “Fișier”	143
5.2.2	Meniul “Prelucrare date UV”	144
5.2.3	Meniul “Prelucrare date IR”	144
5.2.4	Meniul “Help”	144
5.3	Prezentarea zonelor de afișare.	145
5.3.1	Imagine UV și zonele de afișare ce reprezintă date UV	145
5.3.2	Poziționarea punctului de intersecție axe	145
5.3.3	Poziționarea punctului de concentrație de SO ₂ nul	146
5.3.4	Poziționarea punctului de măsurare SO ₂ standard	147
5.3.5	Definirea zonei de măsurare	148
5.3.6	Dimensionare pixel	149
5.3.7	Alegere secțiune pentru calcul flux	150
5.3.8	Calcul viteza vânt	151
5.3.9	Opțiunea “eliminare cer senin”	151
5.3.10	Histograma UV	151
5.4	Rezumat al capitolului Aplicație de procesare a datelor experimentale - Camera UVIR V1.0	152
6	CAMERA UVIR - TEHNOLOGIA FOLOSITĂ	153
6.1	Prezentarea limbajului de programare JAVA.	153
6.2	Scurt istoric	153
6.3	Tehnologii folosite	153
6.3.1	Swing	153
6.3.1.1	Capabilitățile pachetului Swing.	154
6.3.1.2	O scurtă istorie Swing	155
6.3.1.3	Swing și AWT	155
6.3.1.4	Introducere în arhitectura Swing	156
6.3.1.5	Componente Swing folosite.	157
6.3.2	Spring	158
6.3.3	Libraria EAP FITS	159
6.3.4	IzPack	159

8	Cuprins	
6.4	Rezumat capitol Camera UVIR - tehnologia folosită	160
7	CONCLUZII. CONTRIBUȚII.	161
7.1	Concluzii generale	161
7.2	Contribuții practice	163
7.3	Contribuții privind potențiala extindere a domeniului de cercetare	163
	ANEXA 1 – PACHETELE ȘI CLASELE UTILIZATE PENTRU IMPLEMENTAREA APLICAȚIEI CAMERA UVIR	165
	BIBLIOGRAFIE	240

LISTA FIGURILOR

- Figura 2.1 Compoziția stratificată a atmosferei
- Figura 2.2 Reacțiile atmosferice ale SO₂ și emisiile istorice
- Figura 2.3 Schema instalației de extracție cu SO₂
- Figura 3.1 Aparat pentru analiza gazelor sulfuroase
- Figura 3.2 Distribuția gaussiană a norului de poluant
- Figura 3.3 Emisia punctiformă continuă
- Figura 3.4 Unitatea de analiză Testo
- Figura 3.5 Unitatea de analiză
- Figura 3.6 Interconectarea aparatelor pentru măsurători continue complexe
- Figura 3.7 Unitatea de control
- Figura 3.8 Sonda de prelevare a gazelor
- Figura 3.9 Calibrarea la 0, Testo 300
- Figura 3.10 Display, Testo 300
- Figura 3.11 Alegerea combustibilului
- Figura 3.12 Setarea parametrilor, Testo 300
- Figura 3.13 Camera IR în campania de măsurători de la Halânga
- Figura 3.14 Spectrul infrarosu al dioxidului de carbon
- Figura 3.15. Roata cu filtru Cyclops și filtre de interferență.
- Figura 3.16. Spectrele de transmisie atmosferică pentru un nor de cenușă și un nor de fază mixtă măsurate de la senzor prin satelit al AIRS
- Figura 3.17 Imagini rapide cu Cyclops.
- Figura 3.18. Geometria de măsurare pentru vizualizare cu o cameră termică a unui nor vulcanic „ideal” paralel cu planul. Camera se află în poziția Y și toate simbolurile sunt definite în text.
- Figura 3.19 Imaginea Cyclops de diferență de temperatură (11-12 μm) captată într-o locație (stâncă Suicide) din Saipan din Insulele Mariene de Nord.
- Figura 3.20 (a) Variația unghiului de elevație cu diferența de temperatură (11-12 μm) pentru cer senin determinată de modelarea de transfer radiativ. (b) Variația unghiului de elevație cu diferența de temperatură (11-12 μm) pentru cer senin determinată din datele de imagine Cyclops.
- Figura 3.21 Histograma imaginii 2D a diferenței de temperatură (11-12 μm) a datelor indicate. Linia de întrerupere a cenușii este indicată la diferența de temperatură 0K. Anvergura histogramei se datorează unei combinații de efecte de pixeli micști și elevația de vizualizare.
- Figura 3.22. Ilustrarea tehnicii gaussiene de prag pentru setarea alarmei de cenușă. Regiunea gri conține pixeli care se încadrează în acoperirea parțială dintre t-Gaussian; regiunea colorată în roșu indică pixelii care se numără ca pixeli afectați de cenușă.
- Figura 3.23. (a) Diferența de temperatură, raza particulelor, adâncimea optică pentru un unghi zenital de vizualizare de 25°. (b) La fel ca (a), dar pentru un zenit de vizualizare de 75°
- Figura 3.24. Histograma diferenței de temperatură 2D (11-12 μm) obținută la „izvoarele termale” din Rababa când atmosfera era încărcată cu cenușă.
- Figura 3.25. Schema prezintă principalele componente ale camerei de termoviziune în infraroșu „Cyclops”. Roata filtrului, care conține până la 5 filtre, este plasată în spatele obiectivului.
- Figura 3.26 Transmisia de SO₂ atmosferic în regiunea 1000-1500 cm⁻¹ (6,7 - 10 μm)

Figura 3.27. Funcția de răspuns a filtrului pentru canalul Cyclops de 8,6 μm și variația coeficientului de absorbție al SO_2 cu numărul de undă măsurat de către NIST.

Figura 3.28. Temperatura de luminozitate față de variația pe înălțime pentru filtrele de 12 (panoul din stânga), 8,6 (panoul central) și 8,6 - 12 (panoul din dreapta) μm . Liniile drepte sunt forme liniare mai puțin pătrate bazate pe datele de profil de deasupra coloanei, extrapolate prin și sub coloană. ΔT este diferența de temperatură de luminozitate între măsurătorile de 8,6 și 12 μm

Figura 3.29. Camera UV în campania de măsurători de la Halânga

Figura 3.30 Conversia penei imaginilor în concentrație de SO_2

Figura 3.31. Absorbanta relativă (filtru 307 nm / filtru 326 nm) față de concentrația de SO_2 .

Figura 3.32. Iluminarea relativă pentru imaginile camerei UV captate cu șapte diametre diferite de diafragmă (numere stop F). Axa y reprezintă raportul de iluminare relativă, unde 1,00 este valoarea cea mai strălucitoare de pixeli din imagine.

Figura 3.33. Relațiile de calibrare determinate de valorile absorbantei relative (imagini de diferențiere de la filtrul 307 nm și filtrul 326 nm) a celulelor de calibrare de la imaginile captate la timpi de expunere optimi pentru fiecare filtru (A) și timpi de expunere reprezentând un compromis între fiecare filtru (B).

Figura 3.34. Absorbanta față de relațiile concentrației lungimii traiectoriei și dependența lungimii expunerii (descrise aici ca înregistrarea DN maximă din imagine).

Figura 3.35 Schema geometriei de măsurare simulată.

Figura 3.36 Secțiunea transversală a absorbției de SO_2 (linia verde) .Lungimile de undă modelate sunt indicate cu săgeți

Figura 3.37 Transmitanța filtrelor A și B cu absorbție SO_2

Figura 3.38 Interfața programului Vulcamera

Figura 3.39 (a) Imaginea cantității coloanei de SO_2 a coloanei de erupție la ora 14:09:54. Scara din stânga este exprimată în ppm.m de SO_2 . (b) Distribuție orizontală a cantității coloanei de SO_2 la rândul 150 (albastru) și 220 (verde) de pixeli. (c) Cantitatea coloanei verticale de SO_2 la coloana 100 (albastru) și 300 (verde) de pixeli. (d) Rezultatul unei calibrări a camerei SO_2 cu ajutorul celulelor de calibrare.

Figura 3.40. (a) Intercorelările variațiilor temporale în profiluri integrate calculate între a 300-a și a 400-a coloană verticală. Decalajul de timp derivat a fost de 45,6 s pe o distanță de ~ 244 m (100 pixeli la 2,44 m lățime fiecare), rezultând o viteză a vântului de $5,4 \text{ ms}^{-1}$. (b) Variațiile temporale ale fluxului de SO_2 obținute din profilurile celei de-a 400-a coloană verticală.

Figura 3.41. a) Imagine vizibilă la Villarrica făcută cu o cameră digitală de la o distanță de 16,5 km

Figura 3.42 a) Imagine staționară vizibilă a coloanei Santiaguito captată de pe înregistrarea cu camera video.

Figura 3.43. a) Imagine staționară vizibilă a coloanei Fuego captată de pe înregistrarea cu camera video de la o distanță de 7.08 km.

Figura 4.1 CET Arad

Figura 4.2 Romag Termo Halânga

Figura 4.3 Camera, filtrele și roata de filtrare a camerei UV

Figura 4.4 Vedere din față și din spate a camerei UV

Figura 4.5 Controlul camerei UV

Figura 4. 6. Setarea camerei UV

- Figura 4.7 Măsurătoare UV Arad
 Figura 4.8 Măsurătoare UV Halânga
 Figura 4.9 Măsurătoare UV Rovinari
 Figura 4.10 Camera IR
 Figura 4.11 Calibrarea camerei cu ajutorul unui corp negru de referință
 Figura 4.12 Interfața programului Cyclops
 Figura 4.13. Măsurătoare IR Arad
 Figura 4.14 Măsurătoare IR Halânga
 Figura 4.15 Măsurătoare IR Rovinari
 Figura 4.16 Variația concentrației de SO₂ din data de 11.03.2011 cu metoda UV și TESTO
 Figura 4.17 Variația din data de 11.03.2011 a fluxului de SO₂
 Figura 4.18 Variații ale concentrației de SO₂ din data de 11.11.2011 cu metoda UV și TESTO
 Figura 4.19 Variația fluxului de SO₂ din data de 11.11.2011
 Figura 4.20 Variații ale concentrației de SO₂ din data de 10.02.2011 cu metoda UV și TESTO
 Figura 4.21 Variația fluxului de SO₂ din data de 10.02.2011
 Figura 4.22 Variații ale concentrației de SO₂ din data de 28.10.2011 cu metoda UV și TESTO
 Figura 4.23 Variația fluxului de SO₂ din data de 28.10.2011
 Figura 4.24 Variații ale concentrației de SO₂ cu metoda UV și TESTO în data de 29.03.2012
 Figura 4.25 Variația fluxului de SO₂ în data de 29.03.2012
 Figura 5.1 Interfața programului Camera UVIR
 Figura 5.2 Afișarea meniului Prelucrare date UV
 Figura 5.3. Afișarea meniului Prelucrare date IR
 Figura 5.4 Afișarea meniului Help
 Figura 5.5 Poziționare punct de intersecție axe
 Figura 5.6. Poziționarea punctului de concentrație de SO₂ nul
 Figura 5.7. Poziționarea punctului de măsurare SO₂ standard
 Figura 5.8 Zona de măsurare
 Figura 5.9. Dimensionare pixel
 Figura 5.10 Alegere secțiune pentru calcul flux
 Figura 5.11 Calcul viteza vânt
 Figura 6.1. Setul de componente Java
 Figura 6.2 Arhitectura MVC
 Figura 6.3 Componenta UI

LISTA TABELELOR

- Tabelul 1.1 Variația presiunii SO₂ cu temperatura
- Tabelul 2.1: Efectele nocive ale anhidridei sulfuroase (SO₂) în aer, la diferite concentrații.
- Tabelul 2.2. Emisii anuale de SO₂ (t/an)
- Tabelul 2.3. Emisii totale de SO₂ în județul Timis (t/2009)
- Tabelul.2.4 SO₂ - Concentrații medii orare la stații în 2009 μg/m³
- Tabelul 3.1. Numărul de canal, lungime de undă centrală, lățime de bandă, scopul și diferența de temperatură echivalentă zgomotului necesară (NEΔT) pentru Cyclops.
- Tabelul 3.2. NEΔT teoretice (mK) pentru cele cinci canale ale camerei de termoviziune în infraroșu și pentru patru temperaturi de scene diferite.
- Tabelul 3.3. Rezumatul tipurilor de erori și valorilor estimate de eroare.
- Tabelul 3.4. Prezentare generală a scenariilor investigate de transfer radiativ. În fiecare scenariu, doar un singur parametru a fost variat pentru a determina gradul de sensibilitate al transferului radiativ în ceea ce privește această cantitate
- Tabelul 4.1 Rezultatele măsurărilor realizate la Arad din data de 11.03.2011 cu camera UV și Testo
- Tabelul 4.2 Rezultatele măsurărilor realizate la Arad din data de 11.11.2011 cu camera UV și Testo
- Tabelul 4.3 Rezultatele măsurărilor realizate la Halânga din data de 10.02.2011 cu camera UV și TESTO
- Tabelul 4.4 Rezultatele măsurărilor realizate la Halânga din data de 28.10.2011 cu camera UV și TESTO
- Tabelul 4.5 Rezultatele măsurărilor realizate la Halânga din data de 29.03.2012 cu metoda UV și TESTO

NOTAȚII

AIRS = Sondaj electroacustic în Infraroșu Atmosferic
At=atmosferă
BLIR = fotodetector în IR Limitat de Fundal
CCD = circuit cu cuplaj de sarcină
DOAS = Spectroscopie Optică de Absorbție Diferențială
DN = Număr Digital
GOME = Global Ozone Experiment
EMEP(baza de date) = Cooperative Program for Monitoring and Evaluation of the Long Range Transmission of Air Pollutant in Europe
FTIR = Interferometru cu Transformantă Fourier
FOV = Câmp de vizualizare
MIR = Infraroșul Mediu
MODTRAN = model de transfer radiativ
NEDT = diferență de temperatură echivalentă de zgomot
PMT = tub fotomultiplicator
PUFF = nor de poluant
PNA= peroxiacetilnitrați
PM = particule materiale
OMI= Ozone Monitoring Instrument
SNR = Raport Semnal/Zgomot
SZA = Unghi zenital solar
SBDART= Model de Transfer Radiativ Atmosferic Discret Ordonat Santa Barbara
TIR = Infraroșul termic
TOMS = Totale Ozone Mapping Spectrometer

1 Introducere în tematica lucrării

1.1 Generalități

Poluarea atmosferică se definește ca prezența în atmosferă a compușilor ce cauzează un efect dăunător măsurabil asupra sănătății oamenilor, a florei și a faunei sau asupra mediului înconjurător în general. Aceste substanțe nu sunt prezente în mod normal sau sunt prezente la un nivel scăzut în compoziția normală a aerului.

Poluanții pot fi clasificați în antropici și naturali. Ambele tipuri pot fi poluanți primari care sunt eliberați în atmosferă în stare naturală sau poluanți secundari care sunt generați în atmosferă din precursori în urma reacțiilor fizico-chimice.

Poluanții naturali intră periodic sau sporadic în atmosferă și câteodată concentrația acestora poate fi atât de mare încât ei pot afecta grav clima[60].

Combustibilii fosili, biomasa pot provoca poluare datorită apariției de CO, CO₂, SO₂, particule solide și lichide suspendate în atmosferă. Traficul rutier contribuie în mare măsură la emisia acestor poluanți, în special în centrele urbane aglomerate și în zonele unde combustibilii cu un conținut mare de sulf sunt încă utilizați.

Efectul de seră este procesul prin care atmosfera captează o parte din energia solară, încălzind Pământul. Gazele de seră includ dioxidul de carbon rezultat din arderea combustibilului fosil și defrișări, metanul, eliberat de pe plantațiile de orez și locurile de depozitare a gunoaielor, precum și produse rezultate din arderi și diferiți compuși chimici industriali (acid azotos, freoni, hidrogen sulfurat). După vaporii de apă, dioxidul de carbon este principalul gaz de seră.

Emisiile de SO_x sunt calculate pentru simulările numerice actuale din două baze de date. Baza de date EMEP (Cooperative Program for Monitoring and Evaluation of the Long-range Transmission of Air Pollutant in Europe) conține emisii anuale de SO_x provenind din 11 sectoare de activități (centrale termice, trafic, agricultură, etc.). Baza de date, în afara Europei este EDGAR3.2 și conține un inventar al combustibililor, al activităților industriale chimice.[96]

Calitatea aerului este influențată de următorii factori principali: poluarea de fond (în acele zone unde nu se înregistrează un impact direct al surselor de poluare), poluarea de impact (prezentă în zonele afectate în mod direct de surse de poluare), frecvența și distribuția ploilor acide, a ozonului troposferic și a oxidanților fotochimici, precum și de gazele cu efect de seră, care afectează nivelul de absorbție a radiațiilor solare în atmosferă.

Conceptul de aer curat reprezintă o noțiune relativă, întrucât aer lipsit de impurități se găsește greu, chiar în cele mai izolate locuri de pe glob.

Aerul pur este fără miros, fără gust, fără culoare și transparent. El pare albastru dacă este privit în strat gros.

Proprietățile aerului sunt cele ale gazelor din care este format. Rolul principal îl are oxigenul care își păstrează în amestec toate proprietățile lui moderate de prezența azotului. Azotul are rolul important pentru vegetație, iar dioxidul de carbon la nutriția plantelor[9].

Aerul nu se impurifică prin creșterea proporțiilor gazelor care îl compun (cu excepția CO₂) ci prin adaosuri de alte substanțe. Vaporii de apă, chiar dacă rezultă din surse artificiale și concentrația lor poate crește până la 1-3% din volumul total al aerului, nu reprezintă un factor de poluare.

Poluantul este o substanță solidă, lichidă, gazoasă sau sub formă de energie (radiație electromagnetică, ionizantă, termică, fonică sau vibrații) care, introdusă în mediu, modifică echilibrul constituenților acestuia și a organismelor vii și aduce daune bunurilor materiale.

Gazele evacuate pe coșurile de fum ale instalațiilor de ardere, conțin de asemenea, importante cantități de oxizi de sulf, oxizi de azot, monoxid și dioxid de carbon, praf de cenușă, etc. Chiar și prin canalul de fum al coșurilor imobilelor de locuit sunt emise importante substanțe poluante. Amploarea și primejdia acestor emisii se explică, pe de o parte, prin aceea că emisiile de fum se fac la mică înălțime și deci dispersia lor în aer este slabă, iar pe de alta parte, prin conținutul relativ ridicat de hidrocarburi grele și gudroane, explicabil prin temperaturile mici la care se desfășoară procesele de ardere.

La toate acestea se adaugă noxele evacuate în aer odată cu gazele eșapate de autovehicule, și anume: monoxid și dioxid de carbon, oxizi de azot, aldehide, acid clorhidric și bromhidric, acid sulfuric și azotic, combinații ale plumbului, etc.

Dioxidul de sulf este unul dintre cele mai poluante gaze. Oxizii de sulf au un semnificativ impact asupra sănătății umane. Ei sunt responsabili pentru multe probleme respiratorii și cardiovasculare. Oxizii de sulf sunt precursori ai ploilor acide, cu efecte toxice asupra solului și a vegetației. Concentrația de dioxid de sulf poate influența culturile, precum și viața animalelor. SO₂ antropogenic este aproape exclusiv format în timpul combustiei combustibililor fosili cu conținut relevant de sulf și cel mai uzual este cărbunele.

Pentru motivele menționate, este foarte important să monitorizăm concentrația de SO₂ în atmosferă și în special controlul surselor în concordanță cu maximul valorilor admise la evacuare pentru calitatea aerului. Pentru aceasta se folosesc metode de măsurare a poluării atmosferice cu SO₂. În această lucrare sunt prezentate metodele standard și metodele experimentale (UV și IR).

Metoda UV se bazează pe măsurarea absorbției de SO₂ cu ajutorul ultravioletelor, în domeniul lungimilor de undă de până la 320 nm. Camerele UV au senzori de imagine ce detectează lumina în intervalul ultraviolet. Lumina UV este radiația electromagnetică cu lungime de undă care este mai scurtă decât a luminii vizibile dar mai lungă decât a razelor X. Specific, lumina UV acoperă un interval de la 10 nm la 400 nm și de la 3 eV la 124 eV [55].

Camerele UV lucrează ca și alte videocamere dar sunt sensibile la lumină în partea UV a spectrului electromagnetic. La camera UV este specific tipul senzorului imagine, rezoluția, formatul optic, pixelii orizontali, pixelii verticali, mărimea pixelului. CCD-ul camerei și sistemul optic sunt sensibile la condițiile de intensitate a luminii din mediu, deci la unghiul de înălțare a Soarelui și lumina dispersată de la obiectele din jur.

Camera UV s-a dovedit a fi practică și eficientă în cazul utilizării pe teren. Timpul de instalare este mai mic de zece minute, iar în mai puțin de o jumătate de oră se pot capta câteva zeci de imagini. În aplicațiile industriale camerele UV sunt folosite ca echipament de monitorizare a defectelor. În aplicațiile științifice camerele UV sunt folosite la imagistică.

Camerele digitale, sensibile la anumite regiuni ale spectrului de ultraviolete, au fost utilizate în ultimii ani pentru a măsura cantitatea de emisii de dioxid de sulf. Instrumentele se folosesc de absorbția selectivă de către razele UV a moleculelor de dioxid de sulf (SO₂) pentru a determina nivelul concentrației.

Principiul tehnicii camerei UV se bazează pe legea Lambert-Beer pentru a determina concentrația unei clase de absorbție.

16 Introducere în tematica lucrării - 1

Dacă lumina străbate o substanță sau soluția ei, radiațiile de lungimi de undă diferite sunt absorbite în diferite proporții. A determina spectrul de absorbție a unei substanțe înseamnă a măsura micșorarea intensității luminii ce o străbate. Micșorarea intensității luminii de o anumită lungime de undă este independentă de intensitatea inițială absolută și dependentă de grosimea stratului de substanță absorbantă străbătut.

Pentru determinarea concentrației de SO_2 în atmosferă se folosesc celule de calibrare, algoritmi de calcul ai absorbanței unde se aplică legea Lambert-Beer, precum și obținerea unor imagini cu norul de poluant. Din imaginile realizate cu camera UV s-au obținut grafice unde se pot reprezenta concentrații de SO_2 în diferite puncte. Comparând aceste concentrații putem deduce cum are loc dispersia poluantului în timp și în spațiu față de sursa de poluare.

Imaginile UV care încep ca rezoluții de valori de iluminare (reprezentate prin numere digitale, DN), sunt convertite în valori de absorbție folosind legea Lambert-Beer. Imaginile UV pot fi captate doar cu filtrul 307 nm, care este centrat în banda de absorbție a dioxidului de sulf a spectrului electromagnetic, sau cu ambele filtre 307 nm și 326 nm.

Camera poate fi folosită pentru a măsura viteza de transport a penei, o sursă majoră de incertitudine în aceste măsurători. Tehnica de teledetecție este ușor de folosit nefiind legată de coș și oferă mobilitate și accesibilitate pentru monitorizare externă.

O cameră în IR este un dispozitiv care detectează energia infraroșie (căldura) și o convertește într-un semnal electronic care este apoi procesat pentru a produce o imagine termică pe monitorul video.

Termografia reprezintă folosirea imaginii infraroșii și a camerei de măsurare, pentru a detecta și măsura energia termică emisă de un obiect. Energia infraroșie nu este vizibilă deoarece lungimea de undă este prea lungă pentru a fi detectată de ochiul uman. Este partea spectrului electromagnetic pe care noi o percepem sub formă de căldură. Imaginile IR permit să se vadă căldura și cum este ea distribuită.

Camera IR are lentile ca și o cameră cu lumină vizibilă, dar în acest caz lentilele focusează undele din energia infraroșie pe un șir de senzori infraroșii [103].

Mii de senzori ai șirului convertesc energia infraroșie în semnale electrice care sunt convertite ulterior în imagine. În imaginile infraroșii sunt folosite diferite culori pentru a reprezenta temperaturile. Se poate vedea orice temperatura după culoare, folosind scala culoare-temperatură arătată de obicei în dreapta imaginii.

Camera IR detectează și extrage concentrațiile de SO_2 din surse naturale și artificiale (ca și la UV). Ca instrumente de studiu sunt relativ ieftine. Camera de termoviziune Cyclops nerăcită la sol funcționează cu cinci canale spectrale la lungimi centrale de 8.6, 10, 11, 12 μm și un canal în bandă largă (7-14 μm).

Sistemul instrumentului, inclusiv procedura de calibrare și algoritmi de extragere se numește Cyclops. Rezultatele indică faptul că SO_2 se poate identifica și deosebi relativ ușor în coloane, dar concentrația traiectoriei se poate cuantifica mai dificil.

IR utilizează radiologia de termodetecție pasivă în infraroșu și formarea de imagini multispectrale. Sistemul se bazează pe măsurătorile diferențelor de temperatură spectrală fiind sensibil la valorile $\text{NE}\Delta\text{T}$ ale canalului. O limitare fundamentală este valoarea $\text{NE}\Delta\text{T}$ a canalelor filtrate spectral.

Camera poate fi folosită pentru formarea imaginilor gazelor toxice din accidente industriale sau scurgeri de gaze nedorite unde trebuie asigurată securitatea persoanelor.

Cel mai important aspect în designul Cyclops a fost alegerea numărului de filtre necesar, lungimile de undă centrale ale filtrului și lățimile de filtrare. Filtrele de interferență care operează în banda de lungimi de undă 7–14 μm sunt deja disponibile. S-au efectuat o serie de calcule pentru a găsi lungimi de undă și lățimi de filtrare pentru optimizarea semnalului.

S-au făcut măsurători cu camera IR la Rovinari, Arad, Halânga unde s-au obținut diferite imagini în funcție de radiația solară. În funcție de culorile care apar în aceste imagini (acestea fiind date brute, necalibrate) se pot interpreta valorile termice care s-ar putea extinde la calculul concentrației de SO_2 pe diferite porțiuni ale imaginilor. Deoarece nu avem posibilitatea să calibrăm camera IR în condițiile de măsurare (metoda este în curs de cercetare) se folosesc diferite modalități de deducții pentru NEAT.

Camera Cyclops funcționează dintr-o poziție fixă pe teren și vizualizează ținta de la distanță (până la 10 km). Locația actuală a camerei în raport cu ținta este selectabilă. Cu toate acestea, anumite considerații sunt necesare în cazul în care sistemul trebuie să funcționeze util.

Sistemul de cameră Cyclops a fost proiectat pentru a folosi până la cinci filtre spectrale, alese pentru a optimiza detectarea gazelor atmosferice specifice. Pentru a cuantifica concentrațiile traiectoriei de SO_2 de la sol, s-a ales un filtru în bandă de frecvență îngustă centrat la aproape 8,6 μm .

Gazoanalizorul TESTO este un echipament ce determină emisiile gazoase din gazele de ardere cum ar fi: SO_2 , CO, O_2 , NO și NO_x . Aparatul este format din trei subansambluri principale: unitatea de analiză, unitatea de control și sonda de prelevare a gazelor.

Unitatea de analiză este echipată cu două intrări diferite pentru senzorii de temperatură. Unitatea de analiză conține celulele de reacție, bateria de acumulatori, filtrele pentru reținerea impurităților solide din gazele de ardere și aer, decantorul pentru condens și conexiuni electro-pneumatice.

Semnalul electric este preluat și prelucrat în unitatea de control, unde sunt afișate concentrațiile emisiilor de gaze analizate.

Prin conectarea la PC, gazoanalizorul TESTO, poate fi folosit la măsurători de lungă durată (săptămâni) fiind controlat de un program special. Aparatul poate fi echipat cu mai multe sonde de prelevare a gazelor. Acestea diferă în funcție de caracteristicile gazelor prelevate (pot fi sonde încălzite sau nu).

După interconectarea celor trei elemente ale analizorului (unitate de control, unitate de analiză și sondă) se pornește aparatul fie prin conectarea acestuia la rețeaua electrică de 220 V fie la bateria proprie de acumulatori. După intrarea în regim a aparatului, acesta trebuie programat să achiziționeze și să afișeze datele care prezintă interes pentru utilizator. Metoda TESTO măsoară concentrația de SO_2 la baza coșului de evacuare a gazelor. Această metodă este precisă, măsoară direct concentrația de SO_2 dar este mai dificilă măsurătoarea propriu-zisă din motive tehnice. Aparatura este ușor de manevrat și se poate folosi în diferite condiții meteorologice.

Cunoscând efectele nocive ale poluanților în atmosferă, această lucrare consideră că este la îndemâna tuturor celor ce studiază fenomenul de poluare, prin oferirea documentației în măsurarea concentrației de SO_2 la diferiți poluanți. Programul integrat în această lucrare oferă posibilitatea de cercetare și de măsurare a poluării în diferite zone afectate.

Rezultatele măsurătorilor cantităților de SO_2 obținute experimental prin metoda UV și TESTO sunt apropiate, ceea ce înseamnă că metoda UV este eficientă

18 Introducere în tematica lucrării - 1

și este și ușor accesibilă (se pot face măsurători de la distanță de câțiva km de sursa de poluare).

Aceste instrumente au limitări de ordin spectrometric (la calibrare, la semnal-zgomot), dar sunt relativ ieftine și oferă informații legate de dinamica penelor și nivelul emisiilor la rezoluții temporale înalte.

Sursa de lumină pentru instrumentele pasive este lumina Soarelui dispersată în atmosferă. Avantajul acestei surse este faptul că este mereu prezentă pe parcursul zilei.

Coloana de poluant trebuie să fie între sursa de lumină (Soare) și instrumentul de măsurat și de aceea instrumentele trebuie să fie mobile și să se adapteze la diferite poziții față de Soare ceea ce reduce eficiența colectării de date.

Un criteriu tehnic al instrumentelor spectroscopice este lungimea de undă folosită. Cei mai mulți poluanți aeropurtați sunt activați în benzi ale lungimii de undă foarte scurte și bine definite. Astfel, cele mai uzuale lungimi de undă utilizate în măsurarea acestor specii sunt în spectrul *infraroșu și ultraviolet vizibil*.

Emisia continuă poate fi considerată din mai multe emisii instantanee, fiecare generând un puf. Pufurile se deplasează pe o traiectorie mai mult sau mai puțin șerpuită în funcție de fluctuația direcției vântului, se combină aditiv și formează pana de fum.

Pana trebuie să fie între sursa de lumină și instrumentul pentru măsurători. Când se efectuează măsurători de teledetecție a penelor vulcanice folosind lumina împrăștiată, calea de lumină idealizată a fost calea optică efectivă.

Metodele standard de monitorizare la distanță cu ultraviolete (UV) au utilizat corelația și spectrometria de absorbție optică diferențială. Imagistica dioxidului de sulf (SO_2) a fost realizată pe penele de coș, iar în ultimii ani, sisteme de imagistică UV au fost utilizate și în cercetările vulcanice. Aceste instrumente au limitări de ordin spectrometric, dar sunt relativ ieftine, și oferă informații fără precedent legate de dinamica penelor și nivelul emisiilor la rezoluții temporale înalte.

Măsurarea prin teledetecție pentru SO_2 a fost folosită cu succes în trecut pentru activitățile vulcanice și pentru monitorizarea gazelor de ardere. Acum această tehnică este propusă și pentru centrale termice. Tehnologia este bazată pe senzorul unei camere capabil să preia informații în porțiunea UV a spectrului, iar camera este poziționată înspre gazele de ardere.

Influența directă a poluării aerului asupra sănătății populației constă în modificările ce apar în organismul persoanelor expuse, ca urmare a contactului lor cu diferiți poluanți atmosferici. De cele mai multe ori acțiunea directă a poluării aerului este rezultanta interacțiunii mai multor poluanți prezenți concomitent în atmosferă și numai rareori acțiunea unui singur poluant.

Pentru fiecare din cele două tipuri de camere, s-a implementat câte o aplicație de procesare de date Camera UVIR, ce ajută la extragerea de date din fotografiile alese. Aceste aplicații sunt implementate ca parte a acestei lucrări și datele obținute sunt comparate cu cele obținute prin alte modalități de măsurare a poluării atmosferice.

Tehnicile de teledetecție au devenit un instrument folosit pe scară largă pentru măsurarea emisiilor vulcanice. Se pot distinge două tipuri de tehnici de teledetecție care utilizează spectroscopia de absorbție. Instrumentele de teledetecție active folosesc surse artificiale de lumină, în timp ce instrumentele pasive folosesc surse naturale, precum soarele, luna, lumina solară dispersată în atmosferă sau lava fierbinte în infraroșu.

Tehnica de teledetecție va deveni o opțiune pentru măsurătorile de poluare, nu numai pentru vulcani ci și pentru termocentrale.

1.2 Scopul general al lucrării

Scopul general al lucrării se referă la identificarea condițiilor optime de monitorizare cu metode optice de măsurare a calității aerului, care sunt necesare, în sensul oferirii de informații complementare și de confirmare a valorilor măsurate prin metodele standardizate. Lucrarea urmărește identificarea unor avantaje prin utilizarea metodelor optice UV și IR. Tematica prezentei lucrări a fost restrânsă doar la specia poluantă dioxid de sulf.

Se urmărește determinarea poluării atmosferice cu SO₂ ce se poate realiza prin aplicarea simultană a unei metode standardizate și a unei metode optice de monitorizare a calității aerului în zone industrializate cu diferite grade de poluare.

Se dorește desprinderea unor concluzii legate de interpretarea și corelarea rezultatelor obținute prin metode optice și prin metode standardizate.

Pentru atingerea acestor scopuri se va alege metodologia de lucru urmând etape legate de:

- Alegerea instrumentelor, calibrarea și întreținerea lor,
- Culegerea unor informații și stocarea lor
- Definitivarea unui sistem de prelucrare a datelor conform normativelor
- Interpretarea rezultatelor, în conformitate cu legislația

Instrumentele, principiul și metodele de lucru, modul de prelevare, memorare, prelucrare, până la finalitatea obținerii rezultatelor constituie paragrafe independente, din partea aplicativă a tezei.

În această lucrare se urmărește dezvoltarea unor capacități de cercetare într-un domeniu deficitar, cultivarea unor caracteristici manageriale a unor situații de gestionare a calității aerului, precum și informarea, prin diferite mijloace specifice mediului universitar de cercetare, a publicului și autorităților privind necesitatea, modalitatea și costurile pentru a monitoriza mediul.

1.3 Rezumatul capitolului Introducere

Datorită consecințelor sale la nivel global, poluarea atmosferică este o temă ce este studiată în întreaga lume. Poluanții pot fi clasificați în antropici și naturali și pot influența calitatea aerului. Dintre poluanții gazoși se remarcă acțiunea nocivă a dioxidului de sulf asupra florei, faunei, a mediului înconjurător. Își aduce aportul la efectul de seră, la ploile acide și din această cauză este important să monitorizăm concentrația de SO₂ în atmosferă. Pentru aceasta se folosesc metode de măsurare a poluării atmosferice: metode standard, metode experimentale UV și IR (în capitolele 3,4). S-au făcut comparații între metodele standard și metodele experimentale pentru a trage concluzii referitoare la eficiența lor. Pentru metoda UV și IR s-a folosit programul Camera UVIR cu ajutorul datelor măsurate pe teren în localitățile Halânga, Arad și Rovinari. Programul Camera UVIR s-a bazat pe limbajul de programare Java, unde s-a putut evalua concentrația de SO₂, respectiv fluxul de SO₂. Pe baza datelor procesate s-au tras concluzii originale cu privire la măsurătorile efectuate.

2 Stadiul actual de cunoaștere al poluării atmosferice cu SO₂

2.1 Poluarea atmosferică

2.1.1 Atmosfera

Atmosfera este învelișul gazos care se menține în jurul Pământului datorită gravitației. Stratul atmosferic atinge o altitudine de 10000 km și reprezintă 150% din raza medie a Pământului. Compoziția ei este uniformă până la 80 km înălțime, formând un strat care se numește homosferă. Heterosfera se găsește la peste 80 km și se caracterizează printr-o variație semnificativă în compoziția amestecului de gaz în substraturile ei. Compoziția procentuală a gazelor atmosferice raportată la întregul volum al atmosferei este:

- Azot: 78,084% (N₂)
- Oxigen: 20,946% (O₂)
- Argon: 0,934% (Ar, gaz nobil)
- Dioxid de carbon: 0,033% (CO₂)
- urme de gaze, cum ar fi unele gaze nobile (Helium, He; Neon, Ne; Xenon, Xe; Krypton, Kr), hidrogenul (H₂), metanul (CH₄) și protoxidul de azot (N₂O)[77].

Vaporii de apă care reprezintă 4% din volum determină umiditatea atmosferică și au un rol hotărâtor asupra vieții pe pământ, formând norii care produc precipitațiile. Vaporii de apă au rolul de a reflecta și absorbi o parte din radiația solară. O parte a energiei solare, după ce a fost reflectată de către suprafața pământului, este absorbită de vaporii de apă și astfel rămâne pe pământ. Atmosfera acționează ca un strat izolator termic datorită vaporilor de apă. Radiația solară este afectată de prezența unor cantități mari de praf atmosferic care provine din deșerturi, plaje, albiile de râuri, erupții vulcanice, oceane, poluare și incendii. Meteorii pot aduce în atmosferă cantități de praf, datorită dezintegrării prin frecare la intrarea în atmosferă. Frecarea generează o cantitate mare de căldură ce poate distruge particulele. Unul dintre efectele datorate acestui praf este cerul roșu, care poate fi văzut la răsărit și apus. Culoarea roșie a cerului se datorează lungimii de undă a luminii care rezultă după interacțiunea radiației cu particulele de praf[95,76].

Dincolo de stratul denumit stratosferă nu putem găsi în atmosferă particule cum ar fi vaporii, norii sau praful, ci numai componente gazoase.

Atmosfera se divide în straturi în funcție de temperatură care descrește cu altitudinea (cu aproximativ 6.5°C/km în straturile joase).

Troposfera este stratul unde valorile temperaturii se micșorează constant (de la +17°C la -52°C).

Gradientul de temperatură se schimbă brusc într-o zonă de tranziție numită tropopauză, cu o grosime de aproximativ 4 km (care crește în lunile călduroase). Stratosfera începe de la 10 km la Polul Nord și Polul Sud și crește la 17 km la ecuator, în altitudine. Gradientul de temperatură este inversat în stratosferă și temperatura se ridică constant la -3°C. La aproximativ 50 km temperatura atinge 0°C și apare stratopauza, o altă zonă de tranziție. La -90°C la circa 80 km altitudine de la suprafața Pământului apare mezosfera deasupra stratosferei. Mezopauza, zona de tranziție începe de la 90 km. Gradientul de temperatură se schimbă din nou și temperaturile cresc repede și dramatic cu altitudinea, ajungând

la aproximativ 1750°C în termosferă. Termosfera are 2 straturi: ionosfera și exosfera[5].

Ionosfera este stratul ce se întinde de la 80-90km la 400-500km altitudine și în acest strat au loc procesele de ionizare. Ionosfera reflectă radiația solară de o anumită lungime de undă și dă naștere fenomenului numit aurora. Ionizarea este produsă de radiațiile gama și X din radiația solară și se mai numesc radiații penetrante. Aceste raze sunt absorbite de moleculele de azot și hidrogen deoarece au energie mare[57].

În ionosferă fiecare moleculă cedează electroni și se transformă în ioni pozitivi, apare curent electric datorită apariției sarcinilor electrice pozitive și negative. Stratul exterior al ionosferei (peste 400-500 km) se numește exosfera și în acest strat densitatea gazelor atmosferice scade; scade și temperatura la aproximativ -270°C[11,76].

În figura 2.1 este reprezentată straturile atmosferice și variația temperaturii pe înălțime.

Presiunea atmosferică scade treptat cu altitudinea (între temperatură, presiune și densitate există o relație de dependență – legea gazelor). Atmosfera standard este definită pe baza temperaturii la sol, a gradientului vertical de temperatură în troposferă și de presiunea atmosferică.

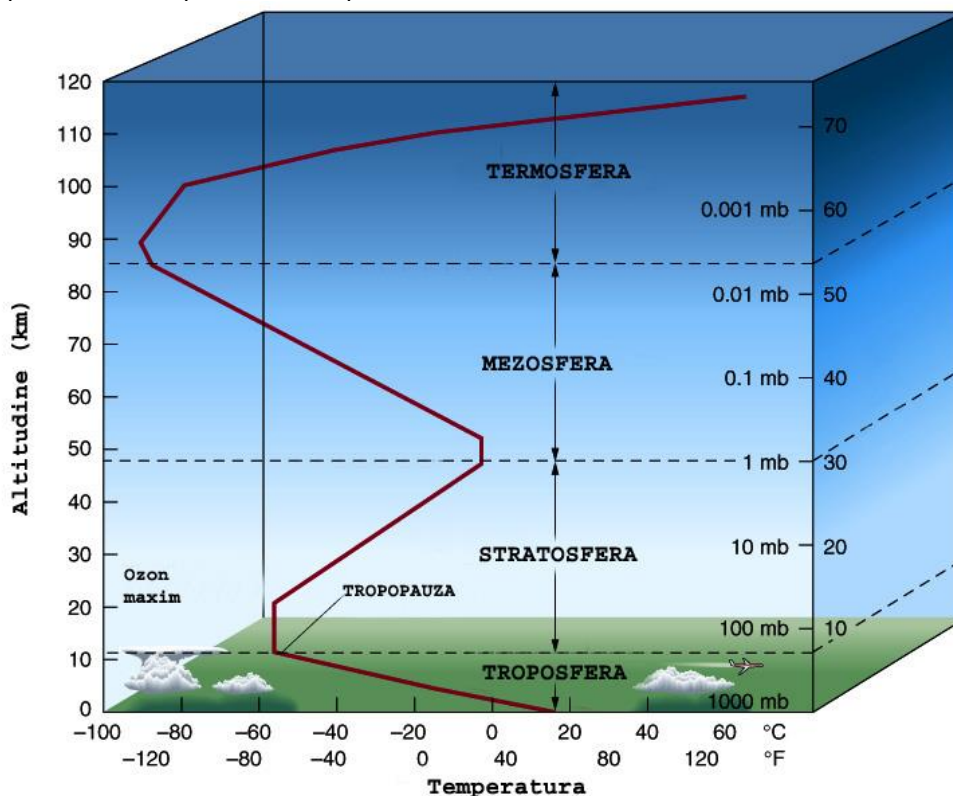


Figura 2.1 Compoziția stratificată a atmosferei[111]

Un alt strat al atmosferei important pentru protecția sănătății oamenilor și pentru mediul înconjurător este denumit ozonosferă sau strat de ozon. Datorită proprietăților sale fizice și chimice, ozonul are o funcție importantă în blocarea radiațiilor ultraviolete, iar în troposferă și în concentrații ridicate este periculos pentru ființele vii.

Radiațiile solare sunt alcătuite din două grupe principale de radiații: radiația termică (electromagnetică) și radiația corpusculară (prin plasma solară). Totalitatea radiațiilor electromagnetice emise de Soare, ordonate după lungime de undă se numește spectru solar.

Tipurile de radiații întâlnite sunt:

Raze γ (gama), (lungime de undă, $\lambda=0,0005-0,14\text{nm}$);

Raze X ($\lambda=0,01-0,10\text{nm}$);

Radiații ultraviolete, UV ($\lambda=10-380\text{ nm}$); pe baza efectului asupra sănătății umane sunt

subdivizate în UV-A ($\lambda=380-315\text{ nm}$), UV-B ($\lambda=315-280\text{ nm}$) și UV-C($\lambda=280-10\text{ nm}$);

Lumina vizibilă ($\lambda=380-780\text{ nm}$);

Radiații infraroșii ($\lambda=0,78-300\text{ }\mu\text{m}$).

Radiațiile prezentate sunt enumerate în ordinea energiei; cele cu lungimi de undă mai scurte au energia cea mai mare. Numai razele gama și X au suficientă energie pentru a fi considerate radiații penetrante.

Aerosolii din troposferă produc o difuzie a radiațiilor primite. Lungimile de undă „cele mai difuzate” sunt cele scurte, cu o culoare albastruie. Aceasta este explicația culorii albastre a cerului[95].

Anumite radiații solare primite sunt reflectate de atmosferă și trimise înapoi în spațiu; astfel acestea se pierd. O altă parte a radiației solare este difuzată spre suprafața Pământului. Dioxidul de carbon și vaporii de apă sunt capabili să absoarbă radiațiile infraroșii(IR). Acesta este așa numitul „efect de seră” care contribuie la creșterea temperaturii aerului.

Procentul absorbției radiației solare IR este variabil în concordanță cu starea vremii. Acesta variază de la aproximativ 10% în condiții de cer senin la aproximativ 30% în condiții de cer noros. Suprafața Pământului reflectă de asemenea o parte a radiației solare. Acest fenomen este denumit albedo, iar intensitatea acestuia depinde de natura suprafeței și de unghiul de incidență al radiației. Valoarea procentuală a albedo este calculată ca 100% înmulțit cu raportul radiației reflectată / radiație primită. Această valoare este în general mică la unghi maxim de incidență (radiații perpendiculare) și mare la unghiuri mici [11].

Poluanții în atmosferă se dispersează datorită vânturilor, turbulențelor și curenților și părăsesc atmosfera sub formă de depozite sau prin descompunere. Studiul difuziunii și reactivității poluanților atmosferici implică cunoașterea calității, cantității și a distribuției emisiilor, precum și a condițiilor atmosferice locale.

2.1.2 Clasificarea poluanților aerului

Poluanții pot fi clasificați în antropici (apăruți odată cu dezvoltarea primelor așezări urbane, sub influența factorului uman) și naturali. Ambele tipuri pot fi poluanți primari care sunt eliberați în atmosferă în stare naturală (ex. dioxid de sulf – SO₂, monoxid de azot - NO, etc.) sau poluanți secundari (ex. ozonul- O₃) care sunt generați în atmosferă din precursori în urma reacțiilor fizico-chimice[77].

Contaminanții aerului provin din combustia în motoarele autovehiculelor, activități industriale și casnice. Formarea și emisia contaminanților în mediul înconjurător sunt foarte variabile. Poluanții secundari (atât antropici cât și naturali) sunt generați din poluanți primari prin multiple modificări precum reacțiile ce implică lumina și oxigenul (fotooxidarea). Câteodată, poluanții secundari pot fi mai toxici și mai persistenți decât compușii originali[8].

Poluanții primari principali cum ar fi: monoxidul de carbon (CO), dioxidul de carbon (CO₂), oxizii de azot (NO_x, în majoritate sub formă de NO), praf și hidrocarburi nu provin din procedee de combustie întotdeauna. Carburanții conțin sulf, emisii de dioxid de sulf (SO₂) și contribuie la fenomenul de ploaie acidă. Fenomenul implică reacții ale dioxidului de sulf cu vaporii de apă din atmosferă, formând acidul sulfuric (H₂SO₄), care face ca ploaia să fie acidă. După ce ajung în atmosferă, poluanții primari suferă procese de difuzie, transport și depunere [4].

Poluanții secundari provoacă reacții chimice particulare în atmosferă (la oxizii de azot și hidrocarburi în prezența radiațiilor solare). Acești compuși sunt implicați în reacții în lanț, care în final oxidează NO la dioxid de azot (NO₂), produc O₃, oxidează hidrocarburi și, de asemenea, produc aldehide și peroxiacetilnitrați (PNA), acid azotic, nitrați și nitro-derivați. Se produc de asemenea sute de produși secundari minori[65,68]. Toți acești compuși, formează ceața fotochimică. Ceața este un contaminant foarte periculos al ecosistemului care afectează ciclul ozonului.

Ceața provoacă reducerea vizibilității, datorită poluării. După natura lor, particulele lichide sau solide suspendate în atmosferă pot fi clasificate astfel:

- Aerosol: particulele lichide sau solide suspendate, diametru (1 μm);
- Păclă: picături, diametru (2 μm);
- Degajări particule solide cu diametru < 1 μm, de obicei eliberate în timpul proceselor chimice sau metalurgice industriale;
- Fum: particule solide cu diametru < 2 μm, purtate de amestecuri de gaze;
- Praf: particule solide cu diametru între 0,2 și 500 μm;
- Nisip: particule solide cu diametru > 500 μm.

Particulele primare provin din sursele naturale, iar particulele secundare provin din cele primare prin una sau mai multe reacții fizice/chimice ce au loc în atmosferă. Particulele mai mari generează praful gros. Particulele care au dimensiunea mai mică de 10 μm (PM10 unde PM reprezintă particulele materiei) afectează sănătatea omului[66].

Aceste particule sunt periculoase, fiind particule inhalabile, care pot pătrunde în tractul respirator superior (de la nări la laringe). În jur de 60% din PM10, de fapt PM2,5 (praf fin) cu diametru < 2,5 μm sunt particule respirabile care pot ajunge la tractul inferior al aparatului respirator (de la trahee la alveola pulmonară)[10].

2.1.2.1 Natura poluanților

Poluarea antropică prezintă un impact major asupra mediului înconjurător și al sănătății la nivel global.

Praful și gazele emise de vulcani, focurile din păduri, descompunerea compușilor organici sunt poluanți naturali care intră periodic sau sporadic în atmosferă și câteodată concentrația acestora poate fi atât de mare încât ei pot afecta grav clima, în special la nivel local (cazul erupțiilor vulcanice)[3].

Poluarea cu poluanți particulari poate depăși cantitativ nivelul surselor antropice. Un exemplu este dioxidul de sulf (SO₂), care este produs de erupțiile

vulcanice, de descompunerea materiei organice și de focurile din păduri. A fost estimat că în 1983, din surse naturale au fost produse 80 - 290 milioane de tone de SO₂ pe an, comparativ cu 69 milioane de tone pe an produse din activitățile oamenilor. Alt exemplu este producerea de oxizi de azot (NO_x) care sunt produși în mod natural de vulcani, oceane, descompuneri organice, furtuni cu trăsnete (interacțiuni între lumină și gazele atmosferice)[34,67].

Particulele provenite din surse naturale (vulcani și furtuni de nisip) au în general un impact limitat asupra mediului înconjurător, a climei și a sănătății. Aceste surse generează particule „mari și grele” neînhalabile, care au o perioadă limitată din punct de vedere al timpului de reanvență în atmosferă. În general, acestor emisii li se atribuie foarte rar fenomene de poluare la scară amplă, deoarece acele evenimente au loc de obicei în zone geografice limitate. Există de asemenea excepții: erupția vulcanică de la Saint Helen's din mai 1980 a înrăutățit calitatea aerului din SUA și din toată zona de nord a Pacificului, în decurs de luni de zile, cu repercusiuni majore asupra climei la nivel mondial, cauzând reducerea valorii insolației, astfel afectând încălzirea maselor de aer și a suprafeței terestre. Vântul transportă praful în marile deșerturi și în zone îndepărtate, nisipul din Sahara poate fi transportat în țări relativ apropiate, ca Italia și Grecia, dar poate ajunge și în locuri mai îndepărtate ca Regatul Unit al Marii Britanii și al Irlandei de Nord dând naștere unor fenomene curioase precum „ploaia roșie” și „zăpada roșie”. Aceste fenomene nu sunt foarte frecvente, iar riscul asupra sănătății și a mediului înconjurător este limitat.

Compușii organici volatili (Volatile Organic Compounds, VOC) sunt produse naturale din plante. De exemplu, izoprenul este un VOC obișnuit, produs de plante, care are diferite funcții, de obicei având rol în protecția plantei. Anumiți cercetători consideră că izoprenul are un impact semnificativ în creșterea alergiilor și a astmului, superior altor compuși de origine antropică. Mai mult decât atât, plantele produc polen, care e parte a particulelor atmosferice. Polenul este foarte cunoscut pentru proprietățile lui alergice [27].

Aerul este un amestec gazos care conține azot, oxigen, dioxid de carbon, gaze rare, urme de praf, vapori de apă. Azotul, carbonul, oxigenul formează un circuit în natură. Compușii gazoși din aer sunt într-un echilibru dinamic și se păstrează concentrația lor în limite constante [12].

2.1.2.2 Sursele poluanților

Populația globului a evoluat în timp, din punct de vedere demografic și s-au diversificat activitățile industriale și casnice care au condus la apariția fenomenului de poluare, fenomen ce s-a accentuat în timp. Explozia demografică și concentrarea populației în centrele urbane, împreună cu dezvoltarea industrială, au dus la o poluare majoră. În trecut, nepăsarea față de mediul înconjurător a fost în general cauzată de lipsa de cunoaștere științifică și tehnologică. În zilele noastre, țările puternic industrializate sau dezvoltate, importante din punct de vedere economic și politic, nu acordă întotdeauna prioritatea necesară protejării mediului înconjurător.

Activitățile umane moderne, implică producerea poluanților care, direct sau indirect, implică procese de combustie pentru: prepararea mâncării, încălzirea locuințelor, utilizarea electricității, utilizarea autovehiculelor, etc.[13].

Sursele fixe pot provoca poluarea aerului și acestea pot fi considerate fabricile și uzinele mici, sistemele de încălzire a locuințelor. Sursele mobile (autovehiculele) pot provoca de asemenea poluarea aerului. Multe surse de

poluare sunt legate în general de producerea și consumul de energie, în special când sunt utilizați combustibilii fosili și derivații lor.

Combustibilii fosili pot provoca poluare datorită apariției de CO, CO₂, SO₂, particule solide și lichide suspendate în atmosferă. Traficul rutier contribuie în mare măsură la emisia acestor poluanți, în special în centrele urbane aglomerate și în zonele unde combustibilii cu un conținut mare de sulf sunt încă utilizați. În locurile în care se folosește benzină cu plumb, se consideră că datorită traficului, în atmosferă se regăsesc 80-90% din acest metal, care este toxic la concentrații mari (produce otrăvirea cu plumb denumită și saturnism). Mai mult decât atât, motoarele cu combustie internă reprezintă sursa majoră de monoxid de carbon (CO) în atmosferă[26].

Principalele surse de poluare ale aerului atmosferic sunt următoarele: industria energetică, industria siderurgică, industria chimică, industria materialelor de construcții, transporturile. Poluarea produsă din industria energetică se referă la introducerea în atmosferă a gazelor naturale din sondele de extracție a petrolului sau a gazelor naturale, a gazelor din prelucrarea cărbunilor. Cărbunii sunt folosiți pentru obținerea energiei electrice la termocentrale. Principalii poluanți emiși de centralele termoelectrice sunt: praf, oxizi de sulf, oxizi de azot, funingine, hidrocarburi, gudroane [61].

Industria siderurgică realizează poluare substanțială în urma proceselor de obținere a fontei, a oțelului, a laminatelor. Se emană în atmosferă oxizi de fier, dioxid de sulf, praf, etc.

Industria lianților(fabricile de ciment) aruncă în atmosferă mari cantități de praf care apoi se depun. Praful rezultat din industria cimentului este împrăștiat până la distanța de peste 3 kilometri față de sursă, concentrația acestuia în apropierea surselor variind între 500 și 2000 t/km²/an. Fumul constituie partea invizibilă a substanțelor ce se elimină prin coșurile întreprinderilor industriale și este constituit din vapori de apă, gaze, produși incomplet arși(cărbune, hidrocarburi, gudroane, etc.). Fumul are o culoare albicioasă dacă arderea este completă. Culoarea neagră indică o ardere incompletă din lipsă de aer și datorită prezenței în cantitate mare a cărbunelui și a funinginei. Culoarea fumului uneori poate fi roșcată, cenușie sau brună, după cum cărbunele conține fier, aluminiu sau mangan.

Industria chimică poluează aerul prin diferite ramuri ale sale: industria produselor clorosodice, industria îngrășămintelor chimice, industria acidului sulfuric, industria acidului azotic, industria amoniacului, industria maselor plastice.

Fabricile de îngrășăminte chimice emit importante cantități de oxizi de sulf, oxizi de azot, acizi ai fosforului, fabricile de sodă emit clor, fabricile de aluminiu emit fluor, fabricile de ciment emană praf, rafinăriile emană hidrocarburi și dioxid de sulf, combinatele de metalurgie neferoasă emit oxizi de sulf, compuși ai metalelor grele, oxizi de azot etc.

Un alt gaz periculos produs prin combustia în motoarele cu ardere internă este monoxidul de azot (NO). Acesta intervine nefavorabil în ciclul ozonului, formând acid azotic gazos prin reacție cu oxigenul și apa. Acidul azotic determină un pH scăzut necesar pentru eliberarea clorului și bromului din poluanți, iar acești halogeni distrug stratul de ozon prin reacții în lanț .

Atmosfera primește mulți poluanți ca produși secundari din procesele industriale. Acești poluanți variază din punct de vedere cantitativ și calitativ. În general, ei au un impact minor la nivel global, deoarece sunt eliberați în atmosferă în concentrații semnificativ mai mici. Unii dintre acești poluanți se pot acumula la

nivel local, astfel crescând efectul lor nociv. Deoarece acești poluanți sunt de obicei toxici, emisia lor duce la creșterea riscului de îmbolnăvire.

Impactul poluanților asupra sănătății depinde de localizarea emisiei și de difuzia ei. Sursele fixe mari, de obicei situate departe de centrele urbane, împrăștie poluanții la înălțimi mari prin coșurile de evacuare.

Centralele termice și autovehiculele eliberează poluanții la nivelul solului în zone aglomerate. Ca o consecință, sursele de poluare mici, atât fixe cât și mobile au un impact negativ major asupra calității aerului în zonele urbane[64].

2.1.2.3 Răspândirea poluanților

Concentrația poluanților atmosferici depinde de mai mulți factori:

- Cantitatea de poluanți din emisii;
- Numărul și abundența surselor de poluare;
- Distanța față de sursă;
- Transformările fizico-chimice ale poluanților în atmosferă;
- Viteza de eliminare din atmosferă;
- Caracteristicile geomorfologice ale zonei poluate;
- Condițiile locale și generale ale vremii.

Precipitațiile atmosferice (furtunile), sunt de asemenea importante, deoarece acestea îndepărtează poluanții din atmosferă și îi transportă la sol. Zonele industriale și urbane sunt cele mai expuse la fenomenele de poluare. Aceste fenomene de poluare sunt întâlnite mai des în zonele industriale în care caracteristicile geomorfologice împiedică circulația aerului, de exemplu, văi înconjurată de munți. Alt aspect important privind poluarea este gradul de insolație și temperatura. În anumite condiții, acești factori pot duce la formarea ceții fotochimice care poate influența multe variabile climatice.

Concentrațiile mici de poluanți sunt avantajate de vânt și de condițiile instabile din straturile inferioare ale atmosferei. Concentrațiile mari de poluanți sunt susținute de ceața densă și rezistentă, absența vântului sau de inversiunea termică[25].

Inversiunea termică previne amestecarea maselor de aer. Acest fenomen are loc datorită încălzirii maselor de aer, cu densitate mai mică, ce se ridică și le înlocuiesc pe cele reci, iar acestea la rândul lor coboară și se încălzesc. Aerul cald este cel mai poluat datorită apropierii de sol unde se găsesc sursele de poluare. Acest mecanism produce o scădere a concentrației poluanților în coloanele de aer datorată amestecării pe verticală. Straturile calde se pot forma la altitudini mari, chiar sute de metri datorită inversiunii termice de tip radiativ sau a unor condiții meteorologice particulare. În aceste cazuri, straturile reci nu se pot ridica și sunt blocate la nivelul solului unde sunt acumulați poluanții. Aceste inversii au loc în timpul nopților senine, imediat după apusul de soare. Solul se răcește repede, determinând astfel o răcire rapidă a aerului. Astfel de inversii se sfârșesc de obicei dimineața, când pământul se încălzește, dar dacă anumite condiții meteorologice împiedică această încălzire, poluanții se pot acumula în decurs de câteva zile, ajungând la concentrații ridicate.

Poluarea poate avea efecte negative, grave și rapide (acute) sau durează mult și au efect cumulativ (cronice). Poluanții pot acționa la nivel local, distrugând pădurile, sau la nivel global, afectând biosfera și clima. Ploaia acidă poate afecta zone întinse, având efect dăunător asupra florei, compoziției solului și poate influența chiar migrarea unor poluanți (ex. metalele din sol). Radiațiile solare, temperatura, vântul și inversiunea termică au o influență semnificativă asupra difuziei și transformării poluanților, relația fiind reversibilă.

2.1.3 Efectul de seră

Efectul de seră este procesul prin care atmosfera captează o parte din energia solară, încălzind Pământul. În domeniul climatic experții sunt de părere că o creștere a nivelului gazelor cu efect de seră, creștere provocată de activitățile umane, accentuează în mod artificial efectul de seră, ducând la creșterea temperaturilor globale și dereglând clima în mod dezastruos. Gazele de seră includ dioxidul de carbon rezultat din arderea combustibilului fosil și defrișări, metanul, eliberat de pe plantațiile de orez și locurile de depozitare a gunoaielor, precum și produse rezultate din arderi și diferiți compuși chimici industriali (acid azotos, freoni, hidrogen sulfurat). După vaporii de apă, dioxidul de carbon este principalul gaz de seră. Carbonul este stocat sub Pământ departe de biosferă în combustibili fosili, însă ciclul organic al carbonului face transferul carbonului în natură. Fără intervenția umană, transferul între rezervoarele de carbon este menținut în echilibru (plantele absorb carbonul în timp ce se dezvoltă, dar îl elimină când mor). Când oamenii taie arbori sau ard combustibili fosili, ei elimină în atmosferă cantități suplimentare de carbon, accentuând astfel efectul de seră. Efectul de încălzire a atmosferei este în principal datorat celor două componente principale ale vieții: apa (ca vaporii de apă) și carbonul (ca dioxid de carbon, CO_2) [77].

Ideea că gazele atmosferice pot reține căldura, a fost pentru prima dată exprimată de Joseph Fourier în jurul anului 1800. El compară atmosfera planetei cu sticla unei sere: ambele lasă razele solare să treacă, apoi captează o parte a căldurii care nu se poate reflecta [16].

Fourier nu cunoștea principiile fizice și chimice care stau la baza acestui fenomen, dar l-a denumit "efect de seră". Circa șaiszeci de ani mai târziu, în Anglia, John Tyndall a demonstrat în urma unor experimente, că apa (vaporii) și dioxidul de carbon absorb căldura primită sub formă de radiații (IR). El a calculat, de asemenea, creșterea de temperatură datorată prezenței acestor molecule în atmosferă.

Astăzi, avem dovezi incontestabile că:

- dioxidul de carbon absoarbe căldura sub forma radiațiilor IR
- concentrația acestui gaz în aer a crescut în ultimii 150 ani
- temperatura medie a pământului a variat în timpul epocilor anterioare.

2.1.4 Calitatea aerului

Calitatea aerului este o importantă și globală problemă a tuturor națiilor. Calitatea aerului este influențată de următorii factori principali: poluare de fond (în acele zone unde nu se înregistrează un impact direct al surselor de poluare), poluare de impact (prezentă în zonele afectate în mod direct de surse de poluare), frecvența și distribuția ploilor acide, a ozonului troposferic și a oxidanților fotochimici, precum și de gazele cu efect de seră, care afectează nivelul de absorbție a radiațiilor solare în atmosferă.

Din punct de vedere al chimiei coloidale, aerul atmosferic este un sistem de particule coloidale, dispersate în gaz, în care componentii solizi și lichizi se găsesc în amestec cu gazele. În acest amestec, moleculele gazelor sunt în continuu schimb datorită interacțiunii diferitelor surse și consumatori[49,59].

Timpul mediu de staționare reprezintă timpul mediu petrecut în atmosferă de moleculele unui gaz. În funcție de timpul de staționare, Junge propune următoarea clasificare a gazelor din atmosferă:

- Gaze cvasipermanente cu $\tau > 10^3$ ani
- Gaze variabile cu τ de ordinul anilor
- Gaze înalt variabile cu un τ de câteva zile sau săptămâni (în general mai puțin de un an)

$\tau = M/F = M/C$, la echilibru pentru un ciclu atmosferic al unui material

M - cantitatea de component din rezervorul atmosferic

F și C - vitezele de formare (la surse) sau de îndepărtare (la consumator)

Gazele principale (O₂, N₂; Ar) au compoziția practic constantă în straturile inferioare ale atmosferei, motiv de bază al menținerii vieții și dezvoltării viețuitoarelor.

Gazele nobile cu excepția heliului sunt strict permanente, neavând nici o sursă sau consumator[6].

Azotul este principalul component al aerului și de aceea masa moleculară a aerului (28,973) este foarte apropiată de a azotului. Această compoziție și masă moleculară sunt constante într-un strat de 80-90 km al atmosferei numit homosferă. Deasupra acestui strat se găsește heterosfera în care masa moleculară este o funcție de înălțime.

Conceptul de aer curat reprezintă o noțiune relativă, întrucât aer lipsit de impurități se găsește greu, chiar și în cele mai izolate locuri de pe glob.

Aerul pur este fără miros, fără gust, fără culoare și transparent. El pare albastru dacă este privit în strat gros.

Aerul uscat este slab conducător de căldură și de electricitate. El poate fi lichefiat, prin răcire la o temperatură foarte joasă. Datorită temperaturii de fierbere foarte joase a componentelor, aerul lichid dacă este lăsat descoperit se evaporă violent. Ținute în aer lichid, substanțele își schimbă brusc proprietățile. De exemplu frunzele sau florile plantelor cufundate în aer lichid devin casante ca și sticla; mercurul se solidifică, plumbul capătă la lovire un sunet clar și metalic; obiectele de cauciuc devin fragile încât se sparg prin lovire[79].

Proprietățile aerului sunt cele ale gazelor din care este format. Rolul principal îl are oxigenul care își păstrează în amestec toate proprietățile lui moderate de prezența azotului. Azotul are rolul important pentru vegetație, iar dioxidul de carbon la nutriția plantelor.

Aerul nu se impurifică prin creșterea proporțiilor gazelor care îl compun (cu excepția CO₂) ci prin adaosuri de alte substanțe. Vaporii de apă, chiar dacă rezultă din surse artificiale și concentrația lor poate crește până la 1-3% din volumul total al aerului, nu reprezintă un factor de poluare[56].

În atmosferă, radiația solară este supusă fenomenelor de absorbție, difuzie și reflexie. Din radiația solară, la suprafața Pământului ajunge o cantitate de 51%. Pierderile se datorează absorbției vaporilor de apă, particulelor de praf și stratului de ozon din aer (16%), absorbției și reflexiei la nivelul formațiunilor noroase (23%), prin dispersie și curbarea direcției (6%) și reflexiei la contactul cu suprafața terestră (4%).

2.1.5 Efectele aerului poluat asupra sănătății populației

În cadrul interrelațiilor dintre om și mediul său ambiant, acesta din urmă exercită asupra omului influențe multiple, dintre care una foarte importantă este acțiunea asupra sănătății. Din acest punct de vedere, se știe că mediul conține factori care au o acțiune favorabilă asupra sănătății cunoscut sub numele de factori sanogeni. Mediul conține și factori care au o acțiune neavorabilă asupra sănătății, care determină îmbunătățirea sau pierdere acesteia, numiți factori patogeni.

Acțiunea mediului poluant asupra organismului este foarte variată și complexă. Ea poate merge de la simple incomodități în activitatea omului până la perturbări puternice ale stării de sănătate. Efectele acute au fost primele asupra cărora s-au făcut observații și cercetări privind influența poluării mediului asupra sănătății populației. Ele se datorează unor concentrații deosebit de mari ale poluanților din mediu care au repercursiuni puternice și brutale asupra organismului uman: marile epidemii de la Hamburg și Petersburg, de la sfârșitul secolului trecut, datorate poluării apei cu vibrii, iar mai recent, Londra, 1952, când datorită poluării cu pulberi și dioxid de sulf s-au înregistrat 4000 decese. Aceste episoade acute au avut meritul de a fi atras atenția specialiștilor asupra sănătății populației expuse, declansând primele cercetări sistematice în domeniul relației sănătate-poluare.

Efectele cronice reprezintă formele de manifestare cele mai frecvente ale acțiunii poluării mediului asupra sănătății umane. În mod obișnuit, diverșii poluanți existenți în mediu nu ating nivele foarte ridicate pentru a produce efecte acute, dar prezența lor continuă, chiar în concentrații mai scăzute nu este lipsită de efecte nedorite.

Influența directă a poluării aerului asupra sănătății populației constă în modificările ce apar în organismul persoanelor expuse, ca urmare a contactului lor cu diferiți poluanți atmosferici. De cele mai multe ori acțiunea directă a poluării aerului este rezultanta interacțiunii mai multor poluanți prezenți concomitent în atmosferă și numai rareori acțiunea unui singur poluant[81].

Cei mai reprezentativi poluanți din atmosferă sunt:

Poluanți cu acțiune iritantă;

Pulberi (sedimentabile sau în suspensie) ce acționează la nivelul căilor respiratorii care, deși prezintă mecanisme de protecție față de efectele nocive ale poluanților (mucus, epitelii ciliate, etc.) pot fi afectate de inflamații, rinite, faringite, laringite, bronșite sau alveolite. Dacă acțiunea poluantului este de lungă durată pot apărea afecțiuni cronice ca bronho-pneumopatia cronică nespecifică.

Oxizii sulfului, ce apar în aer prin arderea combustibililor fosili sau din diferite procese industriale, au un grad mare de solubilitate, produc iritații ale căilor respiratorii ce se traduc prin salivăție, expectorație, spasme și dificultăți în respirație care permanentizate duc la apariția bronșitei cronice [80].

Oxizii azotului rezultă la fel ca cei ai sulfului, produc la nivelul căilor respiratorii blocarea mișcărilor cililor epiteliiilor bronhice și traheale. La nivel sangvin, se combină cu hemoglobina rezultând methemoglobina care împiedică transportul gazelor respiratorii (oxigenului) către țesuturi.

Substanțe oxidante generate din acțiunea radiațiilor ultraviolete asupra unor produși de ardere ai hidrocarburilor, au efect iritant pentru căile respiratorii, ceea ce facilitează suprainfecțiile cu germeni oportuniști.

Poluanți cu acțiune asfixiantă: oxidul de carbon, rezultat din arderi incomplete, se combină cu hemoglobina dând carboxihemoglobina, generând fenomene de lipsa de oxigen cu consecințe dintre cele mai grave asupra respirației diferitelor țesuturi și celule, ce se manifestă clinic prin dureri de cap, amețeli, somnolență, greață, aritmii, etc.

Poluanți cu acțiune toxică sistemică: plumbul, eliminat sub formă de vapori care se condensează relativ repede, poate pătrunde în organismul uman atât pe cale respiratorie (mai periculoasă pentru că ajunge direct în sânge) cât și pe cale digestivă (ficatul are o mare putere de detoxifiere a organismului). Acțiunea nocivă a plumbului se exercită la nivelul sângelui determinând apariția de anemii, și la nivelul sistemului nervos, provocând rămânerea în urmă a dezvoltării la copii.

Poluanți cu acțiune fibrozantă: pulberile mai ales cele cu densitate mare, persistă în plămân, determinând o scădere a elasticității pulmonare ca și o reacție la corp străin, cu formare de țesut nou în jur, ce stă la baza apariției fibrozei

Poluanți cu acțiune cancerigenă: hidrocarburile policiclice aromatice sunt poluanți organici ce rezultă din arderea incompletă a combustibililor solizi și lichizi. Se concentrează în organism în condițiile unei expuneri prelungite.

Poluanți cu acțiune alergizantă: pulberile minerale sau organice ca și gazele (oxizi de azot, sulf, carbon) sau substanțele volatile din insecticide, detergenți, mase plastice, medicamente produc rinite acute, traheite, astm sau manifestări oculare (conjunctivite și blefarite) sau cutanate (eczeme, urticarii, etc).

Poluanții cu acțiune infectantă sunt reprezentați de diverși germeni patogeni din atmosferă. Deși majoritatea germeilor ce cauzează boli infecțioase cu poarta de intrare respiratorie ca: difteria, scarlatina, tusea convulsivă, rujeola, rubeola, varicela, variola, gripa, guturaiul, etc, au o rezistență scăzută în aer datorită unor factori ca: uscăciunea, temperatura scăzută, radiațiile ultraviolete, contaminaera produsă prin aer este responsabilă pentru un număr mare de boli.

2.1.6 Poluarea și echilibrul biologic

Poluarea aerului reprezintă orice modificare a compoziției sale ideale și orice emisie care duce la depășirea unui prag de calitate, depășire considerată periculoasă pentru om. În mod natural poluanții nu fac parte din compoziția aerului (sau se găsesc în proporție mică).

Poluarea aerului este influențată de modul de stratificare a aerului, de inversiunile termice. Astfel există factori de amplificare a poluării și factori care diminuează poluarea. Deasupra orașelor se formează frecvent așa-numitele umbrele de gaze acumulate la o anumită altitudine ca o consecință a circulației rutiere, poluării industriale [83].

Ecologia studiază raporturile pe care organismele le întrețin cu mediul care le înconjoară. Termenul de *ecologie* provine din limba greacă și ar desemna în accepțiunea sugerată de etimologia sa studiul habitatelor, al locuințelor de trai. Încă de la prima sa utilizare, el desemnează, studiul relațiilor dintre viețuitoare [47].

Mediul înconjurător reprezintă totalitatea factorilor naturali și a celor artificiali creați prin acțiuni umane, care, în strânsă interacțiune, influențează echilibrul ecologic, determinând condițiile de viață pentru om și dezvoltarea societății. În Legea Protecției Mediului [39] pentru România se indică: "*mediul* este ansamblul de condiții și elemente naturale ale Terrei: aerul, apa, solul și subsolul, toate straturile atmosferice, toate materiile organice și anorganice, precum și ființele

vii, sistemele naturale în interacțiune cuprinzând elementele enumerate anterior, inclusiv valorile materiale și spirituale".

Mediul are deopotrivă un sens topografic (locul sau biotopul) și, mai ales, un sens funcțional, pe care-l imprimă factorii energetici (căldura, radiațiile), trofici (apa și resursele nutritive) și mecanici (ploaia, vântul).

Ecosistemul este unitatea de bază a ecologiei. Acesta reprezintă ansamblul de acțiuni reciproce ce se petrec între numeroase ființe și mediul lor sau un angrenaj de relații ce se stabilesc între lumea vegetală și animală, sub influența factorilor fizici oferiți de mediu. În sens mai larg prin ecosistem se înțelege complexul dinamic de comunități de plante, animale și microorganisme și mediul lor lipsit de viață, care interacționează într-o unitate funcțională [50].

Structura ecosistemului este caracterizată de patru mari componente, și anume:

- componenta abiotică, reprezentată de resursele energetice și trofice ale mediului;
- componenta producătoare de materie organică;
- componenta consumatoare de materie organică (animalele vegetariene și carnivore);
- componenta formată din microflora și microfauna solului, care descompune și mineralizează deșeurile organice.

Ecosistemele sunt guvernate de mecanisme cu o stabilitate labilă, capabile să reziste prin autoreglare, atât la schimbările minore ale mediului, cât și la variațiile numărului și modificărilor structurale ale populației. Aceste autoreglări au însă limite peste care nu se mai poate trece, fără ca ecosistemul să nu se modifice substanțial, sau chiar să se prăbușească.

Dacă condițiile noi depășesc limitele între care oscilează schimbările obișnuite, echilibrul dinamic al ecosistemelor este rupt, cel mai adesea ireversibil și cu efecte imprevizibile. Catastrofele naturale și, mai ales, activitatea umană produc ecosistemelor cele mai profunde tulburări.

Progresul tehnic aduce cu sine, alături de atâtea minunate realizări, numeroase neajunsuri și o multitudinea de substanțe poluante, care amenință cu distrugerea mediului înconjurător.

Orice substanță rezultată din procese chimice, fizice și biologice, care, răspândită în mediul ambiant, dăunează organismelor vii, bunurilor materiale, operelor de artă și peisajului se numește substanță poluantă[1].

Poluantul este o substanță solidă, lichidă, gazoasă sau sub formă de energie (radiație electromagnetică, ionizantă, termică, fonică sau vibrații) care, introdusă în mediu, modifică echilibrul constituenților acestuia și a organismelor vii și aduce daune bunurilor materiale. Cuantificarea efectelor activității umane și a proceselor naturale asupra mediului, a sănătății și securității omului, precum și a bunurilor de orice fel se realizează prin *evaluarea* impactului asupra mediului. Prejudiciul reprezintă efectul cuantificabil în cost al daunelor asupra sănătății oamenilor, bunurilor sau mediului provocat de poluanți, activități dăunătoare sau dezastre[39].

În urma deteriorării mediului se produce alterarea caracteristicilor fizico-chimice și structurale ale componentelor naturale ale mediului, reducerea diversității și productivității biologice a ecosistemelor naturale și antropizate, afectarea echilibrului ecologic și a calității vieții cauzate, în principal, de poluarea apei, atmosferei și solului, supraexploatarea resurselor, gospodărirea și valorificarea lor deficitară, ca și prin amenajări necorespunzătoare ale teritoriului [58].

Gazele, evacuate pe coșurile de fum ale instalațiilor de ardere, conțin, de asemenea, importante cantități de oxizi de sulf, oxizi de azot, monoxid și dioxid de carbon, praf de cenușă, etc. Chiar și prin canalul de fum al coșurilor imobilelor de locuit sunt emise importante substanțe poluante. Amploarea și primejdia acestor emisii se explică, pe de o parte, prin aceea că emisiile de fum se fac la mică înălțime și deci dispersia lor în aer este slabă iar, pe de alta parte, prin conținutul relativ ridicat de hidrocarburi grele și gudroane, explicabil prin temperaturile mici la care se desfășoară procesele de ardere.

La toate acestea se adaugă noxele evacuate în aer odată cu gazele eșapate de autovehicule, și anume: monoxid și dioxid de carbon, oxizi de azot, aldehide, acid clorhidric și bromhidric, acid sulfuric și azotic, combinații ale plumbului, etc. Cantitatea acestora este deosebit de mare, în acest *secol al civilizației* și depășește, în multe regiuni, pe cea emisă de instalațiile de ardere staționare[33].

Problema rezidurilor activităților umane a luat proporții îngrijorătoare prin acumularea lor provocând alterarea calității factorilor de mediu.

Asigurarea unei calități corespunzătoare a mediului, protejarea lui reprezintă o problemă de interes major și certă actualitate pentru evoluția socială. În acest sens se impune păstrarea calității mediului, diminuarea efectelor negative ale activității umane cu implicații asupra acestuia[82].

