

Optimizarea recunoașterii codurilor Data Matrix în mediul industrial

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea Politehnica din Timișoara
în domeniul Inginerie Electronică și Telecomunicații
de către

Ing. Ion-Cosmin Diță

Conducător științific: prof.univ.dr.ing Marius Oteșteanu
Referenți științifici: prof.univ.dr.ing Corneliu Rusu
prof.univ.dr.ing Franz Quint
prof.univ.dr.ing Ioan Naforniță

Ziua susținerii tezei: 16.03.2012

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. Inginerie Chimică | 10. Știința Calculatoarelor |
| 5. Inginerie Civilă | 11. Știința și Ingineria Materialelor |
| 6. Inginerie Electrică | |

Universitatea Politehnică 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 Politehnică - 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 Politehnice 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

Această teză de doctorat a fost parțial sprijinită de grantul POSDRU/6/1.5/S/13, ID6998, finanțat din Fondul Social European „Investește în oameni!”, prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007 - 2013.

Teza de doctorat a fost elaborată pe parcursul activității mele în cadrul Departamentului de Comunicații al Facultății de Electronică și Telecomunicații, Universitatea „Politehnica” din Timișoara și pe parcursul stagiului de doctorat desfășurat în cadrul Facultății de Electrotehnică și Informatică din cadrul Universității Științelor Aplicate din Karlsruhe, Germania.

La finalul acestei etape, gândurile și recunoștința mea se îndreaptă spre toți cei care m-au sprijinit în elaborarea și finalizarea tezei de doctorat.

Cu această ocazie doresc să-mi exprim mulțumirile domnului Prof. Dr. Ing. Marius Oteșteanu, conducătorul științific al tezei de doctorat, pentru competența cu care mi-a coordonat întreaga activitate desfășurată pe parcursul elaborării tezei precum și pentru suportul oferit în depășirea obstacolelor neprevăzute.

Mulțumesc domnului Prof. Dr. Ing. Franz Quint pentru îndrumarea și susținerea acordată în a elabora teza de doctorat în cadrul Facultății de Electrotehnică și Informatică, Universitatea Științelor Aplicate din Karlsruhe.

Doresc să mulțumesc pe această cale și domnului Prof. Dr. Ing. Vasile Gui pentru sfaturile utile și observațiile aduse în elaborarea și finalizarea acestei teze.

Mulțumesc membrilor Comisiei pentru evaluarea tezei de doctorat și pentru onoarea ce mi-o oferă prin acceptul de a participa la susținerea publică a lucrării.

Doresc să mulțumesc, de asemenea, colegilor din cadrul Departamentului de Comunicații al Facultății de Electronică și Telecomunicații, pentru atmosfera minunată de lucru oferită pe durata acestei perioade de doctorat.

Mulțumesc în mod special prietenului și colegului meu de doctorat, domnul Ing. Wolfgang Proß, pentru toată grija și atenția oferită, pe parcursul stagiului meu de doctorat în Karlsruhe.

Nu în ultimul rând, vreau să transmit mii de mulțumiri familiei mele, care m-a înțeles și sprijinit în toată munca mea.

Timișoara, martie 2012

Ion-Cosmin Diță

Tatălui meu.

Diță, Ion-Cosmin

Optimizarea recunoașterii codurilor Data Matrix în mediul industrial

Teze de doctorat ale UPT, Seria 7, Nr. 44, Editura Politehnică, 2012, 114 pagini, 101 figuri, 10 tabele.

ISSN: 1842-7014

ISBN: 978-606-554-465-9

Cuvinte cheie:

barcode, bidimensional codes, Data Matrix Code, industrial Data Matrix Code, pattern recognition, mean shift, Hough transform.

Rezumat:

Teza de doctorat este dedicată unei probleme de mare interes în amprente digitale și anume achiziția codurilor Data Matrix în mediul industrial. Problema constă în localizarea corectă a amprentei și estimarea caracteristicilor ei pentru a putea fi scanată. Metodele tradiționale de recunoaștere și scanare implementate pentru amprente de coduri Data Matrix standard sunt nepotrivite în cazul de față. Cercetările în această direcție au dus la dezvoltarea unei aplicații de recunoaștere a codurilor destinată în special codurilor Data Matrix industriale.

Analizând literatura de specialitate, s-a considerat că Algoritmii cu Translație la Medie (Mean Shift) ar putea să satisfacă aceste cereri. Încă de la prima implementare a acestuia s-a observat diferența pozitivă a rezultatelor. Pentru a demonstra aceasta, performanțele estimatorului Algoritmii cu Translație la Medie au fost evaluate comparativ cu Metoda Transformării Hough. Pe baza rezultatelor s-a demonstrat eficacitatea Algoritmii cu Translație la Medie în estimarea caracteristicilor amprentei codului Data Matrix industrial.

Pentru scanarea codului, inițial, s-a implementat metoda clasică vectorială pe baza orientării amprentei și distanței dintre module. Această metodă este rapidă și funcționează bine pentru suprafețele plane, unde modulele sunt distribuite uniform. Însă, această metodă întâmpină dificultăți în cazul materialelor curbe. De aceea, pe baza Matricei de transformare s-a implementat o nouă metodă de scanare care funcționează bine pentru toate tipurile de suprafețe. Pentru a evidenția avantajele acestei metode s-a realizat un studiu comparativ, analizând succesiv materiale reale cu suprafețe diferite: plane, cilindrice sau sferice.

Cuprins

1	Introducere	11
1.1	Motivație	11
1.2	Prezentare generală a etapelor procesului de recunoaștere	12
1.3	Structura tezei	13
1.4	Lucrări publicate	15
2	Codul Data Matrix	16
2.1	Codurile unidimensionale	16
2.2	Codurile bidimensionale	17
2.2.1	Codurile suprapuse	18
2.2.2	Codurile matriciale	18
2.3	Compararea codurilor	19
2.4	Codul Data Matrix ECC200	20
2.4.1	Structura generală	21
2.4.2	Caracteristici tehnice	21
2.4.2.1	Prezentarea formei amprentei	21
2.4.2.2	Mărimi și capacități de stocare	22
2.4.2.3	Regiunea datelor	23
2.4.2.4	Metode de corecție a erorilor	23
2.4.2.5	Algoritmul Reed-Solomon pentru corecția erorilor	24
2.5	Technici de inscripționare	24
2.5.1	Transferul termic	24
2.5.2	Imprimarea cu cerneală	25
2