Protecția mediului este o problemă majoră la nivel mondial și acet lucru a impus înființarea unor organizații internaționale ce au ca scop adoptarea unor soluții de diminuare a poluării și creșterea nivelului calității mediului în ansamblu.

2.2 Dioxidul de sulf

Dioxidul de sulf SO₂ este unul din cele mai importante gaze pentru monitorizarea poluării aerului. Oxizii de sulf au un semnificativ impact asupra sănătății umane. Ei sunt responsabili pentru multe probleme respiratorii și cardiovasculare. Oxizii de sulf sunt precursor al ploilor acide și ai altor efecte atmosferice cu efecte toxice asupra solului și a vegetației. Concentrația de dioxid de sulf poate influența comunitățile de plante precum și viața animalelor, dar nu în ultimul rând viața animalelor. SO₂ antropoc este aproape exclusiv format în timpul combustiei combustibililor fosili cu relevant conținut de sulf și cel mai uzual este cărbunele[2,59].

Câțiva combustibili fosili conțin sulf ce este eliberat ca dioxid de sulf prin combustie. Diferiți combustibili sunt caracterizați într-o gamă largă de conținut de sulf: petrolul și produșii săi conțin între 0.1% sulf (parafina) și 3-4% (petrol greu) în forma sulfidelor și tioli[18]. În prezent sunt tehnici comerciale speciale de extragere a sulfului și reducerea șansei de a genera SO₂ din cărbune ce conține 0.1-4%, sulf în principal ca fulgi de pirită(FeS₂). Media de conținut de sulf a rezervelor europene este de 1.7%. Gazul natural (în principal metanul) este fără sulf, dar alte gaze în concordanță cu originile lor pot conține până la 40% din volum cum este hidrogenul sulfurat care generează SO₂ prin combustie[43]. Dacă conținutul de sulf al combustibilului este în totalitate ars, SO₂ este format prin reacția:



În timpul arderii incomplete, cu lipsa oxigenului, sulful și hidrogenul sulfurat pot fi formate la temperaturi înalte.



Hidrogenul sulfurat apare în timpul combustiei cărbunelui de slabă calitate ca lignitul sau în țevile de evacuare ale motoarelor. În timpul combustiei și în canalele de evacuare, SO_2 poate fi oxidat la SO_3 care va forma cu apa din gazele de ardere acidul sulfuric H_2SO_4 . Evacuat în atmosferă, SO_2 reacționează în proporție de 1-2 la mie cu oxigenul în prezența radiației ultraviolete și va forma SO_3 [96].



Dupa aceea, SO_3 va reacționa cu vaporii de apă din atmosferă formând acidul sulfuric H_2SO_4 . În perioadele cu ceață sau zile cu umiditate crescută, rata de transformare a SO_3 în H_2SO_4 poate fi până la 15%.

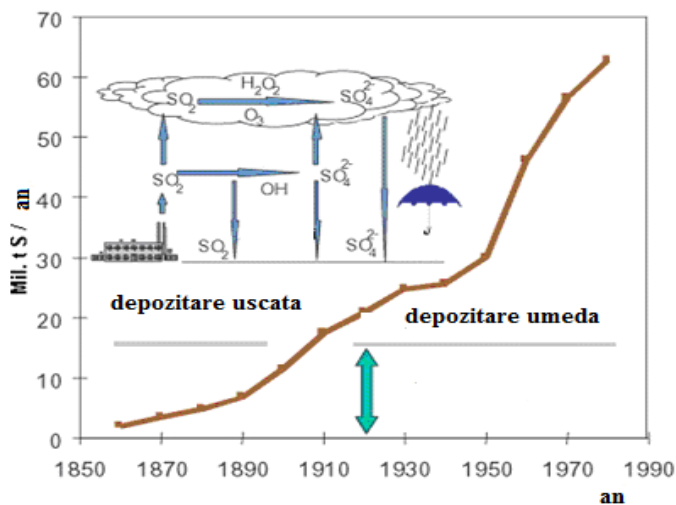


Figura 2.2 Reacțiile atmosferice ale SO_2 și emisiile istorice[43]

În figura 2.2 este prezentată o vedere ilustrată a principalelor reacții ale SO_2 și evoluția istorică a emisiilor poluante în aer.

Pentru motivele menționate este foarte important să monitorizăm concentrația de SO_2 în atmosferă și în special controlul surselor în concordanță cu maximumul valorilor admise la evacuare pentru calitatea aerului. Vulcanii sunt responsabili pentru o mare cantitate de SO_2 natural degajat în atmosferă.

Măsurarea prin teledetectie pentru SO_2 a fost folosită cu succes în trecut pentru activitățile vulcanice și pentru monitorizarea gazelor de ardere. Acum această tehnică este propusă și pentru centrale termice. Tehnologia este bazată pe senzorul unei camere capabil să preia informații în porțiunea UV a spectrului, iar camera este poziționată înspre gazele de ardere.

Dioxidul de sulf se găsește în gaze vulcanice și dizolvat în apele din apropierea vulcanilor. Totodată se mai găsește în atmosfera orașelor industriale [30].

Dioxidul de sulf lichid se obține industrial din gazele sulfuroase rezultate în cadrul fabricilor de acid sulfuric [77,78]. Pentru alegerea cât mai judicioasă a procedurii de lichefiere și a condițiilor de lucru este necesar cunoșterea modului de variație cu temperatura a presiunii de vapori deasupra dioxidului de sulf lichid. Această variație rezultă din următoarele date:

Tabelul 1.1 Variația presiunii SO₂ cu temperatura

Temperatura	-65	-50	-30	-10	0	+10	+30	+50
Presiunea dioxidului de sulf	0,039	0,113	0,376	1	1,529	12,56	4,498	8,175

Din datele menționate reiese ca la -10C dioxidul de sulf lichid are presiunea de vapori egală cu 1 at. În consecință presiunea parțială a dioxidului de sulf din gaze trebuie să fie cel puțin egală cu 1 at.

Întrucât gazele conțin 7% SO₂ este necesară comprimarea acestora la o presiune egală cu 1/0,07=14,3at. Dacă gazele ar fi conținut 14% SO₂ ar fi fost necesară comprimarea acestora doar până la 1/0,14=7,5at iar la un conținut de 100% SO₂ nu ar fi fost necesară nici un fel de comprimare[77]. Plecandu-se de la aceste considerații în practică au fost adoptate 2 categorii de procedee de lichefiere a gazelor obținute în cadrul fabricilor de acid sulfuric și anume:

- Procedee de lichefiere cu concentrarea prealabilă a gazelor
- Procedee directă de lichefiere

2.2.1 Proprietăți ale dioxidului de sulf

Gazul este incolor cu miros inecăcios (toxic, iritant, sufocant). Dioxidul de sulf inhalat într-o atmosferă cu o concentrație mai mare poate produce efecte asupra organismului uman grave. În interiorul plămânilor există vapori de apă și SO₂ se combină cu apa și formează acidul sulfuros; acesta în prezența enzimelor oxidante, prezente în număr mare în mediul intracelular, trece în acid sulfuric care atacă țesutul pulmonar[13]. Reacțiile SO₂ ce au loc după eliberarea acestuia în atmosferă sunt prezentate mai jos.



Pe lângă efectul intern (asupra plămânilor) dioxidul de sulf are efect lacrimogen, provocând totodata dureri de cap în cazul unor intoxicații mai ușoare[31]. Este mai greu decât aerul (se culege într-un vas cu gura în sus). Se lichefiază la -10°C și 3 atmosfere. Este solubil în apă (prin dizolvare vaporii de apă din atmosferă formează o ceață). Nu arde și nu întreține arderea, se combină cu oxigenul la temperatură înaltă cu ajutorul unor catalizatori cum ar fi oxizii de azot.

Dioxidul de sulf este un bun solvent pentru hidrocarburile aromatice. Se știe că petrolul este un amestec de hidrocarburi și prin arderea petrolului brut se degajă mult fum. Au fost numeroase încercări pentru a se separa hidrocarburile aromatice din petrol, deoarece ele prin ardere formează negru de fum. În aceste sens, chimistul român Lazăr Edeleanu a pus la punct în 1907 procedeul acesta la rafinaria Rîfov de lângă Ploiești, procedeul numit *Procedeul Edeleanu*. În acest procedeu prezentat în figura 2.3 are loc dizolvarea selectivă a hidrocarburilor aromatice în dioxid de sulf lichid. La amestecarea SO₂ lichid cu amestecul de hidrocarburi se formează 2 straturi, unul cu densitate mică alcătuit din hidrocarburi nearomatice și cantități mici de SO₂ lichid (rafinatul) și altul cu densitate mai mare format din hidrocarburi aromatice dizolvate în SO₂ lichid (extractul)[14].

Amestecul de hidrocarburi se introduce într-o coloană 1 unde se amestecă în contracurent cu SO₂ lichid. El solubilizează hidrocarburi aromatice și extractul se elimină pe la baza coloanei. Mai trec și hidrocarburi nearomatice în extract, de aceea se spală cu motorină în coloana 2. Extractul în continuare trece în coloana 3 unde se îndepărtează SO₂ prin evaporare și după lichefiere, acesta se reintroduce în proces. Hidrocarburi aromatice se îndepărtează pe la baza coloanei. Hidrocarburi nearomatice și motorina din coloana 2 se separă de SO₂ antrenate în coloana 4 și coloana 5 (prin distilare motorină). Pe la partea superioară a extractorului se îndepărtează rafinatul. Urmele de SO₂ antrenate sunt separate într-un separator 6 și după lichefiere sunt reintroduse în circuit [18,75].

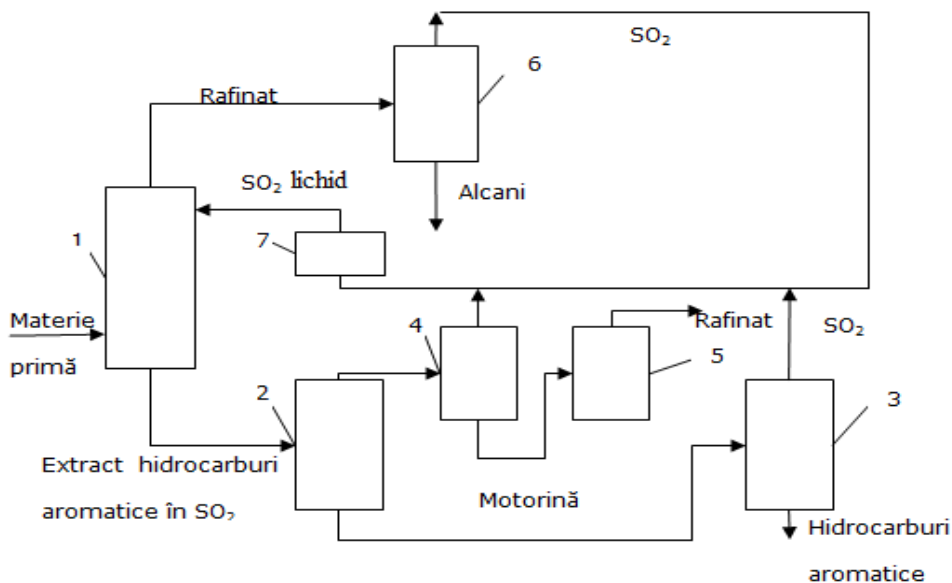


Figura 2.3 Schema instalației de extracție cu SO₂

În figura 2.3 avem următoarele componente: 1-extractor cu SO₂, 2-extractor cu motorină, 3,4,5- coloane de distilare, 6-separator, 7-instalație de lichefiere, 8-dioxid de sulf

2.2.2 Proprietăți corozive

În prezența dioxidului de sulf corозиunea atmosferică a fierului se modifică puternic. Schikorr arată că procedeul de corозиune a fierului decurge în trei etape:

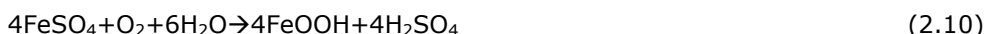
În prima etapă are loc adsorbția SO₂ de către rugina formată în mediul umed; adsorbția se produce cu viteză mare astfel că echilibrul se stabilește foarte repede [15].

36 Stadiul actual de cunoaștere al poluării atmosferice cu SO₂ - 2

Etapa a doua corespunde reacției SO₂ absorbit cu oxigenul atmosferic și dizolvat în filmul de lichid pentru a se forma sulfatul feros conform reacției:

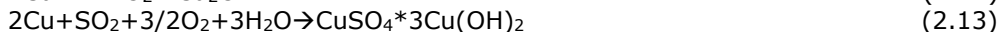


În etapa a treia se transformă sulfatul feros în rugină[8]. Proces ce are loc prin intermediul acțiunii catalitice a H₂SO₄ asupra sulfatului feros conform reacțiilor următoare.



Funinginea catalizează alături de SO₂, procesul de coroziune (se accelerează).

Coroziunea cuprului în atmosferă de SO₂ decurge după următoarele 2 reacții:



Oxidul cupros absoarbe dioxidul de sulf din atmosferă cu formarea unor săruri bazice cu stabilitate ridicată, cu proprietăți protectoare. Funinginea, praful, particulele solide pot perturba formarea straturilor protectoare și favorizează apariția coroziunii în puncte. În atmosfera marină viteza de coroziune a cuprului (de puritate 99%) este de peste 3 ori mai mare decât în atmosfera continentală. Coroziunea cuprului în atmosfera umedă (U_R>55%) implică formarea produșilor de coroziune care depind de impuritățile prezente[22,73].

SO₂ determină procese distructive asupra nemetalelor. SO₂ acționează asupra monumentelor istorice din marmură; se transformă CaCO₃ în gips (CaSO₄·2H₂O). Stratul superficial al monumentelor conține între 85 și 97% gips și între 5-15% CaCO₃. H₂S reacționează cu fierul, formează FeS (etapă intermediară la coroziunea atmosferică) [76].

2.3 Efectele nocive ale poluanților

Orice substanță sau produs care, folosit în cantități, concentrații sau condiții aparent nepericuloase, prezintă risc semnificativ pentru om, mediu sau bunurile materiale (deci pot fi explozive, oxidante, inflamabile, toxice, nocive, corozive, iritante, mutagene sau radioactive) se desemnează drept substanță periculoasă. Prin STAS 12574-87 se stabilesc concentrațiile maxime (medie lunară sau anuală) admise ale unor substanțe poluante în aerul din zonele protejate[42].

Din punct de vedere ecologic, există deosebiri destul de importante între diverse categorii de poluanți. Astfel se deosebesc:

- noxe care dăunează direct organismului uman, ca de exemplu oxizii de azot (NO_x), oxizii de sulf (SO_x), monoxidul de carbon (CO), precum și unele metale grele;
- noxe care acționează direct asupra vegetației, ca de exemplu dioxidul de sulf (SO₂) și combinațiile dintre Cl₂ și H₂;
- noxe care stau la baza formării de acizi, ca de exemplu SO₂, SO₃, NO și NO₂, ce determină formarea ploilor acide și distrugerea pădurilor;

- noxe persistente în soluri, care, în cadrul lanțului biologic plantă-animal-om, se acumulează și devin astfel nocive organismului uman, așa cum este cazul metalelor grele;
 - noxe care devin factori de influență ai climei, ca de exemplu CO₂ și N₂O, precum și factori importanți în declanșarea efectului de seră al pământului sau contribuie la distrugerea stratului natural de ozon.
- Se face observația că prin atmosferă se înțelege masa de aer care înconjoară suprafața terestră, incluzând și stratul de ozon.

2.3.1 Efectele nocive ale oxizilor de sulf

Din oxidarea sulfului combustibil, cea mai mare parte (peste 95%) se transformă în SO₂, restul în SO₃. Conversia SO₂ în SO₃ are loc în flacără, în cazul unui exces mare de oxigen, dar și pe traseul gazelor, în prezența oxizilor de vanadiu și chiar de fier, care joacă rol de catalizator, mai ales la temperaturi de peste 800°C (1073 K).

Evacuat în atmosferă, dioxidul de sulf (SO₂) reacționează în proporție de (1 - 2) 0/00/h cu oxigenul, sub acțiunea radiațiilor ultraviolete solare (ruv), dând naștere anhidridei sulfuroase (SO₃), conform relației:



Aceasta, la rândul ei, se combină cu vaporii de apă din atmosferă și formează acidul sulfuric.



În perioadele de ceață și în zilele foarte umede se atinge un grad de transformare de până la 15,7 % [79]. Dioxidul de sulf reprezintă o substanță toxică, care atrage atenția prin mirosul și acțiunea iritantă asupra mucoaselor, provocând spasm și contracția mușchilor căilor respiratorii superioare. În concentrații ridicate, SO₂ provoacă iritație și senzație de arsură asupra mucoaselor respiratorii și conjunctivale, tuse, tulburări ale respirației, spasm, senzație de sufocare, etc. [38].

Prezența oxizilor de sulf în mediul ambiant se manifestă atât prin leziuni directe ale plantelor, cât și prin modificarea compoziției apei și solului. În tabelul 2.1 sunt prezentate efectele nocive ale SO₂ având diverse concentrații. Astfel SO₂, în concentrație mare, distruge clorofila din frunze, acțiunea sa amplificându-se prin sinergism cu NO₂ [2]. Expunând frunze de diferite plante într-o atmosferă de NO₂, în concentrație de 2 ppm și separat, într-o atmosferă cu SO₂, în concentrație de 0,7 ppm, după 4 ore, nu s-a observat nici o schimbare morfologică în structura frunzelor. Expunând însă aceleași frunze într-o atmosferă cu ambele noxe, dar într-o concentrație individuală mult mai mică decât în primul caz, (0,1 ppm pentru fiecare gaz), s-a observat o continuă modificare a țesutului frunzelor [41].

Tabelul 2.1: Efectele nocive ale anhidridei sulfuroase (SO₂) în aer, la diferite concentrații.

Concentrația [ppm]	Efecte fiziologice	Observații
0,3 ÷ 1,0	Se face simțită prin miros.	Concentrații tolerabile în ateliere și zone de lucru.
1,0 ÷ 10	Este posibilă iritarea nasului și ochilor.	Posibilități de suportare scăzând până la o oră cu creșterea concentrației.
10 ÷ 100	Iritarea accentuată la aceleași organe, ca mai sus.	Posibilități de suportare scăzând până la o oră cu creșterea concentrației.
150 ÷ 650	Atac al aparatului respirator.	O jumătate de oră până la o oră de expunere poate pune viața în pericol, funcție de individ.
10000 sau 1 %	Paralizie respiratorie progresivă.	Concentrație rapid mortală. O iritare vie a părților umede ale pielii, ce apare după câteva minute, este un indiciu.

Oxizii de sulf, respectiv acizii sulfuros și sulfuric, care rezultă prin hidratarea acestora, determină fenomene de coroziune, decolorarea materialelor colorate, reducerea elasticității și rezistenței pentru unii compuși organici (amine, polimeri, textile etc.), unele materiale de construcție și unele tipuri de cabluri electrice.

Oxizii de sulf, alături de cei de azot, sunt astăzi considerați principalele cauze ale ploilor acide, care cauzează distrugerea pădurilor, pe suprafețe îngrijorător de mari [89]. Modificările în compoziția apei și a solului au ca rezultat tulburări de dezvoltare a plantelor, o scădere a producției de masă lemnoasă, respectiv a producției și calității fructelor, cu întregul cortegiu de consecințe economice și de altă natură, ultimele manifestate prin lanțul trofic plantă – animal – om.

Calitatea aerului ambiental sub aspectul acidității este dată de concentrațiile de dioxid de sulf (SO₂) și de oxizi de azot (NO_x) parametrii înregistrați zilnic pentru municipiul Timișoara [108].

2.3.2 Emisii anuale de dioxid de sulf (SO₂)

Dioxidul de sulf este emis în aer în urma proceselor de combustie a materialelor fosile carbonice. Sursele majore de dioxid de sulf sunt marile centrale termice, marile instalații industriale de ardere precum și unitățile de încălzire colective și individuale. Dioxidul de sulf este prezent și în emisiile de la motoarele Diesel, dar contribuția este minora comparativ cu aportul instalațiilor mari de ardere din industria energetică. Aflat în atmosferă, dioxidul de sulf se combină cu vaporii de apă și provoacă ploile acide cu efect distructiv asupra vegetației și clădirilor.

În tabelul 2.2. se prezintă evoluția emisiilor de SO₂ în județul Timiș. Se constată o scădere constantă a cantităților anuale ca urmare a reducerii consumului de combustibili fosili în favoarea celor gazeși și lichizi.

Tabelul 2.2 Emisii anuale de SO₂ (t/an)[106]

Județul	Anul					
	2004	2005	2006	2007	2008	2009
TIMIȘ	3973	3333	3900	3551	3687	1283

În tabelul 2.3 se prezintă sursele emisiilor de dioxid de sulf pentru anul 2009.

Tabelul 2.3. Emisii totale de SO₂ în județul Timis (t/2009)[106]

Grupa	nume grupa	SO ₂ (Mg)
01	Arderi in energetica si industrii de transformare	1232.38
02	Instalatii de ardere neindustriale	35.42
03	Arderi in industria de prelucrare	11.25
08	Alte surse mobile si utilaje	4.01
09	Tratarea si depozitarea deseurilor	0.11
	Total	1283.17

Atmosfera devine un pericol, ca urmare este în pericol întregul mediu de viață. De aceea este nevoie de un control riguros și de măsuri radicale pentru ca viitorul atmosferei să fie sigur. Agenția pentru Protecția Mediului Timiș a monitorizat starea de calitate a aerului prin intermediul analizelor efectuate de către laboratorul propriu[106].

Prin transpunerea legislației europene privind evaluarea și gestionarea calității aerului înconjurător, în decursul anului 2009, în aglomerarea Timișoara s-a continuat monitorizarea calității aerului cu cele 5 stații automate amplasate în zone reprezentative pentru evaluarea calității aerului[110].

Corelarea nivelului concentrației poluanților cu eventualele surse, se face pe baza datelor meteorologice obținute în stație, aceasta fiind prevăzută cu senzori meteorologici de direcție și viteză a vântului, de temperatură, de presiune, de umiditate, precipitații și de radiație solară.

Determinarea nivelului de poluare a aerului cu SO₂ s-a efectuat prin monitorizarea continuă în cele 5 stații automate pentru calitatea aerului. Valorile înregistrate în decursul anului 2009 sunt prezentate în tabelul 2.4.

Tabelul 2.4 SO₂ - Concentrații medii orare la stații în 2009($\mu\text{g}/\text{m}^3$) [105]

Stație	Luna											
	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII
TM-1	8,69	9,07	10,32	5,92	6,76	6,27	6,40	5,91	10,04	9,09	11,96	5,05
TM-2	10,3	13,1	19,6	16,1	10,7	5,91	7,2	7,2	7,8	7,0	6,8	6,9
TM-3	10,7	8,3	6,9	3,9	6,5	6,2	5,6	5,9	8,1	-	-	-
TM-4	17,6	12,9	9,7	9,5	5,7	6,8	7,4	7,1	8,9	10,7	12,3	-
TM-5	6,1	4,8	4,2	4,0	4,0	2,9	2,6	2,6	3,5	6,4	7,1	6,0

2.4 Rezumatul capitolului Stadiul actual de cunoaștere al poluării atmosferice cu SO₂

Se definește atmosfera ca fiind învelișul gazos care se menține în jurul Pământului datorită gravitației. Gazele atmosferice sunt: azot, oxigen, argon, dioxid de carbon, urme de gaze în diferite procente, compoziție care se menține aproape constantă. Dacă se modifică compoziția atmosferei, prin introducerea unor poluanți apare fenomenul de poluare. Poluanții pot fi antropici și naturali. Poluarea antropică prezintă un impact major asupra mediului înconjurător și al sănătății la nivel global. Poluanții naturali pot cauza și ei efecte severe dintre care putem aminti: praful și gazele emise de vulcani, focurile din păduri, descompunerile unor compuși organici, etc. Sursele care pot provoca poluarea antropică a aerului sunt fabricile și uzinele mici, autovehiculele, cele legate de producerea și consumul de energie unde sunt utilizați combustibili fosili și derivații lor. Combustibilii fosili, biomasa pot provoca poluare datorită apariției de CO, CO₂, SO₂ și alte particule, etc. Răspândirea poluanților în atmosferă depinde de mai mulți factori: distanța față de sursă, starea vremii, etc.

Calitatea aerului este o importantă și globală problemă a tuturor națiilor. Calitatea aerului este influențată de următorii factori principali: poluare de fond, poluare de impact, ploile acide, gazele cu efect de seră. Pentru protejarea calității aerului s-au enunțat legi, decrete care să reglementeze factorul aer. Dioxidul de sulf este unul dintre cele mai importante gaze pentru monitorizarea poluării aerului având în vedere efectele asupra sănătății umane, a proprietăților sale, în special cele corozive.

3 Metode de măsurare a poluării atmosferice cu SO₂

3.1 Determinarea concentrației procentuale a dioxidului de sulf în gaze

Determinarea concentrației de SO₂ din gaze se bazează pe oxidarea lui la trioxid de sulf cu ajutorul iodului care se reduce la acid iodhidric.



Analiza se face cu un aparat pentru analiza gazelor prezentat în figura 3.1. Se introduce un volum cunoscut de soluție titrată de iod, cațiva mililitrii cu soluție bicarbonat de sodiu 5%, 5ml soluție de amidon, 1% și se completează cu apă până la jumătate din înălțimea lui (cca 50ml apa distilată). Bicarbonatul de sodiu are rolul de a neutraliza acidul iodhidric format.

Înainte de a începe determinările, se verifică etanșeitarea aparatului. De aceea se închide robinetul d și se deschide robinetul e. Dacă nu curge apa din vasul b umplut inițial cu apă, aparatul este etanș și se pot face determinări. Concentrația procentuală a SO₂ se determină din relația:

$$C_{\text{SO}_2} = \frac{V_{\text{SO}_2} * 100}{V_{\text{SO}_2} + W} \quad (3.2)$$

W – ml apă din soluție

Această metodă este gaz-volumetrică. Se poate folosi și tiosulfatul de sodiu pentru excesul de iod.

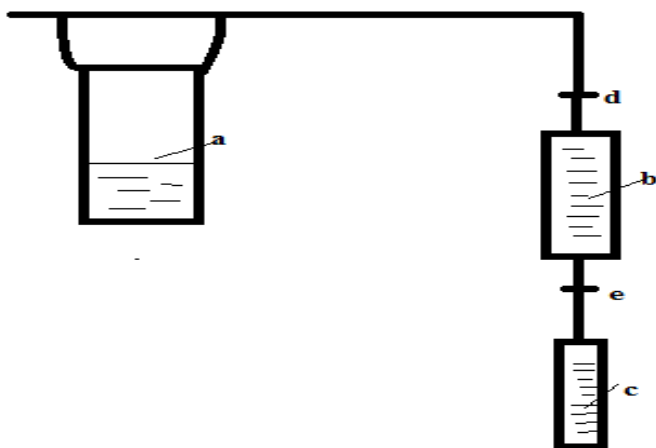


Figura 3.1 Aparat pentru analiza gazelor sulfuroase
a-vas barbotor, b- vas cu apă, c-cilindru gradat, d si e robinete

3.1.1 Legea Lambert – Beer

Dacă lumina străbate o substanță sau soluția ei, radiațiile de lungimi de undă diferite sunt absorbite în diferite proporții. A determina spectrul de absorbție al unei substanțe înseamnă a măsura micșorarea intensității luminii ce o străbate. Micșorarea intensității luminii de o anumită lungime de undă este independentă de intensitatea inițială absolută și dependentă de grosimea stratului de substanță absorbantă străbătut[20].

Dacă substanța este dizolvată într-un solvent transparent, intensitatea absorbției depinde de concentrația c a soluției, de grosimea l a stratului de soluție străbătut.

I_0 -intensitatea luminii înainte de străbaterea soluției

I -intensitatea luminii după străbaterea soluției

Relația care exprimă legea Lambert Beer:

$$\ln \frac{I_0}{I} = \varepsilon cl \quad (3.3)$$

Mărima $E = \ln \frac{I_0}{I}$ se numește extincție (densitate optică sau absorbantă). ε este o proprietate caracteristică a substanței independentă de concentrație, dar depinzând de lungimea de undă.

Dacă l se exprimă în cm și c în mol/l, atunci constanta ε se numește coeficient molar de extincție (absorbivitate molară)

Spectrul de absorbție se obține înscriind într-un grafic variația unei mărimi ce caracterizează absorbția luminii (de exemplu extincția E , coeficientul de extincție ε , sau raportul $\frac{I_0}{I}$). în funcție de lungimea de undă. În spectrele astfel obținute

apar minime și maxime, ultimele numite benzi de absorbție corespund regiunilor de absorbție maximă. Fiecare bandă de absorbție este caracterizată de lungimea de undă λ_{\max} și de lățimea benzii.

3.1.2 Aspecte teoretice legate de fenomenul de fluorescență.

Fluorescența este un fenomen de fotoluminescență care presupune generarea unui fenomen de absorbție inițial. Moleculele luminescente excitate de anumite stări cuantice (în acest caz de iradierea în domeniul ultraviolet), rămân în această stare cel puțin 10^{-9} sec, după care revin la starea fundamentală, prin emisie de luminescență. Această emisie secundară este proporțională cu moleculele de dioxid de sulf SO₂ din probă și captată, transformată și transmisă ca semnal electric de către tubul fotomultiplicator (PMT) al aparatului [20].

$$I = I_0^{-t/\tau} \quad [Cd] \quad (3.4)$$

în care:

I – intensitatea la timpul t , după îndepărtarea sursei de excitație, în Cd;

I_0 – intensitatea constantă aplicată pe parcursul excitației, în Cd;

t – timpul, în s;

τ – intervalul mediu de scădere (timpul necesar ca intensitatea să scadă la 1/e din valoarea inițială), în s.

În cazul în care se vizează aspecte cantitative legate de fenomenul de fluorescență este necesară măsurarea intensității luminii, fie că ne referim la spectrul de fluorescență sau la intensitatea fluorescenței, raportate la o intensitate standard a sursei de lumină (excitație).

Fenomenul de fluorescență apare atunci când molecula, aflată într-un anumit nivel de energie, posedă cel puțin două stări electronice distincte al căror nivel de rotație-vibrație se intersectează în așa fel încât să existe cel puțin un nivel de *rotație-vibrație* de aceeași valoare a energiei, în fiecare stare.

Un criteriu tehnic al instrumentelor spectroscopice este lungimea de undă folosită. Cei mai mulți poluanți aeropurtați sunt activați în benzi ale lungimii de undă foarte scurte și bine definite. Astfel, cele mai uzuale lungimi de undă utilizate în măsurarea acestor specii sunt în spectrul *infraroșu și ultraviolet vizibil*. În consecință, lungimile de undă utilizate în cazul fenomenului de fluorescență sunt cuprinse în intervalul (190 ÷ 320) nm.

3.1.3 Determinări prin analiza Gaussiană

Analiza gaussiană este o metodă de caracterizare a vârfurilor dintr-un semnal. Vârfurile sunt caracterizate prin poziție, semilărgime, intensitate. Pentru descrierea formei acestor vârfuri se folosesc: forma lorentziană corespunzătoare unor fenomene tipic cuantice și forma gaussiană corespunzătoare unor fenomene cu un accentuat caracter dezordonat, numite uneori fenomene de relaxare[32].

Nu sunt informații referitoare la forma vârfului și de aceea semilărgimea și intensitatea vârfului se determină prin aproximare cu un profil gaussian.

Pentru localizarea vârfului, metoda cea mai potrivită este derivarea numerică a semnalelor. Valoarea maximă a vârfului din semnal corespunde intersecției axei absciselor cu graficul derivatei de ordinul întâi. Pentru determinarea vârfului dintr-un semnal se folosesc două metode:

Metoda gradientului(puncte de extrem)

La analiza gaussiană metoda gradientului precizează minimumul negativ absolut și maximumul pozitiv absolut, din derivata de ordinul întâi a semnalului. Diferența dintre valoarea maximă și minimă determină semilărgimea sau adâncimea vârfului.

Poziția vârfului este unde derivata de ordinul întâi atinge valoarea 0. (totodată la zero este și amplitudinea semnalului). Uneori derivata de ordinul întâi intersectează axa absciselor de mai multe ori și în acest caz se determină doar limita inferioară a vârfurilor din semnal.

Derivata a doua a semnalului este o funcție care își schimbă semnul de fiecare dată când derivata de ordinul întâi a semnalului își schimbă direcția. De aceea, derivata de ordinul doi a semnalului intersectează de mai multe ori axa absciselor față de prima derivată(complicând găsirea minimumului). Adâncimea vârfului este dată de diferența dintre două zerouri consecutive ale derivatei de ordinul doi a semnalului care reprezintă punctele de inflexiune ale semnalului[32]. În figura 3.2 este prezentat modelul gaussian de dispersie, pentru care se acceptă o distribuție gaussiană a valorii vectorului viteză, atât pe direcția vântului, cât și pe direcția perpendiculară pe acesta. Defășurarea spațiului are notații consacrate, de care depinde și scrierea ecuațiilor. Modelul penei gaussiene este prezentat mai jos.

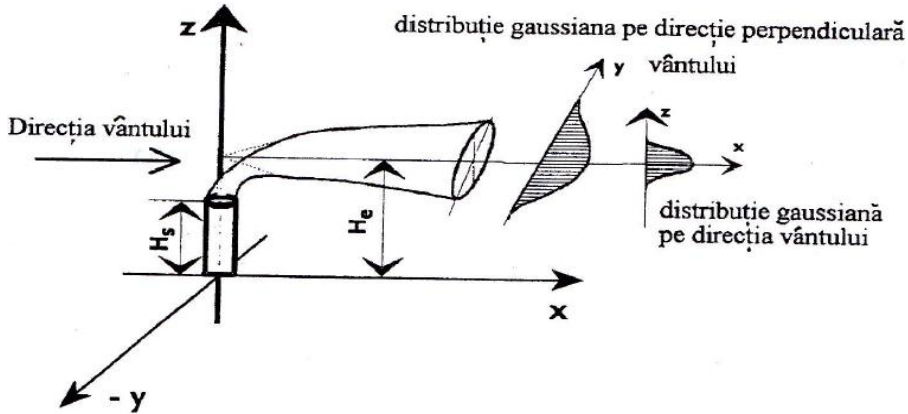


Figura 3.2 Distribuția gaussiană a norului de poluant[32]

Determinarea concentrației noxei la înălțimea z deasupra solului se calculează cu relația:

$$C = \frac{Q}{2\pi\sigma_y\sigma_z} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \left[\exp\left(-\frac{(z-H)^2}{2\sigma_z^2}\right) + \exp\left(-\frac{(z+H)^2}{2\sigma_z^2}\right) \right] \quad (3.7)$$

Unde:

C -concentrația de noxă(kg/m³)

Q -debitul unei surse punctiforme continue(kg/s)

H -înălțimea efectivă a penei deasupra solului(m)

y -distanța laterală de la centrul penei(m)

z -înălțimea deasupra solului

În procesul aplicării modelului gaussian al penei este necesar să se determine valorile parametrilor de difuzie σ_y și σ_z ca funcție de distanța x . Deși s-a dezvoltat o rețea de scheme ale claselor de stabilitate și ale curbelor σ , marea majoritate a meteorologilor se servesc de cele ale lui Pasquill[32], ale cărui observații asupra turbulenței atmosferice au fost folosite pentru estimarea lui σ_y și σ_z .

Următoarele ecuații de similitudine exprimă legătura dintre σ_y și σ_z și turbulență:

$$\begin{aligned} \sigma_y &= \sigma_\theta x S_y \left(\frac{x}{UT_L^v} \right) \\ \sigma_z &= \sigma_\theta x S_z \left(\frac{x}{UT_L^w} \right) \end{aligned} \quad (3.8)$$

Irwin a obținut următoarele aproximații la metoda lui Pasquill:

$$S_y = (1 + 0,031 \cdot x^{0,46})^{-1} x < 10^4 \text{ m, și} \quad (3.9)$$

$$S_y = 33x^{-1/2} \quad x > 10^4 \text{ m.}$$

Draxler[32] a prezentat grafic observațiile asupra $S_z = \frac{\sigma_z}{\sigma_w t}$, ca funcție de timp după emisie și a sugerat formula:

$$S_y = S_z = \left[1 + 0,40 \left(\frac{x}{UT_L^y} \right)^{1/2} \right]^{-1} \quad (3.10)$$

Această formulă este valabilă pentru surse la nivelul solului, în orice condiții de stabilitate pentru S_y și pentru condiții stabile și neutre pentru S_x .

Pentru fiecare clasă de stabilitate a atmosferei se indică câte o curbă: de la A (atmosfera cea mai instabilă) la F (atmosfera cea mai stabilă). Aceste clase sunt bazate pe nebulozitatea atmosferei, gradul de acoperire cu nori, viteza vântului la suprafață și insolarea atmosferei. Inițial curbele σ au fost prezentate grafic așa cum au dovedit Pasquill – Gifford.

Evoluția calculatoarelor a făcut însă posibil ca ecuația gaussiană a penei de noxe să poată fi programată cu ușurință și este preferabil să existe formule analitice (de formă Briggs .) pentru aceste curbe, pentru a calcula deviațiile σ_y și σ_z în cazul surselor aflate în zone rurale:

$$\sigma_y = 465,11628(x) \cdot \text{tg}(\text{TH}) \quad [\text{m}]$$

unde:

$$\text{TH} = 0,017453293 [c \cdot d \cdot \ln(x)]$$

$$\sigma_z = a \cdot x^b [\text{m}]$$

a, c și d sunt coeficienți care sunt funcție de distanța x (km) și de clasa de stabilitate a atmosferei.

În cazul zonelor urbane se utilizează varianta McElroy – Pooler a formulelor pentru calculul deviațiilor σ_y și σ_z .

Norul de poluant(PUFF) se deplasează sub influența vântului și a turbulenței atmosferice. Concentrația poluantului în interiorul norului variază după o distribuție normală. Forma de variație spațială a concentrației în nor este binecunoscutul „clopot Gauss”. Acesta se păstrează de-a lungul întregii traiectorii a norului ,iar forma sa se modifică, aplatizându-se. Dispersia se schimbă, este o funcție crescătoare în timp(σ). Dacă se dorește să se anticipeze concentrația unui poluant dat, ca urmare a unei emisii în atmosferă, se estimează câte o valoare a concentrației pentru fiecare categorie de stabilitate atmosferică. Dependența dintre parametrii de dispersie σ și stabilitatea atmosferică este de forma: $\sigma = ax^b$.

unde x este distanța dintre sursa de poluare și punctul unde se estimează concentrația (receptor), iar a și b sunt constante ale căror valori numerice sunt date de clasa de stabilitate.

Modelul sursei de poluare punctiformă și continuă este prezentat în figura 3.3. Emisia continuă poate fi considerată din mai multe emisii instantanee fiecare generând un puf. Pufurile se deplasează pe o traiectorie mai mult sau mai puțin serpuită în funcție de fluctuația direcției vântului, se combină aditiv și formează pana de fum.

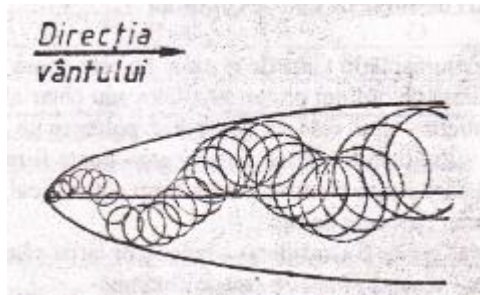


Figura 3.3 Emisia punctiformă continuă

Formula concentrației datorită sursei continue are forma:

$$C_{(x,y,0,H)} = A \exp \left[- \left(\frac{y}{\sqrt{2}\sigma_y} \right)^2 - \left(\frac{H}{\sqrt{2}\sigma_z} \right)^2 \right] \quad (3.11)$$

Unde $C(x,y,0,H)$ este concentrația poluantului la nivelul solului ($z=0$) emis dintr-o sursă punctiformă cu înălțimea efectivă H .

3.2 Metode standard de măsurare a poluării atmosferice cu SO₂

3.2.1 Analizorul de gaze TESTO 300 XXL

Gazoanalizorul TESTO 300 XXL [108] este un echipament de determinare a emisiilor gazoase din gazele de ardere, determinarea acestora realizându-se în celule speciale, în urma unor reacții chimice tip Peltier. Gazele analizate sunt SO₂, CO, O₂, NO și NO_x.

Analizorul de gaze TESTO 300 XXL detectează conținutul de O₂, calculează coeficientul excesului de aer λ și determină concentrația de CO₂, viteza de curgere a gazelor și debitul masic pentru toate speciile de gaze analizate.

Aparatul este format din trei subansambluri principale: unitatea de analiză, unitatea de control și sonda de prelevare a gazelor. Opțional se pot atașa diferiți senzori de temperatură precum și o sondă Pitot-Prandtl. Unitatea de analiză este echipată cu două intrări diferite pentru senzorii de temperatură: una pentru determinarea temperaturii gazelor prelevate și una pentru altă utilizare, spre exemplu pentru determinarea temperaturii ambiante.

Unitatea de analiză conține celulele de reacție, bateria de acumulatori, filtrele pentru reținerea impurităților solide din gazele de ardere și aer, decantorul pentru condens și conexiuni electro-pneumatice, așa cum reiese din figurile 3.4 și 3.5.

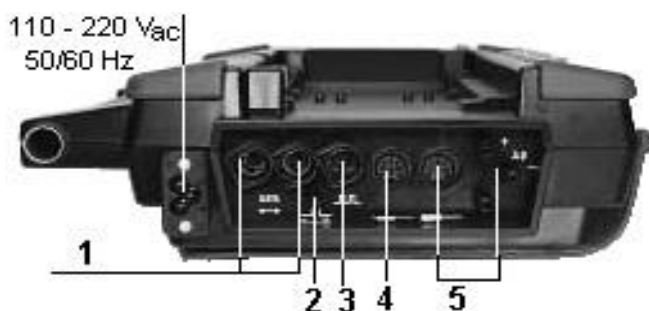


Figura 3.4 Unitatea de analiză Testo

1 - conexiuni transfer de date; 2 - diluție; 3 - semnal de alarmă;
4 - senzor opțional de temperatură; 5 - conectare sondă.

Gazul aspirat prin sondă este introdus în celulele de reacție când pompa de gaz este pornită manual sau automat. Înainte, însă, gazul de analizat este răcit brusc la $4 \div 8$ °C, având loc precipitarea condensatului cu absorbție scăzută de NO₂ și SO₂, condensat ce este evacuat la intervale regulate de timp de o pompă de condens. Gazul uscat trece apoi printr-un filtru special, în vederea reținerii particulelor solide. Acest filtru funcționează și ca o capcană pentru apă: dacă mai există apă în gazele aspirate filtrul devine impermeabil, prin închiderea porilor în contact cu apa, protejând astfel senzorii și pompa.

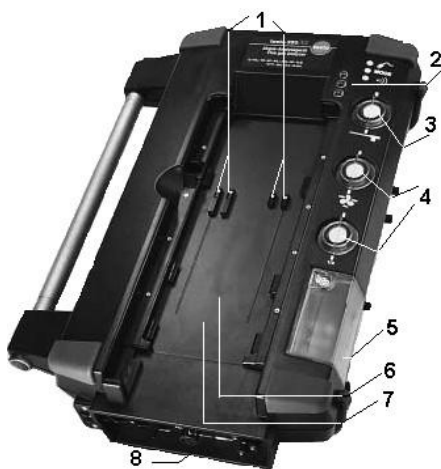


Figura 3.5 Unitatea de analiză

1 - contacte electrice; 2 - leduri de control; 3 - filtru particule solide; 4 - filtre reținere particule din aerul aspirat; 5 - colectare condens; 6 - celule de analiză; 7 - sistem integrat de determinare a vitezei și presiunii gazelor; 8 - conexiuni

În urma reacțiilor Peltier se emite un semnal electric preluat și prelucrat în unitatea de control, fiind afișate valorile concentrațiilor emisiilor de gaze analizate. Surplusul de gaz este evacuat continuu.

Celula de C_mH_n este diferită, concentrația de hidrocarburi fiind determinată prin oxidare catalitică, ceea ce presupune existența unei cantități minime de O₂ în gaze de aproximativ 2 %. Celula este închisă automat dacă valoarea O₂ este sub 2 %. De asemenea, este posibilă și închiderea manuală a celulei de hidrocarburi.

Pentru monitorizarea mai multor surse se pot interconecta până la 8 unități de analiză, echipate identic sau diferit și până la 20 unități separate de achiziție de date (pentru determinări de viteză, temperatură, umiditate, etc.) ca în figura 3.6, toate conectate la aceeași unitate de control și/sau la un PC.

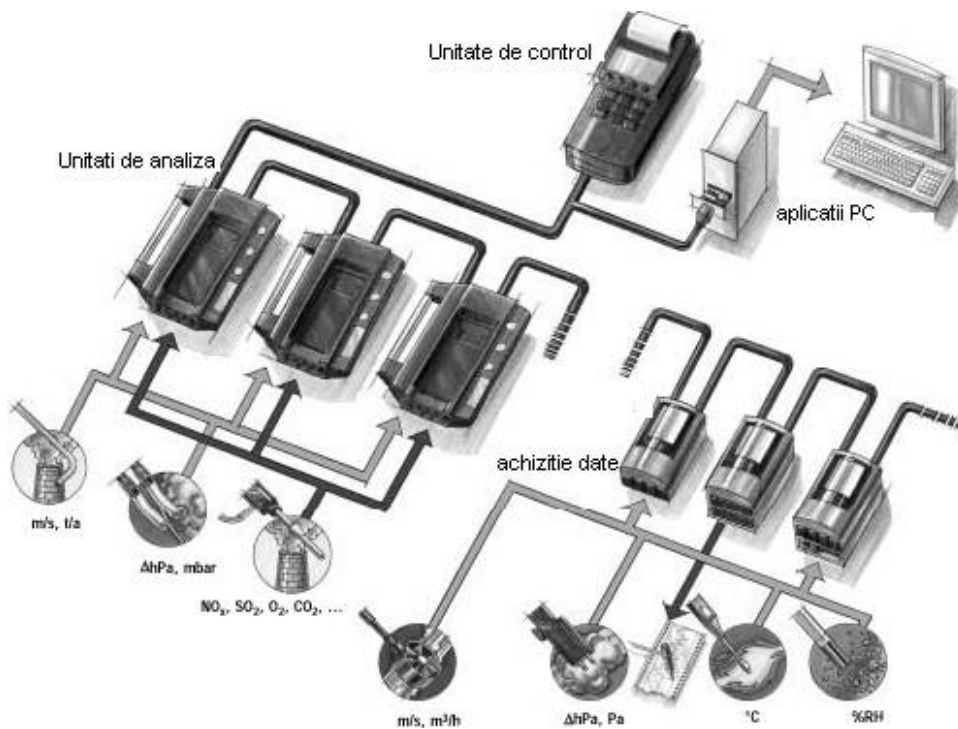


Figura 3.6 Interconectarea aparatelor pentru măsurători continue complexe

Unitatea de control (figura 3.7) este un dispozitiv de măsurare ce poate fi utilizat și independent de unitatea de analiză, însă nu pentru emisiile gazoase. Este echipat cu mufe de intrare la care pot fi conectați senzori de temperatură, umiditate, viteză, turbulență, presiune, curent și tensiune, precum și turație. Pot fi afișate grafic până la șase canale simultan. Unitatea de control poate fi operată fie cu ajutorul tastaturii fie al unui „touch-pen”. Pe lângă valorile datelor măsurate mai sunt afișate informații cum ar fi configurarea sistemului și locația. Aceasta este echipată cu sursă de lumina pentru operarea în condiții de întuneric. Prin conectarea unui sistem de achiziție de date se pot extinde intrările pentru senzori.



Figura 3.7 Unitatea de control

1 - imprimantă, 2 - touch-pen (creion electronic),
 3 - bară pentru informații de sistem, 4 - afișare valori măsurate,
 5 - bară pentru informații legate de funcționare, 6 - taste operare funcții,
 7 - tastatură, 8 - conectare sondă presiune,
 9 - conectare probă,
 10 - conectare unitate de analiză, 11 - interfață tip serial.

Unitatea este echipată cu memorie proprie, în care se pot stoca până la 250.000 de date, precum și cu o imprimantă încorporată pentru tipărirea acestora. Datele măsurate pot fi transferate și către un PC printr-o interfață tip serială.

Prin conectarea la PC gazoanalizorul TESTO 300 XXL poate fi folosit la măsurători de lungă durată (săptămâni), fiind controlat de un program special, ComSoft 3.

Aparatul poate fi echipat cu mai multe sonde de prelevare a gazelor. Acestea diferă în funcție de caracteristicile gazelor prelevate. Astfel există diferențe între sondele destinate prelevării gazelor de eşapament sau gazelor evacuate la coș, de intervalul de temperaturi de operare, de dimensiunile canalului de curgere a gazelor și pot fi sonde încălzite sau nu. În figura 3.8 este prezentată sonda de prelevare a gazelor evacuate la coș, sondă cu tubulatură încălzită la 180° C și care operează la temperaturi de maximum 1200 °C.



Figura 3.8 Sonda de prelevare a gazelor

Odată cu prelevarea și determinarea concentrației noxelor gazoase din gazele de ardere, aparatul calculează instantaneu următorii parametri:

- dioxidul de carbon, CO₂:

$$CO_2 = \frac{CO_{2,\max} \cdot (21 - O_{2\text{mas}})}{21} \% \quad (3.12)$$

în care: $CO_{2,\max}$ - valoarea maximă de CO_2 specifică combustibilului, în %

- pierderea de căldură cu gazele evacuate, qA (conf. ecuației Siebert de mai jos):

$$qA = \left[(FT - AT) \cdot \left[\frac{A_2}{(21 - O_2)} + B \right] \right] \cdot KK \quad (3.13)$$

în care:

FT - temperatura gazului prelevat;

AT - temperatura mediului ambiant;

A_2 și B - factori specifici combustibilului;

O_2 - conținutul de oxigen măsurat în gazele de ardere, în %;

KK - factor ce convertește diferența de temperatură la coș în pierderi relative la coș. Dacă factorii A_2 și B sunt nuli se corectează ecuația Siebert cu factorul „f”, specific combustibilului:

$$qA = f \cdot \frac{(FT - AT)}{CO_2} \quad (3.14)$$

- randamentul arderii, η , ținând cont doar de pierderile de căldură la coș:

$$\eta = 100 - qA \quad (3.15)$$

- coeficientul excesului de aer, λ :

$$\lambda = \frac{CO_{2,\max}}{CO_2} \quad (3.16)$$

- oxizi de azot, NO_x (cu o bună aproximație):

$$NO_x = NO + NO_2 \quad (3.17)$$

- viteza de curgere a gazelor, w [m/s]:

$$w = \frac{575 \cdot \Delta p \cdot (FT + 273,15)}{P_{\text{Abs}}} \cdot \alpha \quad (3.18)$$

în care:

P_{Abs} - presiunea absolută;

Δp - presiunea dinamică, [mbar];

α - factorul tubului Pitot, între 0 ÷ 1.50.

- debitul de gaze evacuate, V [m³/s]:

$$\dot{V} = w \cdot \frac{A1 \times A2}{10.000} \quad (3.19)$$

în care $A1 \times A2$ – secțiunea canalului de curgere, în [cm²].

După interconectarea celor trei elemente ale analizorului (unitate de control, unitate de analiză și sondă) se pornește aparatul fie prin conectarea acestuia la rețeaua electrică de 220 V fie la bateria proprie de acumulatori. În acest moment aparatul intră automat în procedura de calibrare la 0 și spălare a celulelor de reacție, figura 3.9. Sonda aparatului NU trebuie să fie introdusă în canalul de evacuare a gazelor arse.

După intrarea în regim a aparatului acesta trebuie programat să achiziționeze și să afișeze datele care prezintă interes pentru utilizator. În figura 3.10 se prezintă afișajul aparatului.

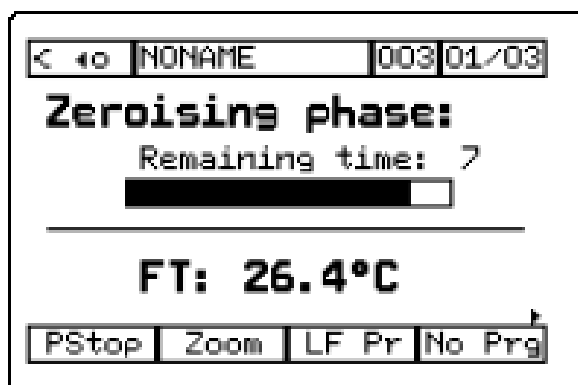


Figura 3.9 Calibrarea la 0, Testo 300

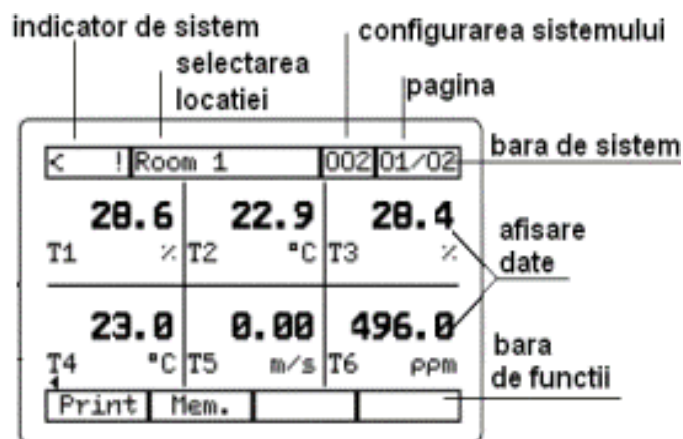


Figura 3.10 Display, Testo 300

Diferitele meniuri pot fi afișate în extenso fie prin atingerea ecranului cu un creion de contact (touch-pen), fie prin apăsarea tastelor din dreptul meniurilor. Fiecărui meniu îi corespunde unul sau mai multe submeniuri. În figura 3.11 se prezintă detaliat fiecare submeniu în parte și parametrii caracteristici fiecăruia. Astfel, se observă că aparatul poate fi programat să pornească automat, pot fi alese unitățile de măsură pentru fiecare parametru în parte, se poate seta tipul canalului de curgere a gazelor, etc.

Este obligatorie setarea aparatului pentru tipul de combustibil corespunzător, din baza de date a aparatului (lignit, cărbune brun, gaz, GPL, etc.), figura 3.12 și alegerea corectă a valorilor pentru oxigenul de referință, O_{2ref} , și CO_{2max} .

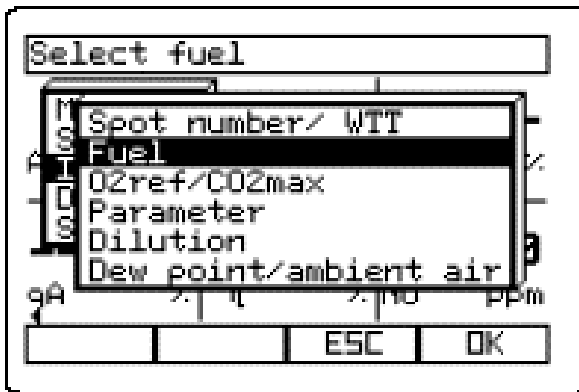


Figura 3.11 Alegerea combustibilului

După setarea parametrilor pe care dorim să-i măsurăm, figura 3.11, se introduce sonda în canalul de gaze și se pornește măsurarea. Datele măsurate pot fi memorate sau notate. Astfel se va întocmi un tabel în care se va nota compoziția gazelor de ardere.

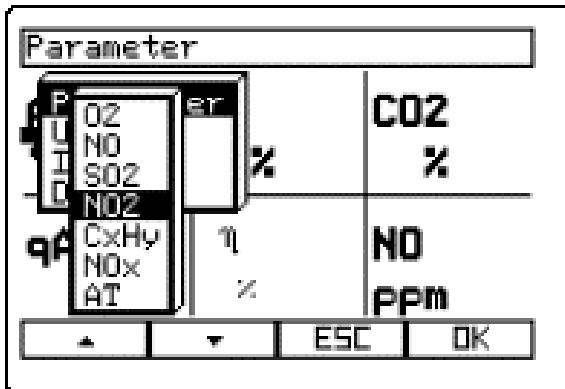


Figura 3.12 Setarea parametrilor, Testo 300

Pentru raportarea valorilor măsurate la legislația în vigoare trebuie făcută conversia acestora din [ppm] în [mg/m³], cu relațiile:

$$C_{CO} [mg / m^3] = \frac{M_{CO}}{22.4} \cdot \frac{21 - O_2^{ref}}{21 - O_2} \cdot CO [ppm] \quad (3.20)$$

$$C_{NO_2} [mg / m^3] = \frac{M_{NO_2}}{22.4} \cdot \frac{21 - O_2^{ref}}{21 - O_2} \cdot (NO + NO_2) [ppm] \quad (3.21)$$

$$C_{SO_2} [mg / m^3] = \frac{M_{SO_2}}{22.4} \cdot \frac{21 - O_2^{ref}}{21 - O_2} \cdot SO_2 [ppm] \quad (3.22)$$

unde 21 reprezintă concentrația oxigenului în aer, în [%].

Valorile astfel calculate se vor compara cu valorile limită prevăzute în legislația în vigoare, pentru instalația de ardere analizată și se vor analiza rezultatele.

3.3 Metode experimentale de măsurare a poluării atmosferice cu SO₂

Două tipuri de tehnici de teledetecție angajând spectroscopia de absorbție pot fi distinse. Instrumentele de teledetecție activă folosesc sursele artificiale de lumină în timp ce instrumentele pasive folosesc sursele naturale precum soarele, luna, lumina solară împrăștiată în atmosferă, sau lava fierbinte în infraroșu. Tehnicile de teledetecție pasive și mai ales cele active nu măsoară concentrația c a substanței, ci mai degrabă coloana de densitate oblică S . Densitatea coloanei S este integrala concentrației absorbite pe care radiația o ia pe drumul său de la sursa de lumină la instrument. Instrumentele active au avantajul că lumina în calea lor este bine definită. Radiația părăsește sursa artificială și poate fi redirecționată către un sistem optic ce trece prin pana de gaz poluant, este apoi colectată prin optica de recepție și analizată de instrument. Densitatea optică de gaz este măsurată și este proporțională cu densitatea pe coloană S fiind definită ca:

$$S = \int_L c(D) di \quad (3.23)$$

Deoarece lungimea optică L este cunoscută, media concentrației de-a lungul coloanei poate fi ușor calculată în conformitate cu ecuația, $c_{avg} = S/L$. Pana trebuie să fie între sursa de lumină și instrumentul pentru măsurători. Instrumentele trebuie să fie mobile și capabile să se adapteze la soare și diferite poziții ale penei, sau pot fi folosite numai în condiții corespunzătoare reducând în mod considerabil achiziția eficientă. Prezența norilor în atmosfera înconjurătoare sau o mare cantitate de aerosoli și picături de apă condensată în pană pot afecta căile optice bine definite[97].

Sursa de lumină alternativă pentru instrumentele pasive este lumina soarelui împrăștiată în atmosferă. Avantajul acestei surse este faptul că este mereu prezent în timpul zilei astfel încât instrumentele staționare pot măsura emisiile vulcanice ori de câte ori pana este în raza de vedere a instrumentului. Pentru măsurători ale luminii împrăștiată, calea de lumină de la soare la instrument nu este

bine definită. Folosim un model numeric pentru a simula un transfer radiativ în și în jurul unui nor vulcanic virtual care conține diferite cantități de SO₂ și aerosoli.

Când se efectuează măsurători de teledetecție a penelor vulcanice folosind lumina împrăștiată, calea de lumină idealizată a fost calea optică efectivă. În acest caz fotonii sunt împrăștiați spre instrument în atmosferă în spatele penei. Ei trec apoi pana în linie dreaptă și sunt detectați de instrument[6].

În atmosferă, lumina soarelui este împrăștiată de către moleculele din aer(Rayleigh) și de particulele de aerosoli și picăturile de apă condensate în nori.

Camerele digitale, sensibile la anumite regiuni ale spectrului de ultraviolete, au fost utilizate în ultimii ani pentru a măsura cantitatea de emisii de dioxid de sulf. Instrumentele folosesc absorbția selectivă de către razele UV a moleculelor de dioxid de sulf (SO₂) pentru a determina nivelul concentrației [7].

Există multe avantaje legate de monitorizare utilizând această tehnică, dar nu au fost investigate amănunțit elementele legate de acuratețe și limitările pe care le implică. Efectele unor parametri de control ai utilizatorului, cum ar fi durata de expunere a imaginii, diametrul diafragmei lentilelor, frecvența de calibrare a celei de imagine, și utilizarea uneia sau doua benzi de filtru, nu au fost încă abordate. Pentru a clarifica consecințele utilizării metodologiei și a stabili acuratețea, au fost efectuate măsurători atât pe teren cât și în laboratoare. Imaginile au fost colectate de celulele de calibrare în diverse condiții aflate sub observație. Concluziile vin în sprijinul îmbunătățirii modului de colectare a acestor imagini. Rezultatele indică faptul că răspunsurile celulelor de calibrare sunt constant fiabile sub 1500 ppm m, dar aceste răspunsuri sunt puternic influențate de schimbările condițiilor de lumină. Durata de expunere care poate produce un număr maxim de imagini digitale de peste 32500 poate reduce zgomotul în imaginea până. Rezultatele măsurătorilor cantităților de dioxid de sulf obținute de la o centrală pe cărbune au fost comparate cu rezultate de măsurători directe și s-a constatat faptul că precizia rezultatelor obținute cu camerele UV este aceeași cu cea a metodelor de măsurare standard [72].

Măsurătorile emisiilor de SO₂ sunt o componentă importantă a monitorizării procesului vulcanic, oferind informații despre forțele de conducere din spatele erupțiilor. Foarte mulți ani, în țările cu activitate vulcanică, au fost utilizate tehnici de monitorizare la distanță pentru a investiga degazarea și a deduce modificările ce apar. Metodele standard de monitorizare la distanță cu radiație UV au utilizat corelația și spectrometria de absorbție optică diferențială. Imagistica dioxidului de sulf a fost realizată pe penele de coș, iar în ultimii ani, sisteme de imagistica UV au fost utilizate și în cercetările științifice. Aceste instrumente au limitări de ordin spectrometric, dar sunt relativ ieftine, și oferă informații fără precedent legate de dinamica penelor și nivelul emisiilor la rezoluții temporale înalte.

3.3.1 Camera IR



Figura 3.13 Camera IR în campania de măsurători de la Halânga

O cameră în infraroșu este un dispozitiv care detectează fotonii din domeniul IR și îi convertește într-un semnal electronic, care este apoi procesat pentru a produce o imagine termică pe monitorul video.

Căldura detectată de o cameră în IR poate fi precis cuantificată sau măsurată, permițându-se nu numai monitorizarea performanței tehnice ci identificarea și evaluarea severității relative a problemelor legate de căldură.

Corpurile încălzite prin consum de căldură emit în jurul lor energie radiantă sub formă de unde electromagnetice de diferite lungimi de undă. Această energie este denumită radiație termică și se datorește stării de excitație termică a moleculelor și atomilor corpurilor care o emit[91].

Distribuția electronilor în orbitali se face respectând principiul minimului de energie (electronii ocupă mai întâi pozițiile caracterizate de energia cea mai joasă). Starea fundamentală definește toate proprietățile fizico-chimice ale unui element, inclusiv proprietățile structurale și cele legate de reactivitatea sa chimică. Regulile de mai sus pot fi aplicate și moleculelor, indiferent de complexitatea lor, de la cele simple biatomice, de tipul hidrogenului și oxigenului molecular, la macromolecule ca ADN. Orbitalii moleculari se găsesc la molecule biatomice, triatomice, poliatomice. Atomii sau moleculele care absorb fotoni se transformă într-o altă stare diferită față de cea fundamentală, numită stare activată (stare de excitație), în starea activată se distrug legăturile covalente ale moleculelor. Starea activată poate fi provocată de acțiunea razelor X, gama, UV. Lumina vizibilă poate activa unele molecule, dar nu le poate rupe.

Radiația IR nu produce o excitație a moleculelor, dar poate cauza vibrația legăturilor moleculare. Reactivitatea moleculelor activate este diferită față de cea a

moleculelor din stare fundamentală. O moleculă activată poate să revină la starea fundamentală sau se poate transfera la alte particule prin interacțiunea cu alți atomi sau molecule din sistem. Radiațiile infraroșii au energie mică comparativ cu radiațiile ultraviolete, deci insuficientă pentru ruperea legăturilor covalente. Un atom sau o moleculă pot absorbi numai fotoni care transportă o anumită cantitate de energie, astfel configurația de electroni poate atinge o nouă aranjare, corespunzătoare unui nou nivel de energie permis pentru acea moleculă (salt cuantic). Nivelele de energie sunt discontinue, astfel numai unele salturi cuantice pentru o moleculă specifică sunt permise, în acord cu structura și configurația sa electronică. Starea moleculară de „excitare” se obține atunci când un electron din învelișul exterior poate „sări” la un orbital asociat cu un nivel energetic superior. Razele X-, Y- și UV au suficientă energie, putând chiar distruge legăturile covalente (ex. cazul formării O₂ din O₃ datorită absorbției UV-B și UV-C). Lumina vizibilă poate excita unele molecule, dar nu poate rupe legăturile [78].

În stare excitată moleculele au alte proprietăți față de cele în stare fundamentală. În consecință, reactivitatea moleculelor poate fi semnificativ modificată. După absorbirea energiei transportate de fotoni, o moleculă tinde să revină rapid la un nivel de energie inferior. Astfel, procesele de „relaxare” permit unei molecule să revină la nivelul său minim de energie permis. Procesul poate readuce pur și simplu o moleculă la starea sa fundamentală sau poate conduce la formarea unor noi molecule, prin interacțiunile dintre aceste molecule excitate și/sau reacțiile cu alți atomi/molecule prezente în sistem. Dacă o moleculă revine pur și simplu la starea sa fundamentală nu suferă transformări chimice, surplusul de energie disipându-se în două moduri diferite: prin procese cu emisie și fără emisie de radiație. În primul caz, energia cinetică a moleculelor de obicei crește și cauzează „ciocniri” între molecule, ducând la încălzirea totală a sistemului prin cedarea energiei.

Când molecula disipează surplusul de energie, de obicei emite radiații la lungimi de undă mai mari. De exemplu, absorbția unui foton poate determina „saltul” unui electron la un orbital cu energie mai mare, fără a trece prin diferite nivele energetice intermediare ale moleculei. Revenirea la starea fundamentală se realizează prin trecerea electronului prin toate nivelele energetice intermediare și prin emiterea de fotoni cu energie egală cu diferența dintre două nivele succesive. Absorbția unei radiații este un fenomen posibil. Alți factori determină dacă și cum o moleculă va absorbi o radiație incidentă care este în mod potențial absorbabilă. Radiațiile infraroșii au energie mică comparativ cu cele ultraviolete, insuficientă pentru ruperea legăturilor covalente. Radiațiile IR pot cauza salturi cuantice ce pot corespunde, în baza lungimii lor de undă, vibrației unei legături specifice dintr-o moleculă. Aceste molecule pot fi identificate sau caracterizate pe baza lungimilor de undă IR absorbite, așa cum este cazul spectrului infraroșu al moleculei.

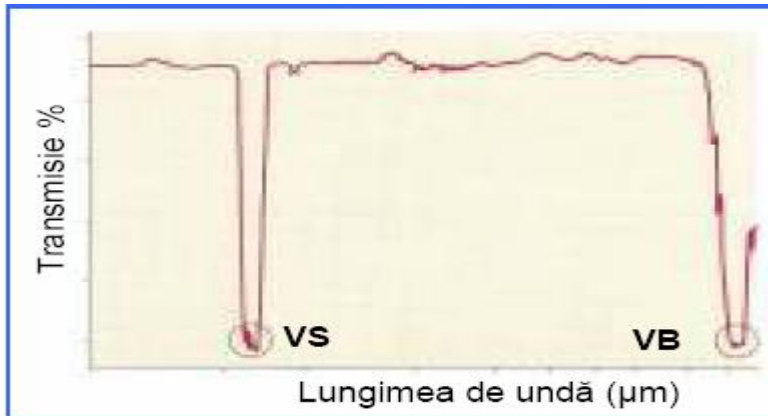


Figura 3.14 Spectrul infrarosu al dioxidului de carbon[77]

Spectrul de absorbție IR este specific unei molecule și poate fi considerat o „amprentă” a moleculei studiate. În figura 3.14 este afișat spectrul dioxidului de carbon. Acesta este detectat cu ajutorul unui spectrometru IR. Radiația emisă de un filament incandescent traversează proba depusă într-un recipient transparent cu grosime definită. Dacă radiațiile IR emise au lungimea de undă corectă (poate fi variată de către analist într-un anumit interval) se va induce un salt cuantic de energie în moleculele probei. Absorbția și reemisia radiațiilor IR pot explica modul în care dioxidul de carbon contribuie la efectul de seră. Moleculele CO₂ absorb energia radiațiilor solare cu lungimi de undă specifice (directă și reflectată). Se înregistrează un salt cuantic la un nivel energetic superior asociat cu vibrația moleculei. Apoi molecula tinde spre starea fundamentală, emițând radiații IR cu lungimi de undă mai lungi comparativ cu cele incidente. O parte dintre acestea se reîntorc spre suprafața terestră, fiind absorbite parțial și provocând creșterea temperaturii. Dioxidul de carbon și vaporii de apă (H₂O absoarbe radiații IR cu $\lambda=2.5 \mu\text{m}$ și $6.5 \mu\text{m}$) sunt principalele gaze cu efect de seră, dar orice moleculă capabilă să absoarbă în IR este un potențial gaz cu efect de seră; de ex. metanul (CH₄), protoxidul de azot (N₂O), ozonul (O₃) și clorfluorcarburile (ex. CFC₃). Moleculele de azot (N₂) și oxigen (O₂) nu sunt gaze cu efect de seră, deoarece pot vibra, sunt simetrice și neutre din punct de vedere electric, fiind formate din atomi identici și cu aceeași electronegativitate și vibrația lor nu induce variații ale momentului de dipol, deci nu absorb radiații IR. Microundele ($\lambda=105 \mu\text{m}$) au o energie mult mai joasă în raport cu radiațiile UV și IR și nu permit nici ruperea și nici vibrația legăturii, dar această energie este suficientă pentru a determina rotația moleculelor în interacțiunile cu lungimile de undă specifice. În cuptoarele cu microunde obișnuite λ este stabilită în vederea creșterii vitezei de rotație a moleculelor de apă din mâncare. Rotirea rapidă a acestor molecule, una contra celeilalte determină fenomenul de fricțiune, care cauzează căldura ce coace alimentele. Interacțiunile dintre radiațiile electromagnetice și materie sunt importante în vederea menținerii condițiilor ideale pentru viața pe planeta noastră. Numai unele lungimi de undă specifice pot fi absorbite de către molecule. Aceste interacțiuni oferă diferite posibilități pentru analiza structurii atomilor și a moleculelor (ex. metode spectrometrice)[77].

Compoziția radiației termice depinde de temperatura la care se află corpul radiant. Pe măsură ce temperatura acestuia crește el emite radiații cu lungimi de

undă din ce în ce mai mici respectiv cu frecvențe din ce în ce mai mari. Spectrul radiației termice este constituit din trei domenii: domeniul IR, domeniul vizibil și domeniul UV.

Imaginile IR lasă să vezi căldura și cum este ea distribuită. Camera IR are lentile ca și o cameră cu lumină vizibilă, dar în acest caz lentilele focusează undele din energia infraroșie pe un șir de senzori infraroșii. Mii de senzori ai șirului convertesc energia infraroșie în semnale electrice care sunt convertite ulterior în imagine.

În imaginile infraroșii diferite culori sunt folosite pentru a reprezenta diferite temperaturi. Poți vedea orice temperatură ce culoare reprezintă folosind scala culoare-temperatură, arătată de obicei în dreapta imaginii.

Camerele IR sunt folosite pentru a monitoriza norii și cerul la rezoluție temporală și spațială mare. Norii au un rol important în stabilirea vremii și a climatului. Proprietăți ca, acoperirea cu nori, temperatura radiativă a norului și temperatura radiativă a cerului senin, au un impact în balanța radiativă a pământului. Camerele termice în infraroșu sunt o tehnologie în curs de dezvoltare și au capacitatea de a detecta condițiile de înorare la mare rezoluție, fără nici o modificare a performanțelor în timpul zilei sau noaptea[88].

O cameră de fotografiat ce lucrează în regiunea infraroșie între 7-14 μm a fost dezvoltată pentru a detecta și cuantifica cenușa vulcanică[85]. Camera folosește radiația infraroșie pasivă măsurată cu până la cinci canale spectrale pentru a determina cenușa din alți absorbânți atmosferici(moleculele de apă), iar un model microfizic de cenușă este folosit pentru a determina distribuția de mărime a particulelor, adâncimea optică infraroșie și masa totală de particule fine. Cenușa vulcanică este majoritar compusă din dioxid de siliciu împreună cu alte minerale în cantități mici.

Radianța penei poate fi considerată ca fiind radiația emisă și radiația din atmosferă, ce este atenuată prin trecerea prin pană. Împrăștierea este ignorată. Radianța de fundal se referă la radianța de la cer, din spatele penei; radianța din față se referă la radianța emanată din atmosferă dintre pană și cameră. Radianța este mărimea fizică egală cu raportul dintre fluxul de lumină emis de suprafața unui corp și aria unei suprafețe. Camera în infraroșu înregistrează imagini calibrate radiometric ale radianței cerului pentru lungimi de undă între 8 și 14 μm din care distribuția spațială a norului poate fi determinată[84].

Instrumentele de teledetecție au fost folosite pentru a identifica, urmări și cuantifica constituenții atmosferici din spațiu de aproape 30 de ani. Datele s-au dovedit extrem de utile pentru detectarea cenușii periculoase și a gazelor (în principal SO₂) emise de vulcani și care au potențialul de a intersecta rutele aeriene la nivel mondial[62].

Un instrument nou de măsurare a gazelor atmosferice, utilizând tehnologia în infraroșu a fost testat precum și un sistem de măsurare a concentrațiilor, folosind imaginile multispectrale[75]. Camera IR a fost folosită la două locații industriale și la doi vulcani unde până de SO₂ a fost prezentă. Instrumentul a dovedit încredere și a fost capabil să detecteze SO₂ în prezența vaporilor de apă. Norii și particulele de cenușă ce interferează cu măsurătorile de SO₂ au putut fi detectate folosind imaginea multispectrală și eliminate din analiză.

Este demonstrat că detectarea și regăsirea concentrațiilor de gaze SO₂ de la surse naturale și provocate de om poate fi realizat prin utilizarea imaginii termice în infraroșu. Descoperirile recente în tehnologia de detector nerăcit oferă posibilitatea de folosire termică relativ ieftină(7-14μm) a dispozitivelor de imagistică ca instrumente pentru studierea și cuantificarea comportamentului gazelor periculoase și a particulelor. O cameră termică ce operează la sol și are 5 canale spectrale la

lungimi de undă centrale de 8,6 , 10, 11,12 și un canal de bandă largă(7-14 μm)a fost testată la câțiva vulcani și la doua locații industriale unde SO₂ a fost principalul constituent al penei de poluant.

3.3.1.1 Camera de formare a imaginilor infraroșii termice

Camerele de formare a imaginilor termice bazate pe tehnologie de rezoluție cu microbolometru nerăcit sunt disponibile în comerț cu sensibilitate de temperatură de 50 mK (8–12 μm), dimensiuni de rezoluție de 320x240 pixeli, sisteme optice F1.0 și operare de 60 Hz. În principiu, o cameră de acest fel poate dobândi imagini care indică schimbări de temperatură mai mici de 0,1 K, la o rată de 10 cadre pe secundă. Ratele cadrelor atât de mari ca aceasta nu sunt neapărat necesare, dar se poate dori obținerea mai multor imagini pe minut, din cauza dinamicii rapide a coloanelor de cenușă vulcanică care erup. Este dificil să se atingă rate foarte înalte de cadre (30–60 Hz) de la aceste camere din cauza prezenței zgomotului ($1/f$, fluctuațiile de temperatură de fundal și internă și zgomot Johnson), care poate fi atenuat prin integrarea cadrelor.

Alți factori pot limita atingerea ratelor ridicate de captare ale imaginilor; de exemplu, extragerea datelor cadrului imaginii impune rapid electronica rapidă și un microprocesor bun, precum și comunicații hardware și software. Camera care s-a folosit încorporează și selecția lungimilor de undă (filtre), iar acest lucru adaugă întârzieri în captarea imaginii. Designul prototipului este limitat la ratele captării imaginii de 1 la fiecare 5–6 minute, care sunt prea lente pentru studierea dinamicii dezvoltării coloanei de cenușă, dar adecvate pentru determinarea microfizicii cenușii. Rata de date lentă a prototipului instrumentului este un produs al sistemului de transfer de date și poate fi ușor depășită cu o mai bună comunicare electronică.



Figura 3.15. Roata cu filtru Cyclops și filtre de interferență.

Roata cu filtru Cyclops din figura 3.15 are 5 filtre cu lungimi de undă centrale la 7,34, 8,55, 10,11, 12 μm. Pentru operațiunile de la sol, filtrul de 7,34 μm a fost înlocuit cu un filtru de bandă largă care acoperă 7-14 μm.

Camerele termice IR cu diferență de temperatură echivalentă de zgomot de 50 mK (NEΔT) sunt disponibile în comerț. Camerele comerciale autonome (COTS) sunt dotate cu un filtru unic, în bandă largă, care acoperă regiunea de lungimi de undă IR de la 7 μm la aproximativ 14 μm, care este considerată o regiune de transparență atmosferică relativ ridicată. Camera a fost concepută să măsoare atât

gazele atmosferice (în special SO₂) și cenușa vulcanică. Pentru utilizarea în detectarea și cuantificarea particulelor la temperaturi tipice atmosferice (230 – 300 K), au fost necesare mai multe modificări ale camerei COTS.

3.3.1.2 Filtrare

Selecția spectrală a radiațiilor în benzi înguste (0,5–1,0 μm) se realizează prin plasarea unei roți de filtrare între sistemul optic anterior și detector. În figura 3.15 se furnizează o fotografie care arată roata de filtrare proiectată să se potrivească pe camera Cyclops. Filtrele sunt atent selectate pentru a se potrivi cu detecția optimă a gazului SO₂ și a particulelor de cenușă vulcanică, cele două componente majore ale emisiilor vulcanice periculoase. Figura 3.16 indică spectrul de absorbție al unor nori de fază micști, comparați cu un spectru de cenușă determinat de senzorul prin satelit al aparatului de sondaj electroacustic în infraroșu atmosferic (AIRS) Se afișează numai regiunea de „fereastră” între 800-1100 cm⁻¹ deoarece aceasta regiune este cel mai puternic afectată de cenușa de silicat și norii meteorologici. Spectrul de absorbție al cenușii crește odată cu creșterea numărului de undă, în timp ce spectrul norului de fază mixtă nu prezintă nici o schimbare sau, probabil, o scădere ușoară cu numărul de undă. În cazul norilor de gheață, panta spectrului de absorbție este sensibil la dimensiunea particulelor și prezintă o scădere marcată odată cu creșterea numărului de undă. Aceste efecte diferite permit identificarea cu ușurință a cenușii din vaporii de apă[85].

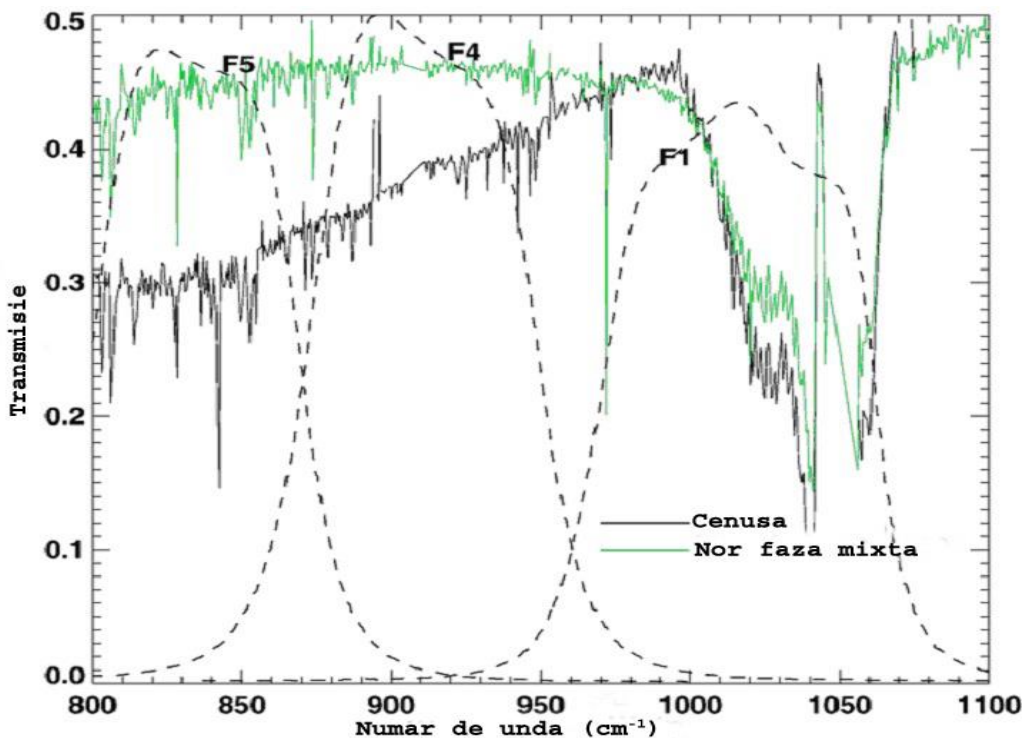


Figura 3.16. Spectrele de transmisie atmosferică pentru un nor de cenușă și un nor de fază mixtă măsurate de la senzor prin satelit al AIRS [85]

Se observă scăderea caracteristică în absorbție odată cu numărul de undă de la aproximativ 800 cm⁻¹ la aproximativ 1000 cm⁻¹ pentru cenușa vulcanică și panta aproape plată a norului de fază mixtă. De asemenea, se indică funcțiile de răspuns ale filtrului (unități arbitrare) pentru trei dintre canalele Cyclops.

Cel mai important aspect în designul Cyclops a fost alegerea numărului de filtre necesar, lungimile de undă centrale ale filtrului și lățimile de filtrare. Filtrele de interferență care operează în banda de lungimi de undă 7–14 μm sunt deja disponibile. S-au efectuat o serie de calcule pentru a găsi lungimi de undă și lățimi de filtrare pentru optimizarea semnalului din cauza cenușii. Principala constrângere a lățimilor filtrelor este raportul semnal-zgomot (SNR) (filtre mai largi oferă mai multă energie și, prin urmare, SNR mai bun), dar filtrele nu trebuie să fie prea largi astfel încât filtrele adiacente să se suprapună introducând corelațiile nedorite între canale. Simulările de transfer radiativ sugerează că pentru a obține NEΔT de <100 mK la 290 K, filtrele trebuie să fie late de ≈1,0 μm presupunând că se poate folosi integrarea timpului suplimentar (calcularea medie a cadrelor de 30–60). Lungimi de undă centrale au fost determinate prin simulări de transfer radiativ. Pentru canalul de SO₂, alegerea a fost condusă prin asigurarea faptului că s-a inclus întreaga caracteristică de absorbție a SO₂ de 8,6 μm. Construirea filtrului mai larg permite prea multă interferență de la vaporii de apă, iar compromisul între NEΔT superior și reducerea efectelor vaporilor de apă a constituit un factor restrictiv în alegerea filtrului de SO₂. Numărul de filtre necesar (3 pentru cenușă și 1 pentru SO₂) a fost, de asemenea, determinat prin calcule de transfer radiativ. Filtrul de bandă largă a fost considerat util, pe baza NEΔT mai scăzut și utilității acestui canal pentru formarea imaginilor termice generale. Tabelul 3.1 prezintă canalele (sau benzile înguste) alese pentru Cyclops pentru detectarea cenușii și SO₂ de la sol. Nedispunând de măsurători în infraroșu cu rezoluție spectrală înaltă, a norilor de cenușă, din detectarea cu senzori de la sol, selecția filtrelor a fost influențată de măsurătorile prin satelit și de transferul radiativ și compensările tehnice menționate anterior. Filtrul de 10 μm (F1 în Figura 3.17) include o caracteristică de absorbție puternică din cauza O₃, pe care nu o vedem de la un senzor de sol. Filtrele F4 și F5 captează schimbarea de absorbție între cenușă și nori meteorologici, Măsurătorile de la un instrument de sol, vizualizează norul de cenușă cu fundal rece, deci variația absorbției cu un număr de undă pentru cenușă va fi opusă celei indicate de măsurătorile din spațiu [93].

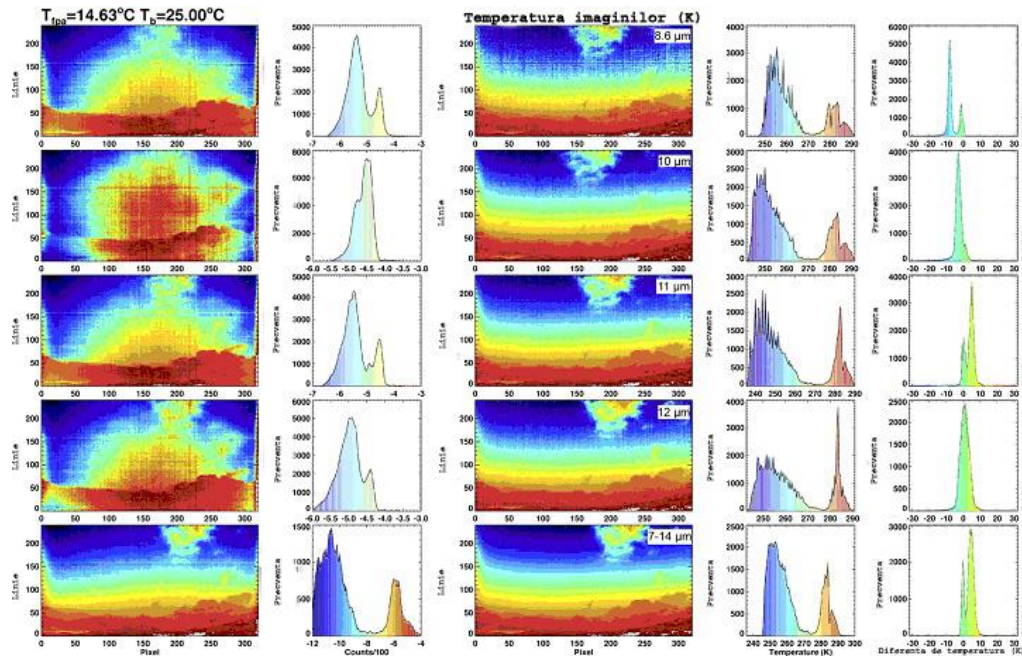


Figura 3.17 Imagini rapide cu Cyclops[86]

Prima coloană din figura 3.17 este o imagine brută cu numere digitale reduse la scară de la cel mai scăzut în albastru la cel mai înalt în roșu. A doua coloană: histograme de imagine pentru date brute. Coloana a treia: imagini de temperatură de luminozitate calibrate (în grade Kelvin), cu valorile cele mai reci în albastru și valorile cele mai calde în roșu. Coloana a patra: histograme de temperatură de luminozitate pentru datele calibrate. Toate cele cinci canale sunt prezentate, începând de sus: 8,55, 10, 11, 12 și 7-14 μm . Coloana a cincea: histograme de diferență de temperatură de luminozitate pentru datele calibrate. Combinațiile indicate, începând din partea de sus, sunt: $T_{11}-T_{12}$, $T_{10}-T_{12}$, $T_{10}-T_{11}$, $T_{8.6}-T_{12}$ și $T_{8.6}-T_{10}$.

Panourile din stânga indică imaginile brute și histogramele lor respective, iar cele din partea dreaptă indică imaginile BT după calibrare și conversie, cu un set de histograme de diferență de temperatură. Toate canalele sunt afișate. De sus în jos acestea sunt: 8,6, 10, 11, 12 μm și în bandă largă (7-14 μm). Se poate observa că imaginile (de temperatură) calibrate conțin în prezent informații cu multe dintre artefactele camerei (distorsiuni optice, neomogenități de pixeli etc.) minimizezate. Imaginile indică, de asemenea, tendința generală a radiației în scădere (la toate lungimile de undă), cu un unghi de elevație a vizualizării crescut. Rata de scădere odată cu unghiul de elevație nu este aceeași la toate lungimile de undă, iar atmosfera induce un efect de absorbție diferențial care depinde de unghiul de vizualizare. Petele semicirculare care apar în centrul imaginilor brute filtrate sunt cauzate de radiațiile nedorite de pe obiective și carcasa Cyclops. De asemenea, există un artefact pe marginea din stânga a datelor necalibrate, care a fost în mare parte eliminat în datele calibrate. Histogramele imaginilor indică faptul că intervalul de temperaturi măsurate în această scenă este destul de vast, de la ~ 250 la ~ 300 K. Histogramele de diferență se folosesc pentru a identifica caracteristici în formarea de imagini, în special diferențele de 11-12 μm fiind utilizate pentru a identifica cenușa

de cerul senin și noros. În cele din urmă, se poate observa că zgomotul de imagine este mai mare de 8,6 μm și cel mai mic în imaginea în bandă largă (panoul cel mai mic)[93].

Aceste observații generale ne conduc la două concluzii foarte importante cu privire la prelucrarea ulterioară a datelor Cyclops. Datele brute, necalibrate nu au practic nici o valoare în identificarea gazelor sau particulelor din aceste imagini IR filtrate termice. Mare parte din informațiile utile sunt conținute în imagini de diferență, prin urmare, reducerea zgomotului și aplicarea unei calibrări coerente sunt fundamentale pentru transformarea datelor în informații.

Vaporii de apă modifică cantitatea de radiații care atinge ținta. Contribuția vaporilor de apă la semnalul de pe detector va depinde de lungimea drumului radiației și de temperatura vaporilor. Aplicarea unei corecții atmosferice este esențială pentru identificarea corectă a cenușii în imagini. Mai mult, corecția trebuie să fie aplicată cu o dependență de unghiul de vizualizare și, de preferință, pe bază, de la pixel la pixel.[45]

Camera disponibilă în comerț FLIR Systems, Inc., Thermovision A40 conține o rezoluție cu microbolometru nerăcită de 320 x 240 pixeli și detectează radiații în intervalul de 7,5-13 μm. Are o diferență de temperatură echivalentă de zgomot de 0,1 K la temperatura camerei și un interval de ieșire de 233-473 K. Acest interval este suficient pentru a observa cerul în majoritatea lunilor, în latitudinile mijlocii. Camera se stabilizează la schimbările în temperatura ambientală prin coborârea diafragmei (a cărei temperatură se măsoară cu un senzor intern) și calibrarea fiecărui răspuns de pixel.

3.3.1.3 Teoria radiației și modelarea

La vizualizarea atmosferei, în regiunea 7,5-13 μm există mai mulți absorbanți/emisători care radiază: apă lichidă sau înghețată sub formă de nori, vapori de apă și ozon. Profilul de temperatură atmosferică T_z determină temperatura de emisie a corpului negru a tuturor compușilor și variază în timp și spațiu[44]. Ecuația generală pentru radianță (I_λ), măsurată la sol dintr-un compus care emite la o lungime de undă dată este

$$I_\lambda = \int_0^{x(0,\infty)} B(T_z) \sec \theta e^{-x(0,z)} dx \quad (3.24)$$

unde $B(T_z)$ este radianța corpului negru la o altitudine z din atmosferă, θ este unghiul dintre linia de vedere și zenit, iar $x(0, z)$ este grosimea optică verticală între z și sol. În ceea ce privește z , dx poate fi scris ca

$$dx = -k_\lambda \rho_z dz \quad (3.25)$$

unde k_λ reprezintă coeficientul de absorbție, ρ_z densitatea. Din aceste expresii se poate vedea că radianța observată va crește pe măsură ce $\sec \theta$ (și, prin urmare, unghiul de vizualizare spre zenit) crește. Amplitudinea și forma exactă a acestei încălziri odată cu unghiul va fi, totuși, dependentă de forma T_z și ρ_z pentru fiecare dintre compușii absorbanți/emitenți.

Pentru a calcula temperaturile așteptate pentru diferite scene de cer, s-a folosit modelul de transfer radiativ atmosferic discret-ordonat Santa Barbara (SBDART)[84]. Acesta este un model conceput pentru studiile radiațiilor atmosferice

senine și noroase și include toate procesele importante care au loc în infraroșu. Calculele sunt efectuate pe un model de transfer radiativ discret ordonat folosind date LOWTRAN 7 (model și cod de radianță și transmitanță atmosferică cu rezoluție scăzută)[84] pentru absorbție / emisie atmosferică pe cer senin. Condițiile noroase sunt tratate prin asumarea dispersiei Mie prin picături de apă lichidă, ale căror dimensiuni urmează o distribuție gamma cu o rază efectivă implicită de 8 μm. Radianțele norilor în fereastra atmosferică s-au dovedit a fi insensibile la schimbările în raza efectivă a picăturilor, variind cu mai puțin de 0,04% atunci când valoarea s-a dublat sau s-a redus la jumătate. Altitudinea norilor, grosimea fizică și adâncimea optică pot fi variate și, în scopul acestui studiu, am presupus că norii conțin doar apă în stare lichidă. O selecție a profilurilor atmosferice de model a fost inclusă și proiectată să fie tipică pentru vara de latitudine mijlocie tropicală, iarna de latitudine mijlocie și climate subarctice de vară. SBDART are o eroare maximă de radianță de undă lungă de 3%, asociate cu cerurile reci, uscate și senine, care se traduc printr-o eroare de temperatură de aproximativ 1,5 K [105].

3.3.1.4 Detectarea IR la sol

Camera Cyclops funcționează dintr-o poziție fixă pe teren și vizualizează ținta de la o oarecare distanță (până la 10 km). Locația actuală a camerei în raport cu ținta este selectabilă. Cu toate acestea, anumite considerații sunt necesare în cazul în care sistemul trebuie să funcționeze util. De exemplu, plasarea camerei astfel încât să aibă vedere fixă a vulcanului aflat în studiu poate fi logică, astfel încât coloanele de cenușă și norii să poată fi simultan observate.

Alternativ, poate fi de preferat ca pe aeroporturi camera să fie amplasată astfel încât să vizualizeze vertical în sus sau la un unghi de elevație mare spre vulcanul din apropiere. Ar putea include obstacole fixe în câmpul vizual (copaci, clădiri, teren etc.), precum și probleme efemere ca urmare a efectelor norilor meteorologici sau vaporilor de apă. Multiplele orientări de vizualizare posibile, fac dificilă furnizarea unei singure formule sau algoritmi, ce țin cont de toate potențialele efecte de vizualizare adverse.

Efectul obstacolelor fixe (copaci, clădiri, teren etc.) poate fi reprezentat prin luarea unei imagini de referință într-o zi senină. Se poate genera o mască din această imagine, iar apoi se poate folosi pentru a evidenția sau delimita porțiunile imaginii care prezintă obstacole.

Efectele lungimii traiectoriei a vaporilor de apă pot fi contabilizate dacă distanța până la țintă (de exemplu, coloana de erupție vulcanică sau nor) și orientarea camerei sunt cunoscute. O estimare a structurii vaporilor de apă din atmosfera locală trebuie să fie, de asemenea, cunoscută, putând fi, de obicei, obținută dintr-un radiosondaj din apropiere. Orientarea camerei poate fi găsită din considerente geometrice simple. Figura 3.18 indică o schemă a geometriei formării de imagini a unei părți a coloanei dintr-o poziție fixă la sol. Sistemul de coordonate adoptat este cel cartezian cu partea dominantă a coloanei plasată la $y = 0$, camera plasată la $x = 0$, $y = L$, $z = 0$, coordonatele x și y reprezentând axele horizontale, iar z este axa verticală. Camera vede coloana de la o distanță R , măsurată de la centrul detectorului până la partea coloanei cea mai apropiată de cameră, și la un unghi de elevație θ_n și unghi azimutal Φ_n , care variază în funcție de numărul n de pixeli al camerei. În acest sistem de coordonate, linia camerei C_l și numerele C_c ale coloanei sunt legate de elevația camerei și unghiurile azimutale, prin:

$$C_1 = \frac{L}{s_n} (\cos \phi_n \operatorname{tg} \theta_n - \operatorname{tg} \zeta) \tag{3.34}$$

$$C_c = \frac{N_c}{2} + \frac{L}{s_n} \operatorname{tg} \phi_n \tag{3.35}$$

$$n = C_c + N_c (C_1 - 1) \tag{3.36}$$

unde L este distanța de la coloană măsurată în planul x - y ($z = 0$), ζ este elevația camerei măsurată de la nivelul solului (înălțimea deasupra nivelului mediu al mării) la prima linie a imaginii, s_n este dimensiunea pixelului n al imaginii, iar imaginea are N_c coloane cu N_l linii (320x240 în configurarea curentă). Camera este orientată astfel încât unghiul azimutal de $\phi_n = 0$ corespunde cu centrul imaginii sau numărul coloanei $N_c/2$. Numerele de pixeli sunt numărate din stânga jos a imaginii cu linia 1, coloana 1 corespunzând numărului de pixel 1, iar ultima coloană a liniei superioare corespunzând numărului de pixel N_c și N_l .

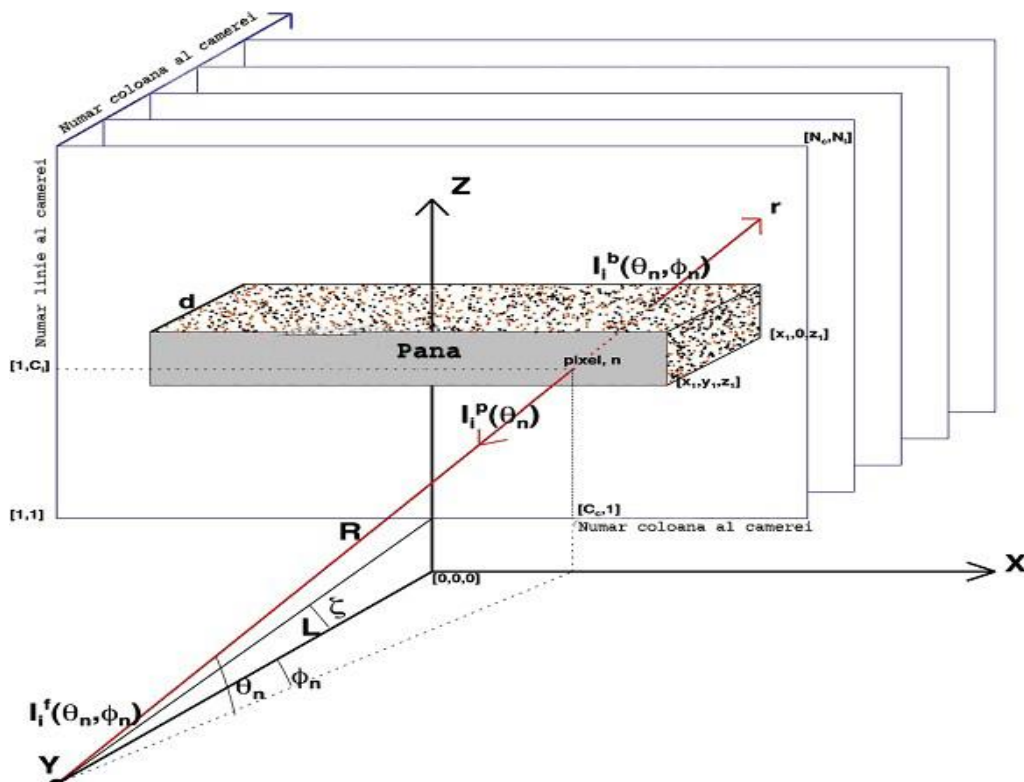


Figura 3.18. Geometria de măsurare pentru vizualizare cu o cameră termică a unui nor vulcanic „ideal” paralel cu planul. Camera se află în poziția Y și toate simbolurile sunt definite în text[85]

Mărimea pixelului variază în funcție de numărul liniei și al coloanei și poate fi determinată de:

$$s_{l,c} = \frac{L}{N_{l,c}} \operatorname{tg} \left(\frac{\psi_{l,c}}{2} \right) \quad (3.37)$$

$$\psi_{l,c} = 2 \operatorname{tg}^{-1} \left(\frac{N_{l,c} \chi}{2F} \right) \quad (3.38)$$

unde F este lungimea focală a camerei, χ este pasul pixelului pe cipul detector ($\sim 45 \mu\text{m}$), iar $\psi_{l,c}$ este câmpul vizual al rezoluției detectorului cu microbolometru pe verticală (ψ_l) sau orizontală (ψ_c).

3.3.1.5 Alarma pe bază de histogramă

Figura 3.19 prezintă o imagine cu diferență de temperatură de luminozitate (11–12 μm), obținută în condiții de vizualizare fără cenușă cu o elevație de 20° deasupra orizontului. Scara de culori din această imagine indică un interval de diferență de temperatură de luminozitate de la –15 K la +10 K, cu pixeli de culoare roșie având diferența de temperatură cea mai pozitivă. Pentru a evidenția regiunea în care poate exista cea mai mare parte a ambiguității, o scară de gri indicând temperatura de la – 0,5 K la +0,5 K este inclusă în scara de culori principală. Astfel, pixeli de culoare gri din imagine de diferență de temperatură pot fi considerați ca fiind marginali, în ceea ce privește detectarea.[92]

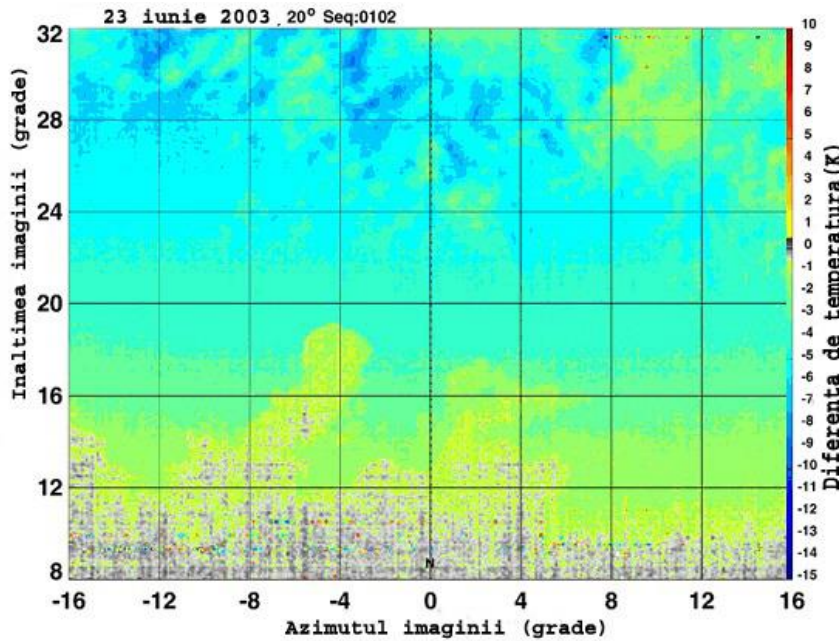


Figura 3.19. Imaginea Cyclops de diferență de temperatură (11-12 μm) captată într-o locație (stânca Suicide) din Saipan din Insulele Mariane de Nord[84]

Elevația camerei a fost de 20° , iar camera vizualiza spre nord (0° azimut). O scară de culori este indicată pe marginea extremă din dreapta a imaginii, pixelii de culoare gri au valori de la $-0,5$ la $+0,5$ K. Aproape toți pixelii au diferențe de temperatură negative, care indică faptul că scena este compusă în întregime din cer senin, vapori de apă și / sau nori meteorologici de fază mixtă.

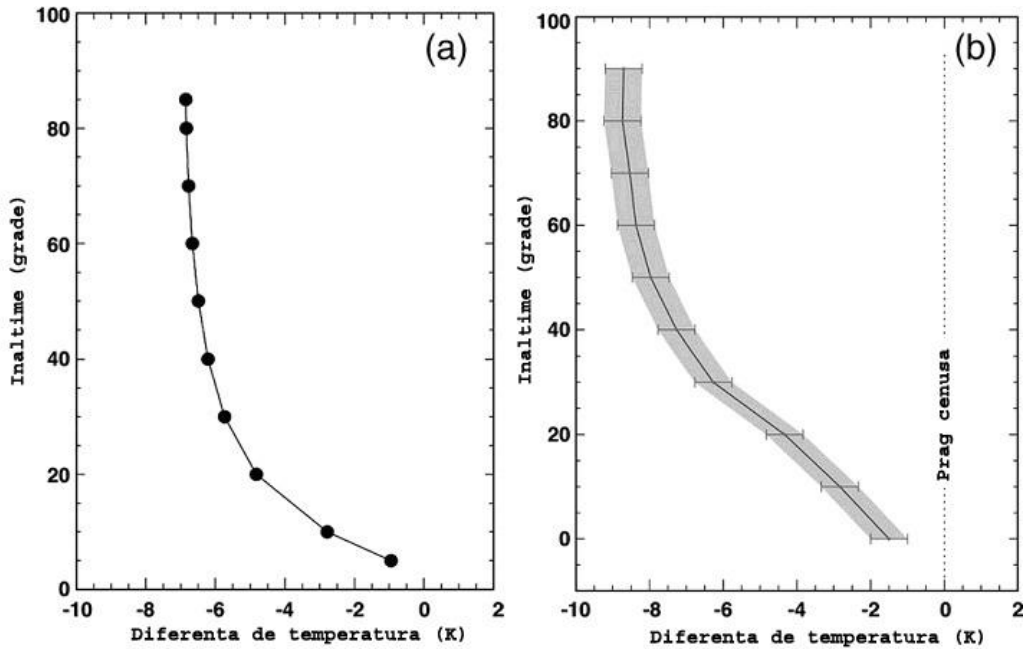


Figura 3.20. (a) Variația unghiului de elevație cu diferența de temperatură (11-12 μm) pentru cer senin determinată de modelarea de transfer radiativ. (b) Variația unghiului de elevație cu diferența de temperatură (11-12 μm) pentru cer senin determinată din datele de imagine Cyclops[84]

În această imagine există câțiva pixeli gri, dar majoritatea pixelilor sunt galbeni, și de la verde la albastru indicând diferențele de temperaturi negative și, prin urmare, condițiile normale (adică, cer senin sau nori de apă/gheață).

Diferența de temperatură scade odată cu unghiul de elevație în toate cazurile studiate. O consecință a acestui comportament este faptul că nu este posibil să se stabilească un prag constant pentru a decide dacă imaginile Cyclops conțin pixeli afectați de cenușă[87].

Histograma are vârfuri proeminente la diferențele de temperatură aproximative de -1 K și -5 K, care corespund norilor și, respectiv, cerului senin. În acest caz, vârful cel mai negativ are o coadă care include câțiva pixeli pozitivi. În imaginea corespunzătoare, acești pixeli vizualizează caracteristici care sunt mici la orizont și includ ținte terestre. Aceste „anomalii” sunt dificil de izolat automat și ar putea da naștere *la alarme false* dacă s-a folosit o tehnică simplă de prag de pixeli [84].

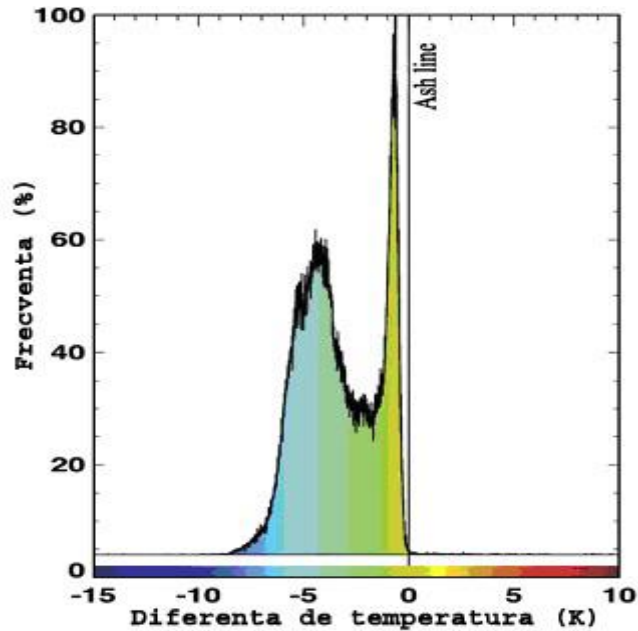


Figura 3.21. Histograma imaginii 2D a diferenței de temperatură (11-12 μm) a datelor indicate. Linia de întrerupere a cenușii este indicată la diferența de temperatură 0K. Lățimea histogramei se datorează unei combinații de efecte de pixeli micști și elevația de vizualizare[84]

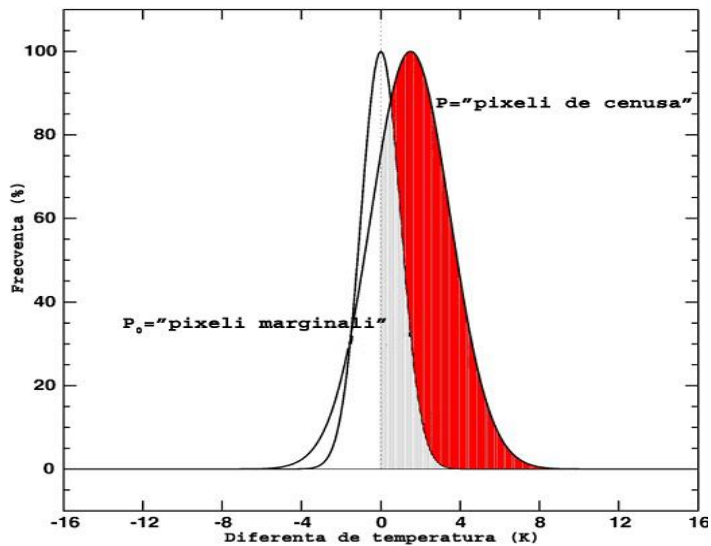


Figura 3.22. Ilustrarea tehnicii gaussiene de prag pentru setarea alarmei de cenușă. Regiunea gri conține pixeli care se încadrează în acoperirea parțială dintre t-Gaussian; regiunea colorată în roșu indică pixelii care se numără ca pixeli afectați de cenușă[84]

3.3.1.6 Fitarea gaussiană

Histograma 2D din figura 3.21 este formată din două vârfuri proeminente cu o răspândire de pixeli în jurul valorii acestor vârfuri. Pentru o scenă relativ uniformă, vârful ar fi înalt și anvergura (sau deviația standard a distribuției) ar fi joasă. O alegere naturală a modelării acestui tip de distribuție este distribuția normală sau distribuția gaussiană. Distribuția gaussiană în termeni matematici este:

$$G(\Delta T) = A_0 \exp \left\{ - \left(\frac{\Delta T - \mu_{\Delta T}}{\sigma_{\Delta T}} \right)^2 \right\} \quad (3.39)$$

unde ΔT este diferența de temperatură, $\mu_{\Delta T}$ este diferența medie de temperatură, $\sigma_{\Delta T}$ este abaterea standard, iar A_0 este frecvența maximă, care apare atunci când $\Delta T = \mu_{\Delta T}$. Fiecare dintre vârfuri ($i = 1 \dots n$) în cadrul distribuției de frecvență se presupune că are o medie la $\mu_{\Delta T,i}$ cu o anvergură de $\sigma_{\Delta T,i}$.

Un set de distribuții gaussiene este potrivit pe datele de distribuție după frecvență și sunt determinați parametrii, $A_{0,i}$, $\mu_{\Delta T,i}$ și $\sigma_{\Delta T,i}$. Combinația liniară a acestor distribuții este modelul potrivit pe date.

Curba rezultată din fitare din figura 3.21 este prezentată în figura 3.22. Trei gaussiene au fost utilizate în potrivirea cu parametrii dați de:

Parametru	$i = 1$	$i = 2$	$i = 3$
$A_{0,i}$	0.742	0.249	0.009
$\mu_{\Delta T,i}$	-4.24 K	- 0.84 K	- 0.67 K
$\sigma_{\Delta T,i}$	± 1.49 K	± 0.33 K	± 0.08 K

Tehnica alarmei apare prin setarea unui prag gaussian (t-gaussian), cu o deviație medie și standard derivate din modelare, și compararea acestuia cu potrivirea de date n -gaussian. Se calculează regiunea dintre pixeli delimitată de valoarea medie t-gaussian și regiunea de suprapunere între cele două gaussiene (pragul și datele de potrivire). Această zonă (sau numărul de pixeli) se scade din numărul de pixeli care depășesc valoarea t-gaussiană medie și se află în cadrul datelor de fitare gaussiană[84]. Raporturile alarmei (în %) se calculează din:

$$R_i = \frac{A_{0,i}}{\sum_{j=1}^n A_{0,j}} \left(\frac{P_i - P_{0,i}}{P_i} \right) \quad (3.40)$$

unde $P_{0,i}$ este numărul de pixeli de suprapunere pentru i gaussian, P_i este numărul de pixeli care depășesc media de prag și $A_{0,i}$ sunt maximele pentru potrivirile gaussiene. Pentru a demonstra modul în care funcționează alarma, vom folosi datele obținute din măsurători pe teren efectuate la vulcan Anatahan (NMI) atunci când se vizualizează un nor de cenușă și în diverse locații din jurul insulei Saipan când se vizualizează cer senin și noros (adică, condiții fără cenușă). S-au stabilit pragurile (separările) care depind de cota de vizualizare și acestea corespund mediei t-gaussian. Anvergura sau deviația standard a t-gaussian depinde, de asemenea, de unghiul de elevație. Valorile pentru acestea au fost determinate prin modelarea

transferului radiativ. Valoarea lor exactă depinde de condițiile atmosferice: în principal, de cantitatea de vapori de apă prezentă în atmosferă.

Figura 3.24 prezintă o histogramă obținută în condiții fără cenușă. Există două vârfuri proeminente (la $\Delta T = -4,24$ K și $\Delta T = -0,84$ K) în histogramă și un vârf minor (la $\Delta T = -0,67$ K). Diferența de temperatură în aceasta și în imaginile ulterioare corespunde cu diferențele de temperatură de 11 -12 μm . Se indică o valoare de separare de -1,0 K (linia roșie punctată) și potrivirea gaussiană (folosind $n = 3$) este suprapusă peste date (linia verde). În acest caz, datele sunt reprezentate prin trei gaussiene. Majoritatea pixelilor (> 74%) se încadrează în distribuția gaussiană cu media $\Delta T = -4,24$ K și deviația standard de $\sigma_{\Delta T} = \pm 1,49$ K. T-gaussian pentru această histogramă are o medie de 0 K și o diferență de ± 2 K. Pentru această imagine nu se generează nici un semnal de alarmă.

Algoritmul de alarmă de cenușă automatizat este un instrument de calitate util pentru procesarea rapidă și comunicarea de informații care să arate prezența de cenușă în câmpul vizual al camerei.

3.3.1.7 Analizele microfizice de cenușă

Cenușa vulcanică conține SiO₂ în cantități diferite. S-au făcut câteva măsurători ale dimensiunilor și distribuțiilor de dimensiune ale particulelor de cenușă din aer, iar acestea sugerează că particulele în intervalul 1 -10 μm (raze) sunt comune. Particulele mai mari tind să cadă rapid și cele mai mici se coagulează provocând un efect de cernere atmosferică care lasă majoritatea particulelor într-un interval de dimensiuni influențat la teledetecție, prin utilizarea radiațiilor infraroșii aproape de 10 μm . Particulele mai mici ($r < 1$ μm) și particulele mai mari ($r > 10$ μm) sunt susceptibile de a fi prezente, dar acestea prezintă un interes scăzut aici, unde accentul se pune pe particulele care pot rămâne în atmosferă pentru a prezenta un pericol pentru aviație și o preocupare pentru sănătatea publică. Aceste două observații principale cu privire la cenușa vulcanică, și anume conținutul ridicat de SiO₂ și dimensiunile tipice de particule în intervalul 1-10 μm , sugerează puternic ca regiunea în infraroșu între 8-14 μm este utilă pentru extragerea microfizicii cenușii.

Un model discret de ordonate de transfer radiativ a fost utilizat împreună cu proprietățile optice ale particulelor de silicat pentru a simula temperaturi pentru benzile Cyclops și condițiile de vizualizare. Calculele oferă temperaturi în funcție de raza particulelor (r), adâncimea optică în infraroșu (τ) și unghiul zenital de vizualizare (θ) pentru un nor vulcanic cu temperatură uniformă T_c și temperatură de fundal T_b . Este puțin probabil că distribuția dimensiunii particulelor într-un nor de cenușă este uniformă, astfel încât o modificare de distribuție γ se folosește în calculele de transfer radiativ. Un cub cu date tridimensionale cu axe r, τ, θ este derivat pentru temperaturile T_c și T_b . Cubul de date se utilizează pentru a efectua o extragere începând de la măsurătorile de temperatură Cyclops și vizualizarea capătului geometric cu variabilele microfizice r și τ . Extragerea microfizică poate fi reprezentată prin:

$$R(r, \tau, \theta; M) \leftarrow G(T_i, \Delta T_{i,j}, \theta; P) \quad (3.41)$$

P include constrângerile fizice furnizate de modelul microfizic (și anume distribuția dimensiunii, indicii reali și imaginari de refracție, densitatea cenușii). Simbolul \leftarrow reprezintă o interpolare între cubul de date (G) și cubul $r\theta$ (R).

Având în vedere că există mai multe ipoteze și aproximări implicate în extragerile de date, acestea sunt sensibile la utilizarea cantităților integrate, care egalizează erorile. O variabilă care prezintă un anumit interes atât pentru pericolul cenușii, cât și pentru înțelegerea proceselor de erupție vulcanică este masa de cenușă fină care a erupt.

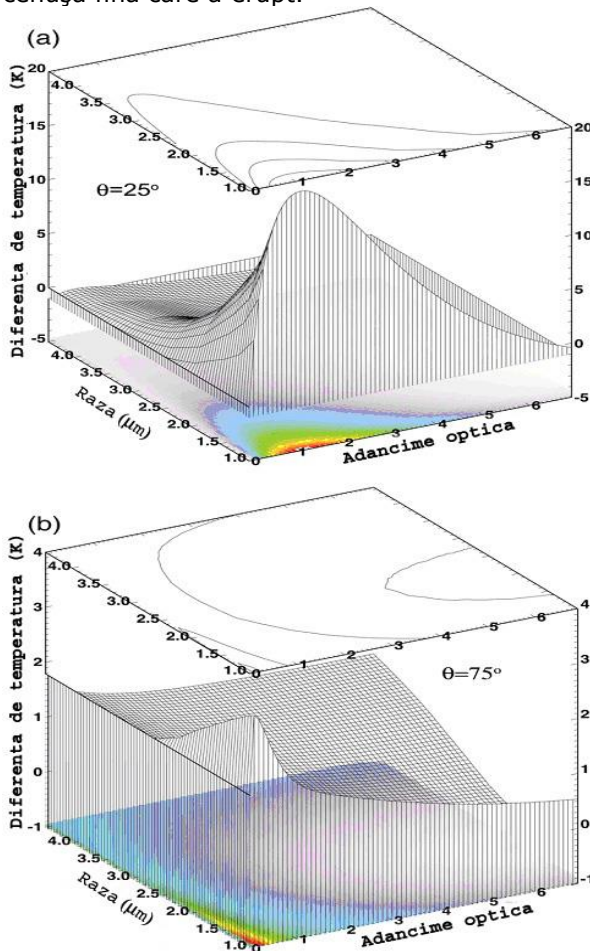


Figura 3.23. (a) Diferența de temperatură, raza particulelor adâncimea optică pentru un unghi zenital de vizualizare de 25° . (b) La fel ca (a), dar pentru un zenit de vizualizare de 75° [84]

Soluția se găsește la intersecția dintre diferența de temperatură și suprafața de adâncime optică a razei. În cele mai multe cazuri, aceasta este o problemă bine-definită, iar punctul de intersecție este unic.

$$N = \int_r^\infty \frac{dn(r)}{dr} dr \quad (3.42)$$

unde $n(r)$ este distribuția (presupusă) a dimensiunii și adâncimea optică a norului este:

$$\tau_\lambda = \pi L \int_0^\infty r^2 Q_{ext}(r, \lambda) n(r) dr \quad (3.43)$$

L este grosimea geometrică a norului. Eficiența extincției polidisperzionale (Q_{ext}) este:

$$\hat{Q}_{ext} = \frac{\int_0^\infty \pi r^2 Q_{ext}\left(\frac{2\pi r}{\lambda}, m\right) \frac{dn(r)}{dr} dr}{\int_0^\infty \pi r^2 \frac{dn(r)}{dr} dr} \quad (3.44)$$

Încărcarea masei de cenușă fină (kg m^{-3}) este:

$$m_l = \frac{4\pi}{3} \rho \int_0^\infty r^3 n(r) dr \quad (3.45)$$

unde ρ este densitatea cenușii. Masa totală M (în kg) poate fi evaluată prin însumarea tuturor pixelilor imaginii afectați folosind extragerile r și τ și înmulțind cu zona A_p a pixelului:

$$M = \sum_p A_p \frac{4\pi}{3} r_p \tau_p \frac{\int_0^\infty \pi r^2 n(r) dr}{\int_0^\infty \pi r^2 Q_{ext}(r, \lambda) n(r) dr} \quad (3.46)$$

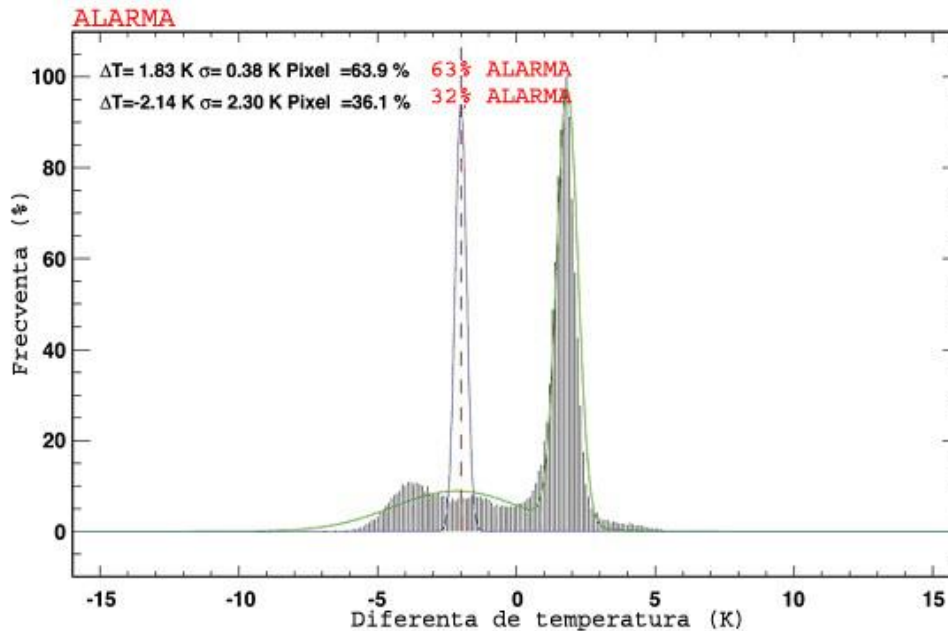


Figura 3.24. Histograma diferenței de temperatură 2D (11-12 μm) obținută la „izvoarele termale” din Rababa când atmosfera era încărcată cu cenușă[84]

S-a folosit un prag scăzut (-2K) pentru separarea cenușii deoarece și atmosfera de fundal conține cenușă. În acest caz, algoritmul automat de alarmă indică faptul că 64% din pixeli sunt afectați de cenușă.

Masa totală a norului se calculează prin însumarea maselor de pixeli individuali conținuți în imagine, fiind indicat în partea superioară a fiecărei figuri. Nu se efectuează compensarea pentru cenușa care iese din câmpul vizual al camerei. Particulele de cenușă în intervalul de dimensiune fină nu par să arate stratificare pe verticală, mai degrabă structura principală sugerează că cele mai mici particule există la periferia norului, iar particulele mai mari rămân în porțiunea centrală.

Interacțiunea dintre dinamica internă a coloanei de cenușă în curs de dezvoltare, efectele gravitaționale și vântul ambiental duc la o distribuție 3D variată și complexă a particulelor, în aceste coloane de cenușă slab eruptive, ceea ce face dificilă efectuarea de concluzii semnificative[104].

Detectarea și extragerea concentrațiilor traiectoriei de gaz SO₂ din surse naturale și artificiale se poate realiza prin utilizarea camerelor de termoviziune în infraroșu. Progresele recente ale tehnologiei detectorului nerăcit oferă acum posibilitatea folosirii camerelor de termoviziune (de la 7 la 14 μm) relativ ieftine ca instrumente de studiu și cuantificare a comportamentului gazelor periculoase și particulelor din coloanele atmosferice. Camera de termoviziune (Cyclops) nerăcită la sol cu viteză rapidă de eșantionare (60 Hz) experimentală, care funcționează cu cinci canale spectrale la lungimi de undă centrale de 8,6, 10, 11 și 12 μm și un canal în bandă largă (7-14 μm) a fost testată în cazul mai multor vulcani și pe două amplasamente industriale, unde SO₂ reprezintă componenta majoră a coloanelor. Sistemul instrumentului, inclusiv procedura de calibrare și algoritmi de extragere, se numește Cyclops. Rezultatele indică faptul că SO₂ se poate identifica și deosebi relativ ușor în coloane, dar concentrația traiectoriei se poate cuantifica mai dificil. Se

furnizează descrierea completă a algoritmilor de extragere, rezultatele ilustrative și analiza detaliată a erorilor. Cu valorile NEΔT ale canalelor spectrale în intervalul 0,4-0,8 K, se obțin erori ale concentrației traiectoriei de 20%. Calcularea mediei cadrelor și valorile îmbunătățite de NEΔT pot reduce această eroare la mai puțin de 10%, ceea ce face exploatarea la distanță, pe timp de zi sau de noapte a unui instrument de acest tip foarte practică atât pentru monitorizarea emisiilor de SO₂ industriale și pentru măsurătorile de flux de SO₂ la vulcanii activi. Sistemul nou de cameră poate fi folosit pentru a studia radiația termică din norii meteorologici și atmosferă.

Măsurarea gazelor vulcanice (de ex., SO₂, HCl, CO₂) prezintă mare interes, pentru a oferi perspective importante în procesele vulcanice, de ex., pentru monitorizarea variațiilor ratelor de flux de gaz vulcanic în vulcanii reactivați. Gazele poluante din sursele industriale trebuie să fie monitorizate, mai ales zonele în curs de dezvoltare unde standardele de emisie industriale pot fi scăzute și dificil de pus în aplicare. Există scurgeri de gaze și din fabricile industriale și conducte, iar emisiile de gaze toxice din accidente industriale sau din acțiunile deliberate necesită identificare și monitorizare de la o distanță de siguranță. În aceste cazuri, precizia și acuratețea ridicate nu sunt necesare, ci mai degrabă identificarea rapidă, cuantificarea concentrației de gaz [21].

Regiunea în infraroșu termică (de la 3 la 15 μm) a spectrului electromagnetic conține mai multe sub-regiuni, care pot fi exploatate pentru studiul gazelor atmosferice. Acestea sunt regiunile fereastră între 3 și 4 μm, care vor desemna infraroșul mediu (MIR), și între 7 și 14 μm, care va desemna infraroșul termic (TIR). MIR se folosește la identificarea „punctelor calde”, regiuni localizate a pixelilor fierbinți din măsurători prin satelit. MIR poate fi folosit de la sol sau de pe platforme aeriene pentru formarea imaginii căldurii provenite din incendii ale pădurii sau a gazelor fierbinți care ies din craterele vulcanice și pentru trasarea hărții temperaturilor în coloane și pe câmpurile de lavă. TIR se folosește mai puțin frecvent pentru studiul proceselor vulcanice. Instrumentele TIR de pe sateliți folosesc sisteme active de răcire și, în aceste cazuri, datele de imagine sunt folosite pentru a monitoriza norii de erupție vulcanică și pentru a-i deosebi de norii meteorologici pentru avertismentele de pericol de aviație și pentru măsurătorile de gaze. Algoritmii lor s-au bazat pe o formulă de fereastră divizată folosind canale centrate la 8,74 μm și 9,56 μm pentru a elimina efectele vaporilor de apă și a determina și abundența SO₂ [48].

S-a folosit detectarea termică pasivă, bazându-se pe emisia sau absorbția de gaz pentru a oferi un semnal pentru măsurare. De asemenea, măsurătorile se pot face în modul de absorbție prin utilizarea soarelui ca sursă sau prin asigurarea unei surse artificiale de radiații.

În aceste aplicații se folosesc interferometre cu un singur câmp de vizualizare (FOV), cu rezoluție spectrală medie (6-0,5 cm⁻¹) pentru a colecta simultan informații cantitative privind speciile multiple de gaz. Interferometrele cu transformantă Fourier (FT-IR) au devenit un dispozitiv pentru studiul gazelor vulcanice. De curând, s-au dezvoltat sisteme care utilizează lumina ultravioletă ca sursă pentru măsurătorile de SO₂ vulcanic, și, de asemenea, pentru măsurătorile de BrO vulcanic.

În ultimii 10-15 ani, s-au realizat progrese importante în producerea bolometrelor de înaltă sensibilitate.

Capacitatea de detectare a acestor dispozitive este limitată de fundal, acestea fiind adesea numite dispozitive BLIP (fotodetector în infraroșu limitat de fundal). Utilizarea semiconductoarelor din siliciu (substrat din nitruură de siliciu cu

materiale de detectare a oxidului de vanadiu) pentru fabricația rezoluțiilor detectoarelor bolometrice a redus foarte mult costurile de producție ale camerelor de termoviziune. Aceste microbolometre, care sunt, de obicei, alcătuite din 10^4 - 10^6 elemente, sunt sensibile la radiațiile în intervalul de lungimi de undă de 7-14 μm și funcționează la 30-60 Hz. Camerele de termoviziune sunt disponibile în comerț, cu sensibilitate la temperaturi de ~ 50 mK (7-14 μm), dimensiuni de rezoluție de 320x240 pixeli (sau mai mare), sistem optic F1.0 și funcționare la 60 Hz. Astfel, în principiu, o cameră de acest fel poate capta imagini care indică modificările de temperatură mai mici de 0,1 K, la o viteză de 10 cadre pe secundă. În practică, acest lucru este greu de realizat din cauza prezenței zgomotului ($1/f$, fluctuațiile de temperatură de fundal și internă și zgomotul Johnson), neuniformitatea rezoluției, necesitatea calibrării și integrarea cadrului. Alți factori pot limita, de asemenea, viteza ideală de captare a imaginii: de exemplu, extragerea informațiilor din cadrul imaginii impune sisteme electronice rapide și un microprocesor bun, precum și hardware și software de comunicare.

Camera cu o singură bandă de trecere (~ 8 -14 μm) se folosește pentru a vedea cerul de deasupra capului în studiul norilor. Aceștia raportează eroarea de calibrare a instrumentului să fie între 0,5 $\text{W m}^{-2} \text{sr}^{-1}$ sau aproximativ 2% din radianța mediului ambiant și, arată că microbolometrul este sensibil la temperaturi scăzute (<240 K) prin observarea schimbărilor în semnale pentru nori rarefiați foarte mari (cirrus).

Camera dezvoltată aici încorporează selecția lungimilor de undă (filtre), acest lucru scăzând sensibilitatea și adăugând întârzieri de timp față de captura imaginii. Camera trebuie să fie sensibilă la temperaturi de 230-300 K, care acoperă intervalul tipic al temperaturilor coloanelor atmosferice[84].

S-a adaptat o cameră termică IR universală disponibilă în comerț (COTS) cu diferență de temperatură echivalentă cu zgomot de 50 mK (NE Δ T), cu un singur filtru în bandă largă, care acoperă regiunea de lungime de bandă IR de la 7 μm la aproximativ 14 μm . În figura 3.25, se furnizează o schemă a componentelor principale ale camerei „Cyclops”. Radiația intră prin sistemul optic anterior (1), care definește câmpul de vizualizare, și se focusează pe rezoluția microbolometrului (2). Sistemul electronic de citire (3) convertește semnalele de radiații detectate în tensiuni, iar indicatorul numeric (4), microprocesorul (5) și portul de comunicații (6) emit rezultatele rezoluției microbolometrului IR către un calculator pentru afișarea imaginii și alte analize.

Pentru utilizarea în detectarea și cuantificarea gazelor atmosferice la temperaturi tipice atmosferice sunt necesare mai multe modificări asupra camerei COTS.

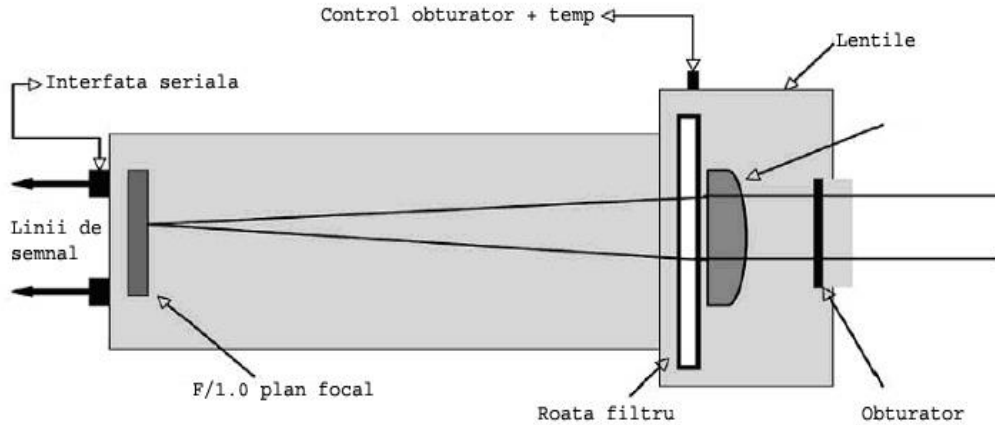


Figura 3.25. Schema prezintă principalele componente ale camerei de termoviziune în infraroșu „Cyclops”. Roata filtrului, care conține până la 5 filtre, este plasată în spatele obiectivului.[85]

Figura 3.26 prezintă intensitățile în linie din baza de date HITRAN-2000 ilustrând principalele caracteristici de absorbție ale SO₂ în regiunea 6,8-10 μm. Cea mai intensă caracteristică la 7,3 μm nu este potrivit pentru detectarea la sol a SO₂, deoarece

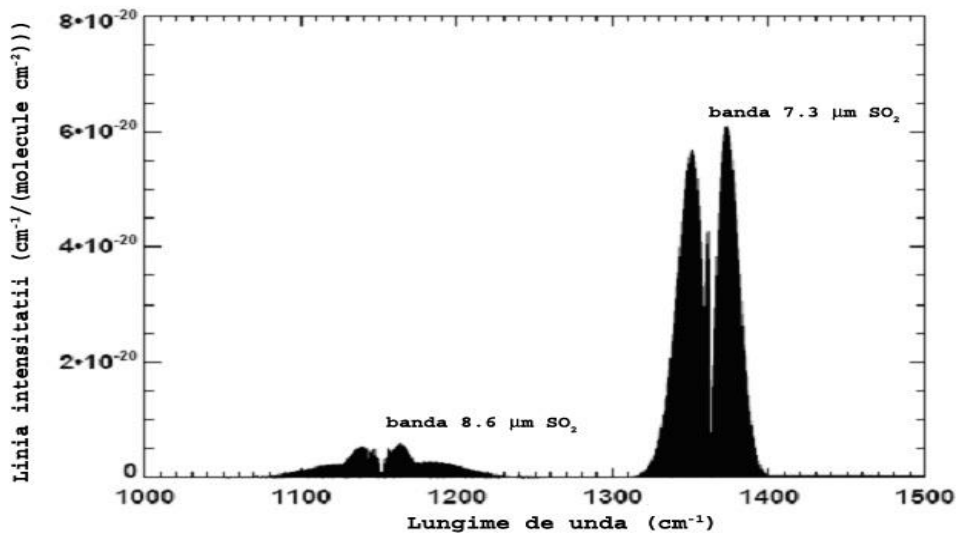


Figura 3.26 Transmisia de SO₂ atmosferic în regiunea 1000-1500 cm⁻¹ (6,7 - 10 μm)[85]

absorbția vaporilor de apă domină în această regiune. Caracteristica de 8,6 μm, deși mai puțin intensă, este mai potrivită pentru detectarea SO₂, deoarece absorbția vaporilor de apă este mult redusă față de 7,3 μm. Cyclops este limitată la măsurarea gazelor, care au caracteristici mari de absorbție (~ 1 μm sau mai mare) în regiunea 7 -14 μm. Un alt gaz vulcanic care îndeplinește acest criteriu este CO₂, dar din cauza abundenței relativ ridicate de CO₂ în atmosfera ambiantă măsurarea acestui gaz folosind radiometria IR termică la sol este problematică.

Designul Cyclops a fost puternic influențat de cunoașterea gazului atmosferic și caracteristicile de absorbție ale particulelor, constrânsă de tehnologia actuală. Tabelul 3.1 arată canalele (sau benzile înguste) alese pentru Cyclops pentru detectarea de SO₂ și cenușă vulcanică de la sol.

Tabelul 3.1. Numărul de canal, lungime de undă centrală, lățime de bandă, scopul și diferența de temperatură echivalentă zgomotului necesară (NEΔT) pentru Cyclops.

Nr. de canal	Lungime de undă	Scop	mK NEΔT
1	7,3/8-12	Formare de imagini SO ₂ /coloană	300/100
2	11,5-12,5	SO ₂ și cenușă vulcanică	200
3	10,4-11,4	Cenușă vulcanică	200
4	8,2-9,2	SO ₂	400
5	9,8-10,4	Temperatură nor/coloană	100

Deosebirea gazului și a particulelor și cuantificarea necesită imagini termice de înaltă fidelitate de la Cyclops. Pentru a atinge fiabilitate și precizie, camera trebuie să fie calibrată.

Aplicarea corecției atmosferice este esențială pentru identificarea corectă a gazelor și particulelor din imagini. În plus, corecția trebuie să fie aplicată cu o dependență de unghiul de vizualizare și, de preferință, pixel cu pixel [29].

Sistemul de cameră Cyclops a fost proiectat pentru a folosi până la cinci filtre spectrale, alese pentru a optimiza detectarea gazelor atmosferice specifice. Pentru a cuantifica concentrațiile traiectoriei de SO₂ de la sol, s-a ales un filtru în bandă de frecvență îngustă centrat aproape 8,6 μm. Funcția de răspuns a filtrului împreună cu coeficientul de absorbție al SO₂ este măsurat de către NIST.

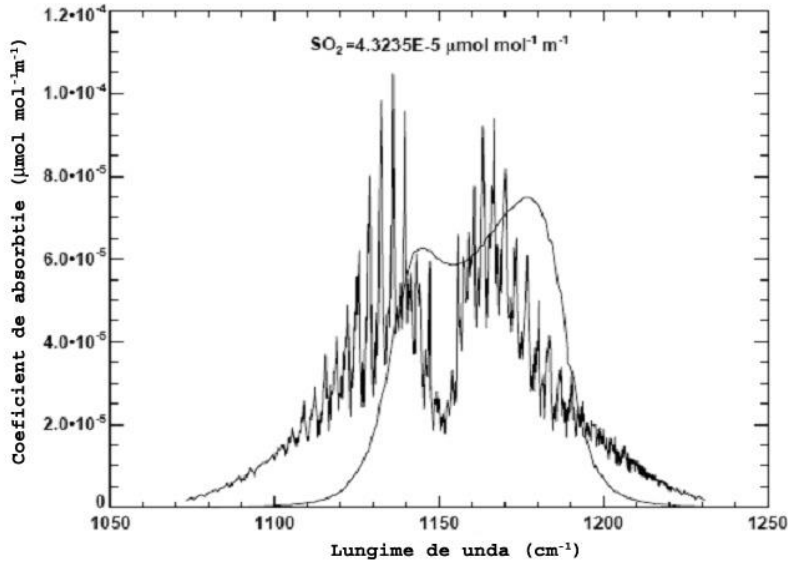


Figura 3.27. Funcția de răspuns a filtrului pentru canalul Cyclops de 8,6 µm și variația coeficientului de absorbtie al SO₂ cu numărul de undă măsurat de către NIST[85]

Coeficientul de absorbtie integrat pe banda de frecvență este de $4,3235 \times 10^{-5} \text{ mol mol}^{-1} \text{ m}^{-1}$. Camera de termoviziune la sol poate vedea o coloană de la o sursă vulcanică sau dintr-un depozit industrial la unghiuri de elevație de 10° sau mai puțin (unghiuri zenitale $> 80^\circ$). Configurația preferată pentru Cyclops este cu un unghi de elevație mare, pentru a reduce efectele absorbtiei vaporilor de apă de-a lungul traiectoriei. Camera are un câmp de vizualizare de $\sim 32^\circ$, iar variația totală unghiulară azimutală este similar cu variația totală zenitală. Fiecare pixel este tratat independent de toți ceilalți și există o trasare simplă între coloana imaginii și numerele de linie și unghiurile azimutale și elevația imaginii. Radiația măsurată la cameră poate fi descrisă prin trei termeni,

$$I_i(\theta) = I_i^f(\theta, \phi) + I_i^p(\theta, \phi) + I_i^b(\theta, \phi) \quad (3.47)$$

unde θ este unghiul de elevație, ϕ este unghiul azimutal, i este numărul de canal, iar indicii se referă la radianța de prim-plan (f), de fundal (b) și radianța coloanei (p). Se poate considera ca radianța coloanei să conste din radiațiile emise și radiațiile din atmosferă care au fost atenuate pe măsură ce traversează prin coloană. Dispersia se ignoră. Radianțele de canal reprezintă integrarea funcțiilor de răspuns de filtru al canalului pentru fiecare pixel în spațiul de imagine 2D. Radianțele de fundal se referă la radianțele de pe cer, din spatele coloanei; radianța de prim-plan se referă la radianța care provine din atmosfera dintre coloană și cameră. Analiza constă în izolarea termenului de radianță a coloanei și apoi estimarea produsului concentrației de gaz și grosimea coloanei. Modelul utilizat nu admite nici o dispersie și variațiile din coeficientul de absorbtie a mediului sunt invariante de-a lungul traiectoriei absorbtiei. Coloana se presupune să fie plană

paralelă și determinată de ecuația Schwarzschild de transfer radiativ. Ecuația rezultată care se utilizează la extragerea cantității concentrației m^* (produsul densității absorbante cu lungimea traiectoriei), se indică aici și se fac unele remarci generale[86].

$$m^* = \rho d = -\frac{1}{k} \cos\theta \cos\phi \ln(1 - \varepsilon) \quad (3.48)$$

unde ε reprezintă emisivitatea efectivă a coloanei și este dată de,

$$\varepsilon = \frac{(\Delta T_{i,j}^p - \Delta T_{i,j}^o) - \Delta T_i^p (1 - \Delta T_{p,j} / \Delta T_{p,i})}{\Delta T_{p,j} (1 - \Delta T_i^p / \Delta T_{p,i})} \quad (3.49)$$

iar k este coeficientul de absorbție mediu pe funcția de răspuns a canalului de măsurare, iar toți ceilalți termeni sunt diferențe de temperatură. Procedura de recuperare folosește 3 din cele 5 canale ale camerei. Informația privind SO₂ din coloană este conținută în canalul de 8,6 μm , în timp ce canalul de 12 μm se folosește la corectarea efectelor atmosferice, iar canalul de 10 μm se folosește la estimarea temperaturii coloanei. Sistemul de extragere utilizează diferențele de temperatură. Cele mai importante dintre acestea sunt contrastul termic, diferențele de temperatură dintre temperatura coloanei și atmosfera de fundal și termenii care implică diferențele dintre temperatura de luminozitate spectrală, cu și fără coloană, și diferențele de temperatură de luminozitate între canalele de 8,6 și 12 μm . Pentru coloanele extrem de opace, aceste diferențe pot fi mici, iar sistemul de extragere devine instabil. Pentru coloanele foarte rarefiate, contrastul termic este redus, iar extragerea devine limitată de zgomot.

Se consideră o coloană paralelă cu planul (placa), cu grosimea d alcătuită dintr-un amestec omogen din două gaze cu densități ρ_1 și ρ_2 . Se presupune că coeficienții de absorbție ai gazelor k_1 și k_2 nu variază în placă și că radiațiile se atenuează prin absorbție și se emit la o temperatură constantă T_p , dar nu se dispersează. Camera vede coloana cu până la cinci canale de bandă îngustă notate prin $i, i = 1, 5$ și vom presupune că toate cantitățile (de ex., radianțe și coeficienții de absorbție) sunt medii pe răspunsurile filtrului de canal.

Ecuația Schwarzschild pentru radianța coloanei pentru un pixel și un canal poate fi scrisă,

$$\frac{dI_i^p(\theta_n)}{k_i \rho dr} = -I_i^p(\theta_n) + B_i(T_p) \quad (3.50)$$

unde r este distanța de-a lungul coloanei în direcția a θ_n , B_i este funcția Planck, i este numărul de canal, iar T_p este temperatura coloanei (care se presupune că nu variază de-a lungul traiectoriei). Această ecuație poate fi integrată de-a lungul traiectoriei pentru a oferi,

$$I_i^p(r_1, \theta_n) = I_i^o e^{-\tau_i(r_1, 0)} + \int_0^T B_i(T_p) e^{-\tau_i(r_1, r)} k_i \rho dr' \quad (3.51)$$

$$\tau_i(r_1, r) = \int_r^{r_1} k_i \rho dr' \quad (3.52)$$

unde I_i^o este radiația din atmosferă în direcția $r, \tau_i(r_1, r)$ este grosimea optică a coloanei între r și r_1 , r_1 este lungimea traiectoriei parcurse de radiații în coloană în direcția r . Se presupune că traiectoria este omogenă, k nu variază în funcție de poziția în coloană, iar coloana se află în echilibru termodinamic. Ecuația (3.43) arată că radiația coloanei măsurată de canalul i constă din termeni reprezentând atenuarea absorbției coloanei și a emisiilor din coloană de-a lungul traiectoriei. Pentru doi absorbanți,

$$\tau_i(r_1, r) = \int_r^{r_1} k_{i,1} \rho_1 + k_{i,2} \rho_2 dr' \quad (3.53)$$

Se scrie o ecuație pentru un canal care nu este afectat de absorbantul ρ_2 ,

$$I_j^p(d, \theta_n) = I_j^o e^{-k_{j,1} m_1} + B_j(T_p)(1 - e^{-k_{j,1} m_1}) \quad (3.54)$$

Radianțele (măsurătorile) se fac la lungimi de undă diferite, fiind utilă transformarea acestora la temperaturi de luminozitate, astfel încât să poată fi efectuate diferențele de canal. Folosim o aproximare de serie Taylor pentru liniarizarea acestor ecuații, iar apoi le combinăm pentru a rezolva pentru m_2 . Liniarizarea radianțelor în jurul unei temperaturi medii s-a folosit, fiind o abordare bună cu condiția ca radianțele să fie similare. Pentru o coloană în echilibru termodinamic cu mediul atmosferic și pentru vizualizarea la unghiuri de elevație mici ($\theta_n < 60^\circ$), radianțele vor fi similare[85]. Liniarizarea în jurul radianței atmosferice (I_i^o), neafectată de coloană,

$$I_i^p = I_i^o + \delta T \left(\frac{\partial B_i}{\partial T} \right) \Bigg|_{T_i^o} \quad (3.55)$$

$$\delta T = T_i^p - T_i^o \quad (3.56)$$

În mod similar,

$$B_i(T_p) = I_i^o + (T_p - T_i^o) \left(\frac{\partial B_i}{\partial T} \right) \Bigg|_{T_i^o} \quad (3.57)$$

Folosind (3.47) - (3.49) și înlocuind radianțele dă,

$$T_i^p - T_i^o = (T_p - T_i^o)(1 - e^{-k_{i,1} m_1}) \quad (3.58)$$

Pentru canalul cu doi absorbanți se obține:

$$T_j^p - T_j^o = (T_p - T_j^0)(1 - e^{-k_{j,1}m_1} e^{-k_{j,2}m_2}) \quad (3.59)$$

Fie $e^{-k_{i,1}m_1} = e^{-k_{j,1}m_1}$. Transmiterea prin vaporii de apă este similară la cele două lungimi de undă alese, și anume 8,6 μm și 12,0 μm. Folosind această aproximare, avem,

$$T_j^p - T_j^o = (T_p - T_j^0)(1 - e^{-k_{i,1}m_1} e^{-k_{j,2}m_2}) \quad (3.60)$$

Scăzând (3.50) din (3.52), după câteva calcule algebrice, avem,

$$m_2 = -\frac{1}{k_{j,2}} \ln(1 - \varepsilon) \quad (3.61)$$

unde,

$$\varepsilon = \frac{(\Delta T_{i,j}^p - \Delta T_{i,j}^o) - \Delta T_i^p (1 - \Delta T_{p,j} / \Delta T_{p,i})}{\Delta T_{p,j} (1 - \Delta T_i^p / \Delta T_{p,i})} \quad (3.62)$$

$$\Delta T_{i,j}^p = T_i^p - T_j^p \quad (3.63)$$

$$\Delta T_{i,j}^o = T_i^o - T_j^o \quad (3.64)$$

și,

$$\Delta T_i^p = T_i^p - T_i^o$$

$$\Delta T_j^p = T_j^p - T_j^o$$

$$\Delta T_{p,i} = T_p - T_i^o$$

$$\Delta T_{p,j} = T_p - T_j^o$$

Ecuatia (3.54) arată că extragerea concentrației traiectoriei depinde în principal de diferența de temperatură a coloanei dintre cele două canale și, de asemenea, de contrastul termic dintre coloană și atmosfera din afara coloanei (ΔT_i^p). Soluția la (3.53) necesită estimări ale variabilelor T_i^p , T_i^o , T_j^p , T_i^o , T_p și θ_n , și ale specificațiilor coeficientului de absorbție $k_{j,2}$. Măsurătorile constau din radianțele coloanei (I_i^p , I_j^p), radianțele de prim-plan (I_i^f , I_j^f) și radianțele de fundal (I_i^b , I_j^b). Se iau în considerare două măsurători, una efectuată prin coloană și alta fără coloana în câmpul vizual. Presupunând că atmosfera nu se schimbă considerabil între aceste două măsurători, se poate scrie pentru prima măsurare (scăzând referința la unghiuri),

$$I_i = I_i^f + I_i^p + I_i^b \quad (3.65)$$

și pentru a doua măsurătoare,

$$I_i^0 = I_i^{f,0} + I_i^{b,0} \quad (3.66)$$

Exponentul ^o se referă la radiațiile atmosferice din „afara” coloanei. Fiecare din aceste cantități poate fi determinată prin rezolvarea integralelor de forma,

$$I_\lambda = \int_z B_\lambda [T(z)] e^{-\int_z k_\lambda(z') \rho(z') dz'} k_\lambda(z) \rho(z) dz \quad (3.67)$$

Fie transmitanța fiecărei traiectorii marcată prin $\tau_{i,q}^f, \tau_{i,q}^p, \tau_{i,q}^b$ pentru prim-plan, coloană și, respectiv, fundal, unde, la fel ca înainte, *i* reprezintă canalul, iar *q* tipul de absorbție (*q* = 1,2). Temperaturile straturilor sunt *T_f*, *T_p* și, respectiv, *T_b* și vom înlocui integralele traiectoriei cu radianțe medii, notate printr-o supralinie. Apoi,

$$I_i = (1 - \tau_{i,1}^f) B_i(T_f) + \tau_{i,1}^f ((1 - \tau_{i,1}^p \tau_{i,2}^p) B_i(T_p) + \tau_{i,1}^p \tau_{i,2}^p \tau_{i,1}^b B_i(T_b)) \quad (3.68)$$

$$I_i^0 = (1 - \tau_{i,1}^f) B_i(T_f) + \tau_{i,1}^f \tau_{i,1}^b B_i(T_b) \quad (3.69)$$

S-a presupus faptul că atmosferele de prim-plan și de fundal nu s-au schimbat între măsurători și că acestea nu conțin SO₂ (absorbant *q* = 2). Scăzând,

$$I_i - I_i^0 = I_i^p - \tau_{i,1}^f \tau_{i,1}^b (1 - \tau_{i,2}^p) B_i(T_b) \quad (3.70)$$

O ecuație similară poate fi obținută pentru un al doilea canal *j*, care nu are nici o absorbție din cauza absorbantului *q* = 2,

$$I_j - I_j^0 = I_j^p - \tau_{j,1}^f \tau_{j,1}^b B_j(T_b) \quad (3.71)$$

Scăzând (3.63) din (3.62),

$$\Delta I_{i,j}^0 = \Delta I_{i,j}^p + \delta I_{i,j}^0 \quad (3.72)$$

unde,

$$\Delta I_{i,j}^0 = (I_i - I_i^0) - (I_j - I_j^0) \quad (3.73)$$

$$\Delta I_{i,j}^p = I_i^p - I_j^p \quad (3.74)$$

$$\delta I_{i,j}^0 = I_i^{b,0} (1 - \tau_{i,1}^p \tau_{i,2}^p) - I_j^{b,0} (1 - \tau_{j,1}^p) \quad (3.75)$$

Cantitățile din (3.74) sunt măsurabile și, prin urmare, (3.75) poate fi rezolvată după aplicarea corecției $\delta I_{i,j}^0$ și calcularea analogiilor de temperatură și luminozitate. Trimiterea la unghiul de elevație θ a fost abandonată din conveniența

notației. Deoarece cantitățile necesare sunt diferențe de temperatură (adică $\Delta T_{i,j}^0$), variația pe verticală se elimină prin prelucrarea diferențelor. Trebuie estimate cantitățile $\Delta T_{i,j}^p$, ΔT_i^p și ΔT_j^p . Aceste cantități se obțin prin prelucrarea fiecărei imagini pentru a elimina variația pe verticală a temperaturii luminozității de-a lungul fiecărei coloană de imagine. Se obține o potrivire liniară mai puțin pătrată pentru fiecare coloană de imagine folosind datele cu câteva linii mai sus de coloana de erupție până la câteva linii sub partea de sus a imaginii. Din moment ce fiecare imagine este înaltă de 240 de linii, potrivirea folosește de obicei între 100 și 150 de linii[90]. Variațiile în numărul de linii utilizate în potrivire apar din cauza faptului că, uneori, coloana este ridicată și deoarece unele imagini conțin date de zgomot spre partea de sus a imaginii.

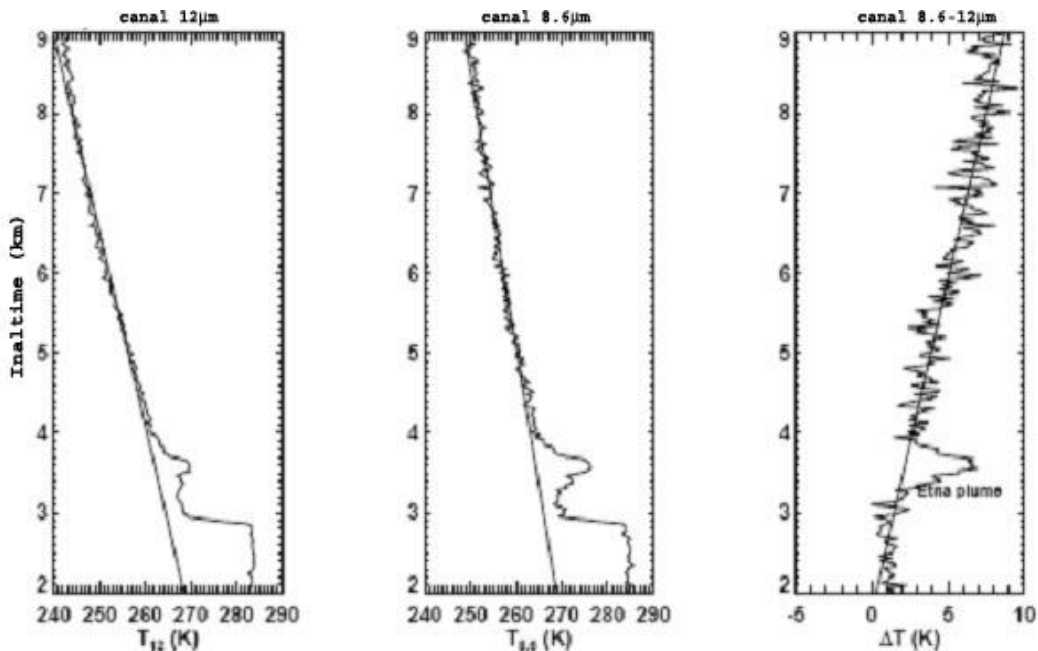


Figura 3.28. Temperatura de luminozitate față de variația pe înălțime pentru filtrele de 12 (panoul din stânga), 8,6 (panoul central) și 8,6 - 12 (panoul din dreapta) μm . Liniile drepte sunt forme liniare mai puțin pătrate bazate pe datele de profil de deasupra coloanei, extrapolate prin și sub coloană. ΔT este diferența de temperatură de luminozitate între măsurătorile de 8,6 și 12 μm [85]

Potrivirea liniară elimină variațiile zenitale și oferă estimări ale T_i^0 și T_j^0 . Deoarece fiecare coloană a imaginii este tratată diferit, se ține cont de orice variație azimutală din atmosferă. Odată ce s-a aplicat această procedură, temperatura coloanei de erupție se estimează din imaginea de 10 μm (canalul cel mai transparent), după aplicarea unei corecții pentru vaporii de apă bazată pe calculele de transfer radiative cu MODTRAN-4.

Sistemul de extragere de SO₂ face mai multe ipoteze simplificatoare, care pot cauza erori în rezultatele finale. Sistemul depinde în principal de măsurătorile de

temperatură și diferențele de măsurare, dar și de câțiva parametri (de ex., coeficientul de absorbție, unghiuri de vizualizare). Se consideră că sursele de eroare se împart în trei grupe distincte:

- Erori de tip I, datorate zgomotului de măsurare,
- Erori de tip II, care rezultă din ipotezele și aproximările utilizate în sistemul de extragere și,
- Erori de tip III, datorate specificațiilor inexacte sau incomplete ale parametrilor necesari în sistem.

Formula teoretică pentru diferența de temperatură echivalentă zgomotului (NEΔT), care produce un SNR (raport semnal-zgomot) unitar pentru un pixel de microbolometru unic poate

Tabelul 3.2. NEΔT teoretice (mK) pentru cele cinci canale ale camerei de termoviziune în infraroșu și pentru patru temperaturi de scene diferite.

Temperatura (K)	Lungime de undă (μm)			
	8,6	10	11	12
220	275	170	140	120
250	140	100	85	80
270	100	75	70	60
290	75	60	55	55

fi scrisă,

$$NE\Delta T = \frac{4}{\pi} \left[\frac{F_{\#}^2}{D^*} \sqrt{\frac{\Delta f}{A_d}} \right] \left(\frac{dI}{dT} \right)^{-1} \quad (3.76)$$

unde $F_{\#}$ este numărul F al camerei, Δf este frecvența de eșantionare, I este radianța, A_d este zona detectorului, iar D^* este capacitatea de detectare normalizată sau cifra de merit a detectorului. Există mai multe surse de zgomot pentru detectoarele de termoviziune, inclusiv zgomotul Johnson, zgomotul 1/f și zgomotul datorat fluctuațiilor de temperatură. Ultima dintre aceste surse de zgomot reprezintă, de obicei, factor de limitare.

Valorile NEΔT (în mK) pentru cameră au fost calculate pentru un set dat de temperaturi de luminositate din scenă folosind derivata funcției Planck, la lungimi de undă centrale ale canalelor.

Efectuarea mediei cadrelor poate reduce zgomotul prin $\sqrt{N_f}$ unde N_f este numărul de cadre. S-au efectuat experimente de laborator și pe teren pentru a stabili indicatorii de performanță pentru camera de termoviziune. Aceste studii au sugerat că imaginile cu 24 de cadre au fost mult mai zgomotoase decât sugerează rezultatele teoretice[85]. Valorile NEΔT măsurate au variat de la 0,1 K la 290 K pentru canalul în bandă largă până la 1,8 K la 220 K pentru canalul de 8,6μm. O potrivire polinomială (gradul 3) mai puțin pătrată pe datele de laborator s-a efectuat pentru fiecare canal, astfel încât s-a obținut valoarea NEΔT pentru orice temperatură NEAT din scenă arbitrar. Potrivirea este dată de,

$$NE\Delta T = \sum_{i=0}^{i=3} a_i T_s^i \quad (3.77)$$

La 260 K, $NE\Delta T = 0,80$ K pentru canalul de 8,6 μm și 0,41 K pentru canalul 12 μm . Studiile au arătat, de asemenea, că $\sim 0,5\%$ dintre pixeli au fost „pixeli morți” – adică, acești pixeli au fost stinși constant și nu au înregistrat semnal. Odată ce acești pixeli au fost identificați, au fost marcați și nu s-au inclus în nici o analiză ulterioară.

Calibrarea absolută a camerei este descrisă de Bernardo și Prata [84]. Ei estimează erorile absolute între 0,5-2 K, în funcție de temperatura scenei, temperatura mediului și canalul utilizat. Deoarece sistemul de extragere folosește diferențele de temperatură, atâta timp cât canalele se comportă astfel, impactul real al erorii de temperatură absolută nu este mare. Principalul impact apare prin estimarea temperaturii coloanei folosind canalul de 10 μm . Calibrările de deviație ale temperaturii se efectuează la fiecare 5-6 minute, folosind o diafragmă înnegrită, în încercarea de a minimiza efectele schimbărilor de temperatură ale mediului. Sursa de eroare pentru aceste calibrări apare din neînnegrirea diafragmei de calibrare. Performanța diafragmei a fost măsurată prin compararea cu un corp negru de emisivitate în laborator $> 0,99$. S-a stabilit că emisivitatea diafragmei a fost de $\sim 0,98 \pm 0,005$, cu o ușoară dependență de lungimea de undă. O eroare de $\pm 0,005$ în emisivitate duce la o eroare de temperatură $< 0,1$ K, care este mult mai mică decât valoarea $NE\Delta T$ a canalelor filtrate ale camerei. Aceste considerații sugerează că o precizie a calibrării absolute de $\pm 0,5$ K este rezonabilă. În timp ce aceasta este o eroare de interferență, semnul interferenței este probabil variabil și greu de stabilit cu excepția cazului în care măsurătorile temperaturilor de mediu, ale carcasei camerei și ale scenei sunt disponibile [37].

Aceste erori se datorează ipotezelor făcute în derivarea schemei de extragere.

Aceste ipoteze includ:

1. Transferul radiativ paralel cu planul, fără dispersie, modelul RT,
2. Liniarizarea radianțelor la temperaturi de luminozitate,
3. Temperatura constantă a coloanei,
4. Lipsa variației spațiale a coeficientului de absorbție al SO₂,
5. Echivalența coeficienților de absorbție a vaporilor de apă la 8,6 μm și 12 μm ,
6. Invarianța structurii atmosferice, cu sau fără coloană.

Ipoteza 1 include ipotezele făcute de obicei pentru rezolvarea problemelor de transfer radiativ în regiunea în infraroșu. Pentru geometrii în care coloana este mică în comparație cu curbura Pământului, traiectoriile radiațiilor sunt aproape identice cu traiectoriile reale. Alte aspecte ale transferului radiativ includ utilizarea codului MODTRAN-4, care a fost supus controlului detaliat și inter-comparațiilor. Estimarea precisă a impactului probabil al erorilor în modelarea transferului radiativ pe extragere este dificil de făcut. Considerăm $\pm 2\%$ ca estimare pentru erorile de modelare. Ipoteza [2] arată că o condiție necesară pentru ca această aproximare să fie valabilă este că radianțele trebuie să fie similare. Acest lucru se analizează ușor prin compararea radianței calculate direct prin funcția Planck, cu radianța calculată folosind o aproximare de serie Taylor de gradul întâi, de ex., (3.49). Se arată dependența erorii de radianță (în %) pe abaterea temperaturii de la o valoare medie și se demonstrează că eroarea este mai mică de 2,5% în radianță, pentru abateri de la temperatura medie de până la ± 10 K. Această eroare a radianței duce la o eroare de concentrație a traiectoriei de mai puțin de jumătate din aceasta, datorită măsurării $NE\Delta T$ sau $\sim 5\%$.

Impactul presupunerii că temperatura este constantă prin coloană ar putea fi semnificativ, deoarece contrastul termic al coloanei contribuie în mod semnificativ

la semnalul SO₂ prin (3.54). În primele etape de generare, coloana este neomogenă și în dezechilibru termic. Când coloana a fost generată de o mare erupție explozivă, poate rămâne neomogenă timp de mai multe ore. O idee a variației de temperatură a coloanei poate fi obținută dintr-o analiză a datelor de canal în bandă largă (7-14 μm). Aceste date sunt cele mai puțin zgomotoase, iar variația poate fi folosită ca reprezentant pentru variația temperaturii termodinamice. Există o presiune mică și dependența de temperatură a coeficientului de absorbție, dar, dat fiind faptul că intervalul de variabilitate al presiunii și temperaturii este mic pentru condițiile de observare, această dependență poate fi neglijată.

Ipoteza 5 a fost examinată prin utilizarea unui model de transport al vaporilor de apă. Modelul susține că transmisia vaporilor de apă (τ_λ) în regiunea ferestrei de 8-12 μm este condusă de,

$$\tau_\lambda = \exp\left\{-k_\lambda (P^* w)^{a_\lambda}\right\} \quad (3.78)$$

unde λ este lungime de undă, w este cantitatea de apă precipitabilă (în cm³), P^* este presiunea efectivă, $P^* = P/P_s$, P = presiune (mb), P_s este presiunea de suprafață, k_λ sunt coeficienții de absorbție, iar a_λ sunt coeficienții determinați prin compararea modelului cu măsurători experimentale. Coeficienții k_λ și a_λ sunt clasificați la intervale de 25 cm⁻¹ de la 800-1200 cm⁻¹. Modelul a fost utilizat pentru a calcula transmisia peste funcțiile de răspuns ale filtrului la 8,6 μm și 12 μm ca funcție a cantității de vapori de apă, până la 5,5 cm³ de apă precipitabilă.

Ipoteza că atmosfera este aceeași, indiferent de prezența sau absența coloanei, pare intuitivă, deoarece traiectoria atmosferică luată în considerare este mult mai mare decât traiectoria din coloană. Radianța atmosferică se calculează printr-o interpolare liniară sau pătratică a radianței atmosferice de deasupra și de sub coloană. Natura foarte liniară a potrivirii obținute demonstrează că aceasta este o aproximare bună. Cu toate acestea, implică o eroare. Incertitudinea este evaluată pentru canalele de 8,6 și 12 μm și pentru diferență. Incertitudinea 1-σ pentru diferență a fost de ± 0,3 K, care se traduce printr-o eroare de concentrație a traiectoriei de ± 3%.

Câțiva dintre parametri utilizați în sistemul de extragere trebuie să fie specificați. Aceștia includ k_{SO_2} , geometria (altitudinea camerei, dimensiunea câmpului vizual al camerei), funcțiile de răspuns ale filtrului canalelor, precum și utilizarea datelor radiosondei în modelul RT. Coeficientul de absorbție a fost obținut prin integrarea peste funcția de răspuns a filtrului folosind valorile NIST ale coeficientului de absorbție măsurat la rezoluția de 0,125 cm⁻¹. Eroarea probabilă suportată este mică în comparație cu alte erori. O eroare în coeficientul de absorbție se traduce direct într-o eroare de concentrație a traiectoriei extrase. Această eroare este de 1%.

Erorile geometrice apar din specificarea incorectă a câmpului vizual al instrumentului, iar inexactitățile în măsurarea elevației camerei. Aceste erori sunt mici și afectează extragerea numai prin $\cos\theta$ și prin calculul RT, care utilizează datele radiosondei și necesită specificarea geometriei de calcul.

Erorile care rezultă din toate sursele de eroare luate în considerare sunt prezentate în tabelul 3.3. Eroarea finală este suma rădăcinii medii la pătrat a tuturor erorilor individuale aleatorii, adică, fără erorile de calibrare absolută și aproximare

de transmisie. Astfel, eroarea extragerii este estimat a fi $\sim 20\%$, cu o interferență de -5% la 6% .

Tabelul 3.3. Rezumatul tipurilor de erori și valorilor estimate de eroare

Tip de eroare	Sursă de eroare	Eroare în m* (%)
I	NE Δ T	$\pm 9-10$
I	Calibrare absolută	± 5
II	Model RT	± 2
II	Liniarizare	± 5
II	Temperatura coloanei	$\pm 12-14$
II	Variabilitate spațială a coeficientului de absorbție	(<1 ?)
II	Aproximare a transmisiei	+3
II	Invariantă atmosferică	± 3
III	Coeficient de absorbție	<1
III	Geometrie	<0.5
III	Radiosondă	-

Deoarece absorbția prin vaporii de apă este puțin mai mare la $12 \mu\text{m}$ decât la $8,6 \mu\text{m}$, dacă vaporii de apă nu sunt prezenți în coloană, atunci $\Delta T_{12, 8,6}$ ar fi mai puțin negativ. Se observă prezența vaporilor de apă de-a lungul traiectoriei de la obiectivul camerei la marginea principală a coloanei și, prin urmare, în regiunile atmosferice aflate la distanță de coloană, $\Delta T_{12, 8,6}$ este în continuare negativ. Dacă atmosfera ar fi complet lipsită de vapori de apă, atunci $\Delta T_{12, 8,6}$ ar depinde de profilul de temperatură și de absorbția gazelor uniforme mixte, dintre care CO₂ este cel mai important în această bandă de frecvență. Profilul $\Delta T_{12, 8,6}$ prezintă o scădere marcată cu înălțimea în atmosferă. Deoarece semnalul SO₂ pe care dorim să îl extragem este mascat de aceste caracteristici, datorită vaporilor de apă și variației sale pe înălțime, încercarea de eliminare este necesară, pentru determinarea valorii de fundal (atmosferă fără SO₂) pentru $\Delta T_{12, 8,6}$.

Corectarea efectelor norilor și încercarea extragerii de SO₂ în prezența norilor este extrem de dificilă folosind datele termice, deoarece trebuie să se cunoască microfizica (dimensiunea particulelor, forme și distribuțiile de mărime), precum și faza termodinamică a norilor. Abordarea este de a încerca să se detecteze norii și alte substanțe de interferență (de ex., cenușă vulcanică) și de a marca acești pixeli de imagine ca fiind eronați. Figura 3.26 prezintă un exemplu de detectare a norilor în imaginile Cyclops. Când corecțiile pentru variația pe verticală a vaporilor de apă sunt luate în considerare, această caracteristică apare ca o anomalie pozitivă și ar fi extrasă ca o concentrație de traiectorie negativă, fiind marcată drept eronată. Cenușa poate interfera cu sistemul de extragere, iar norii de cenușă sunt des întâlniți cu emisiile de gaze cu SO₂. Erupția coloanei de cenușă a fost identificată în cadre de imagini consecutive (canale spectrale diferite) separate de $\sim 0,5$ s, care au captat evoluția rapidă a norului, în comparație cu o emisie de gaz SO₂. Norul de cenușă este deosebit de atmosfera de fundal și de gazul SO₂.

Fluxul mediu de SO₂ se poate estima din aceste date folosind estimările de viteză a vântului la înălțimea și dimensiunile efective ale coloanei. O estimare a fluxului de SO₂ poate fi extras din,

$$F = \rho Au \quad (3.79)$$

unde F este rata de flux (în kg s^{-1}), ρ este concentrația (în kg m^{-3}), A este aria secțiunii transversale a coloanei (m^2), iar u este viteza vântului (în ms^{-1}) la înălțimea coloanei. Vitezele vântului la 200 m au fost $\sim 3\text{-}5 \text{ ms}^{-1}$, iar lățimea coloanei (măsurată la jumătatea maximului) a fost considerată ca $\sim 20 \text{ m}$. Aceste valori dau fluxuri de $\sim 1,5 - 2,5 \text{ kg s}^{-1}$, ușor mai ridicate decât emisiile medii raportate.

Tehnica de teledetecție va deveni o opțiune pentru măsurătorile de poluare, nu numai pentru vulcani ci și pentru termocentrale.

3.3.2 Camera UV



Figura 3.29. Camera UV în campania de măsurători de la Halânga

Camera cu tehnologie cu ultraviolete oferă posibilitatea de a măsura concentrația de SO₂. Fluxul de SO₂ este pe larg monitorizat la vulcanii activi. Utilizarea pe scară largă folosind tehnici pasive de detectare de la distanță este folosită pentru a cuantifica densitățile de gaz pe coloană în penele vulcanice folosind lumina soarelui împrăștiată ca o sursă de lumină. Exemplele includ DOAS pasiv, COSPEC și camera SO₂ [5].

Camera UV furnizează numeroase beneficii precum rezoluție ridicată, ceea ce permite captura evenimentelor explozive (vulcanii). Camera poate fi folosită pentru a măsura viteza de transport a penei, o sursă majoră de incertitudine în aceste măsurători. Tehnica de teledetecție este ușor de folosit nefiind legată de coș și oferă mobilitate și accesibilitate pentru monitorizare externă.

Camerele UV au senzori de imagine ce detectează lumina în intervalul ultraviolet. Tipic camerele UV sunt folosite în metrologie, controlul calității

semiconductorilor, măsurătorile nanoparticulelor, etc. În aplicațiile industriale, camerele UV sunt folosite ca echipament de monitorizare a defectelor. În aplicațiile științifice camerele UV sunt folosite la imagistică.

Camerele UV lucrează ca și alte videocamere dar sunt sensibile la lumină în partea UV a spectrului electromagnetic. Când selectăm o cameră UV senzorul de imagine al dispozitivului este foarte important. Specificațiile camerei includ tipul senzorului imagine, rezoluția, formatul optic, pixelii orizontali, pixelii verticali și mărimea pixelului. Sunt două tipuri de senzori de imagine: color și monocrom.

Fluxul de SO₂ este un parametru util pentru monitorizarea vulcanilor, deoarece acesta poate fi considerat un substituent pentru fluxul de magmă. Cunoscând concentrația originală S în magma vulcanului țintă permite determinarea volumelor de magmă degazificată, care ajută atât la interpretarea observațiilor geofizice și vulcanologice, cât și la evaluarea pericolelor. În literatura de specialitate s-au înregistrat mai multe exemple de corespondențe între fluxul de SO₂ și activitatea vulcanică. De exemplu, fluxurile săptămânale de dioxid de sulf măsurate pe Etna au indicat creșteri mari de la valorile de bază din timpul activității eruptive paroxistice[3].

Măsurările emisiilor de SO₂ reprezintă o componentă importantă a monitorizării proceselor vulcanice. Tehnici de teledetecție au fost folosite timp de decenii pentru a investiga eliminările de gaze și a deduce schimbările vulcanice. Metodele de teledetecție standard cu ultra-violete (UV) la sol au folosit spectrometria optică diferențială de absorbție. Formarea imaginilor de SO₂ s-a realizat la coloanele din coșuri, iar în ultimii ani, sistemele de formare a imaginii UV au fost aplicate cercetării vulcanice. Aceste instrumente au limitări în intervalul spectrometric, dar sunt relativ ieftine, abordează unele dintre limitările metodelor anterioare și oferă perspective fără precedent în dinamica coloanei și ratelor de emisie la rezoluție temporală mare[54].

Componentele camerei UV sunt detectorul Apogee e2V CCD47-10 iluminat din spate de 1024×1024 pixeli. Filtrele sunt centrate la 310 nm, 315 nm, 325 nm și 330 nm. Principiul tehnicii camerei UV rămâne același ca și pentru metodele de teledetecție UV anterioare: aplicarea legii Beer-Lambert pentru a determina concentrația unei clase de absorbție.

Imaginile UV sunt convertite în valori de absorbție folosind legea Beer-Lambert:

$$A = -\ln(I/I_0) \quad (3.80)$$

unde A = absorbanța, I = intensitatea luminii după trecerea prin SO₂, iar I_0 = intensitatea luminii înainte de trecerea prin SO₂ sau intensitatea luminii de fundal. Pentru a traduce absorbanta în concentrații de lungime de traiectorie a SO₂, sunt captate imagini ale celulelor de calibrare (celule cu concentrație cunoscută a lungimii de traiectorie a SO₂), fiind modelată relația dintre absorbanta calculată și concentrație[36].

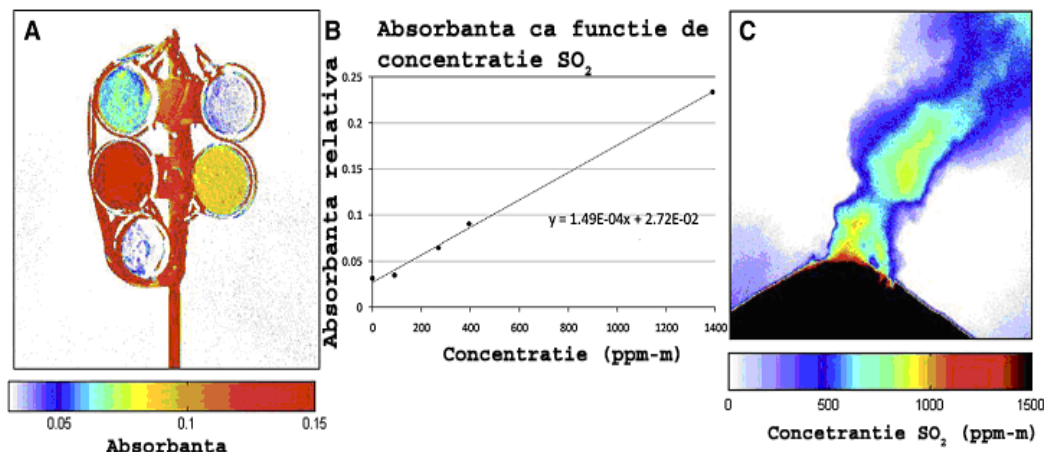


Figura 3.30 Conversia penei imaginilor în concentrație de SO₂[17]

Beneficiile tehnicii camerei UV pentru formarea de imagini a coloanelor de erupție vulcanică sunt multiple: perspective noi în dinamica coloanei din măsurători bidimensionale; viteza coloanei derivată intern obținută din secvențe de imagini; posibilitatea extragerii măsurătorilor la scară redusă (pixeli de dimensiuni mici și secțiuni transversale de coloană) și a imaginilor complete ale coloanei simultan; și rezoluția temporală înaltă a setului de date rezultat.

La colectarea imaginilor de măsurare cu camera UV, parametrii controlați de utilizator includ lungimea expunerii imaginii, diametrul diafragmei obiectivului, regularitatea formării de imagini cu celulele de calibrare. De asemenea se ține cont dacă se folosește filtrul unic 310 nm sau se folosesc ambele filtre 310 nm și 325 nm. În cazul în care se folosesc ambele filtre se va ține cont dacă se aplică timpi de expunere diferiți sau identici pentru fiecare imagine cu filtrul.

3.3.2.1 Introducere în folosirea camerelor

Relația dintre absorbantă și concentrație, conform celor calculate în cazul celor 14 concentrații ale celulei de presiune, este ilustrată în figura 3.30. La concentrații mai mici de aproximativ 1500 ppm, relația este liniară, cu r^2 de 0,96. La concentrații mai mari, datele se potrivesc cel mai bine cu o relație logaritmică (r^2 de 0,914).

Au fost captate șapte imagini din celula de presiune, la o concentrație de 814 ppm în condiții meteorologice dificile. Mediul formării de imagini a variat de la niveluri de lumină ambientală înalte la scăzute și de la cer senin la parțial noros până la zăpadă. Deviația standard pentru aceste măsurători este o absorbantă de 0,024. Dacă pe această deviație se aplică linia care se potrivește cel mai bine din figura 3.31, aceasta este echivalentă cu 296 ppm sau 36%.

Efectul de vignetație în imagini se observă în scăderea de iluminare în colțurile imaginilor. Valorile relative de iluminare descriu cât s-a schimbat iluminarea între centrul imaginii și colțuri, iar acest lucru variază în funcție de diafragma obiectivului. Se remarcă o scădere ușoară, dar coerentă a valorilor DN în colțurile imaginii, pe măsură ce diametrul diafragmei crește de la 3,3 mm (F/32) la 23,33 mm (F/4,5), indicând vignetație constantă. Modificarea iluminării este mai mică de

10% pentru jumătatea centrală a imaginii. Analiza relației dintre valoarea de iluminare și concentrația lungimii traiectoriei de SO₂ (un reprezentant pentru relația absorbantă) indică faptul că acesta rămâne regulat, indiferent de diametrul diafragmei[40]. Pantele liniare ale valorilor intensității celulelor de calibrare (DN) la șapte diafragme diferite, au avut o deviație standard de 1,03DN/ppm.

Timpul de expunere poate varia fundamental pe teren – o valoare constantă pentru acest parametru poate genera intervale foarte diferite de DN în imaginile captate în condiții de cer diferite. În consecință, pentru a ilustra cel mai bine efectul modificării timpului de expunere pe imaginile din acest experiment, datele sunt prezentate luând în considerare DN maxim din fiecare imagine; timpii de expunere mai lungi duc la DN mai mare. În figura 3.31 este evidențiată relația dintre absorbantă și concentrația lungimii traiectoriei de SO₂. Aceasta se modifică odată cu modificarea timpului de expunere. Pantele liniilor cu cea mai bună potrivire cresc cu durata timpului de expunere, la fel ca diferența de înălțime y (decalată din cauza celulei de calibrare). Ratele la care pantele și diferențele de înălțime y se modifică scad pe măsură ce DN maxim al imaginii se apropie de valoarea maximă de înregistrare CCD (65536 înregistrări).

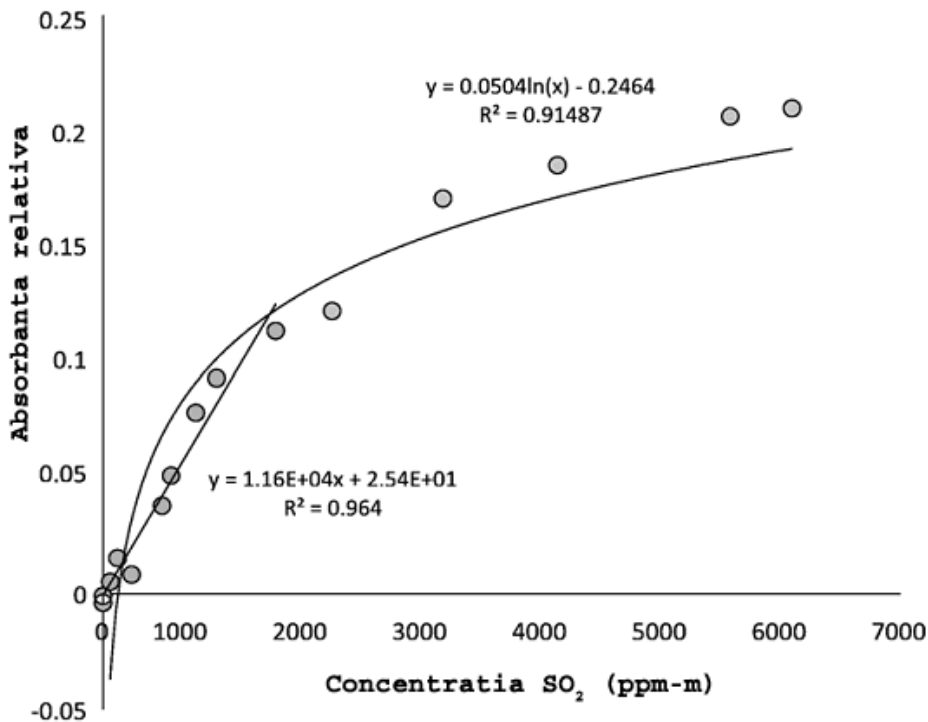


Figura 3.31. Absorbanta relativă față de concentrația de SO₂. [17]

Imaginile UV au fost captate cu filtrul 307 nm și 326 nm a unei celule de presiune la concentrații ppm de SO₂ diferite. Secvența a fost fotografiată la timpi de expunere optimi pentru fiecare filtru. Curbele de calibrare sunt liniare la concentrații mai mici de SO₂, dar se abat de la legea Beer-Lambert la concentrații de peste aproximativ 1500 de ppm.

Pantele pentru imaginile cu un DN maxim între 4400 și 7300 se modifică cu $2,13 \times 10^{-5}$, iar diferențele de înălțime y diferă cu 0,0128, o variație de aproximativ 20%. Timpii de expunere pentru aceste imagini sunt la distanță de 0,2 s. Modificările în pantă și diferența de înălțime y între imaginile cu un DN maxim de 55000 și 65000 înregistrări sunt numai de $2,00 \times 10^{-6}$ și respectiv, $1,20 \times 10^{-3}$. Timpii de expunere ai acestor imagini diferă cu 0,5 s, dar modificarea procentuală în ceea ce privește panta și diferența de înălțime y este de maxim 1%. Imagini cu DN mai mare de 32500 înregistrări converg spre o singură curbă de calibrare.

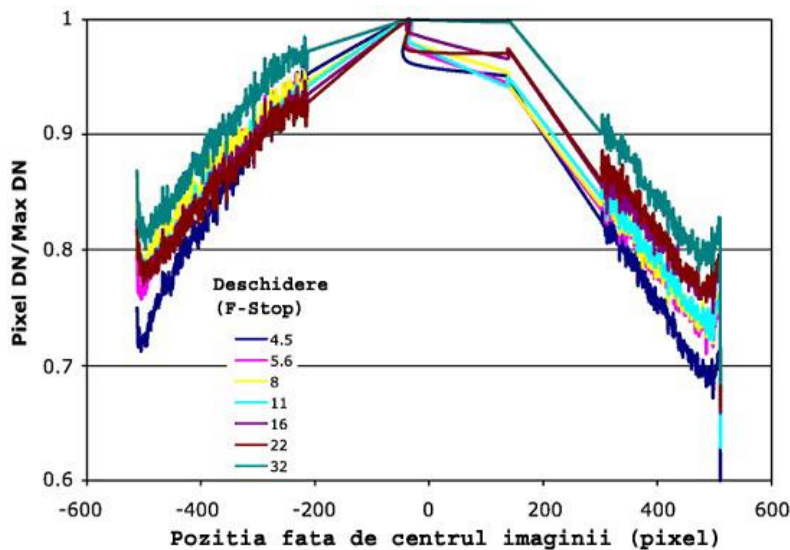


Figura 3.32. Iluminarea relativă pentru imaginile camerei UV captate cu șapte diametre diferite de diafragmă (numere stop F). Axa y reprezintă raportul de iluminare relativă, unde 1,00 este valoarea cea mai strălucitoare de pixeli din imagine[17]

În figura 3.32 se indică rezultatele experimentelor abordând variația procedurală în timp de expunere optime sau identici în timp ce se folosesc filtrele 307 nm și 326 nm [63]. Ambele grafice din figura 3.33 ilustrează șase imagini cu două filtre (un filtru 307 nm scăzând efectul filtrului 326 nm), fiecare set fiind captat în condiții identice de formare de imagini; singura variație este într-o metodă de timp de expunere (optim sau identic).

Liniile cu cea mai bună potrivire ale expunerilor optime sunt mai constante în pantă și diferență de înălțime y : abaterile standard pentru pantă și diferența de înălțime y pentru expunerea optimă sunt $8,54 \times 10^{-6}$ și, respectiv, 0,005, în timp ce setul captat cu aceiași timp de expunere au abateri standard pentru pantă și diferența de înălțime y de $1,47 \times 10^{-5}$ și, respectiv 0,009. Valoarea medie r^2 pentru liniile cu cea mai bună potrivire cu expunere optimă este de 0,949, iar valoarea medie r^2 pentru liniile cu cea mai bună potrivire cu aceeași expunere este de 0,767. Rezultate metodei cu expunere optimă indică un răspuns mai sigur pentru relația absorbție-concentrație, observându-se o dispersie mai redusă în punctele de date ale celulei de calibrare pentru fiecare imagine.

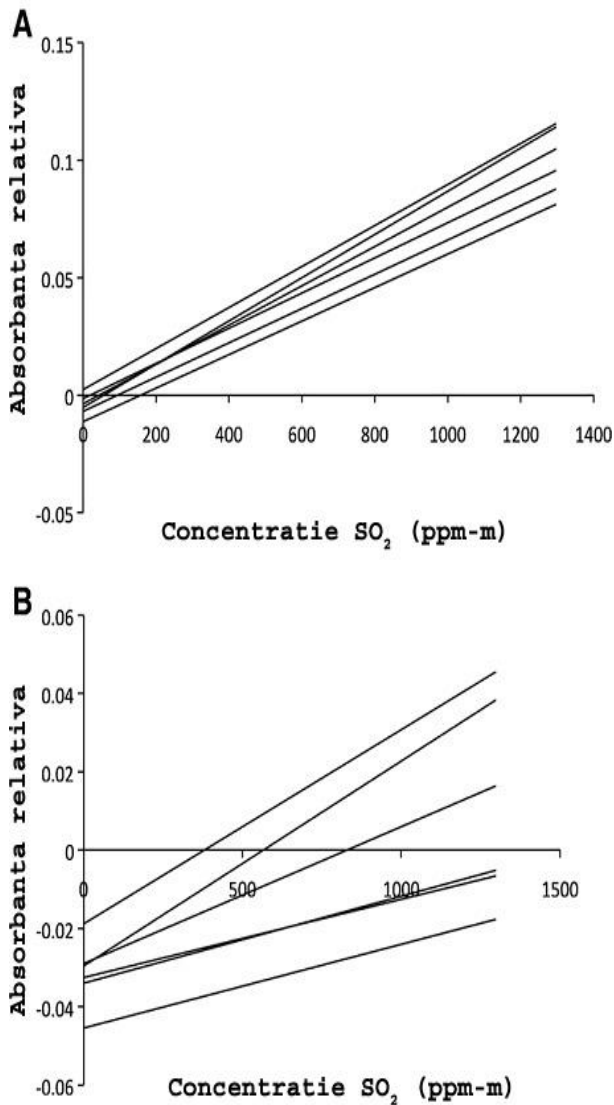


Figura 3.33. Relațiile de calibrare determinate de valorile absorbanței relative (imagini de diferențiere de la filtrul 307 nm și filtrul 326 nm) a celulelor de calibrare de la imaginile captate la timpi de expunere optimi pentru fiecare filtru (A) și timpi de expunere reprezentând un compromis între fiecare filtru (B)[17]

3.3.2.2 Testarea pe teren

În timpul testării pe teren a camerei UV la centrala termică pe cărbune, datele CEMS au raportat concentrații de SO₂ cuprinse între 429,1 ppm și 458,2 ppm.

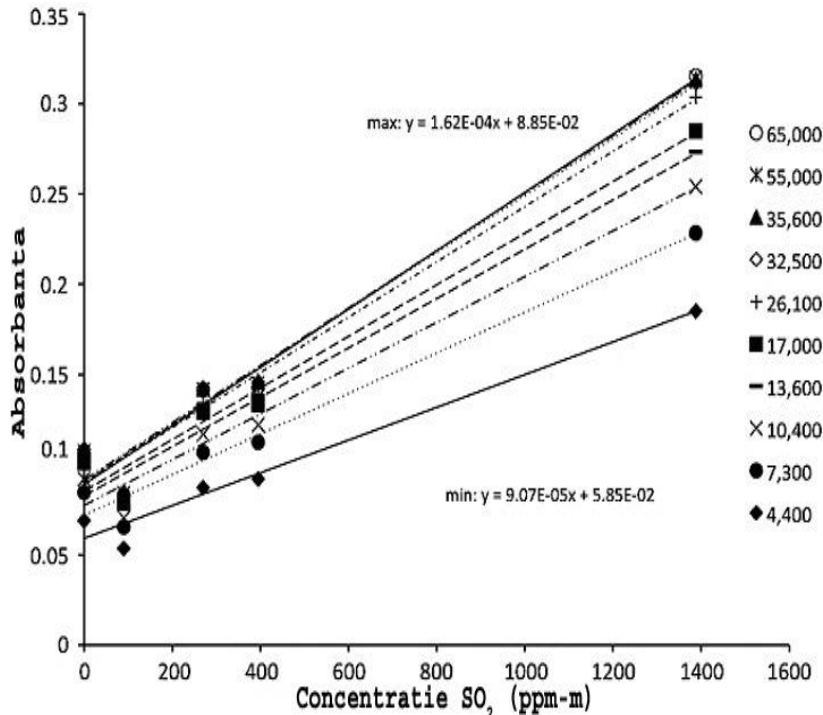


Figura 3.34. Absorbanta față de relațiile concentrației lungimii traiectoriei și dependența lungimii expunerii (descrise aici ca înregistrarea DN maximă din imagine)[17]

Eterogenitatea coloanei se poate observa ilustrând inaplicabilitatea comparației unui singur pixel cu proba CEMS într-un singur punct. În figura 3.34 sunt reprezentate variațiile concentrației în funcție de absorbantă.

Tehnicile de teledetecție au devenit un instrument folosit pe scară largă pentru măsurarea emisiilor de SO₂. Se pot distinge două tipuri de tehnici de teledetecție care utilizează spectroscopia de absorbție. Instrumentele de teledetecție active folosesc surse artificiale de lumină, în timp ce instrumentele pasive folosesc surse naturale, precum soarele, luna, lumina solară dispersată în atmosferă sau lava fierbinte în infraroșu.

Toate tehnicile de teledetecție pasive și majoritatea celor active nu măsoară concentrația c a unei substanțe, ci mai degrabă densitatea coloanei înclinată S . Densitatea coloanei înclinată S reprezintă integrala concentrației absorbantului pe traiectoria pe care o urmărește radiația în drumul său de la sursă de lumină la instrument. Instrumentele active au avantajul că traiectoria lor de lumină este bine definită. Radiația pornește de la sursa artificială, poate fi redirecționată către un anumit sistem optic, trece prin coloana de erupție vulcanică, iar apoi este captată de sistemul optic de recepție și analizată de către instrument. Astfel, se măsoară densitatea optică τ a liniilor de absorbție caracteristice a urmelor de gaz, care este proporțională cu densitatea coloanei înclinată S definită ca

$$S = \int_L c(l)dl \quad (3.81)$$

Lungimea traiectoriei optice L se cunoaște, concentrația medie de-a lungul traiectoriei poate fi ușor calculată ca $c_{avg} = S/L$. Exemple pentru astfel de configurații de teledetecție includ Spectroscopia activă de Absorbție Diferențială pe Traiectorie Lungă (LP-DOAS) și sisteme active în Infraroșu cu Transformantă Fourier (FTIR).

Unele instrumente pasive de teledetecție, precum FTIR cu ocultație solară (sau DOAS cu soare direct, măsoară lumina directă de la o sursă de lumină naturală. Și pentru aceste configurații, traiectoria optică este bine definită. Cu toate acestea, coloana trebuie să fie între sursa de lumină (de ex. soare) și instrumentul pentru măsurătorile care urmează să fie efectuate. Prin urmare, instrumentele trebuie să fie mobile și capabile să se adapteze la poziții diferite de soare și coloană sau pot fi folosite numai în condiții corespunzătoare, ceea ce reduce considerabil eficiența colectării. Prezența norilor în atmosfera înconjurătoare sau a unei cantități mari de aerosoli și picături de apă condensată în coloană, de asemenea, pot afecta traiectoria optică bine definită, crescând foarte mult procesele de dispersie.

O sursă de lumină alternativă pentru instrumentele pasive este lumina soarelui dispersată în atmosferă. Avantajul acestei surse este faptul că este mereu prezentă pe parcursul zilei, astfel încât instrumentele staționare să poată măsura emisiile de SO₂ ori de câte ori coloana se află în raza de vizualizare a instrumentului. Lumina soarelui dispersată s-a folosit în cazul tehnicilor precum spectroscopia de corelație (COSPEC, DOAS pasiv în modurile de scanare și mobile, DOAS cu formare de imagini și noua cameră SO₂).

Obiectivul acestui studiu este să cuantifice efectele transferului radiativ privind rezultatele obținute prin măsurători pasive de teledetecție folosind lumina soarelui dispersată. Folosim un model direct numeric pentru a simula transferul radiativ în și în jurul unei coloane de erupție vulcanică virtuale care conține diferite cantități de SO₂ și aerosoli și a determina efectele parametrilor, precum geometria de măsurare și condițiile SO₂/aerosoli, în cazul unei măsurători ipotetic de teledetecție de SO₂ în zona lungimii de undă a ultra-violetelor. Rezultatele modelului sunt ulterior comparate cu măsurători de pe teren și se prezintă prima metodă de corecție cantitativă a erorilor de măsurare cauzate de transferul radiativ.

Conceptul de „factor de masă de aer” (AMF) se folosește pentru a descrie gradul de sensibilitate al unui instrument de teledetecție la abundența de urme de gaze în interiorul unei coloane de erupție vulcanică. AMF este definit ca raportul dintre densitatea coloanei înclinată măsurată $S_{measured}$ și densitatea coloanei teoretică $S_{straight}$ pe o linie dreaptă prin coloana de erupție vulcanică de-a lungul direcției de vizualizare a instrumentului.[7]

$$AMF = \frac{S_{measured}}{S_{straight}} \quad (3.82)$$

Conform definiției date de ecuația 3.82, un AMF de 1 indică faptul că traiectoria efectivă a fotonului este egală cu lungimea unei linii drepte prin coloană de-a lungul direcției de vizualizare a instrumentului. Valorile mai mari decât unitatea indică traiectoriile extinse ale fotonului, de ex., cauzate de fenomene multiple de dispersie în interiorul coloanei. Cu toate acestea, AMF sub 1 sunt de asemenea

posibile și implică o traiectorie medie scurtată în coloană, de exemplu cauzată de o parte din fotonii măsurați care ratează în întregime coloana.

Traietoria optică de la soare la instrument este diferită pentru fiecare foton individual măsurat în aplicațiile luminii dispersate. Deși este imposibil de știut traiectoria optică a unui foton specific, acest lucru nu este necesar deoarece semnalul înregistrat de instrument reprezintă deja media multor măsurători statistice independente. În unele cazuri, o linie dreaptă de-a lungul direcției de vizualizare a instrumentului poate fi presupusă ca traiectoria luminii efective a radiațiilor incidente în regiunea atmosferică de interes. Până în prezent, această abordare a fost folosită pentru a evalua măsurătorile DOAS pasiv, COSPEC și cu camera SO₂ a coloanelor SO₂ de erupție vulcanică.

3.3.2.3 Traietoria de lumină idealizată

În acest caz, fotonii sunt dispersați numai spre instrument în atmosfera din spatele coloanei. Apoi aceștia trec prin coloană în linie dreaptă și sunt detectați de instrument. Cu toate acestea, chiar și în condiții ideale de măsurare în absența norilor și aerosolilor și în cazul unei coloane transparente, nu toți fotonii urmează traiectoria optică idealizată.

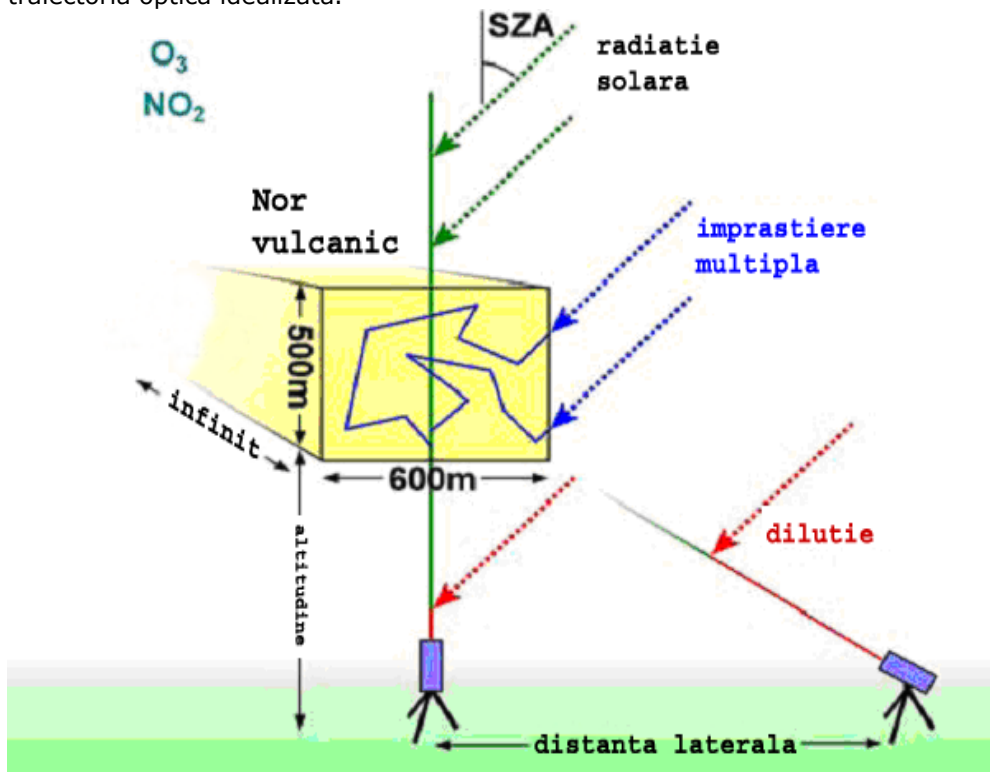


Figura 3.35 Schema geometriei de măsurare simulată[40]

Coloana de erupție investigată din figura 3.35 s-a extins la 500 m în direcție verticală, 600 m în direcție orizontală, a fost infinit lungă și a fost localizat la diferite altitudini deasupra instrumentelor. În interiorul coloanei, s-au presupus concentrațiile variabile de SO₂ și aerosoli. În afara coloanei, s-a luat în considerare absorbția de O₃ și NO₂. Unghiul zenital solar (SZA) a fost de 30° pentru toate scenariile. De asemenea, sunt indicate diferite traiectorii posibile de fotoni de la soare la un instrument de teledetecție. Traiectoria verde indică traiectoria idealizată a fotonilor: lumina este dispersată deasupra coloanei de erupție vulcanică și trece prin coloană în linie dreaptă. Traiectoria roșie reprezintă efectul de diluare. Lumina este dispersată între coloană și instrument și diluează efectiv densitatea măsurată a coloanei. Traiectoria albastră descrie cazul dispersiilor multiple care au loc în interiorul coloanei provocând o prelungire a traiectoriei luminii și, prin urmare, o îmbunătățire a densității măsurate a coloanei.

3.3.2.4 Efectul de „diluare a luminii”

În atmosferă, lumina soarelui este dispersată de moleculele din aer (dispersia Rayleigh și Raman) și de particulele de aerosoli și picături de apă condensate, de exemplu în nori (dispersia particulelor). Instrumentele de teledetecție folosesc acest lucru ca pe o sursă de lumină pentru a măsura compoziția coloanelor de erupție vulcanică de la o distanță sigură de un crater posibil activ. Cu toate acestea, în afară de fotonii dispersați în spatele coloanei, unii fotoni vor fi, de asemenea, dispersați în direcția instrumentului între instrument și coloană. Acești fotoni nu au trecut prin coloană și, prin urmare, această contribuție nu conține structuri de absorbție spectrală care provin din constituenții coloanei. Lumina dispersată între instrument și coloană diluează efectiv densitatea coloanei măsurată.

3.3.2.5 Dispersia multiplă

Coloanele de erupție vulcanică adesea conțin o cantitate mare de vapori de apă. În timp ce este în formă gazoasă, la temperaturi ridicate, acești vapori de apă se pot condensa pe măsură ce se răcesc după ce au fost emiși în atmosferă. Coloanele de erupție vulcanică nu sunt transparente, ci mai degrabă mai mult sau mai puțin opace. Acest lucru, împreună cu prezența particulelor solide, precum cenușa, poate provoca o creștere a dispersiei particulelor în comparație cu coloanele transparente. Astfel, fotonii pot fi dispersați de mai multe ori în interiorul coloanei de erupție vulcanică, iar lungimea traiectoriei din interiorul coloanei poate fi considerabil mai mare decât linia dreaptă. Traiectoriile fotonilor în nori pot fi îmbunătățite cu mai multe zeci de kilometri în norii stratus mari, groși optic. În funcție de mărimea și opacitatea unei coloane de erupție vulcanică specifică, acest lucru poate să apară și aici. În cazul absorbției foarte puternice în interiorul coloanei (fie din urmele de gaze sau aerosolii de absorbție), traiectoria medie a fotonilor în coloane poate fi, scăzută de prezența picăturilor de apă și aerosolilor. Acești constituenți pot crește probabilitatea ca fotonii care au pătruns numai prin părțile exterioare ale unei coloane să ajungă la instrument.

3.3.2.6 Modelarea transferului radiativ

La modelarea transferului radiativ, se simulează o măsurare prin furnizarea modelului cu toate informațiile relevante pentru măsurare și calculul ulterior a ceea ce ar detecta instrumentul virtual. Avantajul studiului de modelare față de o măsurare adevărată este faptul că permite implementarea condițiilor atmosferice definite de utilizator. Astfel se oferă o platformă perfectă pentru studiile de sensibilitate, pe măsură ce toți parametrii relevanți pentru o măsurare simulată poate fi variată individual păstrând în același timp ceilalți constanți.

3.3.2.7 Descrierea modelului

În acest studiu, s-a aplicat modelul transferului radiativ TRACY-II pe mai multe scenarii vulcanice. TRACY-II este un model Monte Carlo invers care generează un ansamblu reprezentativ de traiectorii individuale de fotoni prin atmosfera simulată, fiecare reprezentând o soluție a ecuației de transfer radiativ monocromatic. Modelul tridimensional a fost scris la Institutul de Fizică a Mediului de la Universitatea din Heidelberg, fiind validat cu alte modele de transfer radiativ și măsurători. Avantajul abordării Monte Carlo este faptul că modelul este aproape de procesele fizice fundamentale, prin urmare rezultatele modelului fiind cele mai realiste.

Pentru calcul, atmosfera este împărțită într-o rețea discretă tridimensională de voxelii. Un instrument se plasează într-o anumită poziție în această atmosferă simulată, calculându-se un număr reprezentativ N de traiectorii posibile între instrument și soare conform proceselor fizice care determină transferul radiativ în atmosferă. Apoi, intensitatea radiației I care ajunge la instrument se calculează din suma contribuțiilor individuale w_i care ajung de-a lungul fiecărei traiectorii i .

$$I = \frac{1}{N} \sum_i w_i \quad (3.83)$$

Deoarece atenuarea intensităților traiectoriei individuale w_i prin absorbție poate fi exprimată în funcție de Legea Beer-Lambert, derivata intensității totale I în ceea ce privește coeficientul de absorbție α_v în fiecare voxel v (definită ca produsul dintre secțiunea transversală a absorbției σ și concentrația absorbantului c_v) produce relația:

$$\frac{d}{d\alpha_v} I = \frac{1}{N} \sum_i w_i L_{i,v} \quad (3.84)$$

$L_{i,v}$ reprezintă lungimea traiectoriei luminii a traiectoriei i în voxelul v . Valoarea probabilității ponderate a intensității a lungimii traiectoriei luminii L în voxelul v poate fi determinată prin calcularea derivatei negative a logaritmului intensității I .

$$-\frac{d}{d\alpha_v} \ln(I) = \frac{\sum_i w_i L_{i,v}}{\sum_i w_i} = \langle L_v \rangle \quad (3.85)$$

Densitatea simulată a coloanei înclinată S este apoi ușor de obținut prin însumarea densităților coloanei ponderate a intensității din fiecare voxel individual

$$S = \sum_v c_v \langle L_v \rangle \quad (3.86)$$

Odată ce se obține densitatea coloanei S , împărțirea prin densitatea coloanei drepte dată de geometria de măsurare a scenariului respectiv produce AMF în conformitate cu ecuația 3.82. Datorită importanței SO₂ în calitate de traser al activității vulcanice, în general, încercări ale modelului transferului radiativ au fost efectuate pentru lungimi de undă între 300 nm și 325 nm, intervalul spectral în care absorbția de SO₂ a luminii solare dispersate este cea mai proeminentă. TRACY-II este un model monocrom, astfel doar o lungime de undă poate fi modelată într-un moment dat. Acest proces a fost repetat timp de 21 de lungimi de undă între 300 și 325 nm pentru a cuantifica dependența lungimii de undă a transferului radiativ. Conform celor indicate în figura 3.36, lungimi de undă au fost alese pe și între liniile de absorbție ale SO₂ pentru a permite o descriere a cum ar arăta fiecare bandă de absorbție individuală într-un spectru de măsurare DOAS.

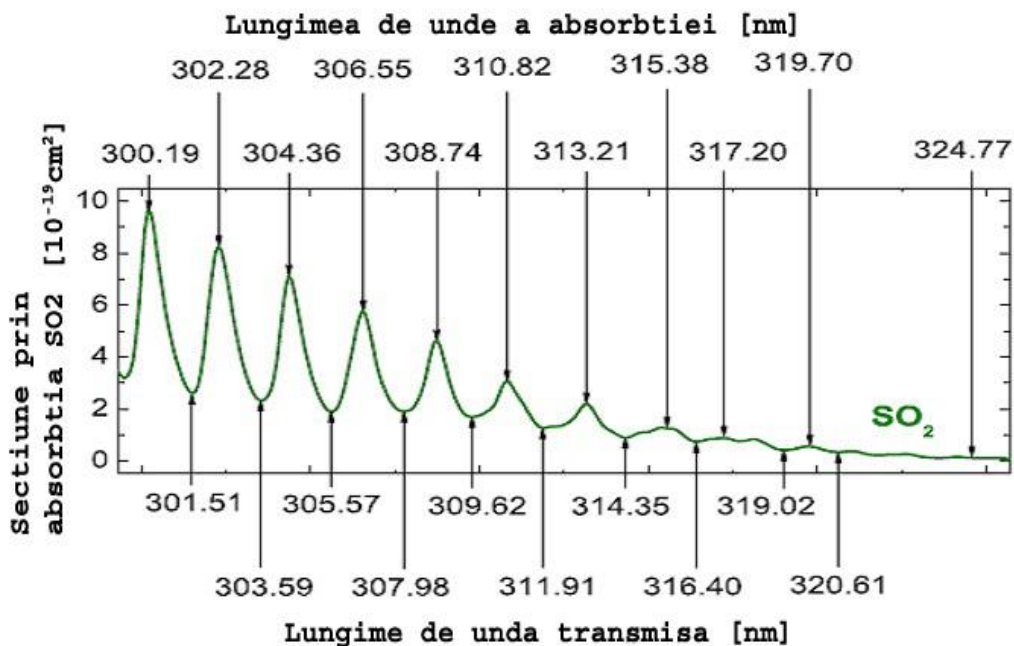


Figura 3.36 Secțiunea transversală a absorbției de SO₂ (linia verde). Lungimile de undă modelate sunt indicate cu săgeți[40]

Tabelul 3.4 Prezentare generală a scenariilor investigate de transfer radiativ. În fiecare scenariu, doar un singur parametru a fost variat pentru a determina gradul de sensibilitate al transferului radiativ în ceea ce privește această cantitate

Scenariu	Altitudinea coloanei	Distanța laterală a instrumentului	Densitatea drepte de SO ₂ a coloanei	Încărcătura cu aerosoli de a coloanei	Încărcătura cu aerosoli a stratului marginal
Coloană transparentă, geometrie verticală (TV)	variabil	0 km	20 ppm m	nici una	nici una
Coloană transparentă, geometrie diagonală (TD)	500 m	variabil	~ 20 ppm m	nici una	nici una
SO ₂ variabil, geometrie verticală (SV)	1 km	0 km	Variabil	nici una	nici una
Aerosoli variabili, geometrie verticală (AV)	1 km	0 km	1,000 ppm m	variabil	nici una
Aerosoli ambiantali, geometrie diagonală (AA)	1 km	3 km	1,000 ppm m	8 km ⁻¹ AEC	variabil

La simularea spectrelor reale de măsurare trebuie luat în considerare AMF pe lungimile de undă de transmisie între liniile de absorbție.

S-au propus metode de corelație, în care două sau trei spectrometre UV statice măsoară variațiile în cantitatea de SO₂ dintr-o coloană de erupție. Seriile de timp de SO₂ rezultate sunt ulterior corelate pentru a determina viteza de transport. Soluția ideală pentru ambele provocări este formarea imaginii conținutului de SO₂ al coloanei de erupție. Imagistica de înaltă frecvență ar permite determinarea simultană atât a cantităților de profil SO₂ instantanee complete cât și a vitezei vântului, utilizând abordarea de corelație pe zeci până la sute de puncte. Dezvoltarea tehnicilor de scanare automată folosind sistemul de spectrometru UV miniatural a permis schimbarea treptată în rezoluția temporală a măsurătorilor de flux, reducând perioada de la câteva zile la câteva minute.

Prima abordare a formării imaginii a SO₂ la sol a coloanelor de erupție a fost tehnica DOAS de imagistică în care o linie în spațiu este proiectată spectral pe un CCD (circuit cu cuplaj de sarcină) sensibil la ultraviolete care permite determinarea exactă a SO₂ (și teoretic a BrO) de-a lungul liniei. Prin scanarea fizică a instrumentului linie cu linie (push-broom), se poate crea o imagine a conținutului de SO₂ din câmpul de vizibilitate. Acest lucru s-a demonstrat cu succes pe Etna, dar cu o frecvență de formare de imagini destul de lentă, de ordinul a aproximativ 20 de minute.

COSPEC sau sistemele de spectrometru UV miniatural (de exemplu mini-DOAS utilizate în măsurătorile fluxului de SO₂ al gazelor vulcanice utilizează lumina solară UV dispersată ca sursă de radiații și de absorbție UV a SO₂ între 300 și 320 nm. După cum se poate observa în figura 3.37, lungimile de undă transmise de filtrul A sunt puternic afectate de prezența SO₂, în timp ce, cele transmise de filtrul B nu sunt. Această diferență este sursa fundamentală a sensibilității la SO₂. Măsurarea luminii solare dispersate (lumina cerului) cu un detector de bandă largă

sensibil la UV folosind fiecare filtru separat produce două intensități măsurate ale căror raport depinde de cantitatea de SO₂ prezentă pe traiectoria optică.

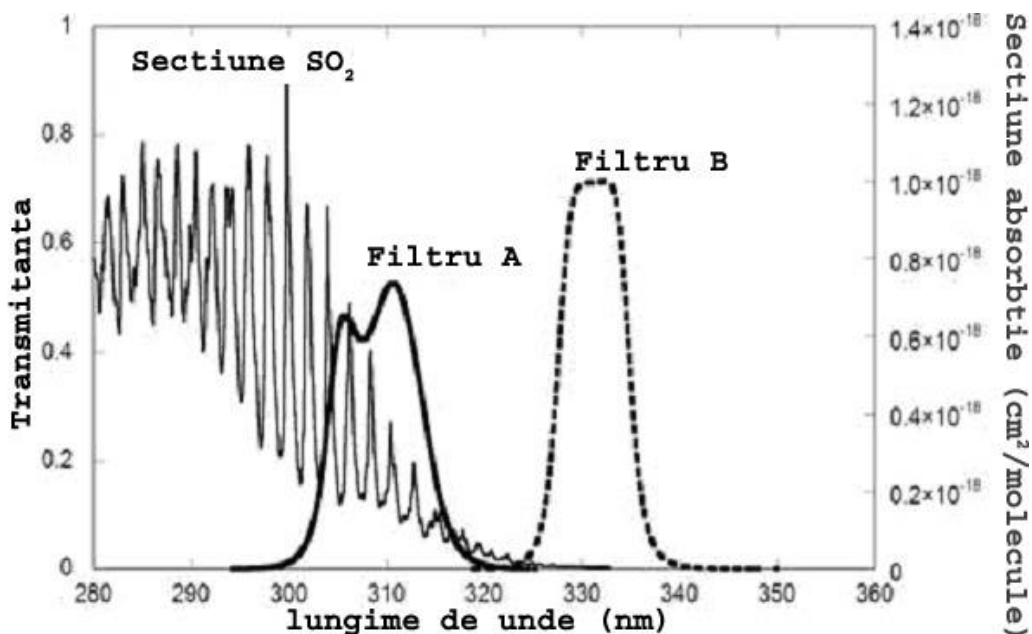


Figura 3.37 Transmitanța filtrelor A și B cu absorbție SO₂[53]

Campaniile pe teren cu COSPEC și DOAS au contribuit la înțelegerea proceselor de erupție și monitorizării înainte de erupții. COSPEC și DOAS folosesc lumina solară difuză ca sursă de lumină UV și absorbția selectivă a luminii de către SO₂. Secțiunile coloanei emise sunt luate dintr-o poziție fixă, prin rotirea instrumentului sau menținând senzorul stabil și scanând coloana, traversând-o cu un automobil, o barcă sau un avion. Fluxurile de SO₂ sunt derivate din măsurătorile sarcinii transversale și vitezei coloanei[35].

Tehnicile de scanare prezintă limitări semnificative și acestea pot fi diminuate dacă se măsoară coloanele cu mai multe instrumente. SO₂ începe să se disperseze și să se transforme în aerosoli de acid sulfuric după intrarea sa în atmosferă și ca atare măsurătorile pe direcția vântului pot subestima progresiv ratele efective de emisie. Limitările metodei de scanare pot influența două obiective comune ale monitorizării de SO₂: (1) necesitatea colectării continue și corecte de date pentru o perioadă lungă de timp; (2) nevoia identificării și evaluării fluctuațiilor de emisii.

Îmbunătățirile semnificative la monitorizarea de SO₂ vulcanic pot fi posibile prin imagistică (formare de imagini) care surprinde în mod ideal mare parte dintr-o coloană de erupție vulcanică într-o simplă imagine și trasează variațiile spațiale în conținutul de SO₂. Au fost luate imagini staționare vizibile la coloanele următorilor vulcani: Villarica, Santiaguito, Fuego și Pacaya. S-au constatat diferite fluxuri de SO₂. Cei patru vulcani au fost folosiți ca locații pentru testările inițiale la camera UV. Camera UV s-a dovedit să fie în general practică și eficientă în cazul utilizării pe teren. Timpul de instalare este mai mic de zece minute, iar în mai puțin de jumătate de oră se pot capta câteva zeci de imagini.

Camera digitală cu UV captează imagini ale coloanelor de erupție vulcanică de SO₂ la o distanță de 5-10 km. Camera oferă un câmp vizual de câțiva km pătrați cu o rezoluție de pixeli, aproximativ de 1 respectiv 2 metri. Absorbantele măsurate ale coloanei sunt reduse la scară, concentrație distanță parcursă, utilizând măsurătorile celulei de calibrare pentru a obține o concentrație a coloanei înclinată de SO₂ pentru orice pixel de coloană, iar fluxurile de SO₂ pot fi calculate folosind vitezele măsurate ale coloanei. Deoarece semnalul coloanei scade odată cu distanța se presupune că lumina difuză a atmosferei care intervine diluează semnalul coloanei, reducând contrastul coloană fundal, ducând la o subestimare a concentrațiilor coloanei.

Camera UV se pretează a fi un bun instrument de monitorizare viitoare a SO₂, datorită instalării sale rapide, utilizării practice pe teren, prelucrării în timp util a rezultatelor și prețului relativ scăzut.

Rezoluția spectrală relativ mare (de obicei, mai puțin de 1 nm) a instrumentelor DOAS permite extragerea simultană a diferitelor categorii de urme de gaze. În afară de SO₂, atât BrO, cât și NO₂ au fost, de exemplu, detectate în coloanele de erupție vulcanică în trecut cu ajutorul observațiilor DOAS de lumină dispersată. Primele încercări de a obține imagini 2-D de urme de gaz (de NO₂, SO₂ și BrO) în UV au folosit abordarea Imaging-DOAS (I-DOAS), unde o coloană de pixeli de imagine este înregistrată simultan cu un spectrometru de formare de imagini. Apoi, imaginea 2-D se construiește prin scanarea direcției de vizualizare perpendicular pentru această coloană.

Tehnica I-DOAS combină avantajele DOAS cu capacitățile imagisticii, timpul de captare este relativ lung (de ordinul a 15 min) și instalarea instrumentului este relativ complexă. Mori, Burton și Bluth au demonstrat capacitatea măsurării distribuțiilor de SO₂ 2-D în coloanele de erupție vulcanică folosind o cameră simplă sensibilă la UV și filtre de bandă care transmit doar radiații la lungimi de undă semnificativ influențate de absorbția de SO₂. Sunt identificate anumite probleme și sunt descrise metode pentru îmbunătățirea atât a preciziei cât și a implementării tehnice a metodei[28].

Datorită robusteții sale mult îmbunătățite în ceea ce privește influența efectelor dispersiei și a absorbției în bandă largă, normalizarea densității optice este necesară ori de câte ori o încărcătură semnificativă de aerosoli este prezentă sau se produce condensarea apei într-o coloană de erupție vulcanică. Condiția ca SO₂ să fie dominant în absorbantul de bandă îngustă în fereastra de lungime de undă a filtrului A rămâne importantă. Numai atunci se pot extrage cu exactitate densitățile coloanei de SO₂.

Principiul de măsurare al camerei SO₂ se bazează pe abilitatea de a calcula densitatea coloanei de SO₂ S din densitatea optică τ . Datorită lățimii de bandă finită a filtrelor de interferență folosite, se poate determina numai o densitate optică ponderată τ^A (Ec. 3.25). Cu toate acestea, intensitatea de fundal I_N la fiecare pixel al imaginii este o condiție prealabilă a extracției. Obținerea acestei imagini de fundal este adesea dificilă, deoarece, în principiu, este nevoie de o imagine exactă a scenei observate în absența coloanei de erupție vulcanică.

Din moment ce acest lucru este, de obicei, imposibil de obținut, o abordare este să se aproximeze intensitatea de fundal I_N înregistrând o imagine de lângă coloana de erupție vulcanică. Această abordare funcționează bine pentru unghiuri zenitale solare mici și dacă fundalul este omogen, adică în absența norilor. Cu toate acestea, deoarece ambele funcții de fază de dispersie, Rayleigh și Mie, nu sunt simetric sferice, înregistrarea unei imagini de fundal într-o direcție de vizualizare

diferită decât în cea în care se măsoară coloana va produce inevitabil o distribuție de intensitate diferită, în special în cazul unghiurilor zenitale solare mari.

O a doua abordare pentru obținerea unei imagini de fundal adecvate este interpolarea valorilor de intensitate obținute pe fiecare parte a coloanei în timpul măsurării coloanei. Această metodă are avantajul că nu trebuie mișcată direcția de vizualizare a camerei. Având în vedere faptul că înregistrarea frecventă a imaginii de fundal este favorabilă din cauză că este influențată de unghiul zenital solar și condițiile de aerosoli/nori, această metodă este în mod clar avantajoasă. Cu toate acestea, tehnica are și un dezavantaj: la folosirea primei abordări, un sistem de corecție sofisticat de câmp plat nu este necesar, deoarece diferențele de sensibilitate de radiații pe detector cauzate de vignetarea optică, eficiența cuantică a pixelilor, precum și unghiul variabil de iluminare al filtrului se anulează la calcularea densității optice normalizate. Trebuie corectate numai decalajul detectorului și curentul de întuneric, ambele putând fi ușor de măsurat prin captarea de imagini întunecate. Totuși, în cazul celei de-a doua abordări, corecția câmp plat precisă trebuie aplicată oricăror imagini înainte ca intensitatea de fundal să poate fi interpolată din pixelii de pe fiecare parte a coloanei. Prin urmare, sensibilitatea relativă a fiecărui pixel de pe detector trebuie caracterizată cu atenție.

Camera SO₂ nu măsoară intensitățile spectrale $I(\lambda)$, ci numai intensitățile integrate $I_{M,A}$ și $I_{N,A}$. Acestea pot fi calculate prin integrarea intensităților spectrale $I(\lambda)$ în intervalul de transmitență al filtrului. Densitatea optică ponderată τ^*_A poate fi calculată. La unghiuri zenitale solare mai mari, eroarea va deveni și mai mare. Prin urmare, acest exemplu ilustrează faptul că o calibrare corectă a densității optice ponderate τ^*_A va depinde de unghiul zenital solar în momentul măsurării. O variație în coloana totală de ozon stratosferic va avea același efect, deoarece totalul densității coloanei de O₃ de-a lungul traiectoriei optice determină spectrul $I_S(\lambda)$ [51].

Iluminarea neperpendiculară poate face necesară o calibrare spațial neomogenă dacă unghiul de incidență de pe filtru nu este egal pentru toate radiațiile măsurate. Într-o configurare simplă a camerei SO₂, de exemplu filtrul de bandă poate fi pus în fața obiectivului camerei. Lumina provenind din diferite zone din câmpul vizual va trece prin filtru sub unghiuri diferite. Sensibilitatea (și calibrarea corectă) a camerei SO₂ va depinde direct de distanța r_d din centrul imaginii pe detector. În această configurare, unghiul de incidență θ pe filtru este fixat pentru fiecare pixel de pe detector. Această proprietate poate fi folosită pentru a face o corecție de prim ordin pentru efectul incidență-unghi. Corecția nu depinde numai de θ , ci și de alți parametri, ceea ce face corecția exactă dificilă. Pe măsură ce fiecare pixel primește lumină dintr-o serie de unghiuri, iar transmisia este suprimată pentru unghiuri mari, deplasarea lungimii de undă a curbei de transmisie efectivă este, mai puțin dependentă de poziția pixelului decât în configurarea inițială[71].

În afară de cele două configurări simple care utilizează fiecare un singur obiectiv de cameră, sunt posibile mai multe modele. Prin utilizarea obiectivelor multiple sau grupurilor de obiective, se poate dezvolta un sistem optic care minimizează unghiul de iluminare al filtrului. Un sistem optic ideal va conserva radianța. Radianța B care intră în sistem este o funcție a unghiului de vizualizare total a (B este proporțional cu a^2) și a primei diafragme în sistemul d_A (B este proporțional cu d_A^2). Odată ce a intrat în sistem, divergența traiectoriei luminii poate fi scăzută, dar crescând în același timp diametrul diaframelor/sistemului optic (altfel lumina se pierde). Propagarea perfect paralelă a luminii ar putea fi realizată numai pentru sistemele optice infinit mari în diametru sau pentru o diafragma infinit mică în partea din față a sistemului. Prin urmare, este imposibilă proiectarea unui

sistem optic cu un unghi de iluminare al filtrului perpendicular θ pentru toate fasciculele de lumină incidente[24].

Un fascicul paralel permite aplicarea unui filtru optic cu transmitanță periodică. În loc să se folosească filtrele de bandă menționate mai sus, transmitanța unui interferometru Fabry-Pérot dimensionat convenabil ar putea fi reglat astfel încât fiecare vârf de transmitanță să se întindă pe o bandă de absorbție a SO₂ în intervalul de lungimi de undă de aproximativ 300-315 nm (preselectat eventual prin utilizarea unui filtru de absorbție UV de emisie, vizibil suplimentar), îmbunătățind astfel în mod semnificativ raportul de zgomot al măsurării și făcându-l mult mai selectiv față de SO₂, chiar și în prezența altor absorbanți[99,100].

3.3.2.8 Sistem optic avansat

Un sistem optic avansat s-a proiectat pentru camera SO₂ pentru a reduce problemele de calibrare dependente de unghiul de iluminare. Sistemul optic este compus din două grupuri de obiective între care este localizat filtrul de bandă de interferență. O diafragmă reglabilă este centrată în punctul focal al primului grup de obiective[17]. Unghiul total de vizualizare al camerei este, dat de relația:

$$\alpha = 2 \arctg \left(\frac{R_{eff}}{f_1} \right) \approx \frac{2R_{eff}}{f_1} \quad (3.87)$$

unde R_{eff} reprezintă raza efectivă a primului obiectiv ($R_{eff} = R_1 - d_A/2$), f_1 reprezintă lungimea focală, iar d_A reprezintă diametrul diafragmei frontale. Presupunând că obiectul vizat (de ex. coloana de erupție vulcanică) se află la o distanță mare de cameră (comparativ cu lungimile focale utilizate de sistemul optic), o imagine virtuală a peisajului vizualizat este creat în planul focal opus al primului grup de obiective. Filtrul de bandă de interferență este poziționat în spatele acestei imagini, astfel încât praful sau neomogenitățile de pe filtru nu sunt în cadrul focalizării. În spatele filtrului, se folosește un al doilea grup de obiective pentru a proiecta imaginea virtuală pe detector. Poziția acestui grup de obiective în raport cu detectorul și imaginile virtuale poate fi variată, astfel încât imaginea să fie redusă la scară pentru a se potrivi pe detector. Marele avantaj al acestei configurații față de configurarea simplă este că unghiul de incidență θ de pe filtrul de bandă nu mai variază în funcție de pozițiile individuale ale pixelilor de pe detector. Deși se realizează o serie de unghiuri de iluminare, acest interval este identic pentru toți pixelii, ceea ce conduce la o curbă de transmisie a filtrului efectivă identică. Unghiul maxim de incidență θ_{max} este controlat de intrarea diafragmei d_A și de lungimea focală a primului grup de obiective f_1 , și, prin urmare, poate fi ușor modificat.

$$\theta_{max} = \arctg \left(\frac{2d_A}{f_1} \right) \approx \frac{2d_A}{f_1} \quad (3.88)$$

Ecuția (3.88) demonstrează explicit că o reducere la unghiul maxim de iluminare θ_{max} poate fi realizată numai prin reducerea razei B care trece prin sistem. Reducerea dimensiunii diafragmei d_A scade direct debitul de lumină ($B \sim$

d_A^2), iar o creștere a f_1 duce la o scădere a unghiului total de vizualizare α și la o reducere pătratică în radianța care intră în sistem ($B \sim \alpha^2 \sim 1/f_1^2$).

Datorită diafragmei relativ mici în partea din față a sistemului optic, radianța acceptată B a fost considerabil mai mică în acest sistem optic decât în cazul configurărilor anterioare (o diafragmă de intrare tipică pentru o configurare mai simplă este de 25 mm). Cu toate acestea, prin utilizarea unui detector CCD iluminat din spate sensibil la UV în locul unui cip standard iluminat din față, raportul semnal-zgomot a fost suficient pentru a obține o rezoluție temporală de ordinul Hz. S-a constatat că limitările principale ale măsurării temporale a rezoluției sunt digitalizarea camerei și timpul de citire, precum și timpul necesar pentru schimbarea filtrelor de bandă, și nu timpul de expunere în sine [70].

Din moment ce se folosește o singură cameră CCD pentru două filtre, s-a folosit același timp de expunere pentru cele două măsurători cu filtrul pentru operația simplă. S-a constatat că semnalul de la filtrul B a avut, de obicei un ordin de mărime mai mare decât cel de la filtrul A, astfel timpul de expunere pentru CCD a fost ales astfel încât semnalele de la filtrul B să nu fie saturate. Timpul de expunere a variat în funcție de diverși parametri, inclusiv meteo și ora din zi.

Pentru a calcula concentrațiile de gaz sau fluxurile de emisii vulcanice, cunoașterea drumului optic prin pană este necesar. În trecut, un foton prin calea penei a fost presupus, deși este cunoscut faptul că acest lucru nu a fost întotdeauna adevărat. Rezultatele arată că măsurătorile efectuate fără informații suplimentare în medie, căile de foton pot fi inexacte, în anumite condiții, cu posibile erori care acoperă mai mult de un ordin de mărime.

Erorile efective depind de parametrii cum ar fi distanța dintre instrument și pană, concentrația penei de SO₂, încărcarea penei de aerosol, precum și condițiile de aerosoli în atmosferă. Ca un exemplu, la o măsurare cu o cameră SO₂ se discută rezultatele care pot fi interpretate corect doar dacă transferul radiativ este considerat. O metodă care este prezentată pentru prima dată permite extragerea de fotoni în spectroscopie (adică DOAS) a măsurătorilor cu o rezoluție corespunzătoare. Prin permiterea pentru o lungime de undă ce depinde de densitatea pe coloană în timpul evaluării măsurătorilor DOAS, ne arată cum efectul transferului radiativ poate fi corectat folosind informațiile moștenite, disponibile în spectrul măsurat, sporind astfel precizia măsurătorilor DOAS de emisii vulcanice. Tehnicile de detectare de la distanță au devenit un instrument răspândit pentru măsurarea emisiilor vulcanice. Zona lor de desfășurare variază de la monitorizarea vulcanică și de risc până la studii științifice, specifice de activitate vulcanică și până chimică. Posibilitatea de a furniza imagini spațiale ale distribuției de SO₂ vulcanic pe o rată de cadre de 1 Hz, face din camerele SO₂ o tehnologie foarte promițătoare pentru monitorizările vulcanice și pentru studierea dinamicii penelor vulcanice în atmosferă [52].

Monitorizarea pericolului vulcanilor se face stabilind ca bază dioxidul de sulf SO₂ și interpretând deviațiile de la acea linie de bază. Modificările emisiilor de SO₂ din fundal pot indica modificări în sistemul vulcanic, adesea o prefigurare în activitatea de erupție. SO₂ este direcționat pentru teledetectie datorită abundenței în pană vulcanică, concentrației atmosferice scăzute (uzual mai puțin de 10 ppb) și banda de absorbție în ultraviolet (UV). Măsurarea emisiilor de SO₂ din vulcanii activi a devenit un pilon în monitorizarea și eforturile științifice de la dezvoltarea corelării spectrometrice COSPEC de la începutul anilor 1970. Monitorizarea SO₂ de la sol a fost invocată la început cu două instrumente: COSPEC și mai recent spectrometrul compact UV a fost adaptat la măsurătorile de la sol, folosind în comun tehnica DOAS de măsurare. Aceste instrumente folosesc lumina solară împrăștiată ca sursă de lumina UV și ia avantajele absorbției selective a luminii UV de către SO₂ [23].

Vulcamera este un program pentru determinarea fluxului de SO₂ al vulcanilor folosind camere cu ultraviolet[98]. Codul permite achiziția imaginii și prelucrarea ei după procesare. Observarea spectrometrică de la distanța a fluxurilor vulcanice de SO₂ sunt pilonul de observație vulcanică. Asemenea date consolidează înțelegerea dinamicii vulcanice și impactul degajării vulcanice în atmosferă. Camerele UV pot furniza numeroase beneficii precum: rezoluție ridicată, care permite captura evenimentelor explozive; posibilitatea de a rezolva operații eterogene și operații cu un singur punct. Imaginile cu camera pot fi utilizate la măsurarea directă a vitezei de transport, potențial o sursă majoră a incertitudinii în aceste măsurători. Codul de la Vulcamera conține 2 elemente: Vulcamera_aq și Vulcamera_post ce administrează achiziția imaginii și toate elementele postprocesării. Vulcamera este proiectat să opereze cu două camere simultan cu filtre centrate pe 310 nm și 330 nm. Este necesar să utilizăm 2 filtre în aceste observații pentru a compensa împrăștierea aerosolului și această abordare minimizează nepotrivirile temporale asociate cu schimbarea filtrului pe o singură cameră. Codul include caracterizarea vignetei prin colectarea imaginilor cu cer senin pentru a compensa dependența unghiulară de iluminare a pixelului; determinarea relației de calibrare între absorbanta și concentrația celulei de SO₂ ca să permită conversia din imaginile măsurate în hărți de concentrație ppm m; folosirea simultană a fluxului de date spectrometric de SO₂ pentru a calibra imaginile; în final reacția acestor operații în pagina principală a Vulcamera_post, conduce la calculul seriei timp a fluxului de SO₂ și a maselor de gaz asociate cu exploziile.

Vulcamera a fost de multe ori testată pe teren la vulcanii din sudul Italiei și se speră ca alții vor realiza semnificativul potențial al teledetecției folosind tehnologia camerei UV [98].

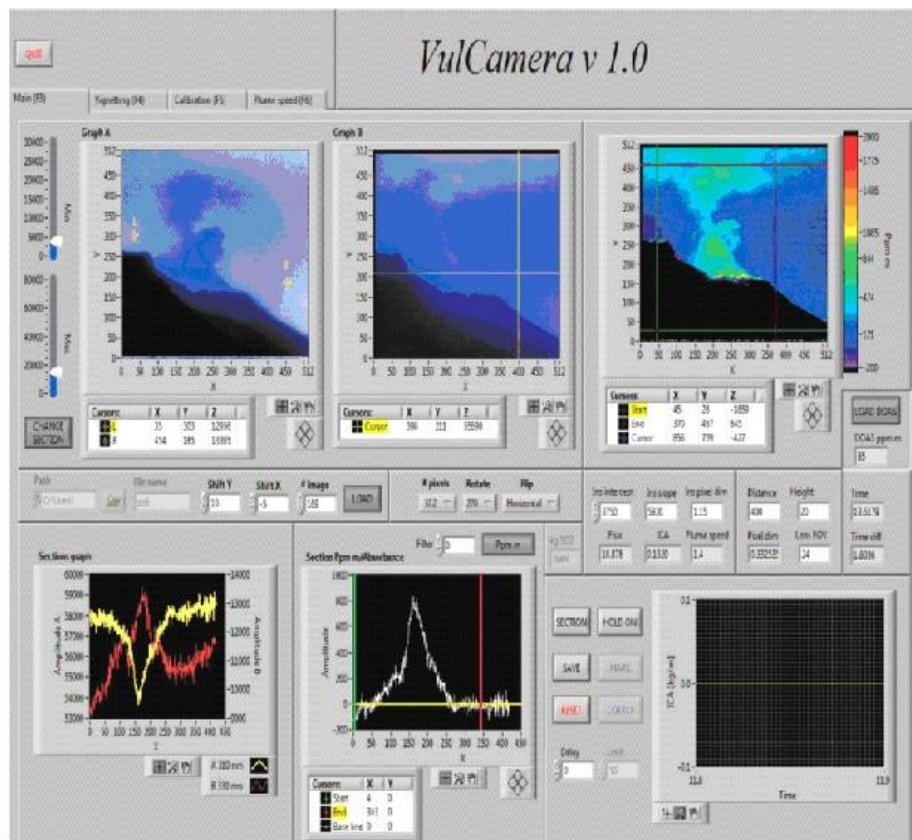


Figura 3.38 Interfața programului Vulcamera[99]

Erupțiile vulcanice pot emite mari cantități de dioxid de sulf și alte gaze în atmosferă. Aceste emisii gazoase sunt întâlnite mai ales în troposfera joasă (sub 3 km), dar pot de asemenea să atingă troposfera înaltă (înălțimi peste 3 km) sau stratosfera joasă (înălțimi de 8-15 km) unde gazele pot avea viață lungă (mai multe zile), în cazul SO₂ se transformă în aerosol acid și afectează bugetul radiativ al atmosferei. Deoarece erupțiile vulcanice au loc sporadic și sunt poziționate geografic la distanțe mari, măsurătorile din sateliții de pe orbită sunt mai potrivite pentru această misiune. Total Ozone Mapping Spectrometer (TOMS) și mai recent Global Ozone Experiment (GOME) și Ozone Monitoring Instrument (OMI) folosesc lumina solară ultravioletă pentru a determina cantitatea de SO₂ tip coloană [2].

Spectrometria de raze ultraviolete a fost implementată de peste 30 de ani pentru a monitoriza emisiile de SO₂ ale vulcanilor. Aceste date au furnizat informații valoroase despre condițiile magmatice din subteran care au fost utile în prognoza erupțiilor vulcanice. În ultima decadă spectrometrele au fost modernizate cu interfață USB deschizând o serie de posibilități pentru înțelegerea vulcanilor și impactul lor asupra atmosferei.

Fluxul de SO₂ poate fi calculat prin integrarea sumei pe o linie într-o imagine perpendiculară pe direcția vântului. Mărimea fiecărui pixel la o distanță a penei este

necesar pentru a calibra scala lungime; acest lucru este ușor de obținut prin comparație cu caracteristicile geometrice ale edificiului[53].

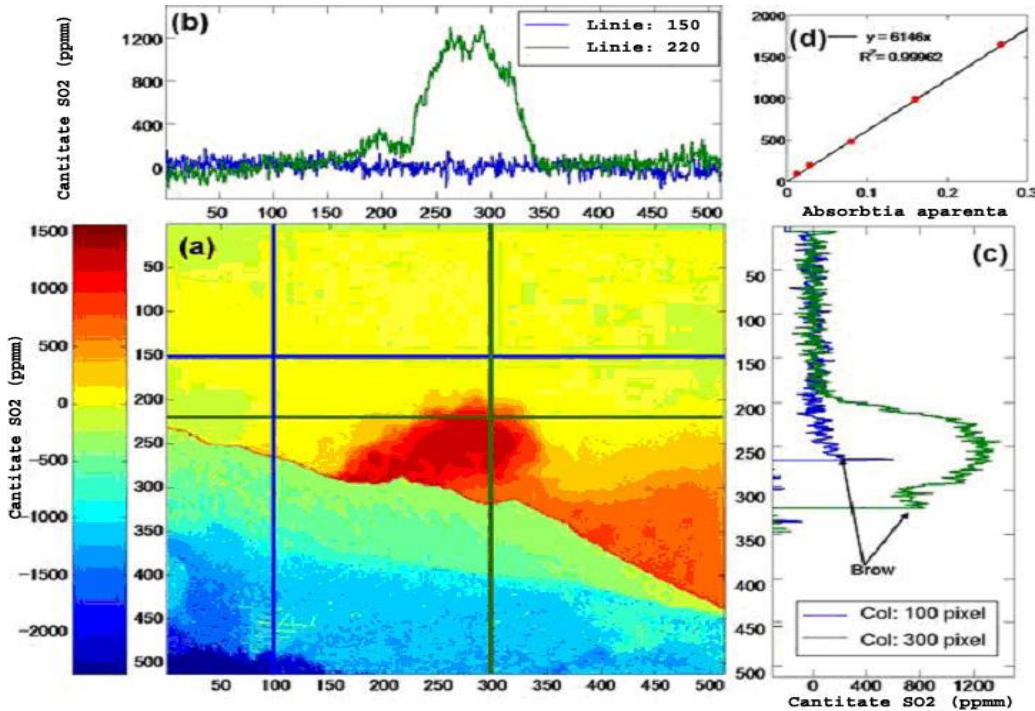


Figura 3.39 (a) Imaginea cantității coloanei de SO₂ a coloanei de erupție la ora 14:09:54. Scara din stânga este exprimată în ppm.m de SO₂. (b) Distribuție orizontală a cantității coloanei de SO₂ la rândul 150 (albastru) și 220 (verde) de pixeli. (c) Cantitatea coloanei verticale de SO₂ la coloana 100 (albastru) și 300 (verde) de pixeli. (d) Rezultatul unei calibrări a camerei SO₂ cu ajutorul celulelor de calibrare[53]

S-au calculat aceste profiluri folosind coloana 400 a imaginilor obținute la Sakurajima, pentru care dimensiunea pixelilor la punctul de măsurare a fost calculată la 2,44 m. Viteza vântului a fost derivată folosind intercorelările calculate pentru profilurile de SO₂ determinate la coloanele 300 și 400 sau la o distanță de 244 m (Figura 3.39a). Decalajele tipice de 45,6 s s-au dovedit a fi identice pentru diferite perechi de coloane ducând la o viteză a vântului constantă de 5,4 ms⁻¹. Deoarece fiecare pixel este lat de ~2,4 m, fiecare coloană din imagine este o înregistrare a emisiilor de gaze la fiecare 2,4 m/5,4 ms⁻¹ = 0,44 s sau ~ 2 Hz. Dacă direcția coloanei nu este perpendiculară pe direcția câmpului de vizualizare, atât cantitatea de profil a SO₂, precum și viteza coloanei constituie valori aparente și nu valori reale. Cu toate acestea, corecțiile unghiului pentru cantitățile de profil și vitezele coloanei se anulează în calculul fluxului; simpla înmulțire a celor două valori dă fluxul de SO₂ din vulcan. Figura 3.39b indică variațiile temporale ale fluxului de SO₂ de la craterul sudic al vulcanului Sakurajima între ora 14:06 și 14:12 pe data de 18 noiembrie 2005. Datele indică valorile fluxului de SO₂ la fiecare 5,7 s. Modificarea fluxului de la 1,2 kg/s la 5,8 kg/s s-a observat în numai ~35 s.

Erorile tipice pentru cantitățile de traiectorie calculată sunt de ~ 100 ppm.m, dar acestea se pot îmbunătăți în detrimentul rezoluției temporale prin medierea coloanelor adiacente și prin utilizarea timpilor de expunere automați și ajustabili.

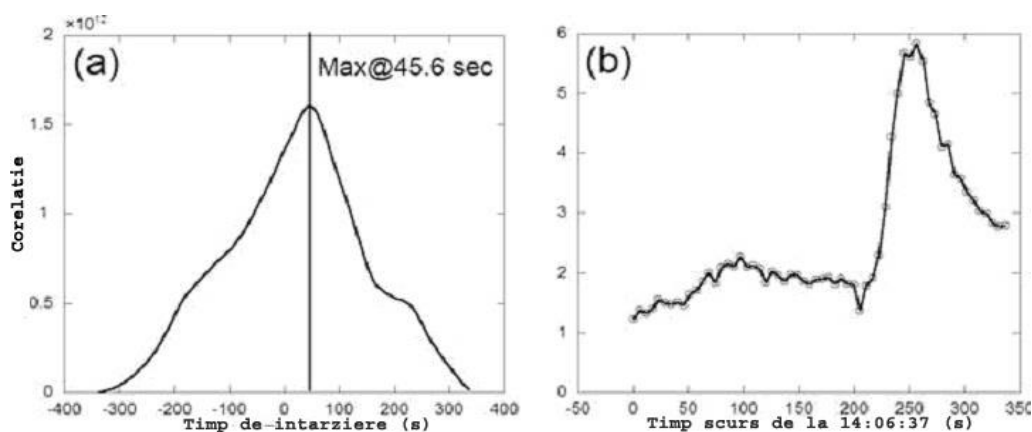


Figura 3.40. (a) Intercorelările variațiilor temporale în profiluri integrate calculate între a 300-a și a 400-a coloană verticală. Decalajul de timp derivat a fost de 45,6 s pe o distanță de ~ 244 m (100 pixeli la 2,44 m lățime fiecare), rezultând o viteză a vântului de $5,4 \text{ ms}^{-1}$. (b) Variațiile temporale ale fluxului de SO₂ obținute din profilurile celei de-a 400-a coloană verticală[53]

Automatizarea sistemului de schimbare a filtrelor ar trebui să fie simplă, permițând măsurători continue. Limita finală pe rezoluția temporală a fluxului de SO₂, măsurată cu camera SO₂, este lățimea pixelului la o distanță a coloanei împărțită cu viteza vântului. La o frecvență înaltă de luare de probe, difuzia ar fi factorul de limitare pentru soluționarea caracteristicilor variabile în timp. Erorile de 100 ppm se estimează pentru condiții de vizualizare bune, cu o condensare redusă de aerosoli în coloană, fără nori și observare la mică distanță (<3 km). Măsurătorile simultane cu un spectrometru UV ar permite corecția cantitativă pentru atenuarea prin aerosoli. O sursă potențială de interferență vine de la BrO, care absoarbe în regiunea filtrului B la 330 nm. Pentru un raport tipic SO₂/BrO de 1000, absorbanta relativă integrată de SO₂ din filtrul A comparativ cu BrO în filtrul B este de 50 la 1 și, prin urmare, efectul BrO se află în limitele de eroare ale tehnicii. Camera SO₂ reprezintă un pas înainte important pentru monitorizarea emisiilor de gaze vulcanice.

Tehnicile de scanare prezintă limitări semnificative. De exemplu, calculul vitezei vântului poate contribui până la 40% la indicele de eroare global, deoarece trebuie derivat independent din dispozitive de mână sau stații meteorologice locale care pot sau nu reprezenta datele despre vânt la înălțimea și locația reală a coloanei. Măsurarea coloanelor cu mai multe instrumente poate reduce semnificativ această problemă. Câmpul vizual este relativ mic pentru COSPEC și spectrometrele UV compacte, datele reprezentând o secțiune transversală localizată a coloanei în deplasare calculat în medie pe durata scanării. Prin urmare, variațiile în compoziția coloanei pe parcursul scanării sau modificările fluxului care apar mai repede decât rata de eșantionare nu pot fi recunoscute. Fluctuațiile ratelor de emisie calculate pot

duce la modificări ale condițiilor meteorologice și / sau modificări în procesele sau interacțiunile subterane magmatice. SO₂ începe să se disperseze și să se convertească chimic în aerosoli de acid sulfuric imediat după intrarea în atmosferă, prin urmare, măsurătorile pe direcția vântului pot subestima progresiv ratele efective de emisie.

Limitările metodei de scanare pot influența serios două obiective comune ale monitorizării de SO₂: (1) necesitatea colectării continue și corecte de date pentru o perioadă lungă de timp; și (2) nevoia identificării și evaluării fluctuațiilor de emisii. Emisiile măsurate de SO₂ pot varia datorită proceselor vulcanice sau ca urmare a amestecurilor neomogene cauzate de procesele de dispersie fizice și chimice în atmosferă. În timpul unui proces vulcanic, este dificil de evaluat modificările în degazeificarea de SO₂ pe baza măsurărilor sporadice sau de comparat aceste estimări sporadice de rate de emisii de SO₂ pe măsurători continue, precum seismicitate sau deformarea solului[69].

Îmbunătățirile semnificative la monitorizarea de SO₂ vulcanic pot fi posibile prin imagistică (formare de imagini), care surprinde în mod ideal mare parte dintr-o coloană de erupție vulcanică într-o singură imagine și trasează variațiile spațiale în conținutul de SO₂. Formarea de imagini UV prin satelit al SO₂ vulcanogenic a monitorizat cu succes activitatea vulcanică la scară largă de peste 25 de ani, iar cel mai avansat sistem până în prezent este OMI (Instrumentul de monitorizare al stratului de ozon).

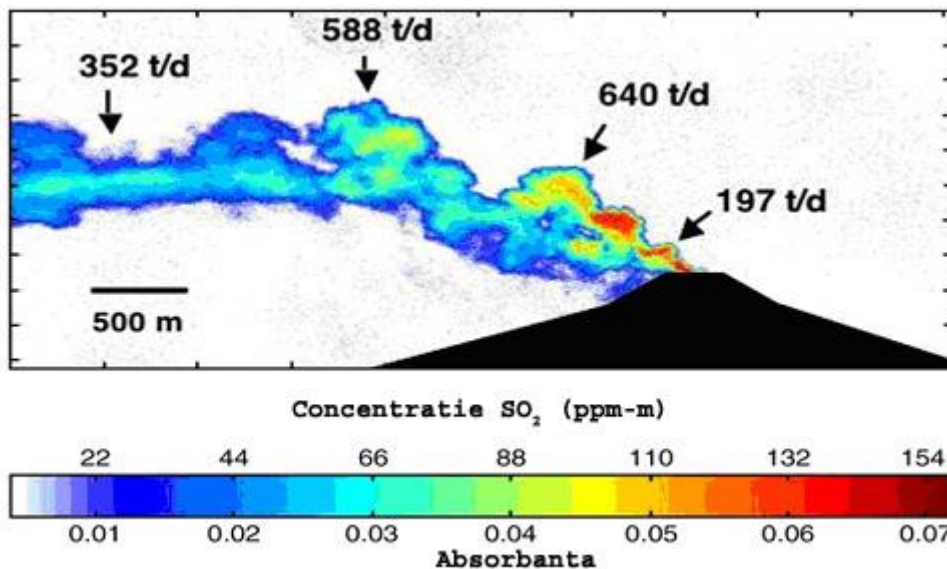


Figura 3.41. Imagine vizibilă la Villarrica făcută cu o cameră digitală de la o distanță de 16,5 km. [3]

Absorbanțele coloanei sunt reduse la scara rezultatelor celulei de calibrare. O viteză a vântului de $4,5 \text{ ms}^{-1}$ s-a calculat din măsurarea circulației porțiunilor distincte ale coloanei pe imagini consecutive. Fluxurile sunt calculate pe porțiuni diferite ale coloanei folosind o corecție ușoară a aerului, săgețile indicând orientarea

pe bucată transversală (t/d = tone pe zi). Cercurile albastre estompate sunt rezultatul unor particule sau pete pe sistemul optic al camerei.

Zonele de măsurare pe teren au fost selectate pentru a optimiza formarea imaginii coloanei, similar criteriilor folosite pentru scanarea amplasării spectrometrului: (1) direcția de vizualizare a camerei este aproape perpendiculară pe direcția coloanei, (2) ca fundal, se folosește o arie suficientă de cer liber, și (3) soarele este direct deasupra capului sau în spatele direcției de vizualizare a camerei. Pe teren, se folosește un colimator de carton pentru a reduce cantitatea de lumină directă și dispersată pe obiectiv; prin urmare, păstrând soarele în spatele camerei, se reduce zgomotul de fond, dar nu este o necesitate pentru formarea de imagini.

Din imaginea brută se generează o hartă de densitate a coloanei înclinată de SO₂. Se calculează media valorilor cerului din fundal pentru a elimina vignetarea – o stare în care intensitatea luminii scade spre marginile imaginii. În mare parte, vignetarea este un efect optic observat în cazul obiectivelor în care lumina oblică nu iluminează colțurile imaginii. Software-ul Maxim include un program interactiv care folosește puncte alese de utilizator pentru a recalcula cerul de fundal la intensitatea luminii aproape uniformă în întreaga imagine. Acest algoritm nu modifică în mod semnificativ magnitudinea coloanei față de raportul de fundal.

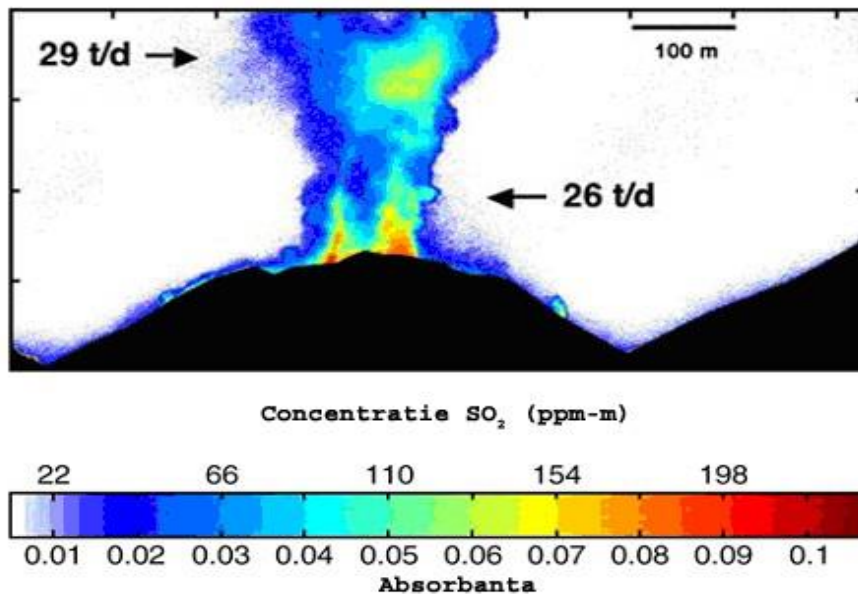


Figura 3.42 Imagine staționară vizibilă a coloanei Santiaguito captată de pe înregistrarea cu camera video. [3]

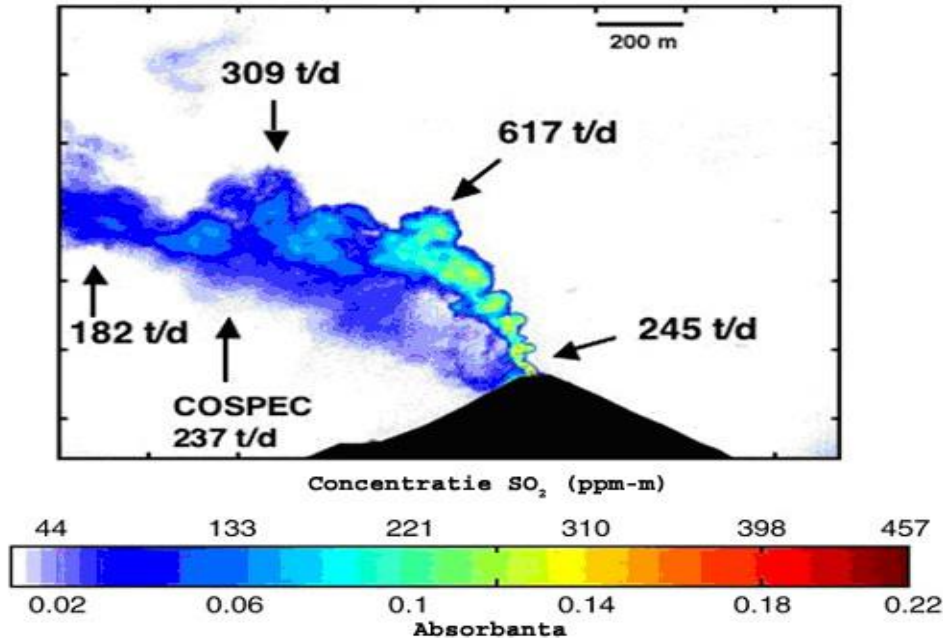


Figura 3.43. a) Imagine staționară vizibilă a coloanei Fuego captată de pe înregistrarea cu camera video de la o distanță de 7.08 km [3]

Dimensiunile unui pixel dintr-o imagine dată luată cu camera UV pot fi ușor de calculat, cunoscând mărimea și dimensiunea rezoluției CCD, distanța focală a obiectivului și măsurând distanța de la cameră la țintă. Prin urmare, viteza coloanei se calculează direct prin măsurarea distanței parcurse a caracteristicilor distinctive ale coloanei pe imagini temporizate consecutiv; cu toate acestea, direcția vântului trebuie să fie determinată independent pentru coloanele în derivă. Fluxurile de SO₂ pot fi calculate în mod similar cu metodele de scanare, prin integrarea abundenței pe transversale (corectare, după caz, pentru unghiuri de vizualizare neperpendiculare ale coloanei) și înmulțirea masei integrate cu viteza coloanei. Deoarece câmpul vizual al camerei cuprinde o secțiune largă a coloanei, mai multe secțiuni transversale (sau zone) pot fi evaluate pentru a elimina problemele de eterogenitate ale coloanei.

CCD-ul camerei și sistemul optic sunt sensibile la condițiile de intensitate a luminii din mediu și, prin urmare, la unghiul solar și lumina dispersată de la obiectele din jur (de ex., arbori, sol, structuri etc.). Calitatea imaginii (claritate, omogenitate la sol) se realizează cel mai bine având soarele în spatele direcției de formare a imaginii (în spatele operatorului). Pentru fiecare locație, cu condiții unice ale unghiului solar, condițiile cerului de fundal și nivelul de iluminare, timpul de expunere a fost reglat pentru a maximiza separarea coloanei de fundal.

Distanța de detecție și unghiul de vizualizare sunt importante, luând în considerare faptul că scanarea cu bază fixă și tehnicile de formare ale imaginilor măsoară „coloanele înclinate” prin pană, reprezentând cantitățile de SO₂ măsurate la unghiuri de vizualizare înclinate față de zenit și care trebuie să fie convertite în coloane verticale pentru a produce măsurători consecvente.

Testele au demonstrat că aceste camere UV pot detecta cu ușurință coloanele de SO₂, dar calculele inițiale ale fluxului de SO₂ au fost mai puțin de un ordin de mărime sub cele calculate de măsurătorile corespunzătoare efectuate cu COSPEC. S-a efectuat un set de teste pe teren la vulcanul Fuego, în care s-a observat că „contrastul” coloanei sau diferența procentuală dintre țintă și luminozitatea de fundal a scăzut exponențial cu distanța. Cu alte cuvinte, semnalul coloanei devine mai luminos (adică absorbanta coloanei scade) cu distanța, în timp ce nivelul de fundal rămâne relativ același. Prin relațiile absorbantei și ale legii lui Beer, absorbanta mai mică a coloanei se traduce printr-o estimare mai mică a abundenței de SO₂ și un flux de masă artificial mai mic.

Cauza pentru reducerea exponențială, în contrastul dintre coloană și fundal, odată cu creșterea distanței se datorează dispersiei moleculare (proporțională cu λ^{-4} , prin urmare mult mai importantă la lungimi de undă UV). Sursa de lumină poate fi lumină solară dispersată direct, lumina dispersată a cerului și lumina reflectată de la sol. Contrastul real/atenuat reprezintă o funcție exponențială a distanței și a coeficientului atmosferei de dispersie a volumului dependent de lungimea de undă. În cazul ultravioletelor, dispersia radiației prin moleculele de aer reprezintă un ordin de mărime mai mare decât în spectrul vizibil. Mori a constatat că acest efect, în intervalul 309-315 nm, a devenit semnificativ la o distanță de 1,5 km și a provocat până la 50% din subestimarea fluxurilor la lungimi de undă mai scurte[53].

Corecția simplă a dispersiei ne permite să cuantificăm fluxurile de SO₂ pentru a testa utilitatea camerei, dar mai multe experimente sunt necesare pentru a valida metoda. În cazul abordării cu filtru de curent în bandă largă, corecția difuziei ar fi necesară pentru fiecare situație pe teren. Activitatea continuă se concentrează pe efectele opacității atmosferice asupra răspunsului camerei, încorporării filtrelor suplimentare pentru camera UV cu bandă îngustă (de ex., în cazul instrumentelor prin satelit, cum ar fi OMI), reducerea interferențelor atmosferice cu ajutorul rapoartelor de bandă în cadrul spectrului de absorbție de SO₂; precum și modelării transferului radiativ pentru a evalua în mod corespunzător răspunsul camerei.

Rezoluția spectrală relativ mare (mai puțin de 1 nm) a instrumentelor DOAS permite extragerea simultană a diferitelor categorii de urme de gaze. În afară de SO₂, atât BrO cât și NO₂ au fost detectate în coloanele de erupție vulcanică în trecut cu ajutorul observațiilor DOAS de lumină dispersată. Spectroscopia în infraroșu cu transformata Fourier (FTIR) și DOAS activă pe traiectorie lungă pot măsura chiar mai multe categorii în intervalele spectrale IR și UV intens, dar necesită geometrii de măsurare mai complexe, deoarece radiațiile la aceste intervale de lungimi de undă nu sunt ușor disponibile în lumina solară dispersată.

Măsurătorile COSPEC și DOAS permit doar determinarea densității urmei coloanei de gaze într-o singură direcție sau, mai degrabă, într-un element de obicei mic de unghi solid, deși, la o sensibilitate tipic ridicată și (în cazul DOAS) specificitate mare. Primele încercări de a obține imagini 2-D de urme de gaz (de NO₂, SO₂ și BrO) în UV au folosit abordarea Imaging-DOAS (I-DOAS), unde o coloană de pixeli de imagine este înregistrată simultan cu un spectrometru de formare de imagini. Apoi, imaginea 2-D se construiește prin scanarea direcției de vizualizare perpendicular pentru această coloană.

Cantitățile mari de SO₂ prezent în coloanele de erupție vulcanică, precum și faptul că SO₂ este de departe absorbantul cel mai dominant în jurul valorii de 300 nm fac inutile măsurătorile înalte de rezoluție spectrală numai pentru cuantificarea de SO₂. S-a demonstrat capacitatea măsurării distribuțiilor de SO₂ 2-D în coloanele de erupție vulcanică folosind o cameră simplă sensibilă la UV și filtre de bandă care

transmit doar radiații la lungimi de undă semnificativ influențate de absorbția de SO₂.

3.4 Rezumat al capitolului Metode de măsurare a poluării atmosferice cu SO₂

Metodele folosite au ca scop final determinarea concentrației de SO₂, respectiv a fluxului de SO₂.

În acest capitol, din literatura de specialitate actualizată la zi, am adus cele mai noi informații cu privire la folosirea metodelor moderne experimentale, respectiv metoda UV și IR.

Am introdus și metoda standardizată TESTO pentru a avea o privire de ansamblu asupra tuturor metodelor de măsurare în ceea ce privește aparatura (construcție, utilizare, eficiență, etc.), aportul lor la îmbunătățirea calității aerului poluat cu SO₂.

Gazoanalizorul TESTO 300 XXL, este un echipament de determinare a emisiilor gazoase din gazele de ardere, în urma unor reacții chimice. Gazele analizate sunt: SO₂, CO, O₂, NO, NO_x. Prin conectarea la PC, gazoanalizorul TESTO 300 XXL poate fi folosit la măsurători de lungă durată (săptămâni), fiind controlat de un program special Comsoft 3.

O cameră IR, este un dispozitiv care detectează energia (căldura) și o convertește într-un semnal electronic, care este apoi procesat pentru a produce o imagine termică pe monitorul video și calculează temperatura. Lumina IR sau termografia este folosirea imaginii infraroșii și a camerei de măsurare pentru a vedea și măsura energia termică emisă de un obiect. Termic, energia infraroșie este lumina care nu este vizibilă deoarece lungimea de undă este prea lungă pentru a fi detectată de ochiul uman. Camera IR are lentile ca și o cameră cu lumină vizibilă dar în acest caz lentilele focusează din energia infraroșie pe un șir de senzori infraroșii. Mii de senzori ai șirului convertesc energia infraroșie în semnale electrice care sunt convertite ulterior în imagine.

În imaginile infraroșii, diferite culori sunt folosite pentru a reprezenta diferite temperaturi. Poți vedea orice temperatură folosind scala culoare-temperatură arată de obicei în dreapta imaginii. La o camera IR se disting anumiți parametri: K_{SO₂} (coeficient de absorbție), geometria camerei (altitudinea camerei, dimensiunea câmpului vizual, funcțiile de răspuns ale filtrului canalelor, utilizarea datelor radiosondei).

Sistemul se bazează pe măsurarea diferențelor de temperatură spectrală, fiind sensibilă la valorile NEΔT ale canalului.

Camera IR detectează și extrage concentrația de SO₂ folosind surse naturale și artificiale. La sistemul Cyclops este important alegerea numărului de filtre, lungimile de undă, lățimile de filtrare.

Metoda UV se bazează pe măsurarea absorbției de SO₂ cu ajutorul ultravioletelor. Aceasta metodă folosește legea Lambert-Beer. Micșorarea intensității luminii de o anumită lungime de undă este independentă de intensitatea inițială absolută și dependentă de grosimea stratului de substanță absorbantă străbatut.

Camera UV este sensibilă la lumină în partea UV a spectrului electromagnetic. Se știe că lumina UV are $\lambda < \lambda$ luminii vizibile dar $\lambda > \lambda$ razelor X.

Camera UV are următoarele specificații: senzorul imagine, rezoluția, formatul optic, pixelii orizontali, pixelii verticali, mărimea pixelului. Senzorii de

imagine pot fi color și monocrom. La colectarea imaginilor de măsurare cu camera UV, parametrii controlați de utilizator sunt: lungime expunere imaginii, diametrul diafragmei obiectivului, regularitatea formării de imagini cu celulele de calibrare, folosirea filtrului unic de 307nm sau a ambelor filtre de 307 nm și 326 nm.

4 Campanii de măsurători

4.1 Descrierea centralelor termice

4.1.1 CET Arad

Din punct de vedere al producerii energiei termice necesare consumatorilor din municipiul Arad, cele două centrale CET Lignit și CET Hidrocarburi funcționează interconectat rezultând o creștere a siguranței și continuității alimentării cu energie termică a consumatorilor. CET Lignit este situat în partea de nord a orașului.

- | | | | | | | | | | | | | | | | |
|---|-----|---------|------|-----|--------|----------|-----|------|---------|------|-------|---|----|--------|---|
| | CET | Lignit | Arad | în | profil | inițial: | | | | | | | | | |
| - | 2 | cazane | abur | CAI | 100 | t/h | - | pe | lignit. | | | | | | |
| - | 3 | cazane | abur | CR | 1244 | - | 420 | t/h, | 137 | bar, | 545°C | - | pe | lignit | ; |
| - | 3 | grupuri | de | 50 | MW | ; | | | | | | | | | |

În anul 1989 s-au pus în funcție CAI 1 și CAI 2 de 100 t/h, 13 bar, 250°C, un boiler de termoficare și magistrala de interconexiune între cele două centrale pentru a asigura alimentarea cu energie termică a orașului din CET Lignit. În 1993 s-a pus în funcție cazanul nr.1 - 420 t/h - pe lignit. În 1994 s-a pus în funcție TA - 1 - DSL 50 cu generatorul TH 60-2 de 55 MW, 3x10,5 kV, 50 Hz.

Actualmente este în faza de investiții cazanul nr.2 - 420 t/h și TA 2 - 50 MW. 2002 - Prin H.G. 105/2002 se decide desprinderea Sucursalei Electrocentrale Arad din cadrul S.C.Termoelectrica S.A. București și trecerea sub autoritatea Consiliului Municipal Arad.



Figura 4.1 CET Arad

4.1.2 Romag Termo Halânga

ROMAG TERMO este situată în partea de nord a municipiului Drobeta Turnu Severin, la Km 5 pe drumul național DN 67 ce leagă municipiul Drobeta Turnu Severin de Târgu Jiu.

Are ca profil activitati productia:

- de abur industrial de 16 și 38 ata necesar Uzinei de apă grea ROMAG PROD;
- energie electrică livrată în SEN;
- energie termică sub formă de apă fierbinte necesară sistemului de termoficare al municipiului Drobeta Turnu Severin.

În figura 4.2 este prezentată o vedere din fața a centralei.



Figura 4.2 Romag Termo Halânga

Centrala ROMAG TERMO folosește din plin principiul cogenerării, ceea ce îi conferă un grad ridicat de utilizare a energiei primare a combustibilului.

CET ROMAG TERMO a fost gândită pentru alimentarea consumatorului de grad "0" ROMAG PROD în vederea obținerii producției de apă grea, în scopul realizării programului nuclear național bazat pe reactoare de tip CANDU care folosesc ca moderator în reacția de fisiune nucleară apa grea.

La CET ROMAG TERMO s-a constatat că există o capacitate mai mare de producere a aburului folosind cele șase cazane decât capacitatea de preluare a celor patru turbine și au avut loc modernizări ale capacităților de producere a energiei electrice. În prezent, CET ROMAG TERMO funcționează cu următoarele capacități de producție:

- cazane de 420 t/h, 140 ata, cu funcționare pe lignit și suport de păcură;
- 1 cazan de 420 t/h, 140 ata, cu funcționare pe păcură;
- turbina nr.1 de tip DSL 50 de condensatie;
- turbina nr.2 de tip DKUL 50 de cogenerare;
- turbina nr.3 de tip DKA-23 de 22 MW cu contrapresiune la 6 bar și priză reglabilă la 38 bar;
- turbina nr.4 de tip DSL 50 de condensatie;
- turbina nr.5 de tip DSL 50 de condensatie;
- turbina nr.6 de tip SC 25 de 25 MW. [3]

Cele șase cazane de 420 t/h sunt considerate instalații mari de ardere, fiind grupate astfel:

- cazanele nr.1,2 și 3 formează instalația mare de ardere nr.1 - IMA 1
 - cazanele nr.4,5 și 6 formează instalația mare de ardere nr.2 - IMA 2
- Cazanul de 420 t/h nr.7 formează instalația mare de ardere nr.3 - IMA 3
- Gazele de ardere provenind de la cazanele de abur de 420 t/h sunt evacuate în atmosferă prin două coșuri de fum din beton armat:
- coșul de fum nr.1, având înălțimea fizică $H=242$ m și diametrul la vârf $D=8,3$ m
 - coșul de fum nr.2, având înălțimea fizică $H=280$ m și diametrul la vârf $D=8,3$ m
- Consumurile de combustibil, păcură și cărbune, au crescut proporțional cu cantitățile de energie electrică și termică produse.

4.2 Măsurători cu camera UV

4.2.1 Prezentarea camerei UV și a filtrelor folosite



Figura 4.3 Camera, filtrele și roata de filtrare a camerei UV

S-a folosit camera UV cu următoarele caracteristici: model Apogee e2V CCD47-10 iluminată din spate, 1024x1024 pixeli, roată de filtre cu lăcașuri pentru 9 filtre. Camera are montate 4 filtre cu lungimile de undă 310, 315, 325 și 330 nm. Filtrele au o lățime de bandă de 10 nm la jumătate din răspunsul maxim (FWHM). Avem 3 lentile UV de dimensiune 50, 78, 105 mm cu F/3.5, F/3.8, F/4.0. Camera are digitizare pe 16 biți. Câmpul de vedere al camerei (FOV) este $15.2^\circ - 7.23^\circ$. Distanța de detecție este până la aproximativ 10 km. Acuratețea pentru SO_2 este de 0.1 gm^{-2} .



Figura 4.4 Vedere din față și din spate a camerei UV

4.2.2 Metode de calibrare

Calibrarea camerei SO_2 s-a făcut de obicei înainte sau după ce au fost efectuate măsurătorile prin introducerea diferitelor celule de calibrare SO_2 cu densități de coloană SO_2 S cunoscute în câmpul vizual al instrumentului și prin determinarea unei relații empirice între densitatea optică normalizată $\hat{\tau}$ (sau absorbantă aparentă AA) și densitatea coloanei SO_2 S . Deoarece densitatea optică normalizată $\hat{\tau}$ depinde de spectrul incident al radiației solare dispersate $I_S(\lambda)$, de transmitanța spectrală a filtrelor de bandă $T(\lambda)$, precum și de eficiența cuantică a detectorului $Q(\lambda)$, această abordare simplă va funcționa corespunzător doar dacă toți cei trei parametri sunt constanți în timp. În timp ce eficiența cuantică $Q(\lambda)$ poate fi presupusă mai mult sau mai puțin constantă în timp și independentă de lungimea de undă în regiunea de interes, ceilalți doi parametri pot varia în funcție de condițiile de măsurare.

S-a aplicat un model pentru studiul gradului de sensibilitate al camerei SO_2 la variațiile distribuției spectrale a radiației solare dispersate incidentale $I_S(\lambda)$ și transmitanța filtrului $T(\lambda)$. S-a simulat transmitanța filtrului de bandă A folosind o distribuție gaussiană de transmitanță centrată la $\lambda_c = 309\text{nm}$, cu o deviere standard de 3 nm. Această curbă se aseamănă mult cu curba de transmitanță a filtrului ales ulterior pentru construirea prototipului camerei SO_2 . Pentru fiecare lungime de undă între 300 și 340 nm, spectrul incident al radiației solare dispersate $I_S(\lambda)$ măsurat a fost înmulțit cu transmitanța filtrului $T_A(\lambda)$. Eficiența cuantică $Q(\lambda)$ nu a fost luată în considerare în continuare, deoarece eficiența cuantică a detectorului spectrometrului este inclusă intrinsec în spectrul solar măsurat $I_S(\lambda)$ și este gândit să semene îndeaproape cu cea a unei camere SO_2 tipice. În acest fel, intensitatea spectrală $I_{0,A}(\lambda)$ a fost calculată.

Camera utilizează un filtru special ce colectează fotoni în regiunea ultravioletă UV unde SO₂ selectiv absoarbe lumina UV. SO₂ se cuantifică prin calibrarea cu celule de concentrație cunoscută.

Spectrul radiației dispersate incidente I_S depinde de unghiul zenital solar (SZA) și, prin urmare, de ora din zi. Această dependență devine importantă atunci când soarele este aproape de orizont. Curba de transmisie a filtrului de bandă T_A depinde de unghiul de iluminare θ . Dacă unghiul de transmisie efectiv nu este constant, acest termen va varia de la pixel la pixel. Acțiunea de calibrare a camerei SO₂ nu este neînsemnată. În timp ce se poate efectua o calibrare empirică prin poziționarea celulelor de calibrare conținând cantități cunoscute de SO₂ în fața camerei, este important să se țină cont de faptul că etalonarea poate depinde de o serie de parametri dependenți de pixeli (de ex., θ , β , $r_{1..n}$, r_d în funcție de configurarea optică aplicată).

Pentru calibrare trebuie colectate cât mai multe puncte de date posibil, iar acestea trebuie să acopere întreaga gamă de valori de măsurare.

O calibrare empirică folosind o celulă de gaz va depinde în mod inevitabil de momentul zilei. Calibrarea empirică va continua, probabil, să fie de mare valoare. Cu toate acestea, în locul calibrării camerei cu celule SO₂, se poate obține o calibrare intrinsecă actualizată prin co-localizarea unui instrument DOAS cu camera SO₂. În acest mod, distribuțiile de SO₂ 2-D măsurate cu camera pot fi normalizate folosind măsurători spectroscopice de înaltă rezoluție efectuate într-unul dintre sub-sectoarele câmpului vizual al camerei. În afară de redarea unei calibrări intrinsece, recente progrese în extragerea transferului radiativ în și în jurul coloanelor de erupție vulcanică pot fi aplicate astfel măsurătorilor cu camera SO₂. După implementarea soluțiilor descrise mai sus, potențialul camerei SO₂ pentru cercetare emisiilor vulcanice este enorm. Instrumentul redă distribuțiile de SO₂ 2-D la rezoluții temporale de ordinul 1 Hz sau mai bune. Costul relativ scăzut al instrumentului (aproximativ 10 000 €) face camera SO₂ să fie un instrument ideal pentru monitorizarea în timp real a activității vulcanice. Cele mai multe observatoare vulcanice au deja camere UV și IR care observă permanent vulcanii potențial activi. Adăugarea unei camere SO₂ oferă instantaneu o imagine de ansamblu a emisiilor de gaze, chiar și pentru un observator neantrenat. De asemenea, imaginile obținute permit o analiză detaliată a interacțiunii dinamice a gazelor vulcanice cu atmosfera ca urmare a craterului de emisie.

4.2.3 Prezentarea tehnicilor de măsurare

Pentru a realiza măsurătoarea se urmează câțiva pași. Se assemblează camera de măsurare și se pune pe trepied. Se pune o lentilă UV. Se conectează camera UV la laptop. Se lansează programul Maxim DL. Din meniul Setup se alege tipul camerei și filtrul ce vor fi utilizate și se apasă butonul Connect.

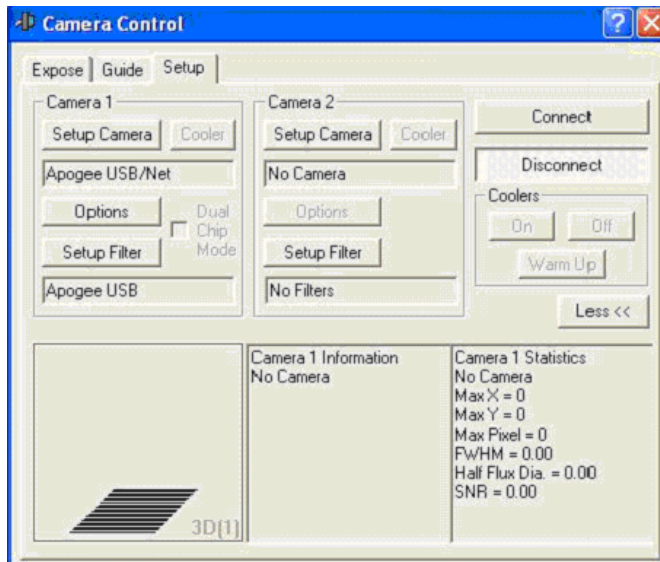


Figura 4.5 Controlul camerei UV

După ce camera a fost conectată se trece în meniul Expose. Se apasă butonul Autosave și se setează parametrii de funcționare ai camerei. (numele, timpul de expunere, numărul de repetiții, etc.) După ce s-au setat parametrii se apasă butonul Start și se începe măsurătoarea.

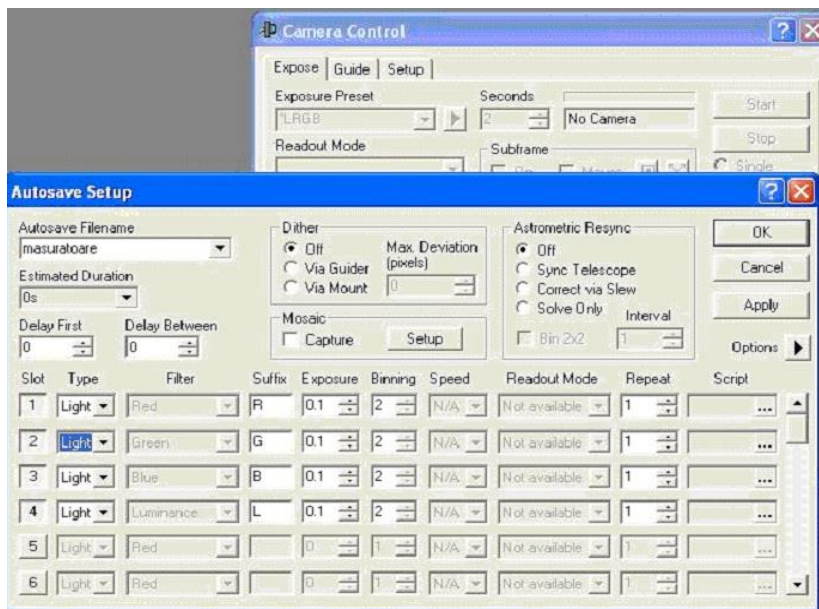


Figura 4. 6. Setarea camerei UV

4.2.4 Campanii de măsurători folosind camera UV

Camera UV a fost folosită în câteva campanii de măsurători la Rovinari, Halânga și Arad în perioada 2010 - 2012. În urma măsurătorilor au fost captate imagini care ulterior au fost prelucrate și interpretate folosind programul Camera UVIR. Mai jos sunt prezentate câteva imagini obținute în urma măsurătorilor.



Figura 4.7 Măsurătoare UV Arad



Figura 4.8 Măsurătoare UV Halânga



Figura 4.9 Măsurătoare UV Rovinari

4.3 Măsurători cu camera IR

4.3.1 Prezentarea camerei IR si a filtrelor folosite

Camera IR este un dispozitiv care se bazează pe detecția energiei infraroșii, convertirea ei într-un semnal electronic, formarea de imagini termice și calcularea temperaturii. Camera IR are lentile ca și o cameră obișnuită, care focusează undele din energia infraroșie pe un șir de senzori infraroșii. Filtrele sunt atent selectate pentru a se potrivi cu detecția de SO₂ și a particulelor de cenușă. Cel mai important aspect la Cyclops este alegerea numărului de filtre necesar, lungimile de undă centrale ale filtrului și lățimile de filtrare. Filtrele alese operează în banda de lungimi de undă 8,6-12μm. Sistemul de cameră folosește la aceste măsurători patru filtre spectrale ca să optimizeze detectarea gazelor atmosferice specifice(8.6, 10, 11, 12). Filtrul de 8.6 μm este special pentru detecția de SO₂. Camera are un câmp de vizualizare de 43°. Camera IR cu microbolometru nerăcit are sensibilitatea de temperatură de 200mK. Dispozitivul prezintă dimensiuni de rezoluție de 640x512 pixeli. Roata de filtrare se plasează între sistemul optic și detector. Distanța de la cameră la țintă poate fi până la 10 km. Acuratețea este de 0.2 g/m² pentru SO₂ și de 0.5 g/m² pentru cenușă.

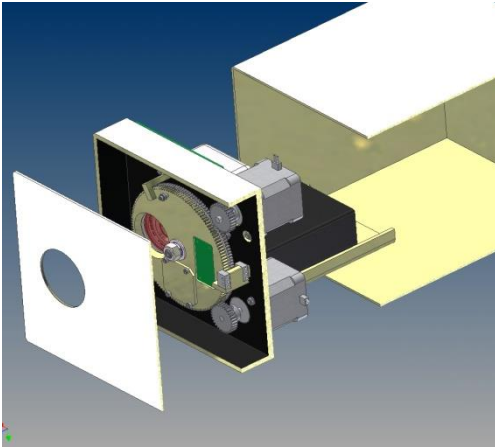


Figura 4.10 Camera IR

4.3.2 Metode de calibrare

O calibrare internă a acurateții absolute a fost efectuată prin luarea temperaturii medii a unui cluster circular central de 570 pixeli dintr-o imagine de referință de corp negru, care are o precizie de până la 0,1 K. Corpul negru a fost plasat la 80 de cm de obiectivul camerei, umplând majoritatea câmpului vizual, fiind reglată camera (împreună cu diafragma) folosind o combinație de azot lichid și radiatoare electrice. Pentru a preveni acoperirea echipamentelor, distanța dintre corpul negru și cameră a fost sigilată cu o folie de plastic, iar diafragma corpului negru a fost ținută aproape de temperatura camerei. Abaterea standard a acestor citiri de pixeli a fost mai mică de 0,1 K. O ecuație liniară de calibrare indică o potrivire bună cu rezultatele, deși diferența între temperatura observată și cea reală a corpului negru este de 7 K la nivelul inferior al intervalului, poate din cauză că producătorii calibrează camerele numai până la 273 K pe producție. Această conversie liniară se aplică tuturor citirilor de pixeli. S-a actualizat anual, iar deviația mică observată s-a luat în considerare în rezultatele imaginii[85].

Discriminarea și cuantificarea gazelor și a particulelor necesită imagini termice de înaltă fidelitate de la Cyclops. Pentru a atinge fiabilitate și precizie, camera trebuie să fie calibrată. Procedura este o calibrare liniară care necesită o estimare a amplitudinii și interceptării care convertește numerele digitale (DNS) la radianțe, iar apoi la temperaturi ale luminozității. Se implementează un proces în două etape: Cyclops este întâi calibrat în laborator, în condiții controlate folosind o sursă de corp negru, iar estimările amplitudinilor și interceptărilor pentru toate canalele sunt determinate pentru o varietate de condiții de mediu și țintă (sursă). Pe teren, condițiile de mediu nu pot fi măsurate suficient de bine pentru a permite utilizarea doar a acestor coeficienți de calibrare. Astfel, se folosește un al doilea pas, care să compenseze pentru schimbările în condițiile de mediu, în special, temperaturile instrumentului, sistemul optic anterior și carcasa exterioară. Acest al doilea pas necesită adăugarea unui obturator de corp negru, plasat în partea din față a sistemului optic, roții filtrului și detectorului. Diafragma controlată de temperatură se mută în fața camerei pe comanda de calculator, pentru a permite un singur punct de calibrare pe linia de calibrare a radianței. Calibrarea poate fi repetată pe cât de des este necesar și se efectuează pentru fiecare dintre cele cinci

filtre separat. Această procedură în două etape, oferă precizii de temperatură de 0,2 – 0,7 K la 280 K, în funcție de canal.

Datele imaginii 2D (320×240 pixeli), captată de pe camera constă din numere digitale primare (DN) pentru fiecare canal (sau filtru). După calibrare, sunt convertite la temperaturi de luminozitate ale scenei (BT), folosind o procedură de tabel de căutare care se referă la DN față de radianța integrată pe funcția de răspuns a filtrului. Intervalul de valabilitate și precizie al acestor proceduri variază în funcție de canal, dar pentru temperaturi cuprinse între 260–290 K, preciziile sunt mai bune de 0,1 K.

După calibrarea și conversia la BT, fiecare secvență de măsurare este formată din 5 imagini BT captate la intervale de 5–6 min. pe perioade de timp de până la mai multe ore. Cea de-a doua limitare a camerei SO₂ este calibrarea sa dificilă. Intensitatea radiației incidente $I(\lambda)$ nu se măsoară în funcție de lungime de undă λ , ci integrat pe întreaga fereastră de transmitanță a filtrului de interferență de bandă. Prin urmare, densitatea optică $\tau(\lambda)$, care este proporțională cu densitatea coloanei de SO₂ nu se poate extrage. În schimb, se obține densitatea optică ponderată τ^{\wedge} . Relația dintre τ^{\wedge} și S nu este liniară, iar calculul coeficientului de calibrare poate fi complex. Calibrarea depinde de unghiul zenital solar (SZA), de totalul coloanei de O₃, de unghiul de iluminare al filtrului θ și de densitatea coloana de SO₂ S în sine.

După ce se permite un timp de calibrare și se obțin până la 5 canale de date de imagini filtrate (o secvență), ratele de eșantionare de o secvență pe minut sunt realizabile. În acest ritm, explorarea dinamicii coloanelor de erupție și norilor este posibilă, chiar și urmărirea caracteristicilor în secvențe de imagini consecutive.

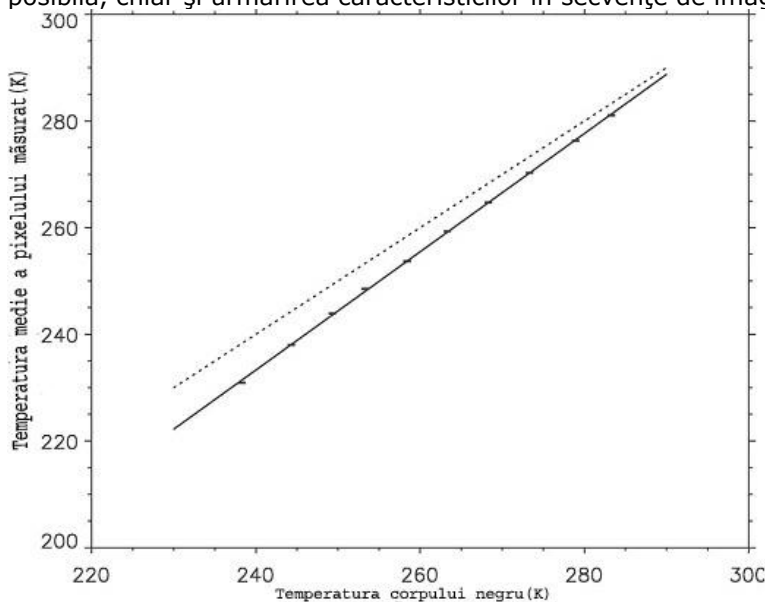


Figura 4.11 Calibrarea camerei cu ajutorul unui corp negru de referință[94]

Eroarea este indicată prin lățimea punctelor de date, iar linia punctată reprezintă o curbă de măsurare perfect corectă (adică, $y = x$).

S-a constatat că calibrarea obiectivului de vizualizare este necesară, din moment ce unghiul solid acoperit de fiecare pixel scade de la 0,83 grade pătrate din centrul imaginii la 0,33 grade pătrate spre marginile imaginii. Pentru a capta imagini operațional, camera a fost fixată astfel încât pixelul central să fie $48,7 \pm 0,1^\circ$ la zenit, fiind dotată cu un obiectiv dându-i un câmp vizual de $60^\circ \times 80^\circ$. Zenitul reprezintă punctul de intersecție al verticalei locului cu sfera cerească, situat deasupra capului observatorului și opus nadirului (punct de pe bolta cerească, opus zenitului, situat la intersecția verticalei locului la care ne referim cu emisfera cerească inferioară).

Unghiul zenital reprezintă unghiul dintre verticala locului și distanța înclinată (cu linia naturală a terenului).

4.3.3 Prezentarea tehnicilor de măsurare

Pentru a realiza măsurătoarea sunt necesari câțiva pași de urmat. Se montează camera pe trepied. Se conectează camera IR la laptop cu ajutorul cablului de date USB. Se alimentează camera. Se lansează programul Cyclops. Se alege filtrul camerei din meniul File → select filterwheel. Se alege camera din meniul File → select camera. Se alege folderul unde vor fi puse imaginile obținute în urma măsurătorilor din meniul File → select data folder. Se apasă butonul Start pentru a începe măsurătorile. Se poate alege intervalul în minute între care are loc măsurătoarea din câmpul interval. Se pot alege totodată și filtrele folosite la măsurători din panoul filter channel. Se poate alege și modul în care se face referința din panoul reference.

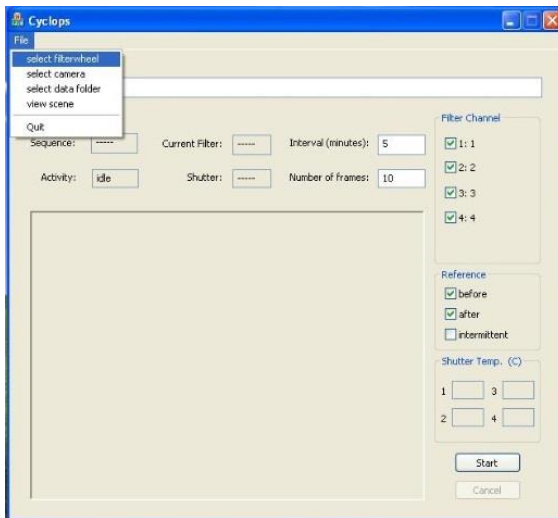


Figura 4.12 Interfața programului Cyclops

4.3.4 Campanii de măsurători folosind camera IR

Camera IR a fost folosită în câteva campanii de măsurători la Rovinari, Halânga și Arad în perioada 2010 – 2012. În urma măsurătorilor au fost captate

imagini care ulterior au fost prelucrate și interpretate folosind programul Camera UVIR. Mai jos sunt prezentate câteva imagini obținute în urma măsurătorilor.



Figura 4.13. Măsurătoare IR Arad

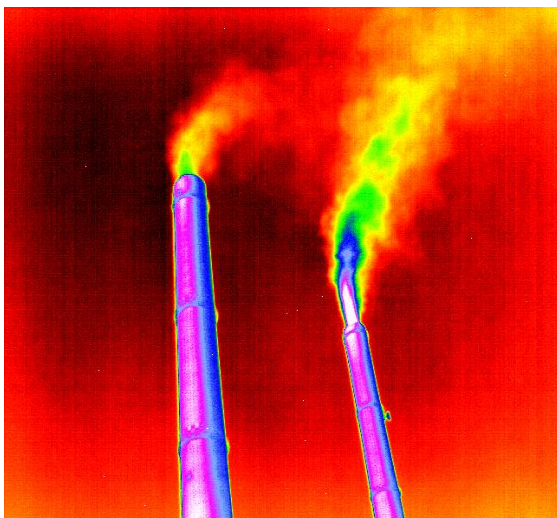


Figura 4.14 Măsurătoare IR Halânga

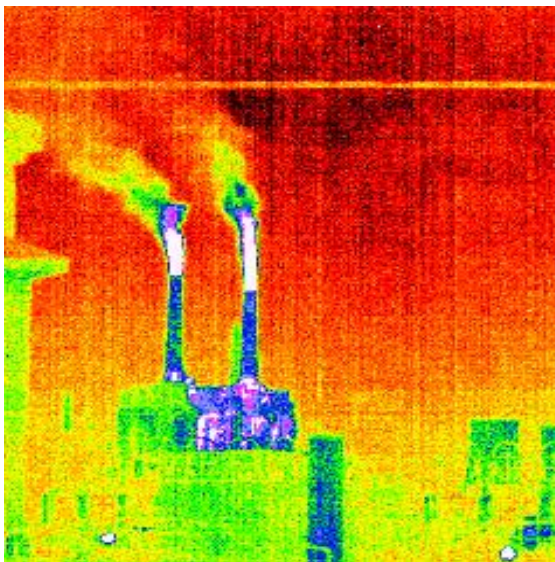


Figura 4.15 Măsurătoare IR Rovinari

4.4 Comparații între metodele de măsurare a poluării atmosferice cu SO₂

4.4.1 Comparație între metodele standard și cele experimentale

Se pot face comparații între metodele standard și metodele moderne UV și IR, privind asemănările și deosebirile în ceea ce privește componența instrumentală, modul de funcționare, de prelucrare a datelor, de utilizare a lor în diferite domenii. Coloana de poluant trebuie să fie între sursa de lumină (Soare) și instrumentul de măsurat și de aceea instrumentele trebuie să fie mobile și să se adapteze la diferite poziții față de Soare ceea ce reduce eficiența colectării de date [74].

Căldura globului pământesc se datorează unui efect asemănător celui care are loc într-o seră. Radiațiile vizibile trec prin fereastra serei încălzindu-i solul. Solul emite radiații infraroșii cu o lungime de undă mult mai mare decât a radiațiilor solare, iar fereastra reflectă aceste radiații. Din acest motiv într-o sferă este mereu mai cald decât afară chiar dacă interiorul nu este încălzit.

Atmosfera datorită vaporilor de apă și de dioxid de carbon alcătuiește un înveliș protector în jurul Pământului. Ea permite trecerea radiațiilor ultraviolete și vizibile, dar reflectă radiațiile infraroșii (termice) venite din scoarța terestră. Deci atmosfera conservă căldura globului terestru [33].

Metodele UV și IR au avantajul că măsoară concentrații SO₂ de la distanță față de metoda TESTO unde instrumentul se pune direct la sursa de poluare.

În urma campaniilor de măsurători efectuate la Arad și la Halânga am luat imagini cu camera UV și Testo pe care le-am procesat cu programul Camera UVIR și am obținut valori ale concentrațiilor de SO₂, a fluxului de SO₂. Aceste valori au fost trecute în tabelele de valori prezentate mai jos. Cu aceste valori am alcătuit grafice

unde am reprezentat variațiile concentrațiilor de SO₂ atât prin metoda Testo cât și prin metoda UV.

Tabelul 4.1 Rezultatele măsurătorilor realizate la Arad din data de 11.03.2011 cu camera UV și Testo

Numar măsur ătoare	Localitate	Data	Ora	Măsurăto are directă (ppm)	Măsurătoa re UV (ppm)	Flux (g/s)	Vite za vânt
1	Arad	11/3/2011	10:28	962	940	23	0.4
2	Arad	11/3/2011	10:33	965	933	14	0.32
3	Arad	11/3/2011	10:37	958	938	21	0.27
4	Arad	11/3/2011	10:38	960	928	27	0.23
5	Arad	11/3/2011	10:43	961	936	18	0.38
6	Arad	11/3/2011	10:46	963	941	16	0.29
7	Arad	11/3/2011	10:47	961	943	22	0.27
8	Arad	11/3/2011	10:49	957	922	15	0.24
9	Arad	11/3/2011	10:59	964	927	12	0.21
10	Arad	11/3/2011	11:10	970	929	29	0.31

Pentru metoda de măsurare TESTO valoarea cea mai mică este de 957 ppm, iar cea mai mare este de 970 ppm.

Pentru metoda de măsurare UV valoarea cea mai mică este 922 ppm, iar cea mai mare este de 943 ppm.

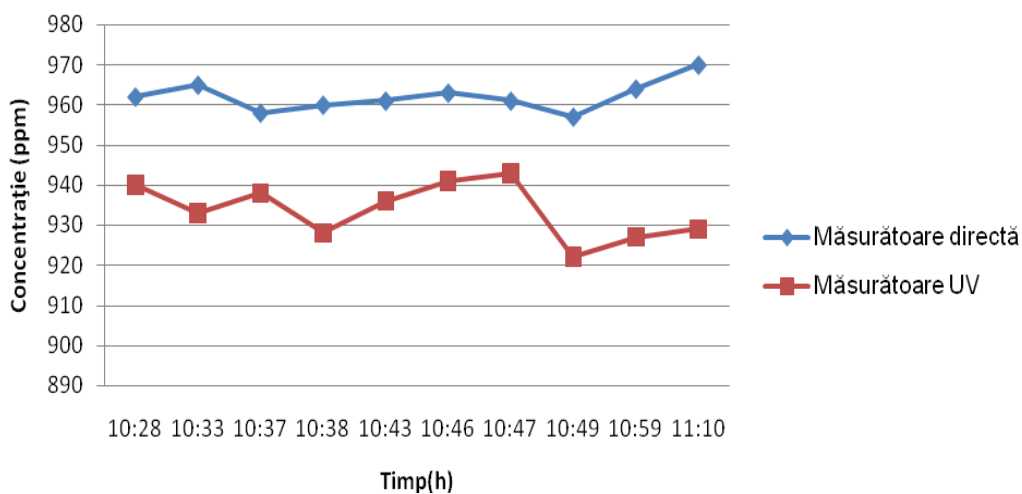


Figura 4.16 Variația concentrației de SO₂ din data de 11.03.2011 cu metoda UV și TESTO

4.4 - Comparații între metodele de măsurare a poluării atmosferice cu SO₂ 131

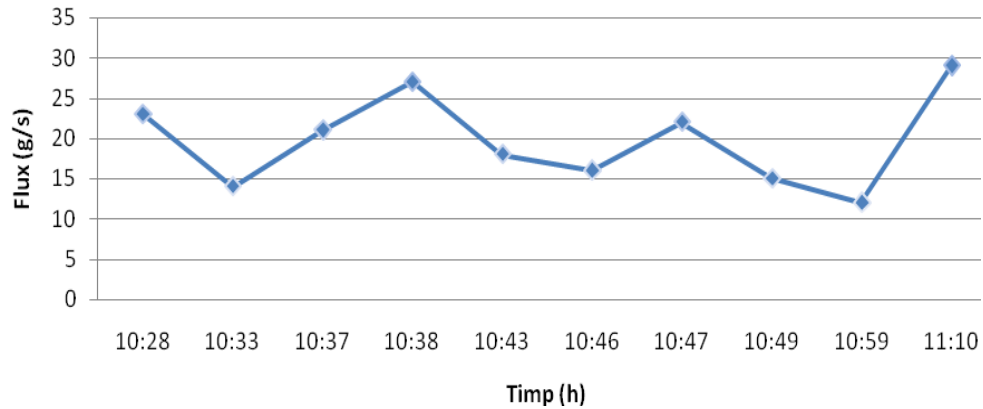


Figura 4.17 Variația din data de 11.03.2011 a fluxului de SO₂

Tabelul 4.2 Rezultatele măsurătorilor realizate la Arad din data de 11.11.2011 cu camera UVși Testo

Numar măsurătoare	Localitate	Data	Ora	Măsurătoare directă (ppm)	Măsurătoare UV (ppm)	Flux (g/s)	Viteza vânt
1	Arad	11/11/2011	8:28	992	975	34	0.72
2	Arad	11/11/2011	8:35	994	973	10	0.33
3	Arad	11/11/2011	8:37	996	978	23	0.29
4	Arad	11/11/2011	8:38	991	970	26	0.23
5	Arad	11/11/2011	8:43	993	976	28	0.29
6	Arad	11/11/2011	8:46	996	956	13	0.22
7	Arad	11/11/2011	8:47	992	962	27	0.37
8	Arad	11/11/2011	8:49	988	952	11	0.34
9	Arad	11/11/2011	8:59	994	967	12	0.34
10	Arad	11/11/2011	9:10	970	979	44	0.52

Pentru metoda de măsurare TESTO valoarea cea mai mică este de 970 ppm, iar cea mai mare este de 996 ppm.

Pentru metoda de măsurare UV valoarea cea mai mică este 952 ppm, iar cea mai mare este de 979 ppm.

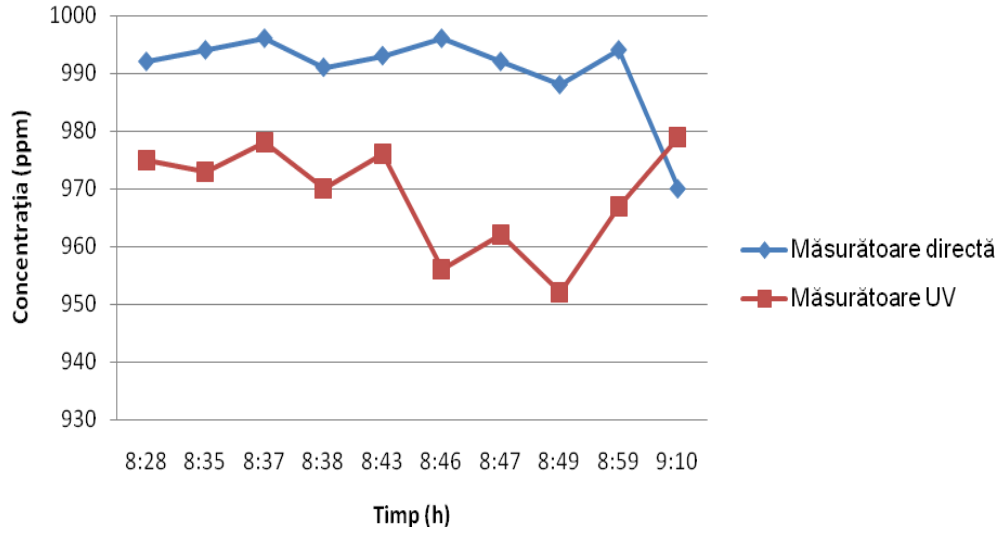


Figura 4.18 Variații ale concentrației de SO₂ din data de 11.11.2011 cu metoda UV și TESTO

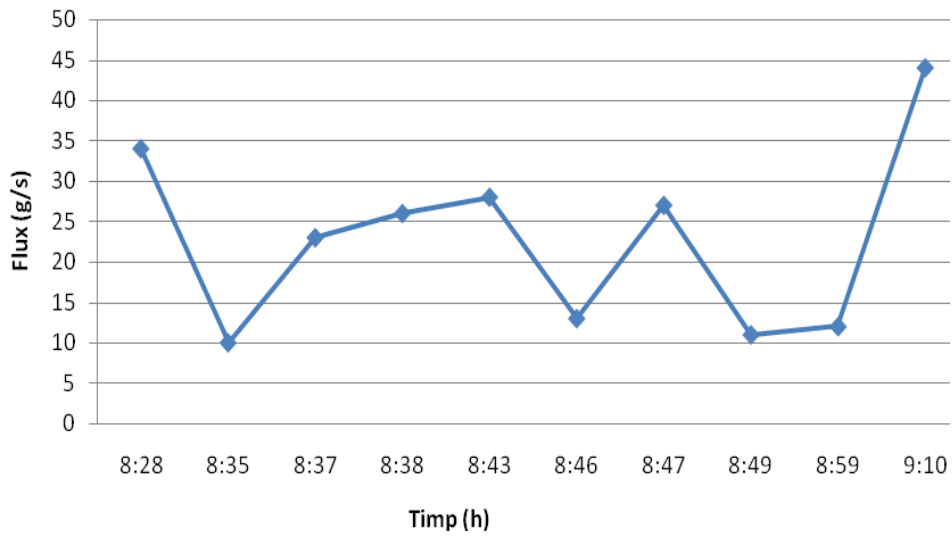


Figura 4.19 Variația fluxului de SO₂ din data de 11.11.2011

Tabelul 4.3 Rezultatele măsurătorilor realizate la Halânga din data de 10.02.2011 cu camera UV și TESTO

Numar măsur ătoare	Localitate	Data	Ora	Măsurătoa re directă (ppm)	Măsurătoa re UV (ppm)	Flux (g/s)	Vitez a vânt
1	Halanga	2/10/2011	11:46	762	743	12	0.87
2	Halanga	2/10/2011	11:49	758	737	17	0.78
3	Halanga	2/10/2011	11:53	756	725	22	0.83
4	Halanga	2/10/2011	11:56	759	733	14	0.95
5	Halanga	2/10/2011	11:58	760	725	20	0.76
6	Halanga	2/10/2011	12:04	763	729	21	0.89
7	Halanga	2/10/2011	12:07	765	722	18	0.73
8	Halanga	2/10/2011	12:09	764	739	19	0.56
9	Halanga	2/10/2011	12:11	757	729	16	0.8
10	Halanga	2/10/2011	12:16	758	737	15	0.81

Pentru metoda de măsurare TESTO valoarea cea mai mică este de 756 ppm, iar cea mai mare este de 765 ppm.

Pentru metoda de măsurare UV valoarea cea mai mică este 722 ppm, iar cea mai mare este de 743 ppm.

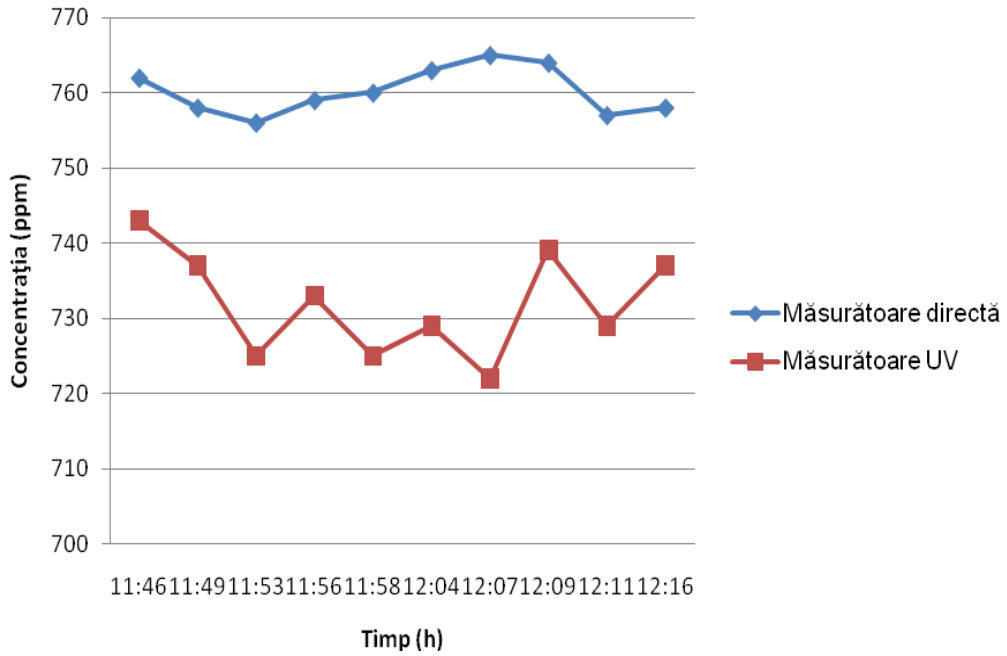


Figura 4.20 Variații ale concentrației de SO₂ din data de 10.02.2011 cu metoda UV și TESTO

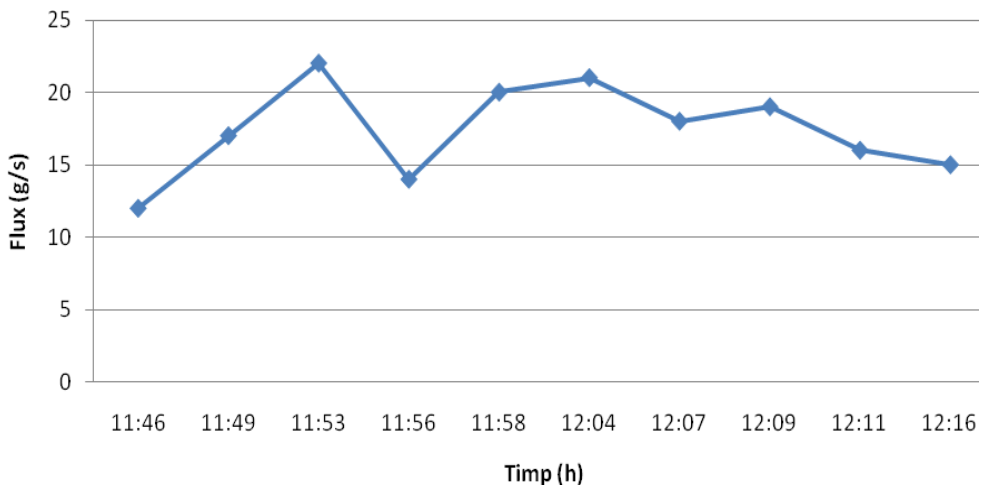


Figura 4.21 Variația fluxului de SO₂ din data de 10.02.2011

Tabelul 4.4 Rezultatele măsurătorilor realizate la Halânga din data de 28.10.2011 cu camera UV și TESTO

Numar măsurătoare	Localitate	Data	Ora	Măsurătoare directă (ppm)	Măsurătoare UV (ppm)	Flux (g/s)	Viteza vânt
1	Halanga	10/28/2011	12:45	778	732	12	0.76
2	Halanga	10/28/2011	12:48	778	747	18	0.51
3	Halanga	10/28/2011	12:55	776	735	14	0.8
4	Halanga	10/28/2011	12:58	779	724	13	0.73
5	Halanga	10/28/2011	12:59	780	745	10	0.64
6	Halanga	10/28/2011	13:07	783	732	11	0.76
7	Halanga	10/28/2011	13:13	775	734	15	0.82
8	Halanga	10/28/2011	13:16	774	752	17	0.71
9	Halanga	10/28/2011	13:19	777	719	16	0.49
10	Halanga	10/28/2011	13:22	778	748	11	0.68

Pentru metoda de măsurare TESTO valoarea cea mai mică este de 775 ppm, iar cea mai mare este de 780 ppm.

Pentru metoda de măsurare UV valoarea cea mai mică este de 719 ppm, iar cea mai mare este de 752 ppm.

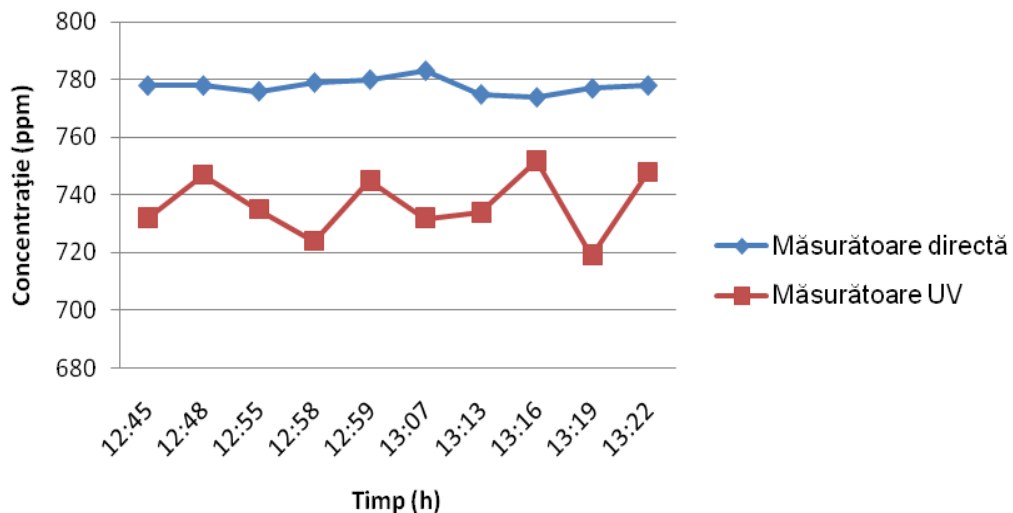
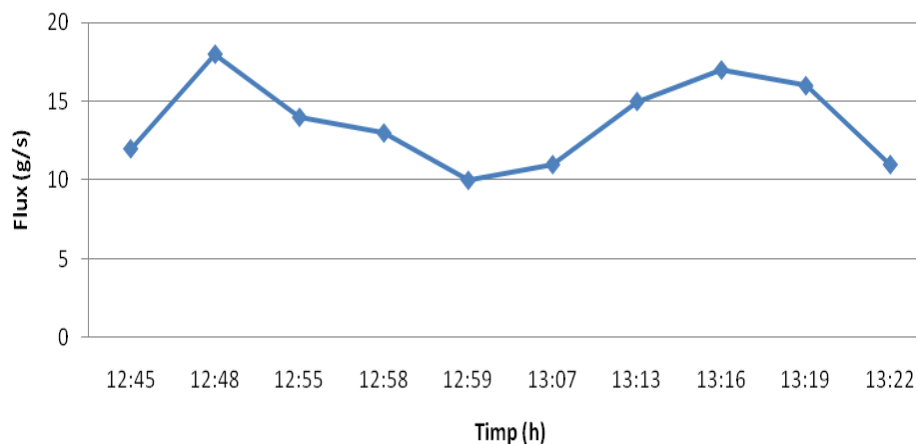


Figura 4.22 Variații ale concentrației de SO₂ din data de 28.10.2011 cu metoda UV și TESTO

Figura 4.23 Variația fluxului de SO₂ din data de 28.10.2011

Tabelul 4.5 Rezultatele măsurătorilor de la Halânga cu metoda UV și TESTO din data de 29.03.2012

Numar măsur ătoare	Localitat e	Data	Ora	Măsur ătoare directă (ppm)	Măsurăto are UV (ppm)	Flux (g/s)	Viteza vânt
1	Halanga	3/29/2012	9:46	717	822	15	1.08
2	Halanga	3/29/2012	9:47	720	747	13	0.91
3	Halanga	3/29/2012	9:49	716	735	12	0.87
4	Halanga	3/29/2012	9:50	715	713	14	0.95
5	Halanga	3/29/2012	9:51	713	755	10	0.8
6	Halanga	3/29/2012	9:52	719	730	11	0.76
7	Halanga	3/29/2012	10:04	710	742	15	0.86
8	Halanga	3/29/2012	10:05	718	759	8	0.87
9	Halanga	3/29/2012	10:06	715	719	16	1.16
10	Halanga	3/29/2012	10:07	715	767	15	0.84

Pentru metoda de măsurare TESTO valoarea cea mai mică este de 710 ppm, iar cea mai mare este de 720 ppm.

Pentru metoda de măsurare UV valoarea cea mai mică este 713 ppm, iar cea mai mare este de 822 ppm.

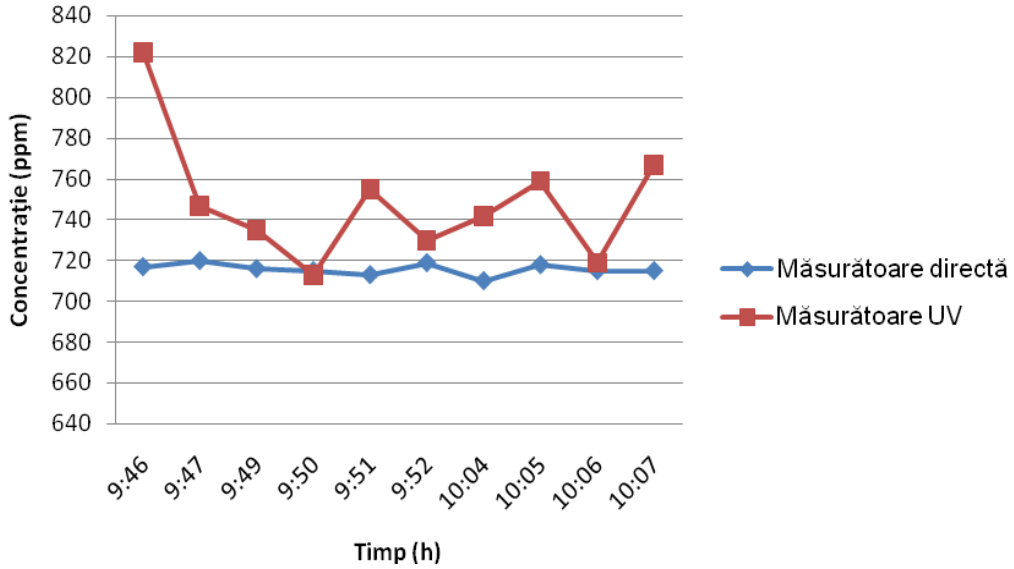


Figura 4.24 Variații ale concentrației de SO₂ cu metoda UV și TESTO în data de 29.03.2012

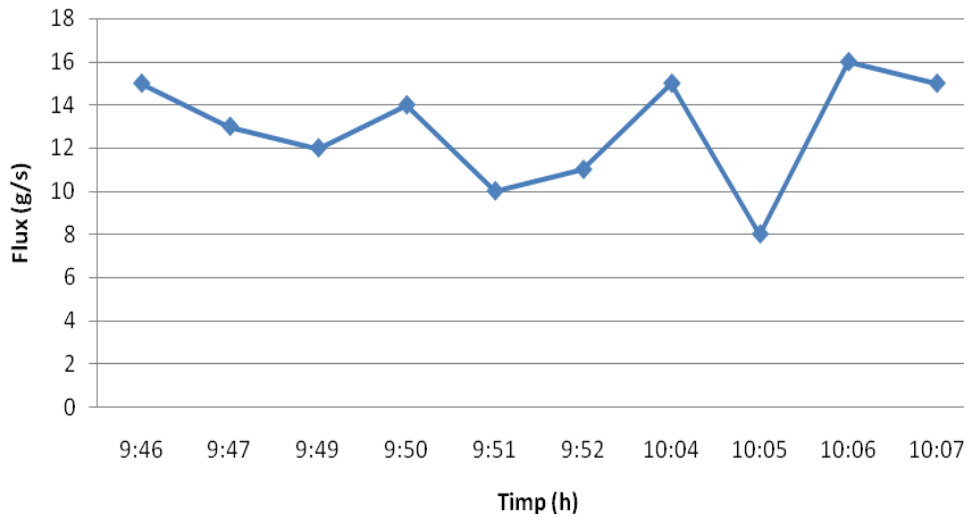


Figura 4.25 Variația fluxului de SO₂ în data de 29.03.2012

Pentru a compara cele două metode de măsurare s-au alcătuit grafice în care sunt consemnate variațiile concentrațiilor măsurate de SO₂ în funcție de timp. Eroarea maximă constatată între cele două metode este sub 10 % ceea ce înseamnă

că metoda de măsurare UV poate fi considerată o metodă suficient de precisă pentru detecția de la distanță a poluantului SO₂.

Algoritmii de calcul folosiți la metoda UV oferă posibilitatea calculării fluxului de SO₂ pe baza aceluiași set de imagini captate cu camera UV. Valorile fluxului de SO₂ din tabel sunt reprezentate grafic pentru fiecare campanie de măsurare.

Se observă oscilații ale valorilor fluxului de SO₂ pe durata fiecărei campanii de măsurare.

4.5 Model matematic de calcul al fluxului și a concentrației de SO₂

4.5.1 Metoda de procesare a datelor experimentale

Algoritmul de calcul utilizat de aplicația Camera UVIR se bazează pe calculul absorbantei aparente a SO₂.

Absorbanta aparentă a SO₂ în imagini ale penei se calculează folosind următoarea ecuație:

$$AA(i, j) = -\log \left[\frac{(PA_s(i, j) - PA_d(i, j)) / (PA_b(i, j) - PA_d(i, j))}{(PB_s(i, j) - PB_d(i, j)) / (PB_b(i, j) - PB_d(i, j))} \right] + offset(i, j) \quad (4.1)$$

Unde AA(i,j) este absorbanta aparentă la linia i și coloana j a pixelului camerei. PA și PB corespunde intensității pixelului(i,j) al camerei cu filtrele A și B. Indicii s, b și d indică imaginile de probă, fundal și întuneric.

Conversia din absorbanta aparentă în ppm.m este realizată prin calibrare folosind celule cu concentrație cunoscută de SO₂.

Cu ajutorul absorbantei aparente se pot obține concentrații ale coloanei de SO₂ pentru orice pixel de coloană, iar fluxurile de SO₂ sunt ulterior calculate folosind vitezele măsurate ale coloanei. În acest sens am realizat un program software Camera UVIR care calculează concentrația și fluxul de SO₂.

Decalajul este scos din imagini pentru a asigura că absorbanta aparentă a cerului senin este 0, necesară datorită micii diferențe între intensitatea de fundal și imaginile de probă rezultate din diverse unghiuri de vizualizare. Acest lucru se realizează prin selectarea de către utilizatorul aplicației a unui punct din imagine unde absorbanta poate fi considerată nulă. Astfel poate fi calculată valoarea offset din formulă.

Conversia din absorbanta aparentă în ppm.m este realizată prin calibrare folosind celule cu concentrație cunoscută de SO₂. Este necesar să utilizăm 2 filtre în aceste observații pentru a compensa împrăștierea aerosolului și această abordare minimizează nepotrivirile temporale asociate cu schimbarea filtrului pe o singură cameră.

Fluxul de SO₂ poate fi calculat prin integrarea sumei pe o linie într-o imagine perpendiculară pe direcția vântului. Pentru acest lucru este necesar să știm viteza vântului la înălțimea coloanei de fum precum și dimensiunea reală a unui pixel de imagine.

Mărirea fiecărui pixel la o distanță a penei este necesar pentru a calibra scala lungime; acest lucru este ușor de obținut prin comparație cu caracteristicile

geometrice ale edificiului. Astfel utilizatorul aplicației poate selecta în imagine o zonă de dimensiune cunoscută oferind astfel programului suficiente informații pentru a calcula dimensiunea reală a unui pixel de imagine.

Pasul următor din algoritmul de calcul al fluxului de SO₂ este calcularea vitezei vântului. Acest lucru este realizat prin selectarea în doua imagini consecutive a unor zone ale penelor de fum asemănătoare. Astfel se poate calcula viteza vântului la înălțimea penei de fum.

Fluxul mediu de SO₂ se poate estima din aceste date folosind estimările de viteză a vântului la înălțimea și dimensiunile efective ale coloanei. O estimare a fluxului de SO₂ poate fi extras din,

$$F = \rho Au \quad (4.2)$$

unde F este rata de flux (în kg s^{-1}), ρ este concentrația (în kg m^{-3}), A este aria secțiunii transversale a coloanei (m^2), iar u este viteza vântului (în ms^{-1}) la înălțimea coloanei.

În forma sa cea mai simplă, o cameră SO₂ se compune exclusiv dintr-o cameră sensibilă la UV și un singur filtru spectral de bandă (numit în cele ce urmează filtrul A), care permite numai radiațiilor într-un interval îngust de lungimi de undă care cuprinde structuri semnificative de absorbție de SO₂ să intre în camera optică. Figura 3.41 arată un exemplu de curba de transmitanță $T_A(\lambda)$ a unui astfel de filtru de bandă de-a lungul secțiunii transversale de absorbție $\sigma(\lambda)$ de SO₂. Pentru fiecare pixel, semnalul de intensitate captat prin intermediul acestui filtru se compară cu intensitatea de fundal, fie măsurată într-o regiune în care coloana nu este prezentă sau interpolată din valorile obținute pe fiecare parte a coloanei.

În absența SO₂, intensitatea luminoasă dependentă de lungimea de undă $I_{0,A}(\lambda)$ care ajunge la fiecare pixel al CCD-ului camerei este dată de intensitatea radiațiilor dispersate $I_S(\lambda)$ care ajung la instrument, la transmitanța filtrului $T_A(\lambda)$ și la eficiența cuantică $Q(\lambda)$ a detectorului.

$$I_{0,A}(\lambda) = I_S(\lambda) \cdot T_A(\lambda) \cdot Q(\lambda) \quad (4.3)$$

În cazul în care SO₂ este prezent pe traiectoria optică a radiației incidente, intensitatea spectrală este atenuată conform ecuației Beer-Lambert.

$$I_A(\lambda) = I_{0,A}(\lambda) \cdot \exp(-\sigma\lambda) \cdot S(\lambda) \quad (4.4)$$

Aici, $\sigma(\lambda)$ este secțiunea transversală de absorbție a SO₂, iar $S(\lambda)$ este densitatea coloanei sau integrala concentrației de SO₂ de-a lungul traiectoriei efective de lumină L (ideal, linia de vizare prin coloană).

$$S(\lambda) = \int_L c(x) dx \quad (4.5)$$

În coloanele de erupție vulcanică, absorbția radiației prin SO₂ este deseori suficient de puternică pentru a influența traiectoria luminii efectivă în interiorul coloanei (Kern et al., 2009a). Prin urmare, densitatea coloanei S este, de obicei, o funcție a lungimii de undă λ . Logaritmul negativ al $I(\lambda)/I_0(\lambda)$ dă densitatea optică $\tau(\lambda)$, care este direct proporțională cu densitatea coloanei de SO₂ $S(\lambda)$, dacă

absorbția de SO₂ este singura cauză a atenuării luminii la lungimea de undă λ de-a lungul traiectoriei optice.[107]

$$\tau(\lambda) = -\ln\left(\frac{I_A(\lambda)}{I_{0,A}(\lambda)}\right) = \sigma(\lambda)S(\lambda) \quad (4.6)$$

Un instrument spectroscopic cu rezoluție suficientă ar măsura densitatea optică τ pentru a extrage densitatea coloanei S , dar camera SO₂ folosește o abordare diferită. Pentru fiecare pixel al detectorului, intensitatea măsurată $I_{M,A}$ este descrisă prin integrala intensității incidente $I_A(\lambda)$ pe intervalul de transport al filtrului (indicele A denotă intensitățile după ce au trecut prin filtrul A , după cum se menționează mai sus).

$$I_{M,A} = \int_{\lambda} I_A(\lambda) d\lambda \quad (4.7)$$

Relația respectivă este valabilă pentru măsurarea fundalului $I_{N,A}$.

$$I_{N,A} = \int_{\lambda} I_{0,A}(\lambda) d\lambda \quad (4.8)$$

Din aceste cantități măsurate, se extrage densitatea optică ponderată $\hat{\tau}_A$.

$$\hat{\tau}_A = -\ln\left(\frac{I_{M,A}}{I_{N,A}}\right) = \ln(I_{N,A}) - \left(\int_{\lambda} I_{0,A} \exp(-\sigma(\lambda)S) d\lambda\right) \quad (4.9)$$

Folosind notația definită în ecuațiile (3.19) și (3.20), densitatea optică ponderată $\hat{\tau}_A$ este dată de:

$$\hat{\tau}_A = -\ln\left(\frac{\int_{\lambda} I_S(\lambda)T_A(\lambda)Q(\lambda)\exp(-\sigma(\lambda)S(\lambda))d\lambda}{\int_{\lambda} I_S(\lambda)T_A(\lambda)Q(\lambda)d\lambda}\right) \quad (4.10)$$

În timp ce densitatea optică ponderată $\hat{\tau}_A$ este, evident, o funcție a densității coloanei de SO₂, dependența $\hat{\tau}_A$ față de S este, în general, neliniară. Dacă intervalul de transmitanță a lungimii de undă a filtrului este suficient de îngust ca secțiunea transversală σ a SO₂ și densitatea coloanei S să poată fi considerate independente de lungimea de undă λ în fereastra de măsurare, densitatea optică integrată poate fi scrisă ca:

$$\hat{\tau}_A = -\ln\left(\frac{I_{N,A} \exp(-\sigma S)}{I_{N,A}}\right) = \sigma S = \tau \quad (4.11)$$

Densitatea optică ponderată τ^A egală cu densitatea optică τ a SO₂ la lungimea de undă măsurată, este proporțională cu densitatea coloanei S.

Relația dintre densitatea optică ponderată τ^A măsurată și densitatea coloanei de SO₂ S este complexă, dar calibrarea empirică poate simplifica procesul de evaluare a datelor. O condiție care trebuie îndeplinită pentru ca principiul de măsurare să rămână valabil este faptul că absorbția de SO₂ este singurul parametru care influențează atenuarea luminii în fereastra lungimii de undă de emisie a filtrului A aplicat. În timp ce această condiție ar putea fi aproximativ îndeplinită în cazul coloanelor de erupție vulcanică pe cer senin și translucid, fără aerosoli, ea nu va fi general valabilă.

În prezența aerosolilor (picături de apă sau cenușă) în coloana de erupție vulcanică, radiațiile care intră în coloană din spate vor fi parțial dispersate în afara câmpului vizual al instrumentului, provocând astfel o scădere în intensitate. Pe de altă parte, lumina care intră în coloană din alte unghiuri va fi parțial dispersată spre instrument, provocând astfel o creștere în intensitate. În funcție de albedoul de dispersie unic al aerosolilor coloanei, și radiațiile vor fi absorbite într-o oarecare măsură de către particule.

În contrast cu absorbția SO₂, care este de departe cea mai puternică din lungimile de undă UV, între aproximativ 250 și 320 nm, dispersia de aerosoli și procesele de absorbție sunt pe bandă largă, adică sunt doar slab dependente de lungimea de undă a radiațiilor.

Prin urmare, un al doilea filtru de bandă cu o fereastră de transmitanță peste 320 nm, numit în cele ce urmează filtrul B, poate fi folosit pentru a cuantifica cantitatea de atenuare (sau creștere) luminoasă, care nu provine din absorbția de SO₂.

Atât imaginile de fundal ($I_{N,B}$), cât și ale coloanei ($I_{M,B}$) sunt înregistrate prin filtrul B. Înainte de a se calcula densitatea optică ponderată τ^A , raportul intensităților pixelilor măsurate prin filtrul A ($I_{M,A}/I_{N,A}$) este normalizat prin raportul intensităților pixelilor măsurate prin filtrul B ($I_{M,B}/I_{N,B}$). Densitatea optică ponderată τ^B poate fi calculată și scăzută din densitatea optică τ^A . Valoarea astfel extrasă este normalizată la densitatea optică τ^A , numită uneori și absorbanta aparentă AA.

$$\hat{\tau} = AA = -\ln\left(\frac{I_{M,A}/I_{N,A}}{I_{M,B}/I_{N,B}}\right) = \ln\left(\frac{I_{M,A}/I_{M,B}}{I_{N,A}/I_{N,B}}\right) = \hat{\tau}_A + \ln(I_{M,B}/I_{N,B}) = \hat{\tau}_A - \hat{\tau}_B$$

(4.12)

4.6 Rezumat al capitolului Campanii de măsurători

Măsurătorile cu camera UV și IR s-au realizat la CET Arat și Romag Termo Halânga în anii 2011 și 2012.

În urma campaniilor de măsurători s-au procesat datele obținute și s-au calculat concentrațiile de SO₂, respectiv fluxul de SO₂.

Valorile obținute prin metoda UV și IR s-au trecut în tabele și s-au comparat între ele pentru a trage concluzii referitoare la metodele experimentale aplicate.

Comparând rezultatele, eroarea este sub 10%, ceea ce înseamnă că se poate aplica cu succes teledetecția pentru măsurarea concentrației de poluant.

Metoda UV, prin algoritmi de calcul își extinde calculele la fluxul de SO₂ (calculare ce nu se face la metoda TEST

5 Aplicație de procesare a datelor experimentale - Camera UVIR V1.0

5.1 Prezentarea generală

Această aplicație a fost dezvoltată pentru a ușura calculul și extragerea de informații din pozele UV și IR obținute de la diverse obiective ce trebuie analizate. Această aplicație implementează calculul de flux SO_2 și se bazează pe poze UV realizate cu camera specială, precum și a unor poze etalon.

Datorită faptului că nu a putut fi momentan specificată exact platforma pe care va rula această aplicație a apărut necesitatea dezvoltării acesteia în Java, acest limbaj oferind posibilitatea rulării programelor multiplatformă.

Clasele implementate pentru această aplicație au fost ierarhizate în următoarele pachete:

pachetul *com.cata.uvcamera.app*. Acest pachet conține clasa *UVCameraApp* (în interiorul acestei clase se află metoda *main* utilizată pentru lansarea aplicației).

pachetul *com.cata.uvcamera.app.process*. Acest pachet conține clasele folosite pentru aplicarea algoritmului de calcul asupra seriei de fotografii încărcate ce fac parte din măsuratoarea curenta.

pachetul *com.cata.uvcamera.app.ui*. Acest pachet conține toate clasele folosite pentru interfața aplicației. Acestea sunt clasele utilizate pentru configurarea elementelor de control pe fereastra principală, precum și clasele ce implementează afișarea imaginilor și a histogramelor.

pachetul *com.cata.uvcamera.app.util*. Acest pachet conține clasele utilitare.

Notă: Această ierarhie împreună cu toate clasele implementate sunt prezentate în Anexa 1.

Aplicația oferă posibilitatea utilizatorului de a deschide un folder în care se află o serie de imagini în formatele acceptate. Acest mod de lucru oferă utilizatorului posibilitatea de a defini și păstra într-o formă ușor de folosit toate datele referitoare la o anumită măsurătoare.

Aplicația se prezintă sub forma unei ferestre principale care conține un meniu și mai multe zone de afișare + numite panels.

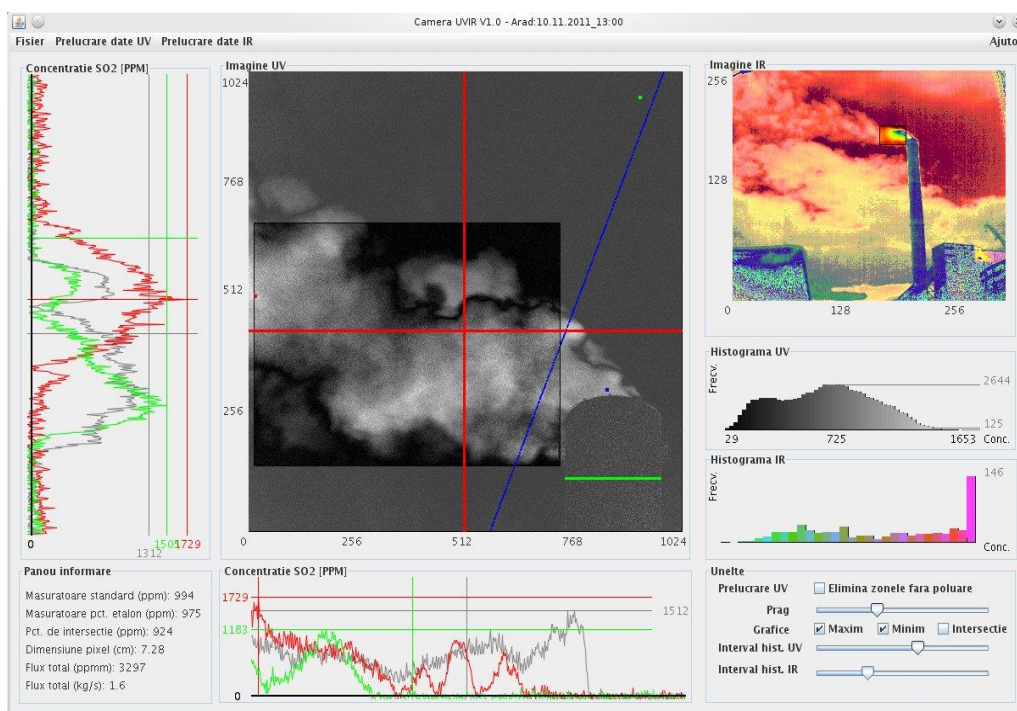


Figura 5.1 Interfața programului Camera UVIR

5.2 Prezentarea meniului.

5.2.1 Meniul "Fișier"

Pentru încărcarea fișierelor corespunzătoare unei măsurători, se accesează meniul fișier → deschide.

Utilizându-se această funcționalitate se va selecta folderul ce conține toate fișierele de date ale măsurătorii.

Aceste fișiere sunt:

- c1l_310* – fișier măsurătoare cer cu celula de calibrare A46 cu filtrul de 310 nm
- c1l_330* – fișier măsurătoare cer cu celula de calibrare A46 cu cu filtrul de 330 nm
- c3h_310* – fișier măsurătoare cer cu celula de calibrare A49 cu cu filtrul de 310 nm
- c3h_330* – fișier măsurătoare cer cu celula de calibrare A49 cu cu filtrul de 330 nm
- m_310* – fișier măsurătoare pană de poluant cu filtrul de 310 nm
- m_330* – fișier măsurătoare pană de poluant cu filtrul de 330 nm
- sky_310* – fișier măsurătoare a cerului cu filtrul de 310 nm
- sky_330* – fișier măsurătoare a cerului cu filtrul de 330 nm
- masuratoare.txt* – fișier text ce conține concentrația de SO₂ măsurată cu Testo, localitatea unde s-au efectuat măsurătorile
- ir.png* – fișier măsurătoare pană de poluant cu camera IR

5.2.2 Meniul "Prelucrare date UV"

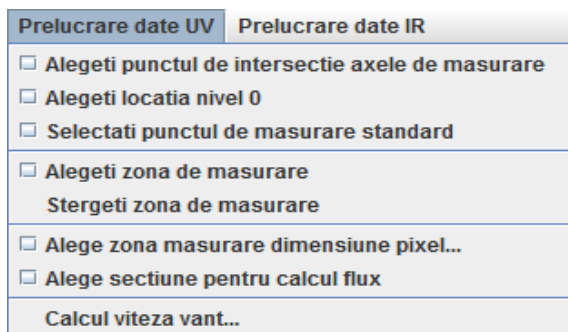


Figura 5.2 Afișarea meniului Prelucrare date UV

Meniul "Prelucrare date UV" oferă:

Alegerea punctului de intersectie al axelor de măsurare.

Alegerea locației nivelului de concentrație considerat nul în poza UV afișată.

Selectare punctului unde se considera ca s-a efectuat măsurătoarea standard.

Alegerea și marcarea zonei unde se va efectua măsurătoarea

Stergerea zonei de măsurare selectate anterior.

Alegerea zonei de măsurare pentru calculul dimensiunii reale a unui pixel din imaginea UV.

Alegerea secțiunii ce va fi utilizată pentru calculul fluxului de SO₂ în pana de fum.

5.2.3 Meniul "Prelucrare date IR"



Figura 5.3. Afișarea meniului Prelucrare date IR

Meniul "Prelucrare date IR" oferă:

Alegerea și marcarea zonei unde se va efectua măsurătoarea. Aceasta zona de măsurare este folosită pentru afișarea histogramei frecvenței.

Stergerea zonei de măsurare selectate anterior.

5.2.4 Meniul "Help"

Acest meniu ofera utilizatorului informații despre autorul aplicației, și poate fi eventual extins pe viitor pentru a oferi utilizatorului și alte informații utile privind algoritmul de calcul folosit în aplicație precum și utilizările ei practice.



Figura 5.4 Afișarea meniului Help

5.3 Prezentarea zonelor de afișare.

5.3.1 Imagine UV si zonele de afișare ce reprezintă date UV

Zona de afișare UV este utilizată pentru afișarea imaginii UV ce este obținută după aplicarea algoritmului de calcul descris în capitolul **Error! Reference source not found.** Imaginile încărcate din folderul selectat sunt citite și apoi folosite ca matrici de valori în algoritmul de calcul. Rezultatul obținut după efectuarea calculelor implementate în clasa *com.cata.uvcamera.app.process.Processor* este tot o matrice de valori. Pentru a putea afișa această matrice sub forma unei imagini se vor scala valorile astfel încât acestea să fie cuprinse între 0-255. Astfel se va afișa o imagine rezultantă folosind nuanțe de gri. Pe această imagine vor putea apoi să fie efectuate restul de calcule și selecții ale zonelor de măsurare, precum și afișarea histogrammei.

5.3.2 Poziționarea punctului de intersecție axe

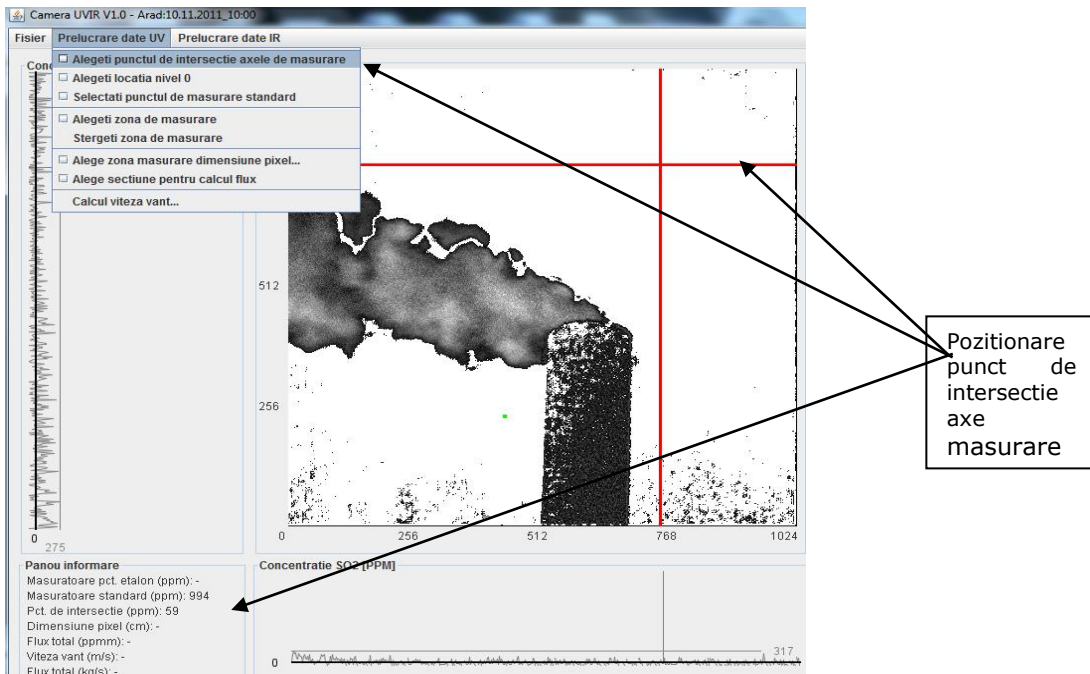


Figura 5.5 Poziționare punct de intersecție axe

Pentru a alege un punct de intersecție, utilizatorul trebuie să acceseze din meniul principal prelucrare date UV → alegeți punctul de intersecție, axele de măsurare și apoi cu cursorul să fixeze punctul de intersecție.

Variația concentrației de SO_2 pe axele verticale și orizontale, în punctul arbitrar ales este afișat în graficele de culoare gri, alăturate imaginii. Valoarea concentrației de SO_2 din punctul ales este afișată în panoul de informare.

5.3.3 Poziționarea punctului de concentrație de SO_2 nul

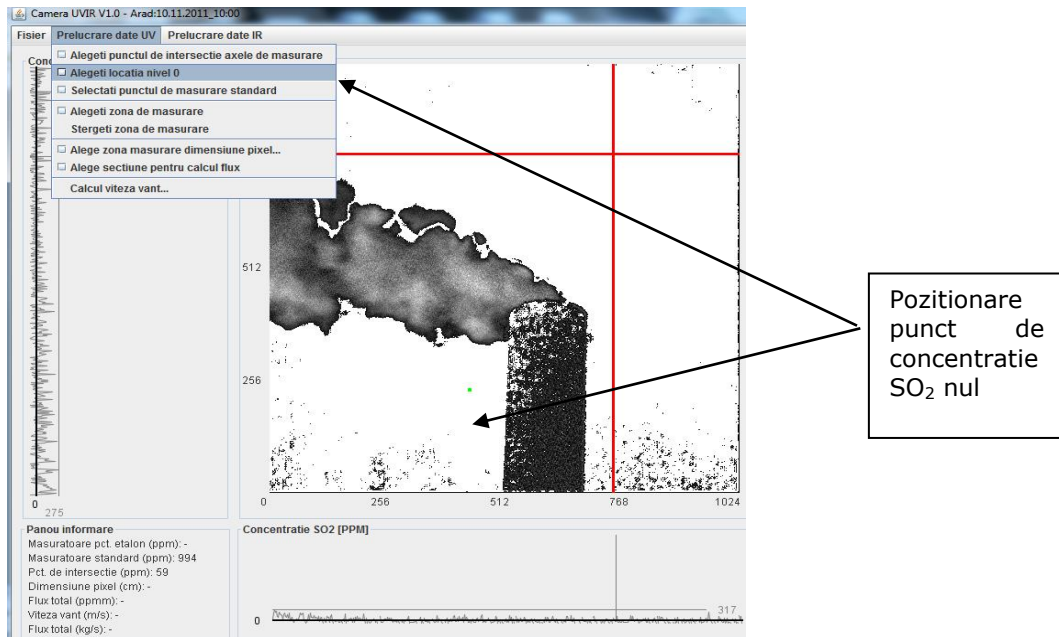


Figura 5.6. Poziționarea punctului de concentrație de SO_2 nul

Pentru a alege punctul de concentrație de SO_2 nul, utilizatorul trebuie să acceseze din meniul principal prelucrare date UV → alegeți locația nivel 0, iar cu cursorul să selecteze un punct din cerul senin, unde concentrația de SO_2 se presupune a fi zero.

Valoarea concentrației de SO_2 calculată în acest punct va fi utilizată în ultimul pas al algoritmului de calcul, prezentat în continuare.

5.3.4 Poziționarea punctului de măsurare SO₂ standard

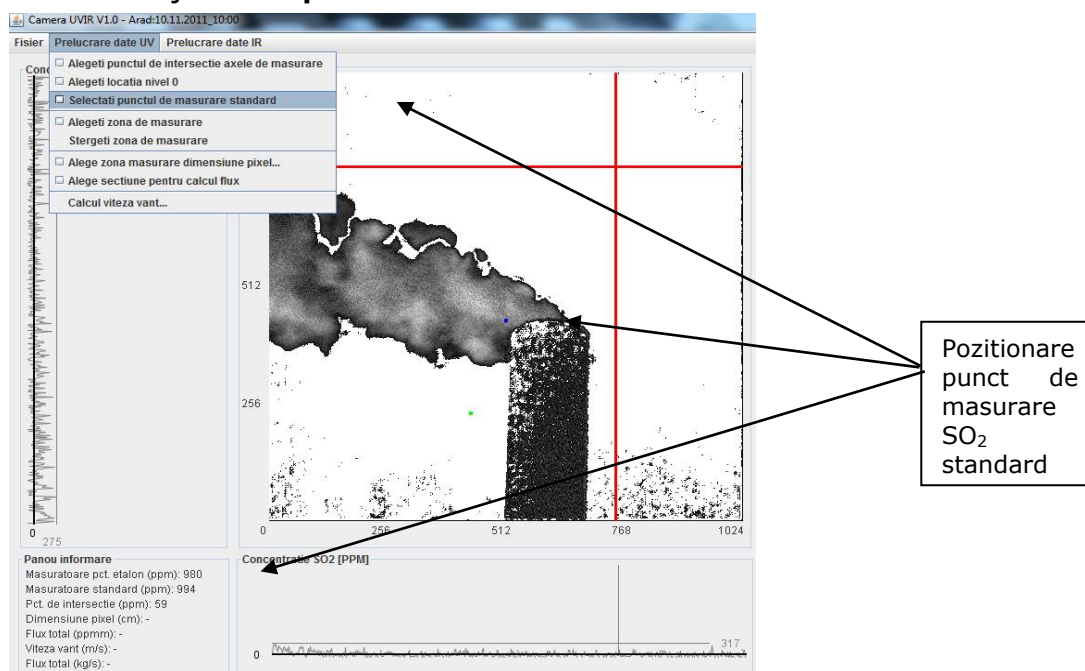


Figura 5.7. Poziționarea punctului de măsurare SO₂ standard

Pentru a alege punctul de măsurare SO₂ standard, utilizatorul trebuie să acceseze din meniul principal prelucrare date UV → selectați punctul de măsurare standard, iar apoi cu cursorul să aleagă un punct de la gura de evacuare a coșului.

Această opțiune oferă posibilitatea comparării concentrației calculate din imaginile UV cu valoarea unei măsurări directe.

Valorile măsurării directe standard, precum și a celei calculate, sunt prezentate pentru comparație, în panoul de informare.

5.3.5 Definirea zonei de măsurare

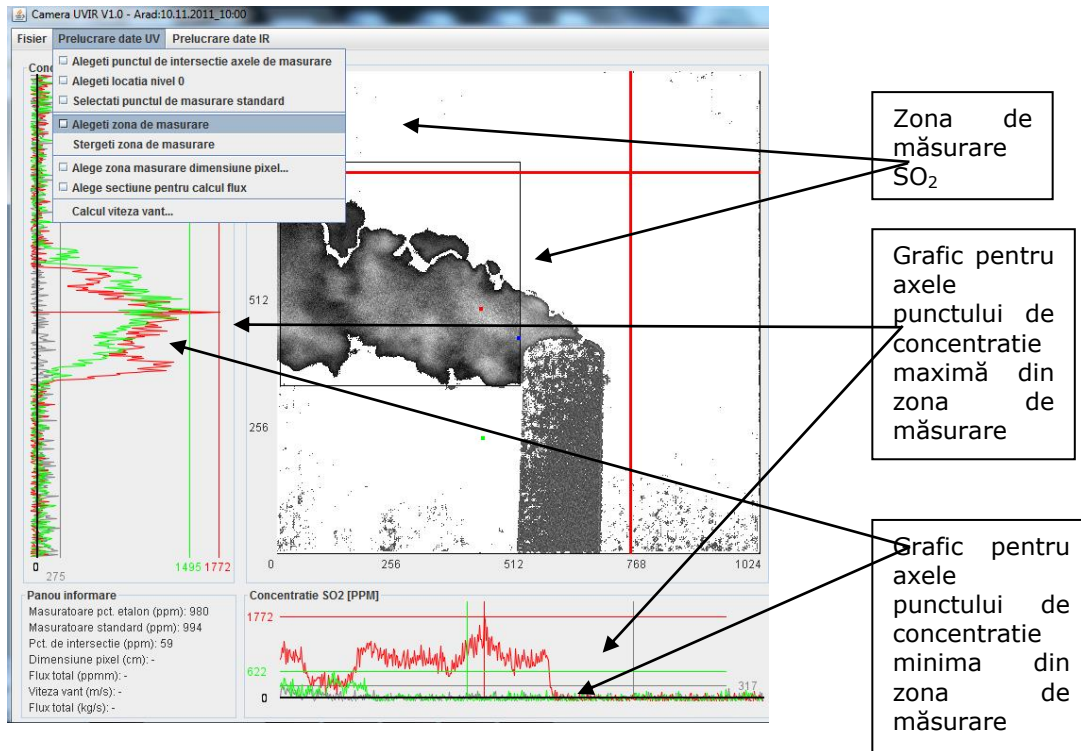


Figura 5.8 Zona de măsurare

Aplicația oferă utilizatorului posibilitatea de a defini o suprafață rectangulară, din zona fumului.

Pentru a alege zona de măsurare SO_2 , utilizatorul trebuie să aleagă din meniul principal prelucrare date UV → alegeți zona de măsurare, iar cu cursorul să definească zona respectivă.

Pentru zona de măsurare SO_2 se afișează în panourile din stânga și din josul imaginii cu linie roșie, variațiile de concentrație SO_2 pentru axele verticală și orizontală din punctul de concentrație maximă, calculat în zona de măsurare.

Totodată cu linie verde se afișează variațiile de concentrație SO_2 , pentru axele verticală și orizontală din punctul de concentrație minimă, calculat în zona de măsurare.

Afișarea graficelor poate fi dezactivată folosind checkboxurile corespunzătoare din panoul unelte.

5.3.6 Dimensionare pixel

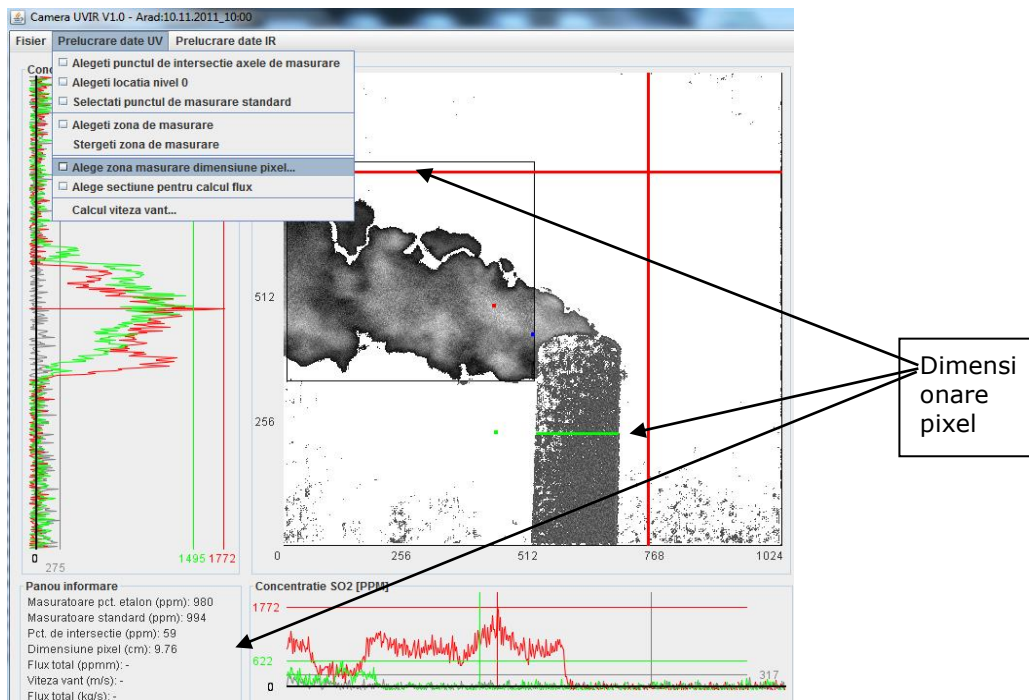


Figura 5.9. Dimensionare pixel

În algoritmul calculului de flux SO_2 este necesar calibrarea scalei lungime. Acest lucru este obținut prin comparația cu caracteristicile geometrice ale unui edificiu de dimensiune cunoscută, putându-se obține astfel mărimea unui pixel.

Pentru a determina dimensiunea pixelului, utilizatorul trebuie să acceseze din meniul principal prelucrare date UV → alegeți zona măsurare dimensiune pixel, iar cu cursorul utilizatorul alege o zonă cunoscută dimensional din fotografie pe care o introduce în cm.

Dimensiunea pixelului în cm este afișată în panoul informare.

5.3.7 Alegere secțiune pentru calcul flux

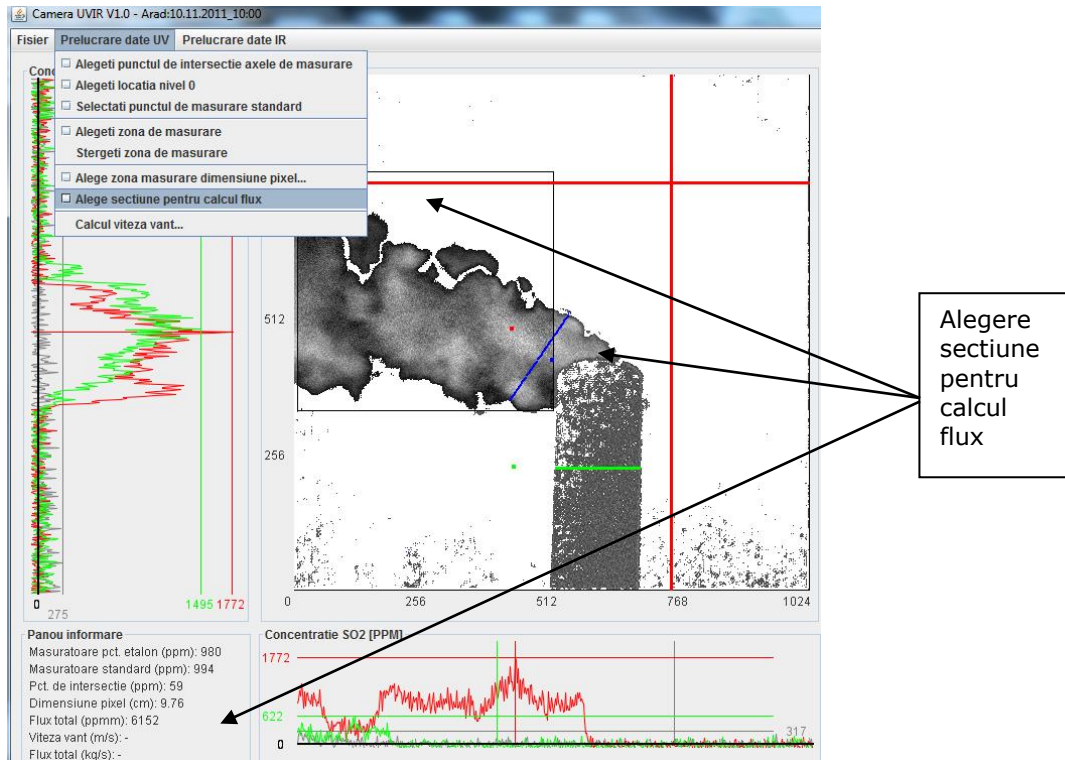


Figura 5.10 Alegere secțiune pentru calcul flux

Fluxul de SO_2 poate fi calculat prin integrarea sumei pe o linie, într-o imagine perpendiculară pe direcția vântului.

Pentru a determina fluxul de SO_2 utilizatorul trebuie să acceseze din meniul principal prelucrare date UV → alege secțiunea pentru calcul flux, iar cu cursorul utilizatorul va alege o secțiune prin norul de fum pentru care se va calcula fluxul de SO_2 . Această secțiune trebuie să fie aleasă perpendiculară pe direcția vântului.

Valoarea fluxului de SO_2 calculat este afișată în panoul informare.

5.3.8 Calcul viteza vânt

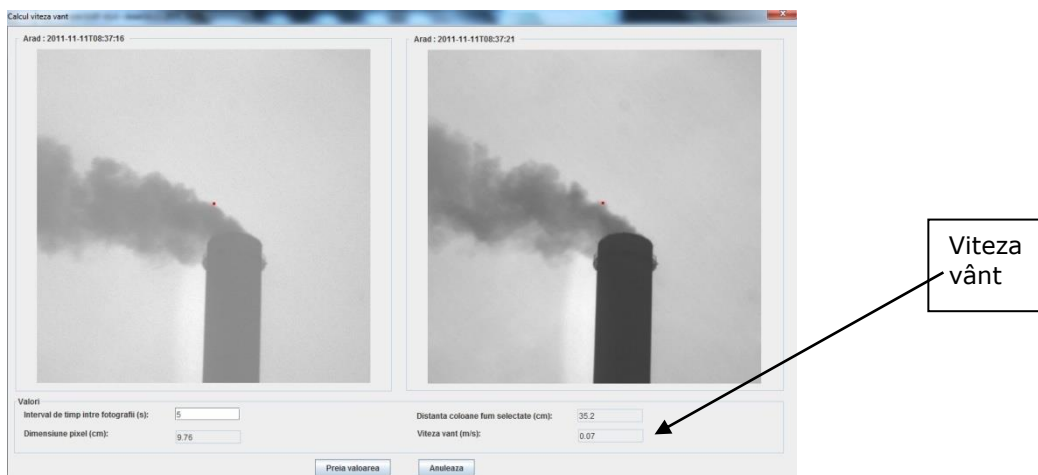


Figura 5.11 Calcul viteza vânt

Pentru a calcula viteza vântului la vârful coșului de fum se aleg două imagini consecutive cu pana de poluant și se caută un element comun al penei de poluant ce se află în cele două imagini. Deplasarea elementului comun în timp ne dă viteza vântului. Timpul scurs între două imagini succesive se cunoaște, fiind citit din headerul fișierelor. Valoarea vitezei vântului se preia folosind butonul Preia valoarea, fiind utilă în calculul de flux. Utilizatorul are posibilitatea de a renunța la calculul vitezei vântului folosind butonul Anulează.

5.3.9 Opțiunea "eliminarea cer senin"

Pentru vizualizarea corectă a penei de fum se poate utiliza opțiunea "eliminarea cer senin". Controlul de tip slider numit "prag" din panoul Unelte oferă posibilitatea utilizatorului de a regla intervalul de valori pentru concentrațiile ce sunt considerate zero.

Punctele de pe imagine ale căror valori se situează în acest interval, nu vor fi luate în considerare în niciunul din algoritmi de calcul implementați în această aplicație.

5.3.10 Histograma UV

După selectarea unei zone de măsurare, precum și după alegerea punctului de concentrație zero și ajustarea intervalului de valori nule se va afișa histograma punctelor de fum din zona de măsurare.

Histograma UV prezintă frecvența de apariție a concentrației de SO_2 în zona selectată.

5.4 Rezumat al capitolului Aplicație de procesare a datelor experimentale - Camera UVIR V1.0

În acest capitol sunt prezentate funcționalitățile programului Camera UVIR. Programul Camera UVIR a fost realizat pentru a procesa datele obținute în urma măsurărilor.

Capitolul începe cu prezentarea interfeței programului Camera UVIR. Urmează prezentarea meniurilor (Fișier, Prelucrare date UV, Prelucrare date IR, Help), luând fiecare meniu în parte pentru a fi discutat. Sunt prezentate în continuare zonele de afișare, precum și funcționalitățile meniului Prelucrare date UV și anume:

- Poziționare punct de intersecție axe
- Poziționare punct de concentrație SO₂ nul
- Poziționarea punctului de măsurare SO₂ standard
- Definirea zonei de măsurare
- Dimensionare pixel
- Alegere secțiune pentru calcul flux
- Calcul viteza vânt

Se poate folosi , de asemenea opțiunea "Eliminare cer senin" pentru vizualizarea corectă a penei de fum. Totodată, se poate realize histograma UV ce prezintă frecvența de apariție a concentrației de SO₂.

O histograma similară se face pentru camera IR, selectând zona de măsurare unde se poate face măsurătoarea.

6 Camera UVIR - tehnologia folosită

6.1 Prezentarea limbajului de programare JAVA.

Java este un limbaj de programare, un sistem runtime, un set de scule de dezvoltare și un *application programming interface* (API). Un programator poate scrie programe în limbajul Java folosind pachete software predefinite ale API-ului Java. Dezvoltatorul compilează programele folosind compilatorul Java. Rezultatul este cunoscut sub numele de cod binar compilat. Acesta este într-o formă care poate fi executată de mașina virtuală Java, inima sistemului runtime Java. Se poate imagina mașina virtuală ca un microprocesor care este implementat în software și rulează folosind capacitățile oferite de sistemul de operare precum și de hardware-ul computerului. Pentru că mașina virtuală Java nu este un microprocesor real, codul binar Java este interpretat în loc să fie executat de instrucțiunile mașinii native a computerului gazdă.[109]

Sistemul runtime Java este compus din mașina virtuală și de librăriile dinamice necesare pentru a implementa API-ul Java pe sistemul respectiv de operare.

6.2 Scurt istoric

Originile Java apar în 1991 când firma SUN investiga produsele electronice de consum. James Gosling, părintele Java, a intenționat să creeze un sistem ieftin, independent de platformă bazat pe C++. Pentru diverse motive C++ a fost abandonat și un alt limbaj de programare OAK a fost creat. Acest limbaj de programare elimina dezavantajele C++, cum ar fi moștenirea multiplă, conversiile automate de tipuri, utilizarea pointerilor și managementul memoriei.

În anul 1994, când WEB a apărut, OAK a fost redenumit în Java. HotJava, Java precum și documentația Java a apărut în varianta alpha la începutul anului 1995.

În ianuarie 1996 Java 1.0 a fost lansată oficial și a fost făcută publică pe Internet.3.3 Swing.

6.3 Tehnologii folosite

6.3.1 Swing

Swing este o nouă colecție de componente GUI (Graphic User Interface) care simplifică și ușurează dezvoltarea de componente și aplicații grafice. Componentele ferestre sunt componente vizuale (cum ar fi meniuri, tool-bars, dialoguri) care sunt folosite în applets și aplicații grafice.

Componentele Swing sunt parte din noua librărie de clase denumită Java Foundation Classes, sau JFC. Swing-ul este principala componentă din JFC. Java Foundation Classes este compusă și din Java 2D, Drag and Drop și Accessibility API.

Până la apariția pachetului Swing 1.0, programatorii Java creau GUI pentru applet-urile și aplicațiile lor utilizând Abstract Windowing Toolkit, sau AWT, care este pachetul standard pus la dispoziție de limbajul de programare Java. Swing 1.0 a fost inițial dezvoltat ca o librărie separată care putea fi folosită împreună cu JDK 1.1.

Versiunea curentă a pachetului Swing (1.1.) este integrată în JDK, care este numit JDK 1.2. Când JDK 1.2. a fost lansat oficial în decembrie 1998, Swing precum și restul pachetului JFC a devenit oficial parte componentă a JDK. Componentele având stilul vechi de dezvoltare bazat pe AWT sunt încă suportate în JDK 1.2. și continuă să fi folosite foarte ușor. Dar componentele Swing sunt mult mai puternice și mai elegante decât cele din AWT ceea ce a făcut ca majoritatea dezvoltatorilor să migreze spre Swing.

6.3.1.1 Capabilitățile pachetului Swing.

Una dintre cele mai importante capabilități ale pachetului Swing este cea de "*look and feel*": aceasta este o facilitate care lasă programatorul să aleagă felul în care componentele grafice pe care le folosesc în aplicațiile lor arată și se comportă (*look and feel*).[46]

Cu această capabilitate a pachetului Swing programatorul poate face componentele din aplicația dezvoltată să aibă ce mod de afișare (*look*) precum și ce comportament (*feel*) dorește. Dacă programatorul dorește, le poate face să apară și să se comporte ca și componentele native de pe calculatorul utilizatorului. Dacă preferă poate face componentele din aplicația dezvoltată să aibă același mod de afișare și de comportament indiferent de platforma pe care aceasta aplicație rulează — asta înseamnă că acest program va arata și se va comporta întotdeauna la fel indiferent de sistemul utilizat. A treia alternativă oferită programatorului Java este aceea de a crea un *look and feel* individual, unic pentru aplicația proprie.

Programatorul poate folosi capabilitățile *look and feel* ale pachetului Swing în multe feluri. Dacă aplicația este proiectată pentru o firmă se pot folosi capabilitățile *look and feel* ale pachetului Swing pentru a insera sigla firmei sau alte imagini proprii firmei în componentele grafice utilizate. Dezvoltatorii de jocuri pe calculator pot dezvolta un set de componente grafice care să reflecte tema individuală a jocului sau a nivelului jocului. Indiferent de ce tip de aplicație este dezvoltată programatorul poate decide să schimbe felul de afișare și de comportament al aplicației în orice moment la unul din tipurile de *look and feels* care este implementat în pachetul Swing. Toate aceste lucruri se pot face datorită posibilităților de individualizare ale componentelor Swing.[102]

Dacă programatorul se așteaptă ca programul să ruleze pe o singură platformă — ca Windows, UNIX sau Macintosh — atunci aplicația specifică *look and feel*-ul folosit și aplicația blochează acest *look and feel* imediat ce este lansată. Dacă aplicația este proiectată pentru a rula pe mai multe platforme programatorul utilizează un *look and feel* diferit pentru fiecare platformă. Dacă programatorul specifică un *look and feel* care nu este valid pe sistemul utilizatorului atunci comportarea implicită a componentelor Swing este cea a *look and feel*-ului Java.

6.3.1.2 O scurtă istorie Swing

Cum a fost prezentat mai înainte, este posibil să se proiecteze o aplicație cu un set de componente Swing care au un *look and feel* exact ca și componentele native utilizate pe oricare platformă pe care programul poate rula. Dacă aplicația este proiectată în modul acesta și un utilizator execută acest program sub Windows el va apare și se va comporta exact ca un program scris pentru Windows. Dacă utilizatorul rulează aceeași aplicație sub UNIX ea va arata și se va comporta exact ca orice program scris pentru UNIX — și așa mai departe.

Componente care apar și se comportă diferit în funcție de sistemul de operare pe care rulează nu este un concept nou. În perioada dinaintea Swing-ului, componentele AWT erau proiectate să aibă *look and feel*-ul de pe sistemul utilizatorului. Dar componentele Swing au îmbunătățit capabilitățile *look and feel* ale sculelor AWT în două direcții importante:

Componentele Swing ușoare: În primul rând componentele Swing sunt proiectate în concordanță cu proiectarea "ușoară" care a fost introdusă în JDK 1.1. Asta înseamnă că componentele Swing nu utilizează nici o implementare specifică platformei. Astfel componentele Swing creează componentele proprii folosind module *look and feel* care au fost scrise din nou și care nu folosesc deloc codul platformei. În consecință componentele Swing pot fi încorporate într-un program folosind mai puțin cod decât vechile componentele denumite "grele". Din cauză că aceste componente Swing folosesc mai puține resurse și produc aplicații mai mici și mai eficiente decât cele care folosesc componente AWT, aceste componente ușoare Swing trebuiesc folosite în toate aplicațiile.[101]

Componentele Swing cu capabilități look and feel: Al doilea avantaj pe care componentele Swing îl au asupra componentelor AWT este că ele pot fi folosite pentru a dezvolta capabilități *look and feel* individuale. Un program Swing poate folosi un *look and feel* identic indiferent pe ce mașină rulează el, sau unul care diferă în funcție de platforma particulară pe care rulează. Swing lasă, pe de altă parte, programatorului posibilitatea să își dezvolte un *look and feel* pentru programele proprii individual exact cum acesta și-a dorit.

6.3.1.3 Swing și AWT

Din perspectiva API (Application Programable Interface) setul de componente Swing extinde, dar nu înlocuiește, setul de componente AWT (Abstract Windowing Toolkit).

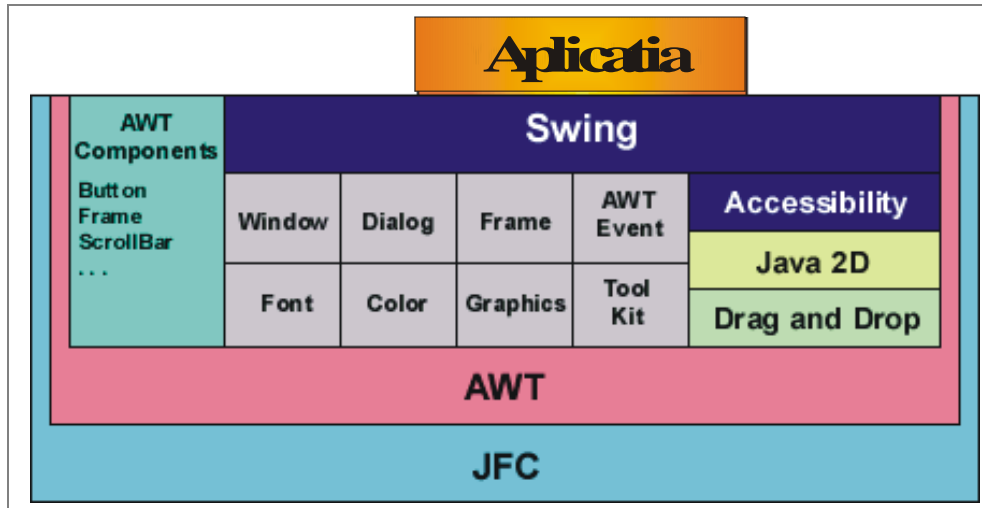


Figura 6.1. Setul de componente Java

Figura prezentată mai sus arată că Swing stă în vârful unui număr de API-uri care implementează multiplele părți ale JFC — incluzând aici Java 2D și Drag and Drop API. Cu toate că aceste API-uri sunt parte a pachetului JFC ele nu sunt parte a Swing-ului. Asta din cauză că anumite părți din comportamentul lor necesită cod nativ, față de componentele Swing care am văzut că nu folosesc deloc cod nativ.[108]

6.3.1.4 Introducere în arhitectura Swing

În prima perioadă a dezvoltării Swing au existat rapoarte care descriau arhitectura Swing ca fiind una care folosește arhitectura MVC (*model-view-controller*). După cum a reieșit aceste rapoarte au fost parțial adevărate dar nu complete.

Arhitectura MVC este bine cunoscută pentru dezvoltarea obiectelor GUI și care a apărut cu limbajul SmallTalk. Arhitectura clasică MVC împărțea fiecare componentă în trei părți: un model, o vizualizare și un controler, cum este prezentat în diagrama următoare.

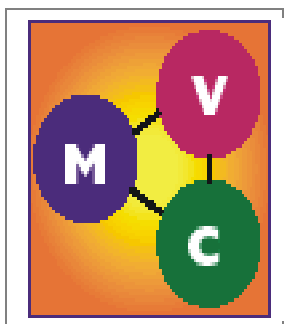


Figura 6.2 Arhitectura MVC

În această arhitectură modelul se ocupa de datele conținute de acea componentă. Componenta de vizualizare controla modul în care acel obiect era afișat utilizatorului, pe când controlerul determina ce se întâmplă când utilizatorul interacționează cu acel obiect.

Swing a modificat această arhitectură. Arhitectura MVC este una simplă și elegantă, și echipa Swing a început să proiecteze și să dezvolte setul de componente Swing folosindu-se de ea. Dar curând ei au descoperit ca arhitectura clasică MVC nu se comporta bine în lumea aplicațiilor Swing. Așa ca au pus bazele componentelor Swing pe o arhitectura MVC modificată. Această nouă arhitectură a fost numită arhitectura modelului separabil (*separable model architecture*)[19].

În această arhitectură, partea de vizualizare și de control este pusă în aceeași componentă numită UI (user interface). Rezultatul este o arhitectură ca aceasta:

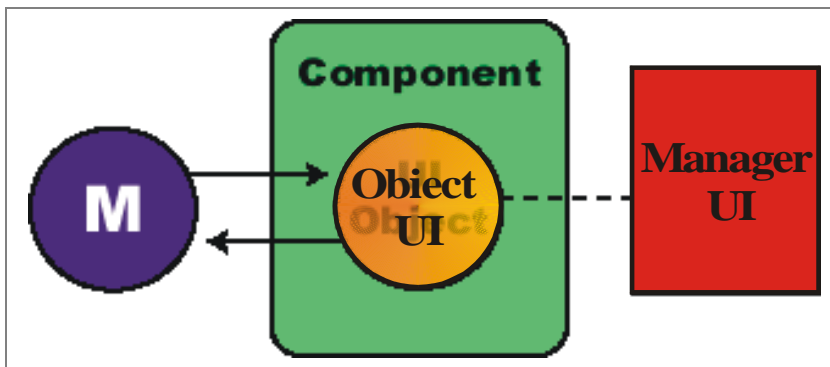


Figura 6.3 Componenta UI

Această arhitectură a fost dezvoltată din cauză că creatorii Swing-ului au găsit că arhitectura tradițională MVC nu se comporta bine în componentele Swing. Asta din cauză ca cele două componente (de vizualizare și de control) necesitau un cuplu foarte strâns care uneori era greu de atins practic. De exemplu arhitectura MVC făcea să fie foarte greu să se creeze un controler generic care să nu știe în partea de proiectare la ce fel de componentă de vizualizare va fi folosit.

Pentru dezvoltatorii de aplicații și applets Swing acesta era un mare avantaj. Cu Swing este ușor de prezentat structuri de date complexe într-o formă atractivă și ușor de înțeles.

6.3.1.5 Componente Swing folosite.

În această aplicație au fost folosite următoarele componente Swing:

javax.swing.JFrame – containerul principal al unei aplicații Swing este *JFrame*. Toate obiectele asociate cu *JFrame* sunt administrate de singurul sau "copil", o instanță *JRootPane*. *JRootPane* este un simplu container pentru mai multe paneluri. Când adăugăm componente la *JFrame*, nu le adăugăm direct la *JFrame* așa cum se făcea în AWT la *Frame*. În loc de acest lucru trebuie specificat în care panel al *JRootPane* vrem să plasăm componenta. În cele mai multe cazuri componentele sunt adăugate la *ContentPane* în următorul mod:

```
getContentPane().add( myComponent );
```

Similar când se dorește setarea unui layout pentru conținutul unui *JFrame* se folosește de obicei:

```
getContentPane().setLayout( new FlowLayout() );
```

javax.swing.JList – această clasă reprezintă o componentă GUI de bază care permite selectarea unuia sau mai multe elemente dintr-o listă de posibilități. Această componentă este folosită în această aplicație pentru afișarea unei liste de mesaje destinate utilizatorului.

javax.swing.JMenuBar – este un container pentru *JMenu* care se așează orizontal pe o linie. Barele meniului sunt așezate de obicei în partea de sus a ferestrei sau a apletului. Se folosește metoda `add(JMenu menu)` pentru a adăuga un nou *JMenu* la acest *JMenuBar*.

javax.swing.JMenu – Această clasă extinde *javax.swing.JMenuItem* și este de obicei adăugată la un *JMenuBar* sau la un alt *JMenu*. Dacă un *JMenu* este adăugat la un alt *JMenu*, el va apare în acel meniu ca un *JMenuItem* cu o săgeată în dreapta. Când acest meniu este activat de mișcarea mouse-ului sau de tastatură atunci un *popup* va apare care va afișa articolele conținute.

javax.swing.JMenuItem – această clasă extinde *AbstractButton* și reprezintă un singur articol de meniu. Se pot atașa icoane la aceste articole de meniu. La aceste articole se pot atașa și acceleratori pentru a le putea activa folosind tastatura.

javax.swing.JCheckBoxMenuItem – această clasă extinde *AbstractButton* și reprezintă un singur articol de meniu. Aceasta opțiune are afisat și un control de tip checkbox pentru prezentarea stării selectat/deselectat al opțiunii respective. La aceste articole se pot atașa și acceleratori pentru a le putea activa folosind tastatura.

javax.swing.JButton – *JButton* este unul dintre cele mai simple componente Swing. La această componentă se pot atașa imagini, se pot specifica texte și aliniamente de imagini, culori, fonturi precum și alte proprietăți. În plus se pot adăuga *ActionListeners*, *ChangeListener*, și *ItemListeners* pentru a primi *ActionEvents*, *ChangeEvent* și respectiv *ItemListeners* când orice proprietate se schimbă.

javax.swing.JFileChooser – Acest control de tip dialog ofera posibilitatea de a selecta un folder. Este folosit pentru selectarea folderului ce conține toate fișierele unei măsurători.

javax.swing.JSlider – Control folosit pentru selectarea unei valori discrete dintr-o plajă de valori selectate.

6.3.2 Spring

Introdus pentru prima dată în 2002 de către Rod Johnson în „Expert One-on-One J2EE Design and Development”, Spring a început ca un cadru simplu ce oferea o modalitate ușoară de a lega aplicațiile între ele. Acum însă poate fi considerat o platformă alternativă unei platforme Java EE complete. Include caracteristici ce se găsesc în mod normal la servere de aplicații complicate și scumpe: suport pentru tranzacții, securitate, mesagerie, acces de la distanță, servicii web, persistență și multe altele.

Cadrul Spring este o platformă pentru dezvoltarea și rularea de aplicații care poate fi împărțită în 2 moduri majore de funcționalitate. În centrul Spring se află un Kernel care este responsabil cu configurarea, administrarea și îmbunătățirea componentelor Spring în aplicații.

Unele beneficii ale IoC, când sunt îmbinate cu abstracțiile enterprise Spring, includ:
Decuplarea componentelor de mediul de execuție;
Separarea datelor de configurare de codul aplicației;
Abilitate facilă de a testa componentele în izolare folosind teste pe unitate, sau în combinație prin teste de integrare, fără modificarea codului existent;
Abilitatea de a schimba POJO-urile existente printr-o modalitate transparentă și declarativă de injectare de funcționalități în interiorul acestora, cum ar fi suport transparent, securitate, audit, creare de loguri, cache și altele.

Containerul spring (cu referire la AOP) este motorul de bază care comandă configurarea și administrarea componentelor, și îmbunătățește utilitatea celorlalte componente din spring. Controlul automat al relațiilor dintre obiecte, este asigurat de containerul ce se ocupă cu Injectia de dependențe (Dependency Injection – DI) sau Inversia controlului (Inversion of Control – IoC) întrucât cadrul are grijă ca obiectele să primească dependențele necesare.

Obiectele administrate spring, numite „beans”, sunt configurate de obicei folosind unul sau mai multe fișiere XML de configurare și/sau adnotări incluse în clasele respective. Spring suportă ambele mecanisme și încă câteva pentru o flexibilitate crescută. Containerul spring se numește Contextul aplicației (Application Context), și este posibil să existe mai multe contexte de aplicații într-o singură aplicație.

Componentele administrate spring, sunt numite „Obiecte Java vechi simple” (Plain Old Java Objects - POJOs). Aceste clase ale aplicației nu necesită implementarea interfețelor speciale, extinderea claselor de bază sau conștientizarea containerului în care sunt rulate. Acest lucru facilitează scrierea de cod portabil și reutilizabil. Cu toate acestea, dezvoltatorii încă pot să intervină în ciclul de viață al componentei și containerului dacă este nevoie.

În momentul execuției, obiectele sunt create în funcție de metadata de configurare și opțional împachetat cu un proxy dinamic într-o manieră transparentă. Acest lucru oferă un punct de extensie pentru cadru prin care se permite injectarea de servicii cheie cum ar fi suport pentru tranzacții, securitate, audit, și logare și este un exemplu al AOP (Aspect Oriented Programming).

6.3.3 Librăria EAP FITS

Pentru citirea fișierelor *.fit a fost folosită librăria eap.fits. Aceasta oferă posibilitatea citirii întregii structuri a unui fișier fits folosind clasa *eap.fits.FitsImageData*. Aceasta clasa oferă conținutul fișierului fit selectat sub formă de matrice de valori double prin citirea valorii fiecru pixel din imagine.

De asemenea se pot extrage și informațiile conținute în header-ul fișierului.

6.3.4 IzPack

Pentru instalarea facilă a aplicației și a librăriilor adiționale este necesară împachetarea claselor java compilate și a tuturor dependențelor într-un fișier

executabil. Acest fișier poate fi apoi folosit ca un program de instalare pe orice calculator ce rulează sistemul de operare compatibil. În versiunea actuală programul de instalare a fost creat doar pentru Windows dar pentru că aplicația a fost implementată în Java va fi ușor de creat un alt program pentru un alt sistem de operare compatibil.

Pentru crearea fișierului de instalare pentru aplicația Camera UVIR a fost necesară folosirea unui utilitar care permite ambalarea, distribuția și implementarea aplicațiilor. A fost ales IzPack – aceasta fiind un utilitar ușor de folosit și care permite operațiile necesare pentru aplicația Camera UVIR.

6.4 Rezumat capitol Camera UVIR - tehnologia folosită

Capitolul începe cu prezentarea limbajului de programare Java, urmat de un scurt istoric. În continuare sunt detaliate tehnologiile folosite (swing, spring, biblioteca EAP Fits, izPack).

Swing este o nouă colecție de componente GUI (Graphic User Interface) care simplifică și ușurează dezvoltarea de componente și aplicații grafice.

Una dintre cele mai importante capabilități ale pachetului Swing este cea de "look and feel": aceasta este o facilitate care lasă programatorul să aleagă felul în care componentele grafice pe care le folosește în aplicații, arată și se comportă (look and feel).

Din perspectiva API (Application Programmable Interface) setul de componente Swing extinde, dar nu înlocuiește, setul de componente AWT (Abstract Windowing Toolkit).

Introdus pentru prima dată în 2002 de către Rod Johnson în „Expert One-on-One J2EE Design and Development”, Spring a început ca un cadru simplu ce oferă o modalitate ușoară de a lega aplicațiile între ele. Acum însă poate fi considerat o platformă alternativă unei platforme Java EE complete.

Pentru citirea fișierelor *.fit a fost folosită biblioteca eap.fits. Aceasta oferă posibilitatea citirii întregii structuri a unui fișier fits folosind clasa *eap.fits.FitsImageData*.

Pentru instalarea facilă a aplicației și a librărilor adiționale este necesară împachetarea claselor java compilate și a tuturor dependențelor într-un fișier executabil. Acest fișier poate fi apoi folosit ca un program de instalare pe orice calculator ce rulează sistemul de operare compatibil.

7 Concluzii. Contribuții.

7.1 Concluzii generale

Originalitatea este conturată prin metodele experimentale complexe folosite și aplicate pe instalația experimentală, prin analizele efectuate, prin studiul corelativ al rezultatelor comparative și prin bogata exemplificare cu date și reprezentări grafice intuitive, atât pentru varianta constructivă, cât și pentru cele numerice.

Parcursul literaturii de specialitate reprezentative a permis găsirea celor mai performante soluții pentru problema abordată și a permis sistematizarea stadiului actual al cunoașterii în domeniu, în cadrul echipei de cercetare a Laboratorului de Combustibili și Investigații Ecologice, al Facultății de Mecanică din Timișoara.

Lucrarea prezintă rezultatele experimentelor efectuate cu metodele standard (TESTO) și cu metodele cu camerele UV și IR și se compară aceste rezultate între ele pentru a se demonstra eficiența lor.

O concluzie generală care rezultă în urma studiilor efectuate este aceea că în anumite condiții se pot obține date cu privire la fenomenul de poluare. Dacă se modifică compoziția atmosferei, prin introducerea unor poluanți apare fenomenul de poluare.

Măsurile de reducere a emisiilor de oxizi de sulf sunt: utilizarea combustibilului cu conținut redus de sulf - reducerea emisiilor de SO₂ la sursă; utilizarea de adsorbantți în sistem de ardere în strat fluidizat - reducerea emisiilor de SO₂ în interiorul cazanului.

Metoda UV este o metodă spectrometrică, ce este utilă în studiul poluării cu SO₂. Această metodă prezintă dificultăți atunci când vremea este nefavorabilă (ceață, ninsoare, ploaie). În lucrare, am obținut informații utile în privința modalităților de măsurare. Măsurătorile efectuate au constituit un pas important pentru măsurarea concentrației de SO₂ și al fluxului de SO₂.

Metoda IR, având la bază termografia, este utilă la studiul poluării cu SO₂. Am obținut informații utile în privința modalităților de măsurare. Sunt dificultăți în ceea ce privește procesul de calibrare.

În studiul efectuat s-a urmărit aplicarea unor metode moderne pentru măsurarea poluării. În acest sens datele experimentale au fost procesate, folosind programul Camera UVIR, fiind creat cu ajutorul limbajului de programare Java. Am folosit următoarele tehnologii ale limbajului Java: swing, spring, biblioteca EAP Fitts, izPack).

Swing este o nouă colecție de componente GUI (Graphic User Interface) care simplifică și ușurează dezvoltarea de componente și aplicații grafice. Una dintre cele mai importante capabilități ale pachetului Swing este cea de "look and feel": aceasta este o facilitate care lasă programatorul să aleagă felul în care componentele grafice pe care le folosesc în aplicațiile lor arată și se comportă (look and feel).

Din perspectiva API (Application Programmable Interface) setul de componente Swing extinde, dar nu înlocuiește, setul de componente AWT (Abstract Windowing Toolkit).

Cadrul Spring este o platformă pentru dezvoltarea și rularea de aplicații care poate fi împărțită în 2 moduri majore de funcționalitate. În centrul Spring se află un

Kernel care este responsabil cu configurarea, administrarea și îmbunătățirea componentelor Spring în aplicații.

Pentru citirea fișierelor *.fit a fost folosită librăria eap.fits. Aceasta oferă posibilitatea citirii întregii structuri a unui fișier fits folosind clasa *eap.fits.FitsImageData*. Aceasta clasa oferă conținutul fișierului fit selectat sub formă de matrice de valori double prin citirea valorii fiecru pixel din imagine.

Pentru instalarea facilă a aplicației și a librăriilor adiționale este necesară împachetarea claselor java compile și a tuturor dependențelor într-un fișier executabil. Acest fișier poate fi apoi folosit ca un program de instalare pe orice calculator ce rulează sistemul de operare compatibil.

Pentru crearea fișierului de instalare pentru aplicația Camera UVIR a fost necesară folosirea unui utilitar care permite ambalarea, distribuția și implementarea aplicațiilor. A fost ales IzPack – aceasta fiind un utilitar ușor de folosit și care permite operațiile necesare pentru aplicația Camera UVIR.

Stabilirea concentrațiilor poluanților specifici(SO₂) s-a realizat prin folosirea instrumentelor de lucru, a algoritmului de calcul ce se bazează pe fenomenul de absorbție.

Măsurătorile la centralele termice luate în studiu au fost repetate la diferite intervale de timp pentru a se urmări evoluția în timp a poluării cu SO₂. În acest sens, imaginile realizate cu camerele UV și IR au fost prelucrate cu programul Camera UVIR.

Programul Camera UVIR are mai multe meniuri (Fișier, Prelucrare date UV, Prelucrare date IR, Help) cu diferite funcționalități. Meniul Prelucrare date UV este cel mai complex și are următoarele funcționalități: poziționarea punctului de intersecție axe, poziționarea punctului de concentrație SO₂ nul, poziționarea punctului de măsurare SO₂ standard, definirea zonei de măsurare dimensionarea pixelului, alegerea secțiune pentru calcul flux, calcul viteza vânt.

Programul Camera UVIR este original și oferă posibilitatea de a arăta în Panoul Informare valorile concentrației de SO₂, dimensiunea pixelului, viteza vântului, valoarea fluxului de SO₂. Pentru zona de măsurare SO₂ se afișează în panourile din stânga și din josul imaginii cu linie roșie, variațiile de concentrație SO₂ pentru axele verticală și orizontală din punctul de concentrație maximă, calculat în zona de măsurare. Totodată cu linie verde se afișează variațiile de concentrație SO₂, pentru axele verticală și orizontală din punctul de concentrație minimă, calculat în zona de măsurare.

Programul de mai sus a fost alcătuit după o documentare amplă din literatura de specialitate(legea Lambert Beer) și am extins cercetările pentru calculul de flux de SO₂, calcul neefectuat prin metoda TESTO.

Valorile concentrațiilor de SO₂, pentru aceeași centrală termică(Arad sau Halânga), au fost trecute în tabele de valori și comparate pentru cele două metode, UV și Testo. Concluzia care s-a desprins prin aplicarea celor două metode a fost că se poate folosi metoda UV, eroarea fiind sub 10% față de metoda Testo.

Am folosit o abordare interdisciplinară a cercetării, prin preluarea de metode de lucru specifice din domeniul mecanic, chimic și al prelucrării datelor pentru studierea și măsurarea poluării cu SO₂.

Am lucrat cu aparatură modernă, semnalele primite de la senzori fiind înregistrate pe calculator și prelucrate cu programe specializate.

7.2 Contribuții practice

Lucrarea cu titlul „Cercetări aplicând spectrometria și termografia pentru determinarea concentrațiilor de SO₂ emise de centralele funcționând pe cărbune” oferă publicului o informare documentată asupra calității aerului.

Identificarea condițiilor optime de monitorizare cu metodele optice UV și IR pentru măsurarea calității aerului, în sensul oferirii de informații complementare și de comparare a valorilor măsurate prin metodele standardizate.

Prezentarea poluării atmosferice și axarea pe poluarea cu SO₂, având în vedere că este nocivă și apare în principal ca rezultat al combustiei combustibililor fosili.

Prezentarea diferitelor metode de măsurare a poluării atmosferice cu SO₂, prin parcurgerea literaturii de specialitate.

Stabilirea modului de efectuare al măsurătorilor cu cele două camere UV și IR.

Dezvoltarea cunoștințelor într-un domeniu de actualitate, cel al poluării cu SO₂.

Implicarea în activitatea de cercetare a departamentului în diverse proiecte, granturi și contracte cu terții.

Realizarea măsurătorilor experimentale cu camera UV și IR cu diverse filtre, cu dimensiuni diferite în vederea interpretării științifice a lor.

Determinarea valorilor concentrațiilor poluanților gazoși din gazele de ardere utilizând echipamentul - gazoanalizorul TESTO.

Stabilirea dependențelor dintre parametrii monitorizați (concentrație, flux de SO₂) cu programul Camera UVIR.

Determinarea concentrațiilor noxelor din gazele de ardere rezultate prin arderea cărbunelui de la centrale termice în urma campaniilor de măsurători.

Realizarea unui program informatic de procesare a datelor măsurate la centralele termice și afișarea concentrației și fluxului de SO₂.

Studiul corelativ al rezultatelor și interpretarea lor în funcție de diverși parametri și factori perturbatori.

Contribuția adusă este de actualitate și răspunde scopului general al societății românești de a cultiva și susține noțiunea de protecție a mediului în general, în particular, echilibrul în zonele urbane în privința calității aerului, al modului în care poate fi alterată și a cauzelor ce o determină.

7.3 Contribuții privind potențiala extindere a domeniului de cercetare

Asigură sprijin în evaluarea nivelului poluării și stării de sănătate a populației.

Constituie o bază de date la elaborarea strategiei locale privind calitatea aerului.

Sprijină autoritățile în limitarea desfășurării unor activități și realizarea unor proiecte cu efecte benefice asupra mediului natural și economic;

Poate fi drept suport în vederea elaborării Planului local de acțiune pentru protecția mediului.

Se realizează tabele și grafice pe variante experimentale, pe baza rezultatelor obținute în urma experimentelor efectuate, cu programul Camera UVIR.

Rezultatelor obținute pe parcursul cercetărilor au stat la baza îndeplinirii sarcinii Universității Politehnica din Timișoara în cadrul proiectului de cercetare RADO.

Dimensiunile și caracteristicile respectiv a punctelor de monitorizare a parametrilor funcționali ai instalației experimentale prezentate în această lucrare îi conferă și valoare didactică, pe lângă cea de cercetare. Studenților li se oferă o imagine la scară redusă a ceea ce înseamnă un sistem performant de măsurare a gazelor de ardere. Echipamentele de măsurare devin și bază materială pentru viitoare proiecte de licență, masterat sau alte teze de doctorat.

Instalația concepută, permite continuarea cercetărilor cu privire la îmbunătățirea metodologiei prezentate în particular și în special, perfecționarea unui sistem de măsurare complex a SO_2 .

Camera UV se pretează a fi un bun instrument de monitorizare viitoare a SO_2 datorită instalării sale rapide, utilizării practice pe teren, prelucrării în timp util a rezultatelor și prețului relativ scăzut.

Un nou instrument de măsurare a gazelor atmosferice utilizând radiometria de termodetecție pasivă în infraroșu a fost testată cu succes și fiind derivat un sistem pentru extragerea concentrației traiectoriei folosind formarea de imagini multispectrale. Instrumentul s-a dovedit fiabil și a putut să detecteze SO_2 în prezența vaporilor de apă.

Activitățile viitoare se vor concentra pe calibrarea corectă a camerei în condiții de laborator, precum și într-o varietate de condiții de mediu și vulcanologice.

Anexa 1 – Pachetele și clasele utilizate pentru implementarea aplicației Camera UVIR



com.cata.uvcamera.app.UVCameraApp.java

```

package com.cata.uvcamera.app;

public class UVCameraApp {

    public static void main(String[] args) {
        ClassPathXmlApplicationContext applicationContext = new
ClassPathXmlApplicationContext("uvcamera-service-context.xml");
        applicationContext.start();
    }
}

```

com.cata.uvcamera.process.Processor.java

```

package com.cata.uvcamera.process;

public class Processor {
    public static final String C3H_330 = "/c3h_330";
    public static final String C3H_310 = "/c3h_310";
    public static final String C1L_330 = "/c1l_330";
    public static final String C1L_310 = "/c1l_310";
    public static final String M_330 = "/m_330";
    public static final String M_310 = "/m_310";
    public static final String SKY_330 = "/sky_330";
    public static final String SKY_310 = "/sky_310";
    public static final String DARK = "/dark";
    private static final int IMAGE_SIZE = Util.UV_X_SIZE;
    private boolean hasResults = false;
    public Map<String, double[][]> data;
    private static final String EXTENSION = ".fit";
    private double[][] conc;
    private double so2C1l = 253;
    private double so2C3h = 873;
    private Map<String, String> fileNames;
    public double[][] resultC1l;
    public double[][] resultC3h;
    private int maxValue;

    private int maxX;
    private int maxY;
    private double[][] data2Return = null;
    private int min;

    final static org.slf4j.Logger log =
LoggerFactory.getLogger(UVCameraApp.class);

    public Processor() {
    }

    private void init() throws FileNotFoundException, IOException {
        fileNames = new HashMap<String, String>();
        try {
            fileNames.put(DARK, new File(Util.getFolder() + DARK +
EXTENSION).getAbsolutePath());
        } catch (Exception ignored) {
        }
        try {
            fileNames.put(SKY_310, new File(Util.getFolder() +
SKY_310
+ EXTENSION).getAbsolutePath());

```

```

        } catch (Exception ignored) {
        }
        try {
            SKY_330 fileNames.put(SKY_330, new File(Util.getFolder() +
                + EXTENSION).getAbsolutePath());
        } catch (Exception ignored) {
        }
        try {
            EXTENSION) fileNames.put(M_310, new File(Util.getFolder() + M_310 +
                .getAbsolutePath());
        } catch (Exception ignored) {
        }
        try {
            EXTENSION) fileNames.put(M_330, new File(Util.getFolder() + M_330 +
                .getAbsolutePath());
        } catch (Exception ignored) {
        }
        try {
            C1L_310 fileNames.put(C1L_310, new File(Util.getFolder() +
                + EXTENSION).getAbsolutePath());
        } catch (Exception ignored) {
        }
        try {
            C1L_330 fileNames.put(C1L_330, new File(Util.getFolder() +
                + EXTENSION).getAbsolutePath());
        } catch (Exception ignored) {
        }
        try {
            C3H_310 fileNames.put(C3H_310, new File(Util.getFolder() +
                + EXTENSION).getAbsolutePath());
        } catch (Exception ignored) {
        }
        try {
            C3H_330 fileNames.put(C3H_330, new File(Util.getFolder() +
                + EXTENSION).getAbsolutePath());
        } catch (Exception ignored) {
        }
        data = new HashMap<String, double[][]>();

        for (String key : fileNames.keySet()) {
            String fileName = fileNames.get(key);
            data.put(key, Util.readFile(fileName));
        }
    }

    private double[][] computeAA(double[][] dark, double[][] x310,
        double[][] x330, double[][] sky310, double[][] sky330,
    double offset) {
        double[][] result = new double[IMAGE_SIZE][IMAGE_SIZE];

        for (int i = 0; i < IMAGE_SIZE; i++) {
            for (int j = 0; j < IMAGE_SIZE; j++) {
                result[i][j] = -Math

```

168 Anexa 1

```

                                                                    .log(((x310[i][j] - dark[i][j]) /
(sky310[i][j] - dark[i][i]))
                                                                    / ((x330[i][j] -
dark[i][j]) / (sky330[i][j] - dark[i][i])))
                                                                    + offset;
        }
    }
    return result;
}

private double computeMean(double[][] data) {
    double mean = 0;

    for (int i = 0; i < IMAGE_SIZE; i++) {
        for (int j = 0; j < IMAGE_SIZE; j++) {
            mean = mean + data[i][j];
        }
    }

    mean = mean / (IMAGE_SIZE * IMAGE_SIZE);
    return mean;
}

private double[][] computeConc(double a, double[][] data, double b) {
    double[][] result = new double[IMAGE_SIZE][IMAGE_SIZE];

    for (int i = 0; i < IMAGE_SIZE; i++) {
        for (int j = 0; j < IMAGE_SIZE; j++) {
            result[i][j] = data[i][j] * a + b;
        }
    }
    return result;
}

public synchronized void startProcessing() throws FileNotFoundException,
    IOException {
    hasResults = false;
    data2Return = null;
    init();

    resultC1l = computeAA(data.get(DARK), data.get(C1L_310),
        data.get(C1L_330), data.get(SKY_310),
data.get(SKY_330), 0);
    resultC3h = computeAA(data.get(DARK), data.get(C3H_310),
        data.get(C3H_330), data.get(SKY_310),
data.get(SKY_330), 0);
    double meanC1l = computeMean(resultC1l);
    double meanC3h = computeMean(resultC3h);

    double a = (so2C3h - so2C1l) / (meanC3h - meanC1l);
    double b = (meanC1l * so2C3h - meanC3h * so2C1l) / (meanC1l -
meanC3h);
    double[][] resultM310 = computeAA(data.get(DARK),
data.get(M_310),
        data.get(M_330), data.get(SKY_310),
data.get(SKY_330), 0);
    conc = computeConc(a, resultM310, b);

    min = Integer.MAX_VALUE;
    int max = 0;
    for (int i = 0; i < IMAGE_SIZE; i++) {
        for (int j = 0; j < IMAGE_SIZE; j++) {
```



```
        if (conc[i][j] < min)
            min = (int) conc[i][j];
        if (conc[i][j] > max) {
            max = (int) conc[i][j];
            maxX = j;
            maxY = i;
        }
    }

    hasResults = true;
    log.debug("Am terminat calculul: " + min + "/" + max);
    maxValue = max;
}

private double[][] adjustValues(double[][] c, Point oPosition) {
    if (oPosition == null)
        return c;
    double offset = Util.getValue(c, oPosition.x, oPosition.y);

    for (int i = 0; i < IMAGE_SIZE; i++) {
        for (int j = 0; j < IMAGE_SIZE; j++) {
            c[i][j] = c[i][j] - offset;
        }
    }
    return c;
}

public void setFileData(String id, File file) {
    if (file == null)
        return;
    try {
        data.put(id, Util.readFile(file.getAbsolutePath()));
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public double[][] getResult() {
    if (hasResults) {
        return applyRatio(conc);
    }
    return null;
}

private double[][] applyRatio(double[][] conc2) {
    if (data2Return != null)
        return data2Return;

    int adjustment = 0;
    double[][] temp = new double[512][512];
    if (IMAGE_SIZE == 512) {
        for (int i = 0; i < IMAGE_SIZE - 1; i = i + 1) {
            for (int j = 0; j < IMAGE_SIZE - 1; j = j + 1) {
                temp[j][i] = conc2[i][j] + adjustment;
            }
        }
    } else {
```

170 Anexa 1

```
        for (int i = 0, posI = 0; i < IMAGE_SIZE - 1 && posI <
512; i = i + 2, posI++) {
            for (int j = 0, posJ = 0; j < IMAGE_SIZE - 1 &&
posJ < 512; j = j + 2, posJ++) {
                temp[posJ][posI] = (conc2[i][j] +
conc2[i][j + 1]
+ conc2[i + 1][j] + conc2[i
+ 1][j + 1]) / 4;
            }
        }
        data2Return = adjustValues(temp, Util.getOPosition());

        for (int i = 0; i < IMAGE_SIZE - 1; i = i + 1) {
            for (int j = 0; j < IMAGE_SIZE - 1; j = j + 1) {
                temp[j][i] = Math.abs(temp[j][i]);
            }
        }

        return data2Return;
    }

    public String getFileData(String id) {
        return fileNames.get(id);
    }

    public int getMaxValue() {
        return maxValue;
    }

    public double getPointValue(int x, int y) {
        return conc[y][x];
    }

    public int getMaxX() {
        return maxX;
    }

    public int getMaxY() {
        return maxY;
    }
}
```

com.cata.uvcamera.ui.AbstractHistogramPanel.java

```
package com.cata.uvcamera.ui;

public abstract class AbstractHistogramPanel extends JPanel {
    protected HistogramDisplayMode displayMode;
    protected int max;
    protected int min;
    protected int displayValues = 20;
    protected Map<Integer, Integer> values = null;
    protected int X_SIZE = 300;
    protected int X_OFFSET = 20;
    protected int Y_OFFSET = 5;
    protected int Y_SIZE = 100;
    protected int displayInterval;
    protected double Yratio = 0;
    protected boolean shouldCompute = true;
}
```

```

public enum HistogramDisplayMode {
    GRAPHICAL_REPRESENTATION, SIMPLE_REPRESENTATION
}

public AbstractHistogramPanel(HistogramDisplayMode displayMode) {
    super();
    this.displayMode = displayMode;
}

protected Map<Integer, Integer> computeDistribution(int min, int max,
    int displayInterval) {

    // now compute distribution
    int minX = Util.getFirstCornerUV().x;
    if (minX > Util.getSecondCornerUV().x) {
        minX = Util.getSecondCornerUV().x;
    }

    int maxX = Util.getFirstCornerUV().x;
    if (maxX < Util.getSecondCornerUV().x) {
        maxX = Util.getSecondCornerUV().x;
    }

    int minY = Util.getFirstCornerUV().y;
    if (minY > Util.getSecondCornerUV().y) {
        minY = Util.getSecondCornerUV().y;
    }

    int maxY = Util.getFirstCornerUV().y;
    if (maxY < Util.getSecondCornerUV().y) {
        maxY = Util.getSecondCornerUV().y;
    }

    java.util.Map<Integer, Integer> barValues = new
java.util.HashMap<Integer, Integer>();

    for (int i = minX; i < maxX; i++) {
        for (int j = minY; j < maxY; j++) {
            if (Util.shouldEliminate(i, j)) continue;
            double value =
Util.getProcessor().getResult()[i][j];
            int interval = min;

            while (value > interval) {
                interval = interval + displayInterval;
            }
            Integer freq = barValues.get(interval);
            if (freq == null) {
                freq = 0;
            }
            barValues.put(interval, freq + 1);
        }
    }
    return barValues;
}

public void setYRatio(int value) {
    Yratio = value;
    repaint();
}

```

```
        public void setInterval(int value) {
            displayValues = value;
            shouldCompute = true;
            repaint();
        }
    }
```

com.cata.uvcamera.ui.CalculVitezaVantDlg.java

```
package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class CalculVitezaVantDlg extends JDialog {

    private static final long serialVersionUID = 1L;
    private ImagePanel jPanelImg1;
    private ImagePanel jPanelImg2;
    private JButton jButtonCancel;
    private JLabel jLabel0;
    private JTextField jTextFieldInterval;
    private JButton jButtonOK;
    private JLabel jLabel1;
    private JLabel jLabel2;
    private JLabel jLabel3;
    private JTextField jTextFieldDimensiunePixel;
    private JTextField jTextFieldDistantaColoane;
    private JTextField jTextFieldVitezaVant;
    private double dimensiunePixel = 0;
    private String distantaColoane = "";
    private double vitezaVant = 0;
    private Point p1 = null;
    private Point p2 = null;
    private int intervalPoze = 15;
    private JPanel jPanelValori;
    private String secondFileName = null;
    private String firstFileName = null;

    private static final String FIRST_FILE = "/m_310.fit";
    private static final String SECOND_FILE = "/m_330.fit";

    public CalculVitezaVantDlg(Frame parent) {
        super(parent);
        initComponents();
    }

    public CalculVitezaVantDlg(Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    public CalculVitezaVantDlg(Frame parent, String title) {
        super(parent, title);
        initComponents();
    }

    public CalculVitezaVantDlg(Frame parent, String title, boolean modal) {
        super(parent, title, modal);
        initComponents();
    }
}
```

```
public CalculVitezaVantDlg(Frame parent, String title, boolean modal,
    GraphicsConfiguration arg) {
    super(parent, title, modal, arg);
    initComponents();
}

public CalculVitezaVantDlg(Dialog parent) {
    super(parent);
    initComponents();
}

public CalculVitezaVantDlg(Dialog parent, boolean modal) {
    super(parent, modal);
    initComponents();
}

public CalculVitezaVantDlg(Dialog parent, String title) {
    super(parent, title);
    initComponents();
}

public CalculVitezaVantDlg(Dialog parent, String title, boolean modal) {
    super(parent, title, modal);
    initComponents();
}

public CalculVitezaVantDlg(Dialog parent, String title, boolean modal,
    GraphicsConfiguration arg) {
    super(parent, title, modal, arg);
    initComponents();
}

public CalculVitezaVantDlg(Window parent) {
    super(parent);
    initComponents();
}

public CalculVitezaVantDlg(Window parent, ModalityType modalityType) {
    super(parent, modalityType);
    initComponents();
}

public CalculVitezaVantDlg(Window parent, String title) {
    super(parent, title);
    initComponents();
}

public CalculVitezaVantDlg(Window parent, String title,
    ModalityType modalityType) {
    super(parent, title, modalityType);
    initComponents();
}

public CalculVitezaVantDlg(Window parent, String title,
    ModalityType modalityType, GraphicsConfiguration arg) {
    super(parent, title, modalityType, arg);
    initComponents();
}

private void initComponents() {
    initFiles();
}
```

```

        setTitle("Calcul viteza vant");
        setFont(new Font("Dialog", Font.PLAIN, 12));
        setBackground(new Color(223, 223, 223));
        setResizable(false);
        setModal(true);
        setForeground(Color.black);
        setLayout(new GroupLayout());
        add(getJPanelImgl(), new Constraints(new Leading(10, 582, 12,
12),
                new Leading(10, 552, 10, 10)));
        add(getJPanelImg2(), new Constraints(new Leading(610, 584, 10,
10),
                new Leading(11, 550, 12, 12)));
        add(getJButtonAnuleaza(), new Constraints(new Leading(631, 10,
10),
                new Trailing(12, 23, 573)));
        add(getJButtonOK(), new Constraints(new Leading(472, 10, 10),
                new Trailing(12, 22, 574)));
        add(getJPanel0(), new Constraints(new Leading(8, 1186, 12, 12),
                new Leading(567, 90, 10, 10)));
        setSize(1218, 730);
    }

    private void initFiles() {
        if (firstFileName == null || secondFileName == null){
            firstFileName = Util.getFolder() + FIRST_FILE;
            secondFileName = Util.getFolder() + SECOND_FILE;

            try {
                String firstDate =
Util.readDateObservation(firstFileName, "DATE-OBS");
                String secondDate =
Util.readDateObservation(secondFileName, "DATE-OBS");

                if (firstDate == null || secondDate == null)
return;

                if (firstDate.compareTo(secondDate) > 0){
                    firstFileName = Util.getFolder() +
SECOND_FILE;
                    secondFileName = Util.getFolder() +
FIRST_FILE;
                }
            } catch (FileNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }

    private JPanel getJPanel0() {
        if (jPanelValori == null) {
            jPanelValori = new JPanel();

            jPanelValori.setBorder(BorderFactory.createTitledBorder("Valori"));
            jPanelValori.setLayout(new GroupLayout());
            jPanelValori.add(getJLabel0(), new Constraints(new
Leading(12, 12,
                12), new Leading(2, 10, 10)));

```

```

jPanelValori.add(getJLabel1(), new Constraints(new
Leading(12, 12,
12), new Leading(31, 10, 10)));
jPanelValori.add(getJLabel2(), new Constraints(new
Leading(615, 10,
10), new Leading(4, 10, 10)));
jPanelValori.add(getJLabel3(), new Constraints(new
Leading(616, 12,
12), new Leading(31, 10, 10)));
jPanelValori.add(getJTextFieldInterval(), new
Constraints(
new Leading(244, 100, 10, 10), new
Leading(0, 12, 12)));
jPanelValori.add(getJTextFieldDimensiunePixel(), new
Constraints(
new Leading(245, 100, 12, 12), new
Leading(35, 10, 10)));
jPanelValori.add(getJTextFieldDistantaColoane(), new
Constraints(
new Leading(862, 100, 10, 10), new
Leading(2, 10, 10)));
jPanelValori.add(getJTextFieldVitezaVant(), new
Constraints(
new Leading(863, 100, 12, 12), new
Leading(33, 10, 10)));
}
return jPanelValori;
}

private JTextField getJTextFieldVitezaVant() {
if (jTextFieldVitezaVant == null) {
jTextFieldVitezaVant = new JTextField();
jTextFieldVitezaVant.setText("" + vitezaVant);
jTextFieldVitezaVant.setEditable(false);
}
return jTextFieldVitezaVant;
}

private JTextField getJTextFieldDistantaColoane() {
if (jTextFieldDistantaColoane == null) {
jTextFieldDistantaColoane = new JTextField();
jTextFieldDistantaColoane.setText(distantaColoane);
jTextFieldDistantaColoane.setEditable(false);
}
return jTextFieldDistantaColoane;
}

private JTextField getJTextFieldDimensiunePixel() {
if (jTextFieldDimensiunePixel == null) {
jTextFieldDimensiunePixel = new JTextField();
jTextFieldDimensiunePixel.setText("" + dimensiunePixel);
jTextFieldDimensiunePixel.setEditable(false);
}
return jTextFieldDimensiunePixel;
}

private JLabel getJLabel3() {
if (jLabel3 == null) {
jLabel3 = new JLabel();
jLabel3.setText("Viteza vant (m/s):");
}
}

```

```
        return jLabel3;
    }

    private JLabel getJLabel2() {
        if (jLabel2 == null) {
            jLabel2 = new JLabel();
            jLabel2.setText("Distanta coloane fum selectate (cm):");
        }
        return jLabel2;
    }

    private JLabel getJLabel1() {
        if (jLabel1 == null) {
            jLabel1 = new JLabel();
            jLabel1.setText("Dimensiune pixel (cm):");
        }
        return jLabel1;
    }

    private JButton getJButtonOK() {
        if (jButtonOK == null) {
            jButtonOK = new JButton();
            jButtonOK.setText("Preia valoarea");
            jButtonOK.addActionListener(new ActionListener() {

                public void actionPerformed(ActionEvent event) {
                    jButtonOKActionActionPerformed(event);
                }

            });
        }
        return jButtonOK;
    }

    private JTextField getJTextFieldInterval() {
        if (jTextFieldInterval == null) {
            jTextFieldInterval = new JTextField();
            jTextFieldInterval.setText("" + intervalPoze);
        }
        return jTextFieldInterval;
    }

    private JLabel getJLabel0() {
        if (jLabel0 == null) {
            jLabel0 = new JLabel();
            jLabel0.setText("Interval de timp intre fotografii
(s):");
        }
        return jLabel0;
    }

    private JButton getJButtonAnuleaza() {
        if (jButtonCancel == null) {
            jButtonCancel = new JButton();
            jButtonCancel.setText("Anuleaza");
            jButtonCancel.addActionListener(new ActionListener() {

                public void actionPerformed(ActionEvent event) {
                    jButtonCancelActionActionPerformed(event);
                }

            });
        }
        return jButtonCancel;
    }
}
```



```

    }

    private JPanel getJPanelImg2() {
        if (jPanelImg2 == null) {
            jPanelImg2 = new ImagePanel(this, secondFileName);

        jPanelImg2.setBorder(BorderFactory.createTitledBorder("Poza 2"));
            jPanelImg2.setLayout(new GroupLayout());
        }
        return jPanelImg2;
    }

    private JPanel getJPanelImg1() {
        if (jPanelImg1 == null) {
            jPanelImg1 = new ImagePanel(this, firstFileName);

        jPanelImg1.setBorder(BorderFactory.createTitledBorder("Poza 1"));
            jPanelImg1.setLayout(new GroupLayout());
        }
        return jPanelImg1;
    }

    public CalculVitezaVantDlg() {
        initComponents();
    }

    private void jButtonOKActionActionPerformed(ActionEvent event) {
        Util.setVitezaVant(vitezaVant);
        this.hide();
    }

    private void jButtonCancelActionActionPerformed(ActionEvent event) {
        this.hide();
    }

    public void setDimensiunePixel(double dimensiunePixelValue) {
        this.dimensiunePixel = dimensiunePixelValue;
        jTextFieldDimensiunePixel.setText("
Util.round(dimensiunePixel));
    }

    public void calculeazaVitezaVant() {
        if (jPanelImg1.getSelection() == null)
            return;
        if (jPanelImg2.getSelection() == null)
            return;

        if (jPanelImg1.getSelection().equals(p1)
            && jPanelImg2.getSelection().equals(p2))
            return;
        try {
            intervalPoze
Integer.parseInt(jTextFieldInterval.getText());
        } catch (Exception e) {
            return;
        }

        p1 = jPanelImg1.getSelection();
        p2 = jPanelImg2.getSelection();

        double distanta = Math.sqrt((p1.y - p2.y) * (p1.y - p2.y)

```

```
        + (p1.x - p2.x) * (p1.x - p2.x)
        * dimensiunePixel;
jTextFieldDistantaColoane.setText("" + Util.round(distanta));

vitezaVant = distanta / 100 / intervalPoze;
jTextFieldVitezaVant.setText("" + Util.round(vitezaVant));
}

public double getVitezaVant() {
    return vitezaVant;
}
}
```

com.cata.uvcamera.ui.EnterValueDlg.java

```
package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class EnterValueDlg extends JDialog {

    private static final long serialVersionUID = 1L;
    private JLabel jLabel0;
    private JTextField jTextFieldValue;
    private JButton jButton0;
    public JTextField getjTextFieldValue() {
        return jTextFieldValue;
    }

    public EnterValueDlg() {
        initComponents();
    }

    public EnterValueDlg(Frame parent) {
        super(parent);
        initComponents();
    }

    public EnterValueDlg(Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    public EnterValueDlg(Frame parent, String title) {
        super(parent, title);
        initComponents();
    }

    public EnterValueDlg(Frame parent, String title, boolean modal) {
        super(parent, title, modal);
        initComponents();
    }

    public EnterValueDlg(Frame parent, String title, boolean modal,
        GraphicsConfiguration arg) {
        super(parent, title, modal, arg);
        initComponents();
    }

    public EnterValueDlg(Dialog parent) {
        super(parent);
        initComponents();
    }
}
```

```
}

public EnterValueDlg(Dialog parent, boolean modal) {
    super(parent, modal);
    initComponents();
}

public EnterValueDlg(Dialog parent, String title) {
    super(parent, title);
    initComponents();
}

public EnterValueDlg(Dialog parent, String title, boolean modal) {
    super(parent, title, modal);
    initComponents();
}

public EnterValueDlg(Dialog parent, String title, boolean modal,
    GraphicsConfiguration arg) {
    super(parent, title, modal, arg);
    initComponents();
}

public EnterValueDlg(Window parent) {
    super(parent);
    initComponents();
}

public EnterValueDlg(Window parent, ModalityType modalityType) {
    super(parent, modalityType);
    initComponents();
}

public EnterValueDlg(Window parent, String title) {
    super(parent, title);
    initComponents();
}

public EnterValueDlg(Window parent, String title, ModalityType
modalityType) {
    super(parent, title, modalityType);
    initComponents();
}

public EnterValueDlg(Window parent, String title,
    ModalityType modalityType, GraphicsConfiguration arg) {
    super(parent, title, modalityType, arg);
    initComponents();
}

private void initComponents() {
    setTitle("Enter etalon value");
    setFont(new Font("Dialog", Font.PLAIN, 12));
    setBackground(new Color(223, 223, 223));
    setResizable(false);
    setModal(true);
    setForeground(Color.black);
    setLayout(new GroupLayout());
    add(getJLabel0(), new Constraints(new Leading(14, 10, 10), new
Leading(14, 10, 10)));
}
```

180 Anexa 1

```
        add(getJTextField(), new Constraints(new Leading(230, 58, 10,
10), new Leading(12, 12, 12)));
        add(getJButtonOK(), new Constraints(new Leading(121, 10, 10), new
Leading(37, 12, 12)));
        setSize(301, 103);
    }

    private JButton getJButtonOK() {
        if (jButton0 == null) {
            jButton0 = new JButton();
            jButton0.setText("OK");
            jButton0.addActionListener(new ActionListener() {

                public void actionPerformed(ActionEvent event) {
                    jButton0ActionActionPerformed(event);
                }
            });
        }
        return jButton0;
    }

    private JTextField getJTextField() {
        if (jTextFieldValue == null) {
            jTextFieldValue = new JTextField();
            jTextFieldValue.setText("");
        }
        return jTextFieldValue;
    }

    public JLabel getJLabel0() {
        if (jLabel0 == null) {
            jLabel0 = new JLabel();
        }
        return jLabel0;
    }

    private void jButton0ActionActionPerformed(ActionEvent event) {
        this.hide();
    }
}
```

com.cata.uvcamera.ui.HGraphicPanel.java

```
package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class HGraphicPanel extends JPanel {
    private static final int _Y = 90;
    private static final int _X = 512;

    private static final int IMG_X = 512;
    private static final int IMG_Y = 512;

    private double[][] data;

    public HGraphicPanel() {
        super();
    }

    public void paint(Graphics g) {
```

```
super.paint(g);

Graphics2D g2 = (Graphics2D) g;

g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);

if (Util.getProcessor() == null)
    return;

if (Util.getProcessor().getResult() != null) {
    data = Util.getProcessor().getResult();
}

if (data == null) {
    return;
}

double ratio = computeRatio();

int vPosition = Util.getVPosition();
int hPosition = Util.getHPosition();
if (vPosition <= 0)
    return;

if (Util.isGraficIntersectie()) {
    g2.setPaint(Color.gray);
    showGraphic(g2, ratio, vPosition, hPosition, true);
}

if (Util.getMaxPositionUV() != null && Util.isGraficMaxim()) {
    vPosition = (int) Util.getMaxPositionUV().getX();
    hPosition = (int) Util.getMaxPositionUV().getY();
    if (vPosition <= 0)
        return;
    g2.setPaint(Color.red);
    showGraphic(g2, ratio, vPosition, hPosition, false);
}

if (Util.getMinPositionUV() != null && Util.isGraficMinim()) {
    vPosition = (int) Util.getMinPositionUV().getX();
    hPosition = (int) Util.getMinPositionUV().getY();
    if (vPosition <= 0)
        return;
    g2.setPaint(Color.green);
    showGraphic(g2, ratio, vPosition, hPosition, false);
}
}

private double computeRatio() {
    double max = 0;
    for (int i = 0; i < IMG_X; i++) {
        for (int j = 0; j < IMG_Y; j++) {
            if (max < data[i][j]) {
                max = data[i][j];
            }
        }
    }

    double ratio = _Y / max;
    return ratio;
}
```

```
    }

    private void showGraphic(Graphics2D g2, double ratio, int vPosition,
        int hPosition, boolean isValueShifted) {
        int startX = 0;
        int startY = 0;
        int offsetX = 40;
        int endX = 0;
        int endY = 0;

        int mean = 0;

        int position = hPosition;

        if (position <= 0)
            return;

        int min = Integer.MAX_VALUE;
        for (int j = 0; j < _X; j++) {
            int x = (int) (data[j][position] * ratio);

            if (x < min) {
                min = x;
            }
        }

        int offsetY = 30; // min - 20;
        int maxValue = 0;
        int minValue = Integer.MAX_VALUE;

        for (int j = 0; j < _X; j++) {
            int x = (int) data[j][position];

            if (maxValue < x)
                maxValue = x;
            if (minValue > x)
                minValue = x;

            mean += x;

            x = (int) (x * ratio);
            if (x < 0)
                x = 0;
            if (x > _X)
                x = _X;

            startX = endX;
            startY = endY;
            endX = j;
            endY = x;

            if (startY != 0) {
                g2.draw(new Line2D.Double(startX + offsetX, _Y
                    - (startY - offsetY), endX +
offsetX, _Y
                    - (endY - offsetY)));
            }
        }

        int y = _Y - (int) (maxValue * ratio) + offsetY;
        g2.drawLine(offsetX, y, _X, y);
        // g2.setPaint(Color.black);
    }
}
```

```

        int shift = 0;
        if (isValueShifted) {
            shift = 520;
        }
        g2.drawString("" + (int) (maxValue/* * Util.getCoefficient() */),
            5 + shift, y + 5);

        int x = vPosition + offsetX;
        g2.drawLine(x, 15, x, _Y - 5 + offsetY);

        if (Util.getOValue() > 0) {
            g2.setPaint(Color.black);
            y = (int) (Util.getOValue() * ratio);
            y = _Y - y + offsetY;
            g2.drawLine(offsetX, y, _X + offsetX, y);
            g2.drawLine(offsetX, y - 1, _X + offsetX, y - 1);

            g2.drawString("0", offsetX / 2, y + 5);
        }
    }
}

```

com.cata.uvcamera.ui.ImagePanel.java

```

package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class ImagePanel extends JPanel {
    private static final int _X = 512;
    private static final int _Y = 512;
    private static final int IMG_X = 512;
    private static final int IMG_Y = 512;
    private static final int OFFSET_Y = 25;
    private static final int OFFSET_X = 35;
    private CalculVitezaVantDlg parent = null;

    BufferedImage image;
    boolean isPrepared = false;
    private String fileName;
    private double[][] data = null;
    private Point selection = null;

    public ImagePanel(CalculVitezaVantDlg parent, String fileName) {
        super();
        this.parent = parent;
        this.fileName = fileName;
        isPrepared = false;

        this.addMouseListener(new MouseListener() {
            @Override
            public void mouseReleased(MouseEvent e) {
                selection = new Point(e.getX() - OFFSET_X,
e.getY() - OFFSET_Y);
                isPrepared = false;
                repaint();
            }

            @Override
            public void mousePressed(MouseEvent e) {

```

```

        }

        @Override
        public void mouseExited(MouseEvent e) {
        }

        @Override
        public void mouseEntered(MouseEvent e) {
        }

        @Override
        public void mouseClicked(MouseEvent e) {
        }
    });
}

public void init() throws FileNotFoundException, IOException {
    isPrepared = false;

    if (data == null) {
        data = Util.readFile(fileName);
    }

    if (data == null) {
        return;
    }

    setBorder(BorderFactory.createTitledBorder(" " +
Util.getLocation() + " : " + Util.readDateObservation(fileName, "DATE-OBS") + "
"));

    image = new BufferedImage(_X, _Y, BufferedImage.TYPE_INT_RGB);

    double max = 0;

    for (int i = 0; i < IMG_X; i++) {
        for (int j = 0; j < IMG_Y; j++) {
            if (max < data[j][i]) {
                max = data[j][i];
            }
        }
    }

    double ratio = 255 / max;

    for (int i = 0, posX = 0; i < IMG_X; i = i + 1, posX++) {
        for (int j = 0, posY = 0; j < IMG_Y; j = j + 1,
posY++) {
            int color = (int) data[i][j];
            if (color < 0) {
                image.setRGB(posY, posX, (new Color(0, 0,
0)).getRGB());
                continue;
            }
            color = (int) (color * ratio);
            image.setRGB(posY, posX, getPixelColor(i, j,
color));
        }
    }

    int size = 2;

```



```

        if (selection != null) {
            this.parent.calculeazaVitezaVant();

            for (int i = selection.x - size; i < selection.x + size;
i++) {
                for (int j = selection.y - size; j < selection.y +
size; j++) {
                    try {
                        image.setRGB(i, j,
Color.RED.getRGB());
                    } catch (ArrayIndexOutOfBoundsException
ignored) {
                    }
                }
            }
            isPrepared = true;
        }

private int getPixelColor(int i, int j, int color) {
    return (new Color(color, color, color)).getRGB();
}

public void paint(Graphics g) {
    super.paint(g);

    if (!isPrepared)
        try {
            init();
        } catch (FileNotFoundException e) {
            isPrepared = true;
        } catch (IOException e) {
            isPrepared = true;
        }

    if (data == null) {
        g.drawString("Nu se poate afisa poza. ", 20, 200);
        g.drawString(fileName, 20, 220);
        return;
    }

    g.drawImage(image, OFFSET_X, OFFSET_Y, _X, _Y, null);

    public Point getSelection() {
        return selection;
    }
}

```

com.cata.uvcamera.ui.InformationPanel.java

```

package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class InformationPanel extends JPanel {

    public InformationPanel() {
        super();
    }
}

```

```

    public void paint(Graphics g) {
        super.paint(g);

        g.drawString(
            "Masuratoare pct. etalon (ppm): "
            + (Util.getMasuratoarePctEtalon()
            == 0?"-":Util.getMasuratoarePctEtalon()), 10, 30);
        g.drawString("Masuratoare standard (ppm): " +
            (Util.getEtalonValue()==0?"-":Util.getEtalonValue()),
            10, 47);
        g.drawString(
            "Pct. de intersectie (ppm): " +
            Util.getIntersectionValue(),
            10, 64);
        g.drawString("Dimensiune pixel (cm): " +
            Util.getDimensiunePixel(), 10,
            81);
        g.drawString("Flux total (ppmm): " + Util.getFlux(), 10, 98);
        g.drawString("Viteza vant (m/s): " + Util.getVitezaVant(), 10,
            115);
        g.drawString("Flux total (kg/s): " + Util.getFluxViteza(), 10,
            132);
    }
}

```

com.cata.uvcamera.ui.IRHistogramPanel.java

```

package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class IRHistogramPanel extends AbstractHistogramPanel {
    private static final int OFFSET_IMG_X = 10;
    private static final int OFFSET_IMG_Y = 15;

    public IRHistogramPanel(HistogramDisplayMode displayMode) {
        super(displayMode);
    }

    public void paint(Graphics g) {
        super.paint(g);
        Graphics2D g2 = (Graphics2D) g;

        Y_OFFSET = 0;

        if (Util.getFirstCornerIR() == null || Util.getSecondCornerIR()
            == null) {
            g.drawString("Histograma nu poate fi calculata.", 50,
                50);
            g.drawString("Selectati mai intai zona de masurat. ", 50,
                70);
            return;
        }

        shouldCompute = true;
        int maxValue = 0;
        if (shouldCompute) {
            max = (int) Util.getMaxValueIR();
            min = (int) Util.getMinValueIR();
        }
    }
}

```

```

        if (min > max) {
            int t = min;
            min = max;
            max = t;
        }

        int noOfIntervals = Util.getIntervalHistogramaIR();
        if (noOfIntervals == 0) noOfIntervals = 1;
        displayValues = (max - min) / noOfIntervals;

        if (displayValues == 0) {
            g.drawString("Histograma nu poate fi calculata.",
50, 50);

                return;
            }

        if (noOfIntervals == 0) {
            return;
        }
        displayInterval = X_SIZE / noOfIntervals;

        values = computeDistribution(min, max, displayValues);
        removePeeks(values);

        // now compute y ratio
        // find largests value
        noOfIntervals = 0;
        Iterator<Integer> it = values.values().iterator();
        while (it.hasNext()) {
            Integer i = it.next();
            noOfIntervals++;
            if (i > maxValue) {
                maxValue = i;
            }
        }

        // adjust no of intervals
        displayInterval = X_SIZE / noOfIntervals;

        Yratio = (double)maxValue / (double)(Y_SIZE - 50);
        if (Yratio == 0)
            Yratio = 0.1;
        shouldCompute = true;
    }

    int lastY = 0;
    g2.setPaint(Color.red);

    boolean displayConcentration = false;
    int lastBarValue = 0;
    for (int i = min, j = 0; i < max; i = i + displayValues, j++) {
        Integer currentValue = values.get(i);
        if (currentValue == null)
            continue;
        lastBarValue = i;
    }

    for (int i = min, j = -1; i < max; i = i + displayValues) {
        Integer currentValue = values.get(i);
        if (currentValue == null)
            continue;
    }

```

```

        j++;
        if (displayMode ==
HistogramDisplayMode.SIMPLE_REPRESENTATION) {
            g2.setPaint(Color.black);
            g2.drawRect(X_OFFSET + displayInterval * j, Y_SIZE
- Y_OFFSET
displayInterval,
                    - (int) ((currentValue) / Yratio),
                    (int) (currentValue / Yratio));
            g2.setPaint(Color.GRAY);
            g2.fillRect(X_OFFSET + displayInterval * j, Y_SIZE
- Y_OFFSET
displayInterval,
                    - (int) ((currentValue) / Yratio),
                    (int) (currentValue / Yratio));
        } else {
            if (lastY != 0) {
                g2.drawLine(X_OFFSET + displayInterval *
j, lastY, X_OFFSET
+ displayInterval * j,
Y_SIZE - 0
- (int) (currentValue /
Yratio));
            }
            lastY = Y_SIZE - 10 - (int) (currentValue /
Yratio);
            g2.drawLine(X_OFFSET + displayInterval * j, Y_SIZE
- 10
- (int) ((currentValue) / Yratio),
X_OFFSET
+ displayInterval * (j + 1), Y_SIZE
- 0
- (int) (currentValue / Yratio));
        }

        if (currentValue == maxValue) {
            displayConcentration = true;
        }

        if (displayConcentration) {
            displayConcentration = false;

            g2.setPaint(Color.gray);

            if (X_OFFSET + displayInterval * j < X_SIZE - 30)
            {
                g2.drawLine(X_OFFSET + displayInterval *
j, Y_SIZE
- Y_OFFSET - (int)
((currentValue) / Yratio),
X_SIZE - 30, Y_SIZE -
Y_OFFSET - (int)
((currentValue) / Yratio));
            }
            g2.drawString("" + currentValue, X_SIZE + 30,
Y_SIZE - Y_OFFSET
- (int) ((currentValue) / Yratio));
            g2.setPaint(Color.black);
            g2.drawLine(X_OFFSET + displayInterval * j, Y_SIZE
- Y_OFFSET,

```

```

X_OFFSET + displayInterval * j,
Y_SIZE - 5);
    }
    if (currentValue == min) {
        g2.setPaint(Color.gray);
        g2.drawString("" + min, X_OFFSET + displayInterval
* j, Y_SIZE
        + Y_OFFSET);
    }
}
g2.setPaint(Color.black);
g2.drawString("Conc.", X_SIZE + 30, Y_SIZE + Y_OFFSET + 10);

g2.translate(0, 100);
g2.rotate(-Math.PI / 2.0);
g2.drawString("Frecv.", 45, 15);
}

private void removePeeks(Map<Integer, Integer> values) {
    int mean = 0;
    int sum = 0;
    int noOfValues = 0;

    for (int i = min, j = 0; i < max; i = i + displayValues, j++) {
        Integer currentValue = values.get(i);
        if (currentValue == null)
            continue;
        sum += currentValue;
        noOfValues++;
    }

    for (int i = min, j = 0; i < max; i = i + displayValues, j++) {
        Integer currentValue = values.get(i);
        if (currentValue == null)
            continue;
        mean = (sum - currentValue) / noOfValues;

        if (currentValue > mean * 5) {
            values.remove(i);
            sum = sum - currentValue;
            noOfValues--;
        }
    }
}

@Override
protected Map<Integer, Integer> computeDistribution(int min, int max,
int displayInterval) {

    // now compute distribution
    int minX = Util.getFirstCornerIR().x;
    if (minX > Util.getSecondCornerIR().x) {
        minX = Util.getSecondCornerIR().x;
    }

    int maxX = Util.getFirstCornerIR().x;
    if (maxX < Util.getSecondCornerIR().x) {
        maxX = Util.getSecondCornerIR().x;
    }
}

```

```

        int minY = Util.getFirstCornerIR().y;
        if (minY > Util.getSecondCornerIR().y) {
            minY = Util.getSecondCornerIR().y;
        }

        int maxY = Util.getFirstCornerIR().y;
        if (maxY < Util.getSecondCornerIR().y) {
            maxY = Util.getSecondCornerIR().y;
        }

        java.util.Map<Integer, Integer> barValues = new
java.util.HashMap<Integer, Integer>();

        int[][] data = Util.getIRDataMatrix();

        for (int i = minX; i < maxX; i++) {
            for (int j = minY; j < maxY; j++) {
                // if (Util.shouldEliminate(i, j)) continue;
                int value = data[i][j];
                int interval = min;

                while (value > interval) {
                    // value = value - displayInterval;
                    interval = interval + displayInterval;
                }
                Integer freq = barValues.get(interval);
                if (freq == null) {
                    freq = 0;
                }
                barValues.put(interval, freq + 1);
            }
        }
        return barValues;
    }
}

```

com.cata.uvcamera.ui.IRImagePanel.java

```

package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class IRImagePanel extends JPanel {
    private static int _X = 512;
    private static int _Y = 512;
    private static final int OFFSET_IMG_X = 35;
    private static final int OFFSET_IMG_Y = 15;

    BufferedImage image;
    boolean isPrepared = false;
    private int[][] data;

    public IRImagePanel() {
        super();
    }

    public void init() {
        isPrepared = false;

        int factor = 1;
        if (Util.IR_X_SIZE > 500) {
            factor = 2;
        }
    }
}

```

```

    }

    data = null;
    if (Util.irData == null) {
        return;
    }

    _X = Util.IR_X_SIZE / factor;
    _Y = Util.IR_Y_SIZE / factor;
    data = new int[Util.IR_X_SIZE][Util.IR_Y_SIZE];

    int pixelCount = 0;
    for (int j = 0; j < Util.IR_Y_SIZE; j++) {
        for (int i = 0; i < Util.IR_X_SIZE; i++, pixelCount++) {
            data[i][j] = Util.irData[pixelCount];
        }
    }

    image = new BufferedImage(_X, _Y, BufferedImage.TYPE_INT_RGB);

    for (int i = 0; i < Util.IR_X_SIZE; i = i + factor) {
        for (int j = 0; j < Util.IR_Y_SIZE; j = j + factor) {
            int color = (data[i][j]); // + data[i + 1][j] +
data[i][j + 1] +
            // data[i + 1][j + 1]) / 4;
            image.setRGB(i / factor, j / factor,
getPixelColor((i + 0) / factor, (j + 0) / factor, color));
        }
        image.setRGB(25,45,0);
    }

    private int getPixelColor(int i, int j, int color) {
        int minX = 0;
        int maxX = 0;
        int minY = 0;
        int maxY = 0;

        if (Util.getFirstCornerIR() != null && Util.getSecondCornerIR()
!= null) {
            minX = Util.getFirstCornerIR().x;
            if (minX > Util.getSecondCornerIR().x) {
                minX = Util.getSecondCornerIR().x;
            }

            maxX = Util.getFirstCornerIR().x;
            if (maxX < Util.getSecondCornerIR().x) {
                maxX = Util.getSecondCornerIR().x;
            }

            minY = Util.getFirstCornerIR().y;
            if (minY > Util.getSecondCornerIR().y) {
                minY = Util.getSecondCornerIR().y;
            }

            maxY = Util.getFirstCornerIR().y;
            if (maxY < Util.getSecondCornerIR().y) {
                maxY = Util.getSecondCornerIR().y;
            }
        }
    }

```

192 Anexa 1

```
        if ((i == Util.getFirstCornerIR().x || i ==
Util.getSecondCornerIR().x)
            && (j >= minY && j <= maxY)) {
            return Color.black.getRGB();
        }
        if ((i >= minX && i <= maxX)
            && (j == Util.getFirstCornerIR().y || j ==
Util.getSecondCornerIR().y)) {
            return Color.black.getRGB();
        }
    }
    if (Util.getFirstCornerIR() == null || Util.getSecondCornerIR()
== null) {
        return color;
    }
    if ((i == Util.getFirstCornerIR().x || i ==
Util.getSecondCornerIR().x)
        && (j == Util.getFirstCornerIR().y || j ==
Util.getSecondCornerIR().y)) {
        return Color.black.getRGB();
    }
    if (i > minX && i < maxX && j > minY && j < maxY) {
        color = color;
    } else {
        color = color + (255 - color) / 4;
    }
    return color;
}

public void paint(Graphics g) {
    super.paint(g);

    if (!isPrepared)
        init();

    if (data == null) {
        g.drawString("Nu este incarcat folderul de masuratori. ",
30, 80);
        g.drawString(
            "Utilizati meniul \"Fisier\" pentru a
selecta",
            30, 100);
        g.drawString(
            " folderul de masuratori.",
            30, 120);
        return;
    }

    g.drawImage(image, OFFSET_IMG_X, OFFSET_IMG_Y, _X, _Y, null);

    // X axis
    g.drawString("0", 25, _Y + OFFSET_IMG_Y + 15);
    g.drawString("128", 150, _Y + OFFSET_IMG_Y + 15);
    g.drawString("256", 285, _Y + OFFSET_IMG_Y + 15);
    // Y axis
    g.drawString("128", 10, _Y + OFFSET_IMG_Y - 125);
```



```

        g.drawString("256", 10, _Y + OFFSET_IMG_Y - 240);
    }
}

```

com.cata.uvcamera.ui.MainWindow.java

```

package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class MainWindow extends JFrame {

    private int MAX_X = 512;
    private int MAX_Y = 512;

    private static final String TITLE = "Camera UVIR V1.0";
    private static final long serialVersionUID = 1L;

    private JMenuBar jMenuBarOperation;

    private JMenu jMenuFile;
    private JMenuItem open;
    private JMenuItem jMenuItemExit;

    private JMenu jMenuOperationUV;
    private JMenuItem jMenuItemResetAreaUV;
    private JMenuItem jMenuItemCalculVitezaVant;
    private JCheckBoxMenuItem jCheckBoxMenuItemDynamic;
    private JCheckBoxMenuItem jCheckBoxMenuItem0Level;
    private JCheckBoxMenuItem jCheckBoxMenuItemEtalonLevel;
    private JCheckBoxMenuItem jMenuItemChooseAreaUV;
    private JCheckBoxMenuItem jMenuItemChooseAreaIR;
    private JCheckBoxMenuItem jMenuItemChoosePixelDimension;
    private JCheckBoxMenuItem jMenuItemChooseFluxSection;

    private JMenu jMenuItemOperationIR;
    private JMenuItem jMenuItemResetAreaIR;

    private JMenu jMenuItemHelp;
    private JMenuItem jMenuItemIAbout;

    private VGraphicPanel vGraphicPanel;
    private UVImagePanel uvImagePanel;
    private HGraphicPanel hGraphicPanel;
    private JPanel infoPanel;
    private IRImagePanel irImagePanel;
    private AbstractHistogramPanel jMenuItemUVHistogram;
    private AbstractHistogramPanel jMenuItemIRHistogram;

    private OperationMode operationMode = OperationMode.STATIC;

    private JPanel jMenuItemTools;
    private JLabel jMenuItemLabelUVTools;
    private JCheckBox jMenuItemCheckBoxClearSky;
    private JSlider jMenuItemSliderClearSky;
    private JLabel jMenuItemLabelGrafice;
    private JCheckBox jMenuItemCheckBoxGraficMaxim;
    private JCheckBox jMenuItemCheckBoxGraficMinim;
    private JCheckBox jMenuItemCheckBoxGraficIntersectie;
    private JLabel jMenuItemLabelThreshold;
    private JLabel jMenuItemLabelIntervalUV;

```

194 Anexa 1

```
private JSlider jSliderIntervalUV;
private JSlider jSliderIntervalIR;
private JLabel jLabelIntervalIR;
private static final String PREFERRED_LOOK_AND_FEEL =
"javax.swing.plaf.metal.MetalLookAndFeel";
public MainWindow() {
    initComponents();
}

private void initComponents() {
    setTitle("Camera UVIR V1.0");
    setResizable(false);
    setLayout(new GroupLayout());
    add(getIRImagePanel(), new Constraints(new Trailing(12, 371, 254,
816), new Leading(6, 313, 10, 10)));
    add(getImagePanel(), new Constraints(new Leading(246, 558, 24,
395), new Leading(7, 560, 12, 12)));
    add(getVGraphicPanel(), new Constraints(new Leading(10, 228, 24,
395), new Leading(10, 557, 18, 18)));
    add(getJPanel0(), new Constraints(new Trailing(12, 370, 254,
816), new Leading(325, 120, 10, 10)));
    add(getJPanel1(), new Constraints(new Leading(818, 372, 10, 10),
new Leading(445, 122, 12, 12)));
    add(getInformationPanel(), new Constraints(new Leading(8, 232,
12, 12), new Leading(569, 141, 10, 10)));
    add(getHGraphicPanel(), new Constraints(new Leading(244, 562, 12,
12), new Leading(569, 141, 12, 12)));
    add(getJPanelTools(), new Constraints(new Trailing(12, 370, 256,
818), new Leading(569, 141, 12, 12)));
    setJMenuBar(getJMenuBar0());
    setSize(1200, 726);
}

private JLabel getJLabelIntervalIR() {
    if (jLabelIntervalIR == null) {
        jLabelIntervalIR = new JLabel();
        jLabelIntervalIR.setText("Interval hist. IR");
    }
    return jLabelIntervalIR;
}

private JSlider getJSliderIntervalIR() {
    if (jSliderIntervalIR == null) {
        jSliderIntervalIR = new JSlider();

        Util.setIntervalHistogramaIR(jSliderIntervalIR.getValue());
        jSliderIntervalIR.addInputMethodListener(new
InputMethodListener() {
            @Override
            public void caretPositionChanged(InputMethodEvent
event) {

                Util.setIntervalHistogramaIR(jSliderIntervalIR.getValue());
                refreshAllIR();
            }

            @Override
            public void inputMethodTextChanged(InputMethodEvent event) {

                Util.setIntervalHistogramaIR(jSliderIntervalIR.getValue());
            }
        });
    }
}

private void refreshAllIR() {
    Util.setIntervalHistogramaIR(jSliderIntervalIR.getValue());
}
```

```

        refreshAllIR();
    }
    });

    jSliderIntervalIR.addChangeListener(new ChangeListener()
{
    @Override
    public void stateChanged(ChangeEvent event) {
        Util.setIntervalHistogramaIR(jSliderIntervalIR.getValue());
        refreshAllIR();
    }
});
}
return jSliderIntervalIR;
}

private JSlider getJSliderIntervalUV() {
    if (jSliderIntervalUV == null) {
        jSliderIntervalUV = new JSlider();

        Util.setIntervalHistogramaUV(jSliderIntervalUV.getValue());
        jSliderIntervalUV.addInputMethodListener(new
InputMethodListener() {
            @Override
            public void caretPositionChanged(InputMethodEvent
event) {
                Util.setIntervalHistogramaUV(jSliderIntervalUV.getValue());
                refreshAllUV();
            }

            @Override
            public void
inputMethodTextChanged(InputMethodEvent event) {
                Util.setIntervalHistogramaUV(jSliderIntervalUV.getValue());
                refreshAllUV();
            }
        });

        jSliderIntervalUV.addChangeListener(new ChangeListener()
{
            @Override
            public void stateChanged(ChangeEvent event) {
                Util.setIntervalHistogramaUV(jSliderIntervalUV.getValue());
                refreshAllUV();
            }
        });
        return jSliderIntervalUV;
    }
}

private JLabel getJLabelIntervalUV() {
    if (jLabelIntervalUV == null) {
        jLabelIntervalUV = new JLabel();
        jLabelIntervalUV.setText("Interval hist. UV");
    }
    return jLabelIntervalUV;
}

```

```
    }

    private JCheckBox getJCheckBoxGraficIntersectie() {
        if (jCheckBoxGraficIntersectie == null) {
            jCheckBoxGraficIntersectie = new JCheckBox();
            jCheckBoxGraficIntersectie.setText("Intersectie");
            jCheckBoxGraficIntersectie.addActionListener(new
ActionListener() {

                public void actionPerformed(ActionEvent event) {

                    Util.setGraficIntersectie(jCheckBoxGraficIntersectie.isSelected());
                    refreshAllUV();

                }
            });
        }
        return jCheckBoxGraficIntersectie;
    }

    private JCheckBox getJCheckBoxGraficMinim() {
        if (jCheckBoxGraficMinim == null) {
            jCheckBoxGraficMinim = new JCheckBox();
            jCheckBoxGraficMinim.setSelected(true);
            jCheckBoxGraficMinim.setText("Minim");
            jCheckBoxGraficMinim.addActionListener(new
ActionListener() {

                public void actionPerformed(ActionEvent event) {

                    Util.setGraficMinim(jCheckBoxGraficMinim.isSelected());
                    refreshAllUV();

                }
            });
        }
        return jCheckBoxGraficMinim;
    }

    private JCheckBox getJCheckBoxGraficMaxim() {
        if (jCheckBoxGraficMaxim == null) {
            jCheckBoxGraficMaxim = new JCheckBox();
            jCheckBoxGraficMaxim.setSelected(true);
            jCheckBoxGraficMaxim.setText("Maxim");
            jCheckBoxGraficMaxim.addActionListener(new
ActionListener() {

                public void actionPerformed(ActionEvent event) {

                    Util.setGraficMaxim(jCheckBoxGraficMaxim.isSelected());
                    refreshAllUV();

                }
            });
        }
        return jCheckBoxGraficMaxim;
    }

    private JLabel getJLabelGrafice() {
        if (labelGrafice == null) {
            labelGrafice = new JLabel();
            labelGrafice.setText("Grafice");
        }
        return labelGrafice;
    }
}
```

```
private JSlider getJSliderClearSky() {
    if (jSliderClearSky == null) {
        jSliderClearSky = new JSlider();
        Util.setClearSkyReport(jSliderClearSky.getValue());
        jSliderClearSky.addInputMethodListener(new
InputMethodListener() {
            @Override
            public void caretPositionChanged(InputMethodEvent
event) {
                Util.setClearSkyReport(jSliderClearSky.getValue());
                refreshAllUV();
            }
            @Override
            public void
inputMethodTextChanged(InputMethodEvent event) {
                Util.setClearSkyReport(jSliderClearSky.getValue());
                refreshAllUV();
            }
        });
        jSliderClearSky.addChangeListener(new ChangeListener() {
            @Override
            public void stateChanged(ChangeEvent event) {
                Util.setClearSkyReport(jSliderClearSky.getValue());
                refreshAllUV();
            }
        });
    }
    return jSliderClearSky;
}

private JCheckBox getJCheckBoxClearSky() {
    if (jCheckBoxClearSky == null) {
        jCheckBoxClearSky = new JCheckBox();
        jCheckBoxClearSky.setText("Elimina zonele fara poluare");
        jCheckBoxClearSky.addActionListener(new ActionListener()
{
            public void actionPerformed(ActionEvent event) {
                Util.setClearSky(jCheckBoxClearSky.isSelected());
                refreshAllUV();
            }
        });
    }
    return jCheckBoxClearSky;
}

private JLabel getJLabelUVTools() {
    if (jLabelUVTools == null) {
        jLabelUVTools = new JLabel();
        jLabelUVTools.setText("Prelucreare UV");
    }
    return jLabelUVTools;
}
```

```

private JLabel getJLabelThreshold() {
    if (jLabelThreshold == null) {
        jLabelThreshold = new JLabel();
        jLabelThreshold.setText("Prag");
    }
    return jLabelThreshold;
}

private JPanel getJPanelTools() {
    if (jPanelTools == null) {
        jPanelTools = new JPanel();

        jPanelTools.setBorder(BorderFactory.createTitledBorder("Unelte"));
        jPanelTools.setLayout(new GroupLayout());
        jPanelTools.add(getJLabelUVTools(), new Constraints(new
Leading(12, 12, 12), new Leading(0, 12, 12)));
        jPanelTools.add(getJSliderClearSky(), new Constraints(new
Leading(121, 218, 10, 10), new Leading(24, 12, 12)));
        jPanelTools.add(getJCheckBoxGraficMaxim(), new
Constraints(new Leading(121, 10, 10), new Leading(41, 8, 8)));
        jPanelTools.add(getJCheckBoxGraficMinim(), new
Constraints(new Leading(196, 10, 10), new Leading(41, 8, 8)));
        jPanelTools.add(getJSliderIntervalUV(), new
Constraints(new Leading(121, 218, 10, 10), new Leading(66, 12, 12)));
        jPanelTools.add(getJCheckBoxGraficIntersectie(), new
Constraints(new Leading(267, 10, 10), new Leading(41, 8, 8)));
        jPanelTools.add(getJCheckBoxClearSky(), new
Constraints(new Leading(121, 8, 8), new Leading(-4, 8, 8)));
        jPanelTools.add(getJSliderIntervalIR(), new
Constraints(new Leading(121, 218, 12, 12), new Leading(94, 12, 12)));
        jPanelTools.add(getJLabelThreshold(), new Constraints(new
Leading(12, 12, 12), new Leading(24, 12, 12)));
        jPanelTools.add(getJLabelGrafice(), new Constraints(new
Leading(12, 12, 12), new Leading(46, 12, 12)));
        jPanelTools.add(getJLabelIntervalUV(), new
Constraints(new Leading(12, 12, 12), new Leading(66, 12, 12)));
        jPanelTools.add(getJLabelIntervalIR(), new
Constraints(new Leading(12, 99, 12, 12), new Leading(88, 12, 12)));
    }
    return jPanelTools;
}

private JPanel getJPanel1() {
    if (jPanelIRHistogram == null) {
        jPanelIRHistogram = new IRHistogramPanel(
HistogramDisplayMode.SIMPLE_REPRESENTATION);
        jPanelIRHistogram.setBorder(BorderFactory
.createTitledBorder("Histograma IR"));
        // jPanelIRHistogram.setYRatio(4);
        // jPanelIRHistogram.setInterval(1);
        jPanelIRHistogram.setFocusable(true);
        jPanelIRHistogram.setEnabled(true);
        jPanelIRHistogram.setVisible(true);
        jPanelIRHistogram.setDoubleBuffered(true);
        jPanelIRHistogram.setVerifyInputWhenFocusTarget(true);
        jPanelIRHistogram.setRequestFocusEnabled(true);
        jPanelIRHistogram.setOpaque(true);
        jPanelIRHistogram.setLayout(new GroupLayout());
    }
    return jPanelIRHistogram;
}

```

```

private JPanel getJPanel0() {
    if (jPanelUVHistogram == null) {
        jPanelUVHistogram = new UVHistogramPanel(
HistogramDisplayMode.SIMPLE_REPRESENTATION);
        jPanelUVHistogram.setBorder(BorderFactory
            .createTitledBorder("Histograma UV"));
        // jPanelUVHistogram.setYRatio(10);
        // jPanelUVHistogram.setInterval(2);
        jPanelUVHistogram.setFocusable(true);
        jPanelUVHistogram.setEnabled(true);
        jPanelUVHistogram.setVisible(true);
        jPanelUVHistogram.setDoubleBuffered(true);
        jPanelUVHistogram.setVerifyInputWhenFocusTarget(true);
        jPanelUVHistogram.setRequestFocusEnabled(true);
        jPanelUVHistogram.setOpaque(true);
        jPanelUVHistogram.setLayout(new GroupLayout());
    }
    return jPanelUVHistogram;
}

private JMenuItem getJMenuItemResetAreaUV() {
    if (jMenuItemResetAreaUV == null) {
        jMenuItemResetAreaUV = new JMenuItem();
        jMenuItemResetAreaUV.setText("Stergeti zona de
masurare");
        jMenuItemResetAreaUV.addActionListener(new
ActionListener() {

            public void actionPerformed(ActionEvent event) {
                jMenuItemResetAreaActionPerformedUV(event);
            }
        });
    }
    return jMenuItemResetAreaUV;
}

private JMenuItem getJMenuItemCalculVitezaVant() {
    if (jMenuItemCalculVitezaVant == null) {
        jMenuItemCalculVitezaVant = new JMenuItem();
        jMenuItemCalculVitezaVant.setText("Calcul viteza
vant...");
        jMenuItemCalculVitezaVant.addActionListener(new
ActionListener() {

            public void actionPerformed(ActionEvent event) {
                jMenuItemCalculVitezaVant(event);
            }
        });
    }
    return jMenuItemCalculVitezaVant;
}

private JMenuItem getJMenuItemResetAreaIR() {
    if (jMenuItemResetAreaIR == null) {

```

200 Anexa 1

```

        jMenuItemResetAreaIR = new JMenuItem();
        jMenuItemResetAreaIR.setText("Stergeti zona de
masurare");

        jMenuItemResetAreaIR.addActionListener(new
ActionListener() {

            public void actionPerformed(ActionEvent event) {

                jMenuItemResetAreaActionPerformedIR(event);
            }
        });
    }
    return jMenuItemResetAreaIR;
}

private JCheckBoxMenuItem getJCheckBoxMenuItem0Level() {
    if (jCheckBoxMenuItem0Level == null) {
        jCheckBoxMenuItem0Level = new JCheckBoxMenuItem();
        jCheckBoxMenuItem0Level.setText("Alegeti locatia nivel
0");
        jCheckBoxMenuItem0Level.addActionListener(new
ActionListener() {

            public void actionPerformed(ActionEvent event) {

                jCheckBoxMenuItem0LevelActionActionPerformed(event);
            }
        });
    }
    return jCheckBoxMenuItem0Level;
}

private JCheckBoxMenuItem getJCheckBoxMenuItemEtalonLevel() {
    if (jCheckBoxMenuItemEtalonLevel == null) {
        jCheckBoxMenuItemEtalonLevel = new JCheckBoxMenuItem();
        jCheckBoxMenuItemEtalonLevel
            .setText("Selectati punctul de masurare
standard");
        jCheckBoxMenuItemEtalonLevel
            .addActionListener(new ActionListener() {

                public
                void
                actionPerformed(ActionEvent event) {

                    jCheckBoxMenuItemEtalonLevelActionActionPerformed(event);
                }
            });
    }
    return jCheckBoxMenuItemEtalonLevel;
}

private JCheckBoxMenuItem getJCheckBoxMenuItemChooseAreaUV() {
    if (jCheckBoxMenuItemChooseAreaUV == null) {
        jCheckBoxMenuItemChooseAreaUV = new JCheckBoxMenuItem();
        jCheckBoxMenuItemChooseAreaUV.setText("Alegeti zona de
masurare");
        jCheckBoxMenuItemChooseAreaUV.addActionListener(new
ActionListener() {

            public void actionPerformed(ActionEvent event) {

```



```

        jCheckBoxMenuItemChooseAreaActionActionPerformedUV(event);
    }
    });
}
return jCheckBoxMenuItemChooseAreaUV;
}

private JCheckBoxMenuItem getJCheckBoxMenuItemChooseAreaIR() {
    if (jCheckBoxMenuItemChooseAreaIR == null) {
        jCheckBoxMenuItemChooseAreaIR = new JCheckBoxMenuItem();
        jCheckBoxMenuItemChooseAreaIR.setText("Alegeti zona de
masurare");
        jCheckBoxMenuItemChooseAreaIR.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent event) {
                jCheckBoxMenuItemChooseAreaActionActionPerformedIR(event);
            }
        });
    }
    return jCheckBoxMenuItemChooseAreaIR;
}

private JCheckBoxMenuItem getJCheckBoxMenuItemChoosePixelDimension() {
    if (jCheckBoxMenuItemChoosePixelDimension == null) {
        jCheckBoxMenuItemChoosePixelDimension = new
JCheckBoxMenuItem();
        jCheckBoxMenuItemChoosePixelDimension
        .setText("Alege zona masurare dimensiune
pixel...");
        jCheckBoxMenuItemChoosePixelDimension
        .addActionListener(new ActionListener() {
            public void
actionPerformed(ActionEvent event) {
                jCheckBoxMenuItemChoosePixelDimensionActionActionPerformed(event);
            }
        });
    }
    return jCheckBoxMenuItemChoosePixelDimension;
}

private JCheckBoxMenuItem getJCheckBoxMenuItemChooseFluxSection() {
    if (jCheckBoxMenuItemChooseFluxSection == null) {
        jCheckBoxMenuItemChooseFluxSection = new
JCheckBoxMenuItem();
        jCheckBoxMenuItemChooseFluxSection
        .setText("Alege sectiune pentru calcul
flux");
        jCheckBoxMenuItemChooseFluxSection
        .addActionListener(new ActionListener() {
            public void
actionPerformed(ActionEvent event) {
                jCheckBoxMenuItemChooseFluxSectionActionActionPerformed(event);
            }
        });
    }
}

```

```
    }
    return jCheckBoxMenuItemChooseFluxSection;
}

private JMenu getJMenuOperationUV() {
    if (jMenuOperationUV == null) {
        jMenuOperationUV = new JMenu();
        jMenuOperationUV.setText("Prelucrare date UV");
        jMenuOperationUV.setOpaque(false);
        jMenuOperationUV.add(getJCheckBoxMenuItemDynamic());
        jMenuOperationUV.add(getJCheckBoxMenuItem0Level());
        jMenuOperationUV.add(getJCheckBoxMenuItemEtalonLevel());
        if (Util.getProcessor() != null) {
            jMenuOperationUV.addSeparator();
        }
        jMenuOperationUV.add(getJCheckBoxMenuItemChooseAreaUV());
        jMenuOperationUV.add(getJMenuItemResetAreaUV());
        if (Util.getProcessor() != null) {
            jMenuOperationUV.addSeparator();
        }
    }

    jMenuOperationUV.add(getJCheckBoxMenuItemChoosePixelDimension());

    jMenuOperationUV.add(getJCheckBoxMenuItemChooseFluxSection());
    if (Util.getProcessor() != null) {
        jMenuOperationUV.addSeparator();
    }
    jMenuOperationUV.add(getJMenuCalculVitezaVant());
}
return jMenuOperationUV;
}

private JMenu getJMenuOperationIR() {
    if (jMenuOperationIR == null) {
        jMenuOperationIR = new JMenu();
        jMenuOperationIR.setText("Prelucrare date IR");
        jMenuOperationIR.setOpaque(false);
        jMenuOperationIR.add(getJCheckBoxMenuItemChooseAreaIR());
        jMenuOperationIR.add(getJMenuItemResetAreaIR());
    }
    return jMenuOperationIR;
}

private JCheckBoxMenuItem getJCheckBoxMenuItemDynamic() {
    if (jCheckBoxMenuItemDynamic == null) {
        jCheckBoxMenuItemDynamic = new JCheckBoxMenuItem();
        jCheckBoxMenuItemDynamic
            .setText("Alegeti punctul de intersectie
axele de masurare");
        jCheckBoxMenuItemDynamic.addActionListener(new
ActionListener() {

            public void actionPerformed(ActionEvent event) {

                jCheckBoxMenuItemDynamicActionActionPerformed(event);
            }
        });
    }
    return jCheckBoxMenuItemDynamic;
}

private JMenu getJMenuHelp() {
```

```

        if (helpMenu == null) {
            helpMenu = new JMenu();
            helpMenu.setText("Ajutor");
            helpMenu.setOpaque(false);
            helpMenu.add(getJMenuItemAbout());
        }
        return helpMenu;
    }

    private JMenuItem getJMenuItemAbout() {
        if (jMenuItemAbout == null) {
            jMenuItemAbout = new JMenuItem();
            jMenuItemAbout.setText("Despre");

            ActionListener lst = new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    String message = TITLE + "\n" +
                        "Autor:      Drd.      Catalin
Nisulescu";
                    JOptionPane.showMessageDialog(null,
message,
                        "Despre      UVCamera",
JOptionPane.INFORMATION_MESSAGE);
                }
            };
            jMenuItemAbout.addActionListener(lst);
        }
        return jMenuItemAbout;
    }

    private JPanel getInformationPanel() {
        if (infoPanel == null) {
            infoPanel = new InformationPanel();
            infoPanel.setBorder(BorderFactory
                .createTitledBorder("Panou informare"));
            infoPanel.setLayout(new GroupLayout());
        }
        return infoPanel;
    }

    private JPanel getHGraphicPanel() {
        if (hGraphicPanel == null) {
            hGraphicPanel = new HGraphicPanel();
            hGraphicPanel.setBorder(BorderFactory
                .createTitledBorder("Concentratie      SO2
[PPM]"));
            hGraphicPanel.setLayout(new GroupLayout());
        }
        return hGraphicPanel;
    }

    private UVImagePanel getImagePanel() {
        if (uvImagePanel == null) {
            uvImagePanel = new UVImagePanel();
            uvImagePanel.setBorder(BorderFactory
                .createTitledBorder("Imagine UV"));
            uvImagePanel.setFocusable(true);
            uvImagePanel.setEnabled(true);
            uvImagePanel.setVisible(true);
            uvImagePanel.setDoubleBuffered(true);
        }
    }

```

```

        uvImagePanel.setVerifyInputWhenFocusTarget(true);
        uvImagePanel.setRequestFocusEnabled(true);
        uvImagePanel.setOpaque(true);
        uvImagePanel.setLayout(new GroupLayout());
        uvImagePanel.addMouseMotionListener(new
MouseMotionAdapter() {

            public void mouseMoved(MouseEvent event) {
                imagePanelMouseMotionMouseMoved(event);
            }

        });
        uvImagePanel.addMouseListener(new MouseAdapter() {

            public void mouseClicked(MouseEvent event) {
                imagePanelMouseMouseClicked(event);
            }

        });
    }
    return uvImagePanel;
}

private IRImagePanel getIRImagePanel() {
    if (irImagePanel == null) {
        irImagePanel = new IRImagePanel();

        irImagePanel.setBorder(BorderFactory.createTitledBorder("Imagine IR"));
        irImagePanel.setFocusable(true);
        irImagePanel.setEnabled(true);
        irImagePanel.setVisible(true);
        irImagePanel.setDoubleBuffered(true);
        irImagePanel.setVerifyInputWhenFocusTarget(true);
        irImagePanel.setRequestFocusEnabled(true);
        irImagePanel.setOpaque(true);
        irImagePanel.setLayout(new GroupLayout());
        irImagePanel.addMouseMotionListener(new
MouseMotionAdapter() {

            public void mouseMoved(MouseEvent event) {
                imagePanelMouseMotionMouseMoved(event);
            }

        });
        irImagePanel.addMouseListener(new MouseAdapter() {

            public void mouseClicked(MouseEvent event) {
                imagePanelMouseMouseClicked(event);
            }

        });
    }
    return irImagePanel;
}

private JPanel getVGraphicPanel() {
    if (vGraphicPanel == null) {
        vGraphicPanel = new VGraphicPanel();
        vGraphicPanel.setBorder(BorderFactory
        .createTitledBorder("Concentratie SO2
[PPM]"));
        vGraphicPanel.setLayout(new GroupLayout());
    }
    return vGraphicPanel;
}

private JMenuBar getJMenuBar0() {
    if (jMenuBar0operation == null) {

```

```

        jMenuBarOperation = new JMenuBar();
        jMenuBarOperation.add(getJMenuFile());
        jMenuBarOperation.add(getJMenuOperationUV());
        jMenuBarOperation.add(getJMenuOperationIR());
        if (Util.getProcessor() != null) {
            jMenuBarOperation.add(Box.createHorizontalGlue());
        }
        jMenuBarOperation.add(getJMenuHelp());
    }
    return jMenuBarOperation;
}

private JMenu getJMenuFile() {
    if (jMenuFile == null) {
        jMenuFile = new JMenu();
        jMenuFile.setText("Fisier");
        jMenuFile.add(getJMenuOpen());
        if (Util.getProcessor() != null) {
            jMenuFile.addSeparator();
        }
        jMenuFile.add(getJMenuExit());
    }
    return jMenuFile;
}

private JMenuItem getJMenuExit() {
    if (jMenuExit == null) {
        jMenuExit = new JMenuItem();
        jMenuExit.setText("Iesire");
        ActionListener lst = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.exit(0);
            }
        };
        jMenuExit.addActionListener(lst);
    }
    return jMenuExit;
}

private JMenuItem getJMenuOpen() {
    if (open == null) {
        open = new JMenuItem();
        open.setText("Deschide folder masuratori...");
        ActionListener lst = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JFileChooser chooser = new JFileChooser();
                chooser.setCurrentDirectory(new
java.io.File("."));
                chooser.setDialogTitle("Select folder");

                chooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
                chooser.setAcceptAllFileFilterUsed(false);
                if (chooser.showOpenDialog(null) ==
JFileChooser.APPROVE_OPTION) {
                    Util.resetAll();

                    Util.setFolder(chooser.getSelectedFile()
                                .getAbsolutePath());
                    startProcessing();
                }
            }
        }
    }
}

```

```

        };
        open.addActionListener(lst);
    }
    return open;
}

private static void installLnF() {
    try {
        String lnfClassname = PREFERRED_LOOK_AND_FEEL;
        if (lnfClassname == null)
            lnfClassname =
UIManager.getCrossPlatformLookAndFeelClassName();
        UIManager.setLookAndFeel(lnfClassname);
    } catch (Exception e) {
        System.err.println("Cannot install " +
PREFERRED_LOOK_AND_FEEL
            + " on this platform:" + e.getMessage());
    }
}

public void startMainWindow() {
    installLnF();
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            MainWindow frame = new MainWindow();

            frame.setDefaultCloseOperation(MainWindow.EXIT_ON_CLOSE);
            frame.setTitle("Aplicatie prelucrare imagini
poluare UV/IR");

            frame.getContentPane().setPreferredSize(frame.getSize());
            frame.pack();
            frame.setLocationRelativeTo(null);
            frame.setVisible(true);
            startProcessing();
        }
    });
}

private void startProcessing() {
    Util.setOValue(getOValue());

    if (Util.getFolder() == null)
        return;
    try {
        Util.processFolder();
    } catch (Exception e) {
        JOptionPane.showConfirmDialog(this, e.getMessage(),
            "Exception occurred",
JOptionPane.DEFAULT_OPTION,
            JOptionPane.ERROR_MESSAGE);
        return;
    }
    refreshAllUV();
}

private void refreshAllUV() {
    Util.setOValue(getOValue());

    uvImagePanel.repaint();
    vGraphicPanel.repaint();
    hGraphicPanel.repaint();
}

```

```
        infoPanel.repaint();

        irImagePanel.repaint();
        jPanelUVHistogram.repaint();

        setTitle(TITLE + " - " + Util.getLocation() + ":" +
Util.getDate());

        refreshAllIR();
    }

    private void refreshAllIR() {
        irImagePanel.repaint();
        jPanelIRHistogram.repaint();
    }

    public void resetAll(){
        jCheckBoxClearSky.setSelected(false);
        jCheckBoxGraficIntersectie.setSelected(true);
        jCheckBoxGraficMaxim.setSelected(true);
        jCheckBoxGraficMinim.setSelected(true);
        jSliderClearSky.setValue(50);
        jSliderIntervalIR.setValue(50);
        jSliderIntervalUV.setValue(50);
    }

    public void setProcessor(Processor processor) {
        Util.setProcessor(processor);
    }

    private void jCheckBoxMenuItemDynamicActionPerformed(ActionEvent
event) {
        if (jCheckBoxMenuItemDynamic.isSelected()) {
            operationMode = OperationMode.DYNAMIC;
            jCheckBoxMenuItem0Level.setSelected(false);
            jCheckBoxMenuItemEtalonLevel.setSelected(false);
        } else if (jCheckBoxMenuItem0Level.isSelected()) {
            operationMode = OperationMode.CHOOSE_0_LEVEL;
            jCheckBoxMenuItemDynamic.setSelected(false);
            jCheckBoxMenuItemEtalonLevel.setSelected(false);
        } else if (jCheckBoxMenuItemEtalonLevel.isSelected()) {
            operationMode = OperationMode.CHOOSE_ETALON_LEVEL;
            jCheckBoxMenuItemDynamic.setSelected(false);
            jCheckBoxMenuItemDynamic.setSelected(false);
        } else {
            operationMode = OperationMode.STATIC;
            jCheckBoxMenuItemDynamic.setSelected(false);
            jCheckBoxMenuItem0Level.setSelected(false);
            jCheckBoxMenuItemEtalonLevel.setSelected(false);
        }
    }

    private void imagePanelMouseMotionMouseMoved(MouseEvent event) {
        int x = 0;
        int y = 0;
        if ((event.getPoint().y - 15 < MAX_Y - 1)
            && event.getPoint().y - 15 > 2) {
            y = event.getPoint().y - 15;
        } else {
            return;
        }
    }
}
```

```
    }

    if ((event.getPoint().x - 35 < MAX_X - 1)
        && event.getPoint().x - 35 > 2) {
        x = event.getPoint().x - 35;
    } else {
        return;
    }

    switch (operationMode) {
    case DYNAMIC:
        Util.setVPosition(x);
        Util.setHPosition(y);

        break;
    case CHOOSE_0_LEVEL:
        Util.setOPosition(new Point(x, y));

        break;
    case CHOOSE_ETALON_LEVEL:
        Util.setEtalonPosition(new Point(x, y));

        break;
    case CHOOSE_FIRST_CORNER_UV:
        Util.setFirstCornerUV(new Point(x, y));

        break;
    case CHOOSE_SECOND_CORNER_UV:
        Util.setSecondCornerUV(new Point(x, y));

        break;
    case CHOOSE_FIRST_CORNER_IR:
        Util.setFirstCornerIR(new Point(x, y));
        System.out.println(new Point(x, y));
        refreshAllIR();
        return;
    case CHOOSE_SECOND_CORNER_IR:
        Util.setSecondCornerIR(new Point(x, y));
        refreshAllIR();
        return;
    case CHOOSE_FIRST_POINT_PIXEL_DIMENSION:
        Util.setFirstCornerPixelDimension(new Point(x, y));

        break;
    case CHOOSE_SECOND_POINT_PIXEL_DIMENSION:
        Util.setSecondCornerPixelDimension(new Point(x, y));

        break;
    case CHOOSE_FIRST_POINT_FLUX_SECTION:
        Util.setFirstPointFluxSection(new Point(x, y));

        break;
    case CHOOSE_SECOND_POINT_FLUX_SECTION:
        Util.setSecondPointFluxSection(new Point(x, y));

        break;
    default:
        return;
    }
    refreshAllUV();
}
```



```

private void imagePanelMouseMouseClicked(MouseEvent event) {
    if (operationMode == OperationMode.CHOOSE_FIRST_CORNER_UV) {
        operationMode = OperationMode.CHOOSE_SECOND_CORNER_UV;
        return;
    }

    if (operationMode == OperationMode.CHOOSE_FIRST_CORNER_IR) {
        operationMode = OperationMode.CHOOSE_SECOND_CORNER_IR;
        return;
    }

    if (operationMode == OperationMode.CHOOSE_FIRST_POINT_PIXEL_DIMENSION) {
        operationMode = OperationMode.CHOOSE_SECOND_POINT_PIXEL_DIMENSION;
        return;
    }

    if (operationMode == OperationMode.CHOOSE_FIRST_POINT_FLUX_SECTION) {
        operationMode = OperationMode.CHOOSE_SECOND_POINT_FLUX_SECTION;
        return;
    }

    if (operationMode == OperationMode.CHOOSE_SECOND_POINT_PIXEL_DIMENSION) {
        operationMode = OperationMode.STATIC;

        EnterValueDlg dlg = new EnterValueDlg();
        dlg.getJLabel0().setText("Lungimea distanta selectata (cm)");
        dlg.setTitle("Introduceti valoarea pentru distanta selectata (cm)");
        dlg.show();
        String value = dlg.getjTextFieldValue().getText();
        try {
            Util.setDimensiunePixel(Integer.parseInt(value));
        } catch (Exception ignored) {}
        refreshAllUV();
    }

    if (operationMode == OperationMode.CHOOSE_ETALON_LEVEL) {
        operationMode = OperationMode.STATIC;
        refreshAllUV();
    }

    if (operationMode == OperationMode.CHOOSE_0_LEVEL) {
        JOptionPane.showConfirmDialog(
            null,
            "Ati ales locatia pentru nivel 0 la coordonatele "
            + Util.getOPosition().x + "x"
            + Util.getOPosition().y,
            "Nivel 0",
            JOptionPane.DEFAULT_OPTION,
            JOptionPane.OK_OPTION);
        try {
            Util.getProcessor().startProcessing();
        }
    }
}

```

```
        } catch (Exception e) {
        }
        refreshAllUV();
    }
    jCheckBoxMenuItemDynamic.setSelected(false);
    jCheckBoxMenuItem0Level.setSelected(false);
    jCheckBoxMenuItemEtalonLevel.setSelected(false);
    jCheckBoxMenuItemChooseAreaUV.setSelected(false);
    jCheckBoxMenuItemChoosePixelDimension.setSelected(false);
    jCheckBoxMenuItemChooseFluxSection.setSelected(false);
    jCheckBoxMenuItemChooseAreaIR.setSelected(false);
    operationMode = OperationMode.STATIC;
}

private void jCheckBoxMenuItem0LevelActionPerformed(ActionEvent
event) {
    if (jCheckBoxMenuItem0Level.isSelected()) {
        operationMode = OperationMode.CHOOSE_0_LEVEL;
    } else {
        operationMode = OperationMode.STATIC;
    }
}

private void jCheckBoxMenuItemEtalonLevelActionPerformed(
    ActionEvent event) {
    if (jCheckBoxMenuItemEtalonLevel.isSelected()) {
        operationMode = OperationMode.CHOOSE_ETALON_LEVEL;
    } else {
        operationMode = OperationMode.STATIC;
    }
}

private void jMenuItemResetAreaActionPerformedUV(ActionEvent event) {
    Util.setFirstCornerUV(null);
    Util.setSecondCornerUV(null);
    refreshAllUV();
}

private void jMenuItemCalculVitezaVant(ActionEvent event) {
    CalculVitezaVantDlg dlg = new CalculVitezaVantDlg();
    if (Util.getDimensiunePixelValue() == 0){
        JOptionPane.showConfirmDialog(
            null,
            "Nu ati calculat dimensiunea pixel.",
            "Nu ati calculat dimensiunea pixel.",
            JOptionPane.DEFAULT_OPTION,
            JOptionPane.OK_OPTION);
        return;
    }
    dlg.setDimensiunePixel(Util.getDimensiunePixelValue());
    dlg.show();
    refreshAllUV();
}

private void jMenuItemResetAreaActionPerformedIR(ActionEvent event) {
    Util.setFirstCornerIR(null);
    Util.setSecondCornerIR(null);
    refreshAllIR();
}

private void jCheckBoxMenuItemChooseAreaActionPerformedUV(
    ActionEvent event) {
```

```

        if (jCheckBoxMenuItemChooseAreaUV.isSelected()) {
            operationMode = OperationMode.CHOOSE_FIRST_CORNER_UV;
        } else {
            operationMode = OperationMode.STATIC;
        }
    }

    private void jCheckBoxMenuItemChooseAreaActionActionPerformedIR(
        ActionEvent event) {
        if (jCheckBoxMenuItemChooseAreaIR.isSelected()) {
            operationMode = OperationMode.CHOOSE_FIRST_CORNER_IR;
            Util.setFirstCornerIR(null);
            Util.setSecondCornerIR(null);
            refreshAllIR();
        } else {
            operationMode = OperationMode.STATIC;
        }
    }

    private void jCheckBoxMenuItemChoosePixelDimensionActionActionPerformed(
        ActionEvent event) {
        if (jCheckBoxMenuItemChoosePixelDimension.isSelected()) {
            operationMode =
OperationMode.CHOOSE_FIRST_POINT_PIXEL_DIMENSION;
        } else {
            operationMode = OperationMode.STATIC;
        }
    }

    private void jCheckBoxMenuItemChooseFluxSectionActionActionPerformed(
        ActionEvent event) {
        if (jCheckBoxMenuItemChooseFluxSection.isSelected()) {
            operationMode =
OperationMode.CHOOSE_FIRST_POINT_FLUX_SECTION;
        } else {
            operationMode = OperationMode.STATIC;
        }
    }

    private int getOValue() {
        if (Util.getProcessor() == null)
            return 0;
        if (Util.getProcessor().getResult() == null)
            return 0;

        if (Util.getOPosition() == null)
            return 0;

        int size = 3;
        int value = 0;
        int noOfValues = 0;
        for (int i = Util.getOPosition().x - size; i <
Util.getOPosition().x
            + size; i++) {
            for (int j = Util.getOPosition().y - size; j <
Util.getOPosition().y
                + size; j++) {
                try {
                    value +=
Util.getProcessor().getResult()[Util

```

```
        .getOPosition().x][Util.getOPosition().y];
                noOfValues++;
            } catch (Exception ignored) {
            }
        }
    }
    return value / noOfValues;
}
}
```

com.cata.uvcamera.ui.ToolsPanel.java

```
package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class ToolsPanel extends JPanel {
    private JLabel jLabel0;
    private JCheckBox jCheckBoxClearSky;
    private JSlider jSlider0;

    public ToolsPanel() {
        super();
    }

    private void initComponents() {
        setLayout(new FlowLayout(FlowLayout.LEFT));
        add(getJLabel0());
        add(getJCheckBoxClearSky());
        add(getJSlider0());
        setSize(320, 126);
    }

    private JLabel getJLabel0() {
        if (jLabel0 == null) {
            jLabel0 = new JLabel();
            //
jLabel0.setHorizontalAlignment(SwingConstants.TRAILING);
            jLabel0.setText("Prelucrare UV");
            // jLabel0.setVerticalTextPosition(SwingConstants.TOP);
        }
        return jLabel0;
    }

    private JCheckBox getJCheckBoxClearSky() {
        if (jCheckBoxClearSky == null) {
            jCheckBoxClearSky = new JCheckBox();
            jCheckBoxClearSky.setVisible(true);
            jCheckBoxClearSky.setText("Elimina zona neutra");
        }
        return jCheckBoxClearSky;
    }

    private JSlider getJSlider0() {
        if (jSlider0 == null) {
            jSlider0 = new JSlider();
            jSlider0.setPaintLabels(true);
            jSlider0.setPaintTicks(true);
            jSlider0.setPreferredSize(new Dimension(250, 27));
        }
    }
}
```

```

        return jSlider0;
    }
}

```

com.cata.uvcamera.ui.UVHistogramPanel.java

```

package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class UVHistogramPanel extends AbstractHistogramPanel {
    public UVHistogramPanel(HistogramDisplayMode displayMode) {
        super(displayMode);
    }

    public void paint(Graphics g) {
        super.paint(g);
        Graphics2D g2 = (Graphics2D) g;

        if (Util.getFirstCornerUV() == null || Util.getSecondCornerUV()
== null) {
            g.drawString("Histograma nu poate fi calculata.", 50,
50);
            g.drawString("Selectati mai intai zona de masurat. ", 50,
70);
            return;
        }

        int maxValue = 0;
        if (shouldCompute) {
            max = (int) Util.getMaxValueUV();
            min = (int) Util.getMinValueUV();

            int noOfIntervals = Util.getIntervalHistogramaUV();
            if (noOfIntervals == 0) noOfIntervals = 1;
            displayValues = (max - min) / noOfIntervals;

            if (displayValues == 0) {
                g.drawString("Histograma nu poate fi calculata.",
50, 50);
                return;
            }

            if (noOfIntervals == 0) {
                return;
            }
            displayInterval = X_SIZE / noOfIntervals;

            values = computeDistribution(min, max, displayValues);

            // now compute y ratio
            // find largets value
            Iterator<Integer> it = values.values().iterator();
            while (it.hasNext()) {
                Integer i = it.next();
                if (i > maxValue) {
                    maxValue = i;
                }
            }

            Yratio = maxValue / (Y_SIZE - 50);

```

214 Anexa 1

```

        if (Yratio == 0)
            Yratio = 0.1;
        shouldCompute = true;
    }

    int lastY = 0;
    g2.setPaint(Color.red);

    boolean displayConcentration = true;
    int lastBarValue = 0;
    for (int i = min, j = 0; i < max; i = i + displayValues, j++) {
        Integer currentValue = values.get(i);
        if (currentValue == null)
            continue;
        lastBarValue = i;
    }

    for (int i = min, j = 0; i < max; i = i + displayValues, j++) {
        Integer currentValue = values.get(i);
        if (currentValue == null)
            continue;

        if (displayMode == HistogramDisplayMode.SIMPLE_REPRESENTATION) {
            // g2.setPaint(new Color((int) (i *
            Util.getImageGrayRatio()), // (int) (i * Util.getImageGrayRatio()), (int) (i
            * Util // .getImageGrayRatio())));
            g2.setPaint(Color.black);
            g2.drawRect(X_OFFSET + displayInterval * j, Y_SIZE
            - Y_OFFSET - (int) ((currentValue) / Yratio),
            displayInterval, (int) (currentValue / Yratio));
            g2.setPaint(new Color((int) (i *
            Util.getImageGrayRatio()), (int) (i *
            Util.getImageGrayRatio()), (int) (i * Util
            .getImageGrayRatio())));
            g2.fillRect(X_OFFSET + displayInterval * j, Y_SIZE
            - Y_OFFSET - (int) ((currentValue) / Yratio),
            displayInterval, (int) (currentValue / Yratio));
        } else {
            if (lastY != 0) {
                g2.drawLine(X_OFFSET + displayInterval *
                j, lastY, X_OFFSET + displayInterval * j,
                Y_SIZE - 0 - (int) (currentValue /
                Yratio));
            }
            lastY = Y_SIZE - 10 - (int) (currentValue /
            Yratio);
            g2.drawLine(X_OFFSET + displayInterval * j, Y_SIZE
            - 10 - (int) ((currentValue) / Yratio),
            X_OFFSET + displayInterval * (j + 1), Y_SIZE
            - 0

```

```

        - (int) (currentValue / Yratio));
    }

    if (currentValue == maxValue) {
        displayConcentration = true;
    }

    if (i == lastBarValue) {
        displayConcentration = true;
    }

    if (displayConcentration) {
        displayConcentration = false;

        g2.setPaint(Color.gray);
        if (X_OFFSET + displayInterval * j < X_SIZE + 25)
{
            g2.drawLine(X_OFFSET + displayInterval *
j, Y_SIZE
((currentValue) / Yratio),
Y_OFFSET
((currentValue) / Yratio));
        }
        if (i != lastBarValue) {
            g2.drawString("" + currentValue, X_SIZE +
30, Y_SIZE
((currentValue) / Yratio));
        }
        g2.setPaint(Color.black);
        g2.drawLine(X_OFFSET + displayInterval * j, Y_SIZE
- Y_OFFSET,
Y_SIZE - 5);

            int x = X_OFFSET + displayInterval * j;
            if (x > X_SIZE - X_OFFSET + 10) {
                x = X_SIZE - X_OFFSET + 10;
            }
            g2.drawString("" + i, x, Y_SIZE + Y_OFFSET + 5);
        }

        if (currentValue == min) {
            g2.setPaint(Color.gray);
            g2.drawString("" + min, X_OFFSET + displayInterval
* j, Y_SIZE
+ Y_OFFSET);
        }
    }
    g2.setPaint(Color.black);
    g2.drawString("Conc.", X_SIZE + 30, Y_SIZE + Y_OFFSET + 5);

    g2.translate(0, 100);
    g2.rotate(-Math.PI / 2.0);
    g2.drawString("Frecv.", 45, 15);
}
}

```

com.cata.uvcamera.ui.UVImagePanel.java

```
package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class UVImagePanel extends JPanel {
    private static final int _X = 512;
    private static final int _Y = 512;
    private static final int IMG_X = 512;
    private static final int IMG_Y = 512;
    private static final int OFFSET_Y = 15;
    private static final int OFFSET_X = 35;

    BufferedImage image;
    boolean isPrepared = false;
    private double[][] data;

    public UVImagePanel() {
        super();
    }

    public void init() {
        isPrepared = false;

        if (Util.getProcessor() == null)
            return;

        data = Util.getProcessor().getResult();
        if (data == null) {
            return;
        }

        image = new BufferedImage(_X, _Y, BufferedImage.TYPE_INT_RGB);

        double max = 0;

        for (int i = 0; i < IMG_X; i++) {
            for (int j = 0; j < IMG_Y; j++) {
                if (max < data[i][j]) {
                    max = data[i][j];
                }
            }
        }

        double ratio = 255 / max;
        Util.setImageGrayRation(ratio);

        for (int i = 0; i < IMG_X; i++) {
            for (int j = 0; j < IMG_Y; j++) {
                int color = (int) data[i][j];
                if (color < 0) {
                    image.setRGB(i, j, (new Color(0, 0,
0)).getRGB());
                    continue;
                }
                color = (int) (color * ratio);
                image.setRGB(i, j, getPixelColor(i, j, color));
            }
        }
    }
}
```



```

    }

    // vertical red line
    for (int j = 0; j < _Y; j++) {
        image.setRGB(Util.getVPosition() - 1, j,
Color.RED.getRGB());
        image.setRGB(Util.getVPosition(), j, Color.RED.getRGB());
        image.setRGB(Util.getVPosition() + 1, j,
Color.RED.getRGB());
    }

    // horizontal red line
    for (int j = 0; j < _Y; j++) {
        image.setRGB(j, Util.getHPosition() - 1,
Color.RED.getRGB());
        image.setRGB(j, Util.getHPosition(), Color.RED.getRGB());
        image.setRGB(j, Util.getHPosition() + 1,
Color.RED.getRGB());
    }

    // horizontal for pixel dimension
    if (Util.getFirstCornerPixelDimension() != null
        && Util.getSecondCornerPixelDimension() != null)
        for (int j = Util.getFirstCornerPixelDimension().x; j <
Util
            .getSecondCornerPixelDimension().x; j++) {
            image.setRGB(j,
Util.getFirstCornerPixelDimension().y - 1,
                Color.GREEN.getRGB());
            image.setRGB(j,
Util.getFirstCornerPixelDimension().y,
                Color.GREEN.getRGB());
            image.setRGB(j,
Util.getFirstCornerPixelDimension().y + 1,
                Color.GREEN.getRGB());
        }

    // oPosition
    int size = 2;
    if (Util.getOPosition() != null) {
        for (int i = Util.getOPosition().x - size; i <
Util.getOPosition().x
            + size; i++) {
            for (int j = Util.getOPosition().y - size; j <
Util
                .getOPosition().y + size; j++) {
                try {
                    image.setRGB(i, j,
Color.GREEN.getRGB());
                } catch (ArrayIndexOutOfBoundsException
ignored) {
                    // log.debug("ImagePanel oPosition
OUT: [" + i + "][" +
                        // j
                        // + "]);
                }
            }
        }
    }

    // etalonPosition

```

218 Anexa 1

```
size = 2;
if (Util.getEtalonPosition() != null) {
    for (int i = Util.getEtalonPosition().x - size; i < Util
        .getEtalonPosition().x + size; i++) {
        for (int j = Util.getEtalonPosition().y - size; j
            < Util
                .getEtalonPosition().y + size; j++)
        {
            try {
                image.setRGB(i, j,
                    Color.BLUE.getRGB());
            } catch (ArrayIndexOutOfBoundsException
                ignored) {
                // log.debug("ImagePanel
                etalonPosition OUT: [" + i +
                    // "]"[" + j
                    // + "]"");
            }
        }
    }
}

size = 2;
Point maxPoint = Util.getMaxPositionUV();
if (maxPoint != null) {
    for (int i = maxPoint.x - size; i < maxPoint.x + size;
        i++) {
        for (int j = maxPoint.y - size; j < maxPoint.y +
            size; j++) {
            try {
                image.setRGB(i, j,
                    Color.RED.getRGB());
            } catch (ArrayIndexOutOfBoundsException
                ignored) {
                // log.debug("ImagePanel
                maxPosition OUT: [" + i + "]"["
                    // + j + "]"");
            }
        }
    }
}

private int getPixelColor(int i, int j, int color) {
    int minX = 0;
    int maxX = 0;
    int minY = 0;
    int maxY = 0;

    if (Util.getFirstCornerUV() != null && Util.getSecondCornerUV()
        != null) {
        minX = Util.getFirstCornerUV().x;
        if (minX > Util.getSecondCornerUV().x) {
            minX = Util.getSecondCornerUV().x;
        }

        maxX = Util.getFirstCornerUV().x;
        if (maxX < Util.getSecondCornerUV().x) {
            maxX = Util.getSecondCornerUV().x;
        }

        minY = Util.getFirstCornerUV().y;
```

```

        if (minY > Util.getSecondCornerUV().y) {
            minY = Util.getSecondCornerUV().y;
        }

        maxY = Util.getFirstCornerUV().y;
        if (maxY < Util.getSecondCornerUV().y) {
            maxY = Util.getSecondCornerUV().y;
        }

        if ((i == Util.getFirstCornerUV().x || i ==
Util.getSecondCornerUV().x)
            && (j >= minY && j <= maxY)) {
            return Color.black.getRGB();
        }

        if ((i >= minX && i <= maxX)
Util.getSecondCornerUV().y) {
            && (j == Util.getFirstCornerUV().y || j ==
            return Color.black.getRGB();
        }
    }

    if (Util.isClearSky()) {
        if (Util.shouldEliminate(i, j)) {
            return Color.WHITE.getRGB();
        }
    }

    if (Util.isFluxSection(i, j)) {
        return Color.blue.getRGB();
    }

    if (Util.getFirstCornerUV() == null || Util.getSecondCornerUV()
== null) {
        return (new Color(color, color, color)).getRGB();
    }

    if ((i == Util.getFirstCornerUV().x || i ==
Util.getSecondCornerUV().x)
        && (j == Util.getFirstCornerUV().y || j ==
Util.getSecondCornerUV().y)) {
        return Color.black.getRGB();
    }

    if (i > minX && i < maxX && j > minY && j < maxY) {
        color = color;
    } else {
        color = color + (255 - color) / 4;
    }

    return (new Color(color, color, color)).getRGB();
}

public void paint(Graphics g) {
    super.paint(g);

    if (!isPrepared)
        init();

    if (data == null) {

```

220 Anexa 1

```
g.drawString("Nu este incarcat folderul de masuratori. ",
120, 300);
g.drawString(
    "Utilizati meniul \"Fisier\" pentru a
selecta folderul de masuratori.",
    120, 320);
return;
}

g.drawImage(image, OFFSET_X, OFFSET_Y, _X, _Y, null);
// X axis
g.drawString("0", 25, _Y + OFFSET_Y + 15);
g.drawString("256", 145, _Y + OFFSET_Y + 15);
g.drawString("512", 275, _Y + OFFSET_Y + 15);
g.drawString("768", 405, _Y + OFFSET_Y + 15);
g.drawString("1024", 520, _Y + OFFSET_Y + 15);
// Y axis
g.drawString("256", 5, _Y - 115);
g.drawString("512", 5, _Y - 250);
g.drawString("768", 5, _Y - 370);
g.drawString("1024", 3, _Y - 480);
}
}
```

com.cata.uvcamera.ui.VGraphicPanel.java

```
package com.cata.uvcamera.ui;

//VS4E -- DO NOT REMOVE THIS LINE!
public class VGraphicPanel extends JPanel {
    private static final int _X = 200;
    private static final int _Y = 512;

    private static final int IMG_X = 512;
    private static final int IMG_Y = 512;

    private double[][] data;

    public VGraphicPanel() {
        super();
    }

    public void paint(Graphics g) {
        super.paint(g);

        Graphics2D g2 = (Graphics2D) g;

        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);

        if (Util.getProcessor() == null)
            return;

        if (Util.getProcessor().getResult() != null) {
            data = Util.getProcessor().getResult();
        }

        if (data == null) {
```

```
        return;
    }

    double ratio = computeRatio();

    int vPosition = Util.getVPosition();
    int hPosition = Util.getHPosition();
    if (vPosition <= 0)
        return;

    if (Util.isGraficIntersectie()) {
        g2.setPaint(Color.gray);
        showGraphic(g2, ratio, vPosition, hPosition, true);
    }

    if (Util.getMaxPositionUV() != null && Util.isGraficMaxim()) {
        vPosition = (int) Util.getMaxPositionUV().getX();
        hPosition = (int) Util.getMaxPositionUV().getY();
        if (vPosition <= 0)
            return;
        g2.setPaint(Color.red);
        showGraphic(g2, ratio, vPosition, hPosition, false);
    }

    if (Util.getMinPositionUV() != null && Util.isGraficMinim()) {
        vPosition = (int) Util.getMinPositionUV().getX();
        hPosition = (int) Util.getMinPositionUV().getY();
        if (vPosition <= 0)
            return;
        g2.setPaint(Color.green);
        showGraphic(g2, ratio, vPosition, hPosition, false);
    }
}

private double computeRatio() {
    double max = 0;
    for (int i = 0; i < IMG_X; i++) {
        for (int j = 0; j < IMG_Y; j++) {
            if (max < data[i][j]) {
                max = data[i][j];
            }
        }
    }

    double ratio = _X / max;
    return ratio;
}

private void showGraphic(Graphics2D g2, double ratio, int vPosition,
    int hPosition, boolean offsetValue) {
    int startX = 0;
    int startY = 0;
    int offsetY = 15;
    int offsetX = 10;
    int endX = 0;
    int endY = 0;

    int mean = 0;

    int maxValue = 0;
    int minValue = Integer.MAX_VALUE;
}
```

```

        for (int j = 0; j < _Y; j++) {
            int x = (int) data[vPosition][j];

            if (maxValue < x)
                maxValue = x;
            if (minValue > x)
                minValue = x;

            mean += x;

            x = (int) (x * ratio);
            if (x < 0)
                x = 0;
            if (x > _X)
                x = _X;

            startX = endX;
            startY = endY;
            endX = x;
            endY = j;

            if (startY != 0) {
                g2.draw(new Line2D.Double(startX + offsetX, startY
+ offsetY,
                                        endX + offsetX, endY + offsetY));
            }
        }

        int x = (int) (maxValue * ratio);
        g2.drawLine(x + offsetX, offsetY, x + offsetX, _Y + offsetY);
        int offsetV = 0;
        if (offsetValue) {
            offsetV = 10;
        }
        g2.drawString("" + (int) (maxValue/* * Util.getCoefficient() */),
x - 15
                    + offsetX, _Y + offsetY + 15 + offsetV);

        int y = hPosition + offsetY;
        g2.drawLine(offsetX, y, _X + offsetX, y);

        if (Util.getOValue() > 0) {
            g2.setPaint(Color.black);
            x = (int) (Util.getOValue() * ratio);
            g2.drawLine(x - 1 + offsetX, offsetY, x - 1 + offsetX, _Y
+ offsetY);
            g2.drawLine(x + offsetX, offsetY, x + offsetX, _Y +
offsetY);

            g2.drawString("0", x - 5 + offsetX, _Y + offsetY + 15);
        }
    }
}

```

com.cata.uvcamera.util.OperationMode.java

```

package com.cata.uvcamera.util;

public enum OperationMode {
    DYNAMIC,

```

```

    STATIC,
    CHOOSE_0_LEVEL,
    CHOOSE_ETALON_LEVEL,
    CHOOSE_FIRST_CORNER_UV,
    CHOOSE_SECOND_CORNER_UV,
    CHOOSE_FIRST_POINT_PIXEL_DIMENSION,
    CHOOSE_SECOND_POINT_PIXEL_DIMENSION,
    CHOOSE_FIRST_POINT_FLUX_SECTION,
    CHOOSE_SECOND_POINT_FLUX_SECTION,
    CHOOSE_FIRST_CORNER_IR,
    CHOOSE_SECOND_CORNER_IR
}

```

com.cata.uvcamera.util.Util.java

```

package com.cata.uvcamera.util;

public class Util {
    private static final String LOCATION_TXT = "masuratoare.txt";
    private static Processor processor;
    private static int hPosition = 150;
    private static int vPosition = 150;
    private static Point oPosition = null;
    private static int oValue;
    private static Point etalonPosition = null;
    public static Point fluxSectionFirstPoint = null;
    public static Point fluxSectionSecondPoint = null;
    private static Point firstCornerUV;
    private static Point firstCornerIR;
    private static Point secondCornerUV;
    private static Point secondCornerIR;
    private static Point maxPositionUV = null;
    private static double maxValueUV;
    private static Point minPositionUV = null;
    private static double minValueUV;
    private static Point maxPositionIR = null;
    private static double maxValueIR;
    private static Point minPositionIR = null;
    private static double minValueIR;
    private static int etalonValue;
    private static MainWindow mainWindow;
    private static double ratio;

    public static int UV_X_SIZE = 512;
    public static int UV_Y_SIZE = 512;
    private static String location;
    private static String date;

    public static int IR_X_SIZE = 0;
    public static int IR_Y_SIZE = 0;
    public static int[] irData;
    private static String folder;
    private static Point firstCornerPixelDimension;
    private static Point secondCornerPixelDimension;
    private static Integer lungimeZonaMasurarePixel = null;
    private static Integer fluxValue = 0;
    private static boolean clearSky;
    private static int clearSkyReport;
    private static boolean graficIntersectie = true;
    private static boolean graficMinim = true;
}

```

```
private static boolean graficMaxim = true;
private static int[][] irDataMatrix;
private static int intervalHistogramaIR;
private static int intervalHistogramaUV;
private static double vitezaVant = 0;
private static double fluxViteza;
private static final String IRFileName = "ir.png";

public static void resetAll() {
    hPosition = 150;
    vPosition = 150;
    oPosition = null;
    oValue = 0;
    etalonPosition = null;
    fluxSectionFirstPoint = null;
    fluxSectionSecondPoint = null;
    firstCornerUV = null;
    firstCornerIR = null;
    secondCornerUV = null;
    secondCornerIR = null;
    maxPositionUV = null;
    maxValueUV = 0;
    minPositionUV = null;
    minValueUV = 0;
    maxPositionIR = null;
    maxValueIR = 0;
    minPositionIR = null;
    minValueIR = 0;
    etalonValue = 0;
    ratio = 0;
    irData = null;
    firstCornerPixelDimension = null;
    secondCornerPixelDimension = null;
    lungimeZonaMasurarePixel = null;
    fluxValue = 0;
    graficIntersectie = true;
    clearSky = false;
    clearSkyReport = 50;
    graficMinim = true;
    graficMaxim = true;
    irDataMatrix = null;
    intervalHistogramaIR = 50;
    intervalHistogramaUV = 50;
    fluxViteza = 0;
    vitezaVant = 0;
    if (mainFrame != null) {
        mainFrame.resetAll();
    }
}

public static void processIRImage() {
    String inFile = folder + "/" + IRFileName;
    Image image = Toolkit.getDefaultToolkit().getImage(inFile);
    try {

        PixelGrabber grabber = new PixelGrabber(image, 0, 0, -1,
-1, true);

        if (grabber.grabPixels()) {
            IR_X_SIZE = grabber.getWidth();
            IR_Y_SIZE = grabber.getHeight();

            if (isGreyscaleImage(grabber)) {
```



```
        // irData = (byte[]) grabber.getPixels();
    } else {
        irData = (int[]) grabber.getPixels();
    }
    System.out.println("Am citit IR pixels: " +
irData.length);
    }
    } catch (InterruptedException e1) {
        e1.printStackTrace();
    }
}

public static String getFluxViteza() {
    if (fluxViteza == 0) {
        return "-";
    }

    return "" + round(fluxViteza);
}

public static double round(double value) {
    return ((double) (int) (value * 100) / 100);
}

public static final boolean isGreyscaleImage(PixelGrabber pg) {
    return pg.getPixels() instanceof byte[];
}

public static String getDate() {
    return date;
}

public static void setDate(String date) {
    Util.date = date;
}

public static void setProcessor(Processor processor) {
    Util.processor = processor;
}

public static Processor getProcessor() {
    return processor;
}

public static int getHPosition() {
    return hPosition;
}

public static void setHPosition(int pos) {
    hPosition = pos;
}

public static Point getOPosition() {
    return oPosition;
}

public static void setOPosition(Point pos) {
    oPosition = pos;
}

public static Point getEtalonPosition() {
```

```
        return etalonPosition;
    }

    public static void setEtalonPosition(Point pos) {
        etalonPosition = pos;
    }

    public static Point getFirstCornerUV() {
        return firstCornerUV;
    }

    public static Point getFirstCornerIR() {
        return firstCornerIR;
    }

    public static Point getFirstCornerPixelDimension() {
        return firstCornerPixelDimension;
    }

    public static Point getSecondCornerPixelDimension() {
        return secondCornerPixelDimension;
    }

    public static void setFirstCornerUV(Point pos) {
        firstCornerUV = pos;
        maxPositionUV = null;
        minPositionUV = null;
    }

    public static void setFirstCornerIR(Point pos) {
        if (pos != null) {
            if (pos.x < 0)
                pos.x = 0;
            if (pos.x > IR_X_SIZE)
                pos.x = IR_X_SIZE;
            if (pos.y < 0)
                pos.y = 0;
            if (pos.y > IR_Y_SIZE)
                pos.y = IR_Y_SIZE;
        }
        firstCornerIR = pos;
        minPositionIR = null;
        maxPositionIR = null;
    }

    public static void setFirstCornerPixelDimension(Point pos) {
        firstCornerPixelDimension = pos;
        secondCornerPixelDimension = null;
        lungimeZonaMasurarePixel = null;
    }

    public static Point getSecondCornerUV() {
        return secondCornerUV;
    }

    public static Point getSecondCornerIR() {
        return secondCornerIR;
    }

    public static void setSecondCornerUV(Point pos) {
        secondCornerUV = pos;
        maxPositionUV = null;
    }
}
```

```
        minPositionUV = null;
    }

    public static void setSecondCornerIR(Point pos) {
        if (pos != null) {
            if (pos.x < 0)
                pos.x = 0;
            if (pos.x > IR_X_SIZE)
                pos.x = IR_X_SIZE;
            if (pos.y < 0)
                pos.y = 0;
            if (pos.y > IR_Y_SIZE)
                pos.y = IR_Y_SIZE;
        }
        secondCornerIR = pos;
        minPositionIR = null;
        maxPositionIR = null;
    }

    public static void setSecondCornerPixelDimension(Point pos) {
        pos.y = firstCornerPixelDimension.y;
        secondCornerPixelDimension = pos;
        maxPositionUV = null;
        fluxValue = 0;
    }

    public static int getOValue() {
        return oValue;
    }

    public static void setOValue(int value) {
        oValue = value;
    }

    public static int getVPosition() {
        return vPosition;
    }

    public static void setVPosition(int pos) {
        vPosition = pos;
    }

    public static double getMaxValueUV() {
        if (maxPositionUV == null) {
            maxValueUV = 0;
            getMaxPositionUV();
        }
        return maxValueUV;
    }

    public static double getMinValueUV() {
        if (minPositionUV == null) {
            minValueUV = 0;
            getMinPositionUV();
        }
        return minValueUV;
    }

    public static double getMaxValueIR() {
        if (maxPositionIR == null) {
            maxValueIR = 0;
        }
    }
}
```

```
        getMaxPositionIR();
    }
    return maxValueIR;
}

public static double getMinValueIR() {
    if (minPositionIR == null) {
        minValueIR = 0;
        getMinPositionIR();
    }
    return minValueIR;
}

public static Point getMaxPositionUV() {
    if (maxPositionUV != null)
        return maxPositionUV;

    if (Util.getFirstCornerUV() == null || Util.getSecondCornerUV()
== null) {
        return null;
    }

    int minX = Util.getFirstCornerUV().x;
    if (minX > Util.getSecondCornerUV().x) {
        minX = Util.getSecondCornerUV().x;
    }

    int maxX = Util.getFirstCornerUV().x;
    if (maxX < Util.getSecondCornerUV().x) {
        maxX = Util.getSecondCornerUV().x;
    }

    int minY = Util.getFirstCornerUV().y;
    if (minY > Util.getSecondCornerUV().y) {
        minY = Util.getSecondCornerUV().y;
    }

    int maxY = Util.getFirstCornerUV().y;
    if (maxY < Util.getSecondCornerUV().y) {
        maxY = Util.getSecondCornerUV().y;
    }

    maxValueUV = 0;
    double[][] data = processor.getResult();
    for (int i = minX; i < maxX; i++) {
        for (int j = minY; j < maxY; j++) {
            if (maxValueUV < data[i][j]) {
                maxValueUV = data[i][j];
                maxPositionUV = new Point(i, j);
            }
        }
    }

    return maxPositionUV;
}

public static Point getMaxPositionIR() {
    if (maxPositionIR != null)
        return maxPositionIR;

    if (Util.getFirstCornerIR() == null || Util.getSecondCornerIR()
== null) {
```

```

        return null;
    }

    int minX = Util.getFirstCornerIR().x;
    if (minX > Util.getSecondCornerIR().x) {
        minX = Util.getSecondCornerIR().x;
    }

    int maxX = Util.getFirstCornerIR().x;
    if (maxX < Util.getSecondCornerIR().x) {
        maxX = Util.getSecondCornerIR().x;
    }

    int minY = Util.getFirstCornerIR().y;
    if (minY > Util.getSecondCornerIR().y) {
        minY = Util.getSecondCornerIR().y;
    }

    int maxY = Util.getFirstCornerIR().y;
    if (maxY < Util.getSecondCornerIR().y) {
        maxY = Util.getSecondCornerIR().y;
    }

    maxValueIR = Integer.MIN_VALUE;

    int[][] data = getIRDataMatrix();

    maxPositionIR = Util.getFirstCornerIR();
    for (int i = minX; i < maxX; i++) {
        for (int j = minY; j < maxY; j++) {
            if (maxValueIR < data[i][j]) {
                maxValueIR = data[i][j];
                maxPositionIR = new Point(i, j);
            }
        }
    }

    return maxPositionIR;
}

public static int[][] getIRDataMatrix() {
    if (irDataMatrix != null)
        return irDataMatrix;
    irDataMatrix = new int[Util.IR_X_SIZE][Util.IR_Y_SIZE];

    int pixelCount = 0;
    for (int j = 0; j < Util.IR_Y_SIZE; j++) {
        for (int i = 0; i < Util.IR_X_SIZE; i++, pixelCount++) {
            irDataMatrix[i][j] = Util.irData[pixelCount];
        }
    }
    return irDataMatrix;
}

public static Point getMinPositionUV() {
    if (minPositionUV != null)
        return minPositionUV;

    if (Util.getFirstCornerUV() == null || Util.getSecondCornerUV()
== null) {
        return null;
    }
}

```

```
    }

    int minX = Util.getFirstCornerUV().x;
    if (minX > Util.getSecondCornerUV().x) {
        minX = Util.getSecondCornerUV().x;
    }

    int maxX = Util.getFirstCornerUV().x;
    if (maxX < Util.getSecondCornerUV().x) {
        maxX = Util.getSecondCornerUV().x;
    }

    int minY = Util.getFirstCornerUV().y;
    if (minY > Util.getSecondCornerUV().y) {
        minY = Util.getSecondCornerUV().y;
    }

    int maxY = Util.getFirstCornerUV().y;
    if (maxY < Util.getSecondCornerUV().y) {
        maxY = Util.getSecondCornerUV().y;
    }

    double[][] data = processor.getResult();

    minValueUV = Double.MAX_VALUE;
    for (int i = minX; i < maxX; i++) {
        for (int j = minY; j < maxY; j++) {
            if (minValueUV > data[i][j]) {
                minValueUV = data[i][j];
                minPositionUV = new Point(i, j);
            }
        }
    }

    return minPositionUV;
}

public static Point getMinPositionIR() {
    if (minPositionIR != null)
        return minPositionIR;

    if (Util.getFirstCornerIR() == null || Util.getSecondCornerIR()
== null) {
        return null;
    }

    int minX = Util.getFirstCornerIR().x;
    if (minX > Util.getSecondCornerIR().x) {
        minX = Util.getSecondCornerIR().x;
    }

    int maxX = Util.getFirstCornerIR().x;
    if (maxX < Util.getSecondCornerIR().x) {
        maxX = Util.getSecondCornerIR().x;
    }

    int minY = Util.getFirstCornerIR().y;
    if (minY > Util.getSecondCornerIR().y) {
        minY = Util.getSecondCornerIR().y;
    }

    int maxY = Util.getFirstCornerIR().y;
```

```

        if (maxY < Util.getSecondCornerIR().y) {
            maxY = Util.getSecondCornerIR().y;
        }

        minValueIR = Integer.MAX_VALUE;

        int[][] data = getIRDataMatrix();

        for (int i = minX; i < maxX; i++) {
            for (int j = minY; j < maxY; j++) {
                if (minValueIR > data[i][j]) {
                    minValueIR = data[i][j];
                    minPositionIR = new Point(i, j);
                }
            }
        }

        return minPositionIR;
    }

    public static void setEtalonValue(int intValue) {
        etalonValue = intValue;
    }

    public static int getEtalonValue() {
        return etalonValue;
    }

    public static double getCoefficient() {
        if (etalonValue == 0)
            return 0.0;
        if (etalonPosition == null)
            return 0.0;
        return (double) etalonValue
            /
processor.getPointValue(etalonPosition.x,
                        etalonPosition.y);
    }

    public static Point getCenterLocation(JFrame frame) {
        if (frame == null)
            return new Point(100, 100);
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = frame.getSize();

        if (frameSize.height > screenSize.height) {
            frameSize.height = screenSize.height;
        }
        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;
        }
        return new Point((screenSize.width - frameSize.width) / 2,
            (screenSize.height - frameSize.height) / 2);
    }

    public static JFrame getMainFrame() {
        return mainFrame;
    }

    public static void setMainFrame(MainWindow frame) {

```

```

        mainFrame = frame;
    }

    public static void setImageGrayRation(double ratio2) {
        ratio = ratio2;
    }

    public static double getImageGrayRatio() {
        return ratio;
    }

    public static void processFolder() throws Exception {
        retrieveLocation();
        processIRImage();
        getProcessor().startProcessing();
    }

    private static void retrieveLocation() throws Exception {
        String fileName = getFolder() + "/" + LOCATION_TXT;

        BufferedReader input = new BufferedReader(new
FileReader(fileName));
        try {
            setLocation(getLine(input));
            date = getLine(input);
            etalonValue = Integer.parseInt(getLine(input));
            String oPositionString = getLine(input);
            oPosition = new Point();
            oPosition.x =
Integer.parseInt(oPositionString.substring(0,
oPositionString.indexOf("x")));
            oPosition.y = Integer.parseInt(oPositionString
.substring(oPositionString.indexOf("x") +
1));

            try {
                String line = getLine(input);
                if (line != null)
                    vitezaVant = Double.parseDouble(line);
            } catch (Exception ignored) {
            }
        } catch (Exception e) {
            throw new Exception("Invalid location.txt file format.");
        }
    }

    private static String getLine(BufferedReader input) throws IOException {
        String line = "#";
        while (line != null && line.startsWith("#")) {
            line = input.readLine();
        }
        return line;
    }

    public static void setLocation(String location) {
        Util.location = location;
    }

    public static String getLocation() {
        return location;
    }

    public static String getFolder() {

```



```

        return folder;
    }

    public static void setFolder(String folderName) {
        folder = folderName;
    }

    public static String getDimensiunePixel() {
        String dimensiune = "-";

        if (firstCornerPixelDimension != null
            && secondCornerPixelDimension != null
            && lungimeZonaMasurarePixel != null) {
            double value = round(getDimensiunePixelValue());
            dimensiune = "" + value;
        }

        return dimensiune;
    }

    public static double getDimensiunePixelValue() {
        double dimensiune = 0.0;

        if (firstCornerPixelDimension != null
            && secondCornerPixelDimension != null
            && lungimeZonaMasurarePixel != null) {
            dimensiune = ((double) lungimeZonaMasurarePixel /
(double) Math
                .abs(firstCornerPixelDimension.x
                    -
secondCornerPixelDimension.x));
        }

        return dimensiune;
    }

    public static void setDimensiunePixel(int value) {
        lungimeZonaMasurarePixel = value;
    }

    public static String getMaxValueAsString() {
        // Util.getMaxValue() * Util.getCoefficient();
        return "-";
    }

    public static String getIntersectionValue() {
        if (Util.getProcessor() == null
            || Util.getProcessor().getResult() == null) {
            return "-";
        }
        return ""
            +
            Util.getProcessor().getResult()[Util.getVPosition()][Util
                .getHPosition()];
    }

    public static boolean shouldEliminate(int i, int j) {
        double report = ((double) Util.getClearSkyReport() / 50);
        if (report == 0.0) {
            report = 0.0001;
        }
    }

```

```
report);
int minInterval = (int) (Util.getOValue() - Util.getOValue() /
report);
int maxInterval = (int) (Util.getOValue() + Util.getOValue() /

double[][] data = processor.getResult();

double value = data[i][j];
int noOfValues = 1;
if (i > 1) {
    noOfValues++;
    value += data[i - 1][j];
}
if (j > 1) {
    noOfValues++;
    value += data[i][j - 1];
}
if (i > 2) {
    noOfValues++;
    value += data[i - 2][j];
}
if (j > 2) {
    noOfValues++;
    value += data[i][j - 2];
}
if (i > 2 && j < UV_Y_SIZE - 1) {
    noOfValues++;
    value += data[i - 2][j + 1];
}
if (j > 2 && i < UV_X_SIZE - 1) {
    noOfValues++;
    value += data[i + 1][j - 2];
}
if (i < UV_X_SIZE - 1) {
    noOfValues++;
    value += data[i + 1][j];
}
if (j < UV_Y_SIZE - 1) {
    noOfValues++;
    value += data[i][j + 1];
}
if (i < UV_X_SIZE - 2) {
    noOfValues++;
    value += data[i + 2][j];
}
if (j < UV_Y_SIZE - 2) {
    noOfValues++;
    value += data[i][j + 2];
}
if (i < UV_X_SIZE - 2 && j < UV_Y_SIZE - 1) {
    noOfValues++;
    value += data[i + 2][j + 1];
}
if (j < UV_Y_SIZE - 2 && i < UV_X_SIZE - 1) {
    noOfValues++;
    value += data[i + 1][j + 2];
}
}
if (noOfValues > 0) {
    value = value / noOfValues;
}

if (value > minInterval && value < maxInterval) {
```

```
        return true;
    }
    return false;
}

public static int getMasuratoarePctEtalon() {
    if (getEtalonPosition() == null)
        return 0;
    return (int) getValue(getEtalonPosition().x,
getEtalonPosition().y);
}

public static double getValue(int x, int y) {
    if (processor.getResult() == null)
        return 0.0;
    double data[][] = processor.getResult();

    return getValue(data, x, y);
}

public static double getValue(double data[][], int x, int y) {
    return (data[x][y] + data[x][y + 1] + data[x][y - 1] + data[x -
1][y]
        + data[x - 1][y + 1] + data[x - 1][y - 1] + data[x
+ 1][y]
        + data[x + 1][y + 1] + data[x + 1][y - 1]) / 9;
}

public static void setFirstPointFluxSection(Point point) {
    fluxSectionFirstPoint = point;
    fluxSectionSecondPoint = null;
    fluxValue = 0;
}

public static void setSecondPointFluxSection(Point point) {
    fluxSectionSecondPoint = point;
    fluxValue = 0;
}

public static boolean isFluxSection(int x, int y) {
    if (isFluxLine(x, y))
        return true;
    if (isFluxLine(x + 1, y))
        return true;
    if (isFluxLine(x, y + 1))
        return true;
    if (isFluxLine(x + 1, y + 1))
        return true;
    return false;
}

private static boolean isFluxLine(int x, int y) {
    Point p1 = Util.fluxSectionFirstPoint;
    Point p2 = Util.fluxSectionSecondPoint;
    if (p1 == null)
        return false;
    if (p2 == null)
        return false;

    if (p1.x - p2.x == 0 && p1.y - p2.y != 0)
        return true;
}
```

```
        if (p1.x - p2.x != 0 && p1.y - p2.y == 0)
            return true;
        if (p1.x - p2.x == 0)
            return true;
        if (p1.y - p2.y == 0)
            return false;

        double f1 = ((double) x - (double) p1.x)
                    / ((double) p2.x - (double) p1.x);
        double f2 = ((double) y - (double) p1.y)
                    / ((double) p2.y - (double) p1.y);

        f1 = round(f1);
        f2 = round(f2);

        if (f1 == f2)
            return true;

        return false;
    }

    public static String getFlux() {
        String flux = "-";

        if (getFluxValue() == 0)
            return flux;
        return "" + getFluxValue();
    }

    public static int getFluxValue() {
        if (fluxValue != 0)
            return fluxValue;

        Point p1 = Util.fluxSectionFirstPoint;
        Point p2 = Util.fluxSectionSecondPoint;
        if (p1 == null)
            return fluxValue;
        if (p2 == null)
            return fluxValue;

        double[][] data = processor.getResult();
        if (data == null)
            return fluxValue;

        double pixelDim = getDimensiunePixelValue();
        if (pixelDim == 0)
            return fluxValue;

        int minX = 0;
        int minY = 0;
        int maxX = 0;
        int maxY = 0;
        if (p1.x < p2.x) {
            minX = p1.x;
            maxX = p2.x;
        } else {
            minX = p2.x;
            maxX = p1.x;
        }
        if (p1.y < p2.y) {
            minY = p1.y;
            maxY = p2.y;
        }
    }
}
```

```

        } else {
            minY = p2.y;
            maxY = p1.y;
        }

        double sum = 0;
        int noOfPoints = 0;
        synchronized (fluxValue) {
            fluxValue = 0;
            for (int i = minX; i < maxX; i++) {
                for (int j = minY; j < maxY; j++) {
                    if (!isFluxLine(i, j))
                        continue;

                    if (shouldEliminate(i, j))
                        continue;
                    double value = (int) data[i][j] *
pixelDim;
                    // double v = getValue(i, j) * pixelDim;
                    fluxValue = fluxValue.intValue() + (int)
value;

                    noOfPoints++;
                    sum += data[i][j];
                }
            }
        }
        double mediaConc = sum / noOfPoints;
        double dist = noOfPoints * getDimensiunePixelValue();
        fluxViteza = mediaConc * dist * vitezaVant;

        // distanta este in m
        fluxViteza = fluxViteza / 100;

        // ppm in kg/m3
        fluxViteza = fluxViteza / 1000;

        // change from cm to m
        fluxValue = fluxValue / 100;
        return fluxValue;
    }

    public static void setClearSky(boolean selected) {
        clearSky = selected;
    }

    public static boolean isClearSky() {
        return clearSky;
    }

    public static void setClearSkyReport(int value) {
        clearSkyReport = value;
    }

    public static int getClearSkyReport() {
        return clearSkyReport;
    }

    public static void setGraficIntersectie(boolean
jCheckBoxGraficIntersectie) {
        graficIntersectie = jCheckBoxGraficIntersectie;
    }

```

```
public static boolean isGraficIntersectie() {
    return graficIntersectie;
}

public static void setGraficMinim(boolean jCheckBoxGraficIntersectie) {
    graficMinim = jCheckBoxGraficIntersectie;
}

public static boolean isGraficMinim() {
    return graficMinim;
}

public static void setGraficMaxim(boolean jCheckBoxGraficIntersectie) {
    graficMaxim = jCheckBoxGraficIntersectie;
}

public static boolean isGraficMaxim() {
    return graficMaxim;
}

public static void setIntervalHistogramaIR(int value) {
    intervalHistogramaIR = value;
}

public static int getIntervalHistogramaIR() {
    return intervalHistogramaIR;
}

public static void setIntervalHistogramaUV(int value) {
    intervalHistogramaUV = value;
}

public static int getIntervalHistogramaUV() {
    return intervalHistogramaUV;
}

public static String readDateObservation(String fileName, String
headerKey)
    throws FileNotFoundException, IOException {
    RandomAccessFitsFile file = new RandomAccessFitsFile(
        new RandomAccessFile(fileName, "r"));
    FitsHDU hdu = null;
    hdu = file.getHDU(0);

    return hdu.getHeader().card(headerKey).value().toString();
}

public static double[][] readFile(String fileName)
    throws FileNotFoundException, IOException {
    RandomAccessFitsFile file = new RandomAccessFitsFile(
        new RandomAccessFile(fileName, "r"));
    FitsHDU hdu = null;
    hdu = file.getHDU(0);

    FitsImageData fileData = (FitsImageData) hdu.getData();

    double[][] doubleData = new double[UV_X_SIZE][UV_X_SIZE];
    fileData.goToPixel();

    int[] pointer = new int[2];
    int col = 0;
```

```
int row = 0;
for (int countPixel = 0; countPixel < fileData.pixelCount();
countPixel++) {
    double d = fileData.getNextPixel();
    doubleData[row][col] = d;
    row++;
    if (row >= UV_X_SIZE) {
        row = 0;
        col++;
        if (col == UV_X_SIZE) {
            break;
        }
    }
    pointer[0] = col;
    pointer[1] = row;
    fileData.goToPixel(pointer);
}

// data.put(fileName, doubleData);
file = null;
hdu = null;
fileData = null;
return doubleData;
}
public static void setVitezaVant(double vitezaVant2) {
    Util.vitezaVant = vitezaVant2;
    fluxValue = 0;
}
public static String getVitezaVant() {
    if (vitezaVant == 0)
        return "-";
    return "    " + round(vitezaVant);}
```

Bibliografie

- [1]. Baumbach G., Air quality control, Ed. Springer, Berlin, 1996, ISBN 3-540-57992-3
- [2]. Beral E., Zapan M., Chimie anorganică, Editura Tehnică, București, 1977
- [3]. Bluth G.J.S. , Shannon J.M., Watson I.M., Prata A.J., Realmuto V.J., Development of an ultra-violet digital camera for volcanic SO₂ imaging, Journal of Volcanology and Geothermal Research 161 (2007) 47–56
- [4]. Bran F., Ioan I. , Globalizarea și mediul, Ed. Universitara, ISBN 978-973-749-513-6
- [5]. Brawn I., Probleme globale ale omenirii. Starea lumii, Ed. Tehnică, București, 1992
- [6]. Calhun Y., Seideman D., Air quality, Ed. Chelsea House Publishers, 2005, ISBN 978-143-810-228-3
- [7]. Caltabiano T., Romano R., Budetta G., SO₂ flux measurement at Mount Etna(Sicily), Journal of Geophysic Reasearch, 1994
- [8]. Caluianu S.,Cociorva S., Măsurarea și controlul poluării atmosferei, Ed.Matrixrom, București, 1999, ISBN 973-9390-67-6
- [9]. Călin L., Lonțiș N., Popescu F., **Nisulescu G. C.**, Bria N., Consideration on the impact of air – gas mixture burning corn drying process, Proceeding of the 6th International Conference Integrated Systems for agri food production, SIPA'09, Editura ORIZONTURI UNIVERSITARE, Nyiregyhaza, Hungary, November, 2009, pp.161-167, ISBN: 978-973-638-449-3
- [10]. Călinoiu D. G., Ionel I., Popescu F., Trif-Tordai G., **Nisulescu C.**, Study of microphysical properties of aerosol particles using sun photometer, International conference "Knowledge, culture, science the fundament of quality of life in society" , Timisoara, Romania, Noiembrie 2010.
- [11]. Călinoiu D. G., Ionel I., Popescu F., Trif-Tordai G., **Nisulescu C.**, Characteristics of aerosol types from sun photometer measurements, International Conference Integrated Systems for Agro Food Production, SIPA'11, Editura Laszlo Sikolya, Gabor Pay, Nyiregyhaza, Hungary, November 10-12, 2011, pp.6-10,ISBN: 978-606-569-312-8
- [12]. Călinoiu D. G., Ionel I., Popescu F., Trif-Tordai G., **Nisulescu C.**, Vetreș I., Recent sun photometry measurements in Romania, as part of the AERONET Network, Termotehnica, ISSN-L: 1222-4057, ISSN (online): 2247-1871, Ed. AGIR, Romania, Nr.2, 2011, pp. 34-37
- [13]. Ciobotaru V., Socolescu A.M., Poluarea și protecția mediului, Ed. Economică, 2009, ISBN 978-973-709-409-4
- [14]. Ciucanu I., Metode fizico-chimice de analiză, Ed. Mirton, Timișoara, 1995
- [15]. Constantinescu G.C., Roșca I., Negoiu M., Chimie anorganică, Ed. Tehnică, București, 1986
- [16]. Corega C., Todică M., Săndulache V., Aștilean S., Probleme de fizică, Ed. Facla, Timișoara, ISBN 973-36-0090-3
- [17]. Dalton M. P., Watson I. M., Nadeau P. A., Werner C., Morrow W., Shannon J. M., *Assessment of the UV camera sulfur dioxide retrieval for point source plumes*, Journal of Volcanology and Geothermal Research 188 (2009) 358–366

- [18]. Davis M., Connor O., Chimie-experiență și principii, Ed. Științifică și Enciclopedică, București, 1983
- [19]. December J., O introducere în Java și HotJava, Ed. Teora, ISBN 973-601-654-4
- [20]. Dissescu C.A., Luca I., Tudor M., Fizică și climatologie agricolă, Ed. Didactică și Pedagogică, 1971
- [21]. Dobjanschi Lelia - Contribuții privind aportul centralei pe cărbune ROMAG TERMO la poluarea aerului și măsuri pentru reducerea acesteia, Teză de doctorat, Universitatea Politehnica Timișoara, 2007
- [22]. Drăgulescu C., Petrovici E., Introducere în chimia anorganică modernă, Ed. Facla, Timișoara, 1973
- [23]. Edmonds M., Herd R.A., Galle B., Oppenheimer C., Automated, high time-resolution, measurements of SO₂ flux at Soufriere Hills Volcano, Montserrat, Bulletin of Volcanology, volume 65, pp.578-586, 2003
- [24]. Galle B., Oppenheimer C., Geyer A., McGonigle A. J. S., Edmonds M., Horrocks L., A miniaturized ultraviolet spectrometer for remote sensing of SO₂ fluxes: A new tool for volcano surveillance, Journal of Volcanology and Geothermal Research, Vol. 119, Issue: 1-4, Pages: 241-254, 2003
- [25]. Godish T., Air quality 4th edition, Ed. Lewis Publishers, 2004, ISBN 978-156-670-586-8
- [26]. Harrop O. D., Air quality assessment and management, Ed. Taylor and Francis, 2002, ISBN 0-415-23410-7
- [27]. Hites R.A., Elements of environmental chemistry, Ed. John Wiley & Sons, Inc., Hoboken, New Jersey, 2007, ISBN 978-047-199815-0
- [28]. Horton K.A., Williams-Jones G., Garbeil H., Mouginiis-Mark P., Porter J.N., Elias T., Sutton A.J., Real time measurements of volcanic SO₂ emissions: validation of a new UV correlation spectrometer, Bulletin of Volcanology 68, 323-327, 2006
- [29]. Hristev A., Probleme de fizică, Ed. Icar, București, 1981, ISBN 973-606-159-0
- [30]. Iacobescu S., Chimie tehnologică generală, Ed. Universității din Craiova, 1972
- [31]. Ifrim S., Roșca I., Chimie generală, Ed. Tehnică, București, 1989, ISBN 973-31-0134-6
- [32]. Ionel I., Dispersarea noxelor-Teorie și aplicații, Ed. Politehnica, Timișoara, 2000, ISBN 973-9389-58-9
- [33]. Ionel, I., Ungureanu, C., Termoenergetica și mediul, Ed. Tehnică, București, 1996
- [34]. Iordache V., Ardelean F., Ecologie și protecția mediului, Ed. Matrixrom, ISBN 978-973-755-255-6
- [35]. Kantzas E. P., McGonigle A.J.S., Ground Based Ultraviolet Remote Sensing of Volcanic Gas Plumes, Sensors 2008, 8, 1559-1574
- [36]. Kantzas E.P., McGonigle A.J.S., Tamburello G., Aiuppa A., Bryant R.G., Protocols for UV camera volcanic SO₂ measurements, Journal of Volcanology and Geothermal Research, 194, 2010, 55-60
- [37]. Kearney C. S., Dean K., Realmuto V. J., Watson I. M., Dehn J., Prata F., Observations of SO₂ production and transport from Bezymianny volcano, Kamchatka using the MODerate resolution Infrared Spectroradiometer

- (MODIS), International Journal of Remote Sensing Vol. 29, No. 22, 20 November 2008, 6647–6665
- [38]. Kekedy L., Chimie analitică calitativă, Ed. Scrisul românesc, Craiova, 1982
- [39]. Kern C., Kick F., Lubcke P., Vogel L., Wohrbach M., Platt U., Theoretical description of functionality, applications, and limitations of SO₂ cameras for the remote sensing of volcanic plumes, Atmospheric Measurement Techniques, 3, 733–749, 2010
- [40]. Kern C., Deutschmann T., Vogel L., Wöhrbach L., Wagner T., Platt U., Radiative transfer corrections for accurate spectroscopic measurements of volcanic gas emissions, Bulletin of Volcanology DOI 10.1007/s00445-009-0313-7
- [41]. Kim Y.J., Platt U., Advanced Environmental Monitoring, Ed. Springer, ISBN 978-1-4020-6364-0
- [42]. Kleibohmer W., Environmental Analysis, Ed. Elsevier, Amsterdam, 2001, ISBN 0-444-50021-9
- [43]. Ashok Kumar, Air Quality, Sciyo, Rijeka, Croatia, 2010, ISBN 978-953-307-131-2
- [44]. Lelea D., Cioablă A.E., **Nisulescu C.**, Transfer de căldură cu aplicații în microcanale, Ed. Politehnica, Timișoara, 2011, ISBN 978-606-554-378-2
- [45]. Lelea D., **Nisulescu C.**, The micro-tube heat transfer and fluid flow of water based Al₂O₃ nanofluid with viscous dissipation, International Communications in Heat and Mass Transfer, Volume 38, Issue 6, Pages 704-710 (July 2011), ISSN: 0735-1933, Imprint: Elsevier
- [46]. Lemay L., Cadenhead R., Java 2 fără profesor, Ed. Teora, ISBN 973-20-0378-2
- [47]. Lixandru B., *Ecologie generală*, Ed. Eurobit, Timișoara, 2003, ISBN 973-620-032-9
- [48]. Manda D., Legătura fizicii cu viața, Ed. Didactică și Pedagogică, București, 1974
- [49]. Maroni M., Seifert B., Linvall T., Indoor air quality, Ed. Elsevier, Amsterdam, ISBN 0-444-81642-9
- [50]. Mayr, F., *Kesselbetriebstechnik*, Resch Verlag, 6. Auflage, 1994
- [51]. McGonigle A.J.S., Hilton D.R., Fischer T.P., Oppenheimer C., Plume velocity determination for volcanic SO₂ flux measurements, Geophysical research letters, VOL. 32, L11302, 4 PP., 2005
- [52]. Moffat A.J., Millan M.M., The applications of optical correlation techniques to the remote sensing of SO₂ plumes using sky light, Atmospheric Environment, Volume 5, Issue 8, 677-690, 1971
- [53]. Mori T., Burton M., *The SO₂ camera: A simple, fast and cheap method for ground-based imaging of SO₂ in volcanic plumes*, Geophysical research letters, VOL. 33, L24804, doi:10.1029/2006GL027916, 2006
- [54]. Mori T., Burton M. - Quantification of the gas mass emitted during single explosions on Stromboli with the SO₂ imaging camera, Journal of Volcanology and Geothermal Research
- [55]. Mori T., Mori T., Kazahaya K., Ohwada M., Hirabayashi J., Yoshikawa S., Effect of UV scattering on SO₂ emission rate measurements. Geophysical Research Letter, 2006, 33:L17315. doi:10.1029/2006GL026285
- [56]. Moussiopoulos N., Berge E., Bohler T., Leeuw F., Gronskai K.E. Mylona S., Tombrou M., Ambient air quality, pollutant dispersion and

- transport models, Ed. Office for Official Publications of the European Communities, 1997, ISBN 978-929-167-028-4
- [57]. Muntean O.I., *Ecologie și protecția mediului*, Ed. Emia, ISBN 978-973-753-053-5
- [58]. Muzio, L., Teague, M.: *Errors in Grab Sample Measurements of N₂O from Combustion Sources*, JAPCA, No. 39, 1989, p. 287-293.
- [59]. Nenițescu C. D., *Chimie generală*, Ed. Didactică și Pedagogică, București, 1972
- [60]. Nicu V., *Elemente de poluare și protecție a atmosferei*, Iași , 2000
- [61]. Nimmo, W., Hampartsoumian, E., Sedigh, K., Williams, A.: *Control of NO_x emissions by combustion-air staging; the measurement of NH₃, HCN, NO and N₂O concentrations in fuel-oil flames*, Journal of Institute of Energy, September, 1991, Vol. 64, p. 128-134.
- [62]. Nguyn, C.T., Galais, A., Fortunato, G., *Pollution imagery by optical interferometry :application to SO₂ gas*, Applied Optics, 34, 5398-5405, 1995
- [63]. **Nișulescu C.**, Ionel I., Popescu F., *Emission monitoring highlighted through remote sensing technique*, Proceedings of Optoelectronic Techniques for Environmental Monitoring, Magurele, September 28-30, 2011
- [64]. **Nișulescu G. C.**, Popescu F. *Relevant aspects of the real time monitoring for air pollution using a LIDAR system in Timisoara episode*, *Buletinul Științific al Universității "Politehnica" din Timișoara, Seria Mecanică*, ISSN: 1224- 6077, Ed. Politehnica Timișoara, Romania, vol. 54, no. 4, Dec. 2009, pp. 35-39
- [65]. **Nișulescu C.**, Calinoiu D., Timofte A., Boscornea A., Talianu C., *Diurnal variation of particulate matter in the proximity of Rovinari fossil-fuel power plant*, Proceedings of the 4th International Workshop on Optoelectronic Techniques for Environmental Monitoring, OTEM, ISSN: 2066-8651, Published by INOE, Bucuresti , Romania, Octombrie 2010, pp. 40 - 47
- [66]. **Nișulescu G. C.**, Ionel I., *Air pollution issues highlighted with LIDAR systems*, Proceedings of the International Workshop „Global and Regional Environmental Protection” G.L.O.R.E.P., ISBN: 978-606-554-210-5, Editura Politehnica, Timisoara, Romania, vol.1, Noiembrie 2010, pp.290-293.
- [67]. **Nișulescu C.**, Ionel I. Calinoiu D., Vetres I., *Air pollution monitoring in a town nearby power plant*, *Advances in Biology, Bioengineering and Environment*, ISSN:1792 - 7412, 1792 - 7374, 1792 - 7439, ISBN: 978 - 960 - 474 - 261 -5, Ed. WSEAS Press, Vouliagmeni, Athens, Greece, December 29-31, 2010, pp. 181-185
- [68]. **Nișulescu C.**, Calinoiu D., Timofte A., Boscornea A., Talianu C., *Diurnal variation of particulate matter in the proximity of Rovinari fossil-fuel power plant*, *Environmental Engineering and Management Journal*, ISSN 1582 - 9596 , Ed. Ecoyone, Iasi, January 2011, Vol. 10, No. 1, pp.99 - 107
- [69]. **Nișulescu C.**, Ionel I., Călinoiu D., *Remote sensing measurement technique for SO₂ emissions*, Proceedings of the 35th Annual Congress of the American Romanian Academy of Arts and Science "Science and Art in the Informatics Era" ISBN 978-2-553-01596-0

- [70]. **Nișulescu G. C.**, Remote sensing technique for SO₂ emissions observation, Workshop nr.2 "Interdisciplinaritatea și managementul cercetării", Timisoara, 2011
- [71]. **Nișulescu C.**, Ionel I., Călinoiu D., New methods for determination SO₂ emissions, International Conference Integrated Systems for Agro Food Production, SIPA'11, Editura Laszlo Sikolya, Gabor Pay, Nyiregyhaza, Hungary, November 10-12, 2011, pp.6-10,ISBN: 978-606-569-312-8
- [72]. **Nișulescu G.C.**, Ionel I., Malan B., Dobrin M., Remote SO₂ Monitoring with UV Cameras for Stack Emissions, Revista de Chimie, September, 2012
- [73]. **Nișulescu C.**, Ionel I., Popescu F., Research concerning emissions monitoring by means of UV cameras, Journal of Optoelectronics and Advanced Materials, 2012
- [74]. Oniciu L., Coroziunea metalelor, Ed. Științifică și Enciclopedică, București, 1986
- [75]. Oniciu L., Chimie fizică, Ed. Didactică și Pedagogică, 1973
- [76]. Paduch, M., *Interpretationshilfe für die Olfaktrometrie*, VDI Berichte, Nr. 1059, 1993, p. 593-607.
- [77]. Page J. A., Robinson E. A., Brereton M. E., Chimie – Principii și Experiențe, Ed. Tehnică, București, 1973
- [78]. Pascariu A., Crișan M., Schimbarea globală a climei – Efectul de seră și distrugerea stratului de ozon
- [79]. Pietrzyk D. J., Frank C.W., Chimie Analitică, Ed. Tehnică, București, 1989,
- [80]. Pincovschi E., Baltă P., Floarea M., Avram R., Chimie tehnologică generală, Ed. Didactică și Pedagogică, București, 1977
- [81]. Pluschke P., Indoor Air Pollution, Ed. Springer, Berlin, ISBN 3-540-21098-9
- [82]. Popescu F., **Nisulescu C.**, Vetres I., Air quality simulation as investigation tool for urban development strategies, Buletinul Științific al Universității "Politehnica" din Timișoara, *Seria Mecanică*, ISSN: 1224- 6077, Ed. Politehnica Timișoara, Romania, vol. 55(69), no. 2, Decembrie 2010, pp. 68 -72
- [83]. Popescu M., Popescu M., Ecologie aplicată, Ed. Matrixrom,ISBN 973-685-183-4
- [84]. Postelnicu V., Coatu V., Mică enciclopedie matematică, Ed. Tehnică, București, 1971
- [85]. Prata A.J., Bernardo C., Retrieval of volcanic ash particle size, mass and optical depth from a ground-based thermal infrared camera, Journal of Volcanology and Geothermal Research 186 (2009) 91–107
- [86]. Prata A. J., Bernardo C., Retrieval of SO₂ from a ground-based thermal infrared imaging camera, Atmospheric Measurement Technique.
- [87]. Prata A. J., Satellite detection of hazardous volcanic clouds and the risk to global air traffic, Nat Hazards (2009) 51:303–324 DOI 10.1007/s11069-008-9273-z
- [88]. Prata A.J., Gangale G., Clarisse L., Karagulian F., Ash and sulfur dioxide in the 2008 eruptions of Okmok and Kasatochi: Insights from high spectral resolution satellite measurements, Journal of Geophysical Research, vol. 115, D00L18, doi:10.1029/2009JD013556, 2010
- [89]. Racoceanu C., Căpățână C., Emisiile de noxe ale centralelor termoelectrice, Ed. Matrixrom, ISBN 973-685-882-0

- [90]. Sandsten J., Edner H., Svanberg S., Gas imaging by infrared gas-correlation spectrometry, *Optics Letters* 21, 1945-1947, 1996
- [91]. Shaw J. A., Nugent P. W., Pust N. J., Thurairajah B., Mizutani K., Radiometric cloud imaging with an uncooled microbolometer thermal infrared camera
- [92]. Shaw J. A., Thurairajah B., Edqvist E., Mizutani K., *Infrared Cloud Imager Deployment at the North Slope of Alaska During Early 2002*, Twelfth ARM Science Team Meeting Proceedings, St. Petersburg, Florida, 2002
- [93]. Smith S., Toumi R., *Direct observation of cloud forcing by ground-based thermal imaging*, *GEOPHYSICAL RESEARCH LETTERS*, VOL. 35, L07814, doi:10.1029/2008GL033201, 2008
- [94]. Smith S., Toumi R., Measuring Cloud Cover and Brightness Temperature with a Ground-Based Thermal Infrared Camera, *Journal of applied meteorology and climatology* volume 47
- [95]. Sportisse B., *Fundamentals in Air Pollution*, Ed. Springer, 2008, ISBN 978-90-481-2970-6
- [96]. Stoica L., Constantinescu I., Alexandru R., Lupu I., Nașcu H., Onu P., *Chimie Generală și Analize Tehnice*, Ed. Didactică și Pedagogică, București, 1983
- [97]. Ștefan S., Nicolae D., Caian M., *Secretele aerosolului atmosferic în lumina laserilor*, Ed. Ars Docendi, București, 2008, ISBN 978-973-558-357-6
- [98]. Ștefan S., *Fizica aerosolului atmosferic*, Ed. All, Bucuresti, 1998, ISBN 973-9392-89-x
- [99]. Tamburello G., Kantzas E. P., McGonigle A.J.S., Aiuppa A., *Vulcamera: a program for measuring volcanic SO₂ using UV cameras*, *Annals of Geophysics*, 54, 2, 2011
- [100]. Tamburello G., McGonigle A.J.S., Kantzas E.P., Aiuppa A., Recent advances in ground-based ultraviolet remote sensing of volcanic SO₂ fluxes, *Annals of Geophysics*, 54,2,2011, doi:10.4401/ag-5179
- [101]. Tamburello G., Kantzas E.P., McGonigle A.J.S., Aiuppa A., Giudice G., UV camera measurements of fumarole field degassing, *Journal of Volcanology and Geothermal Research*, 199,2011, 47-52
- [102]. Tanasă S., Andrei S., Olaru C., *Java de 0 la expert*, Ed. Polirom, Iași, 2007, ISBN 978-973-46-0317-6
- [103]. Thurairajah B., Shaw J. A., *Infrared Cloud Imager Measurements of Cloud Statistics from the 2003 Cloudiness Intercomparison Campaign*, Fourteenth ARM Science Team Meeting Proceedings, Albuquerque, New Mexico, March 22-26, 2004
- [104]. Thurairajah B., Shaw J. A., *Cloud Statistics Measured With the Infrared Cloud Imager (ICI)*, *IEEE Transactions on Geoscience and Remote Sensing*, June 2005
- [105]. Williams-Jones G., Horton K.A. Elias T., Garbeil H., Mouginiis-Mark P.J., Sutton A.J., Harris A.J.L., Accurately measuring volcanic plume velocity with multiple UV spectrometers, *Bulletin of Volcanology*, volume 68, 338-332
- [106]. Wright J., *Environmental Chemistry*, Ed. Routledge, 2003, ISBN 0-415-22601-5

- [107]. Yamamoto H., Watson I. M., Phillips J. C., Bluth G. J., Rise dynamics and relative ash distribution in vulcanian eruption plumes at Santiaguito Volcano, Guatemala, revealed using an ultraviolet imaging camera, GEOPHYSICAL RESEARCH LETTERS, VOL. 35, L08314, doi:10.1029/2007GL032008, 2008
- [108]. ***, - www.testo.com, Testo 300 XXL, Manual de utilizare.
- [109]. ***, <http://arpmtm.anpm.ro>
- [110]. ***, www.oracle.com
- [111]. <http://burro.cwru.edu/Academics/Astr201/Atmosphere/structure.jpg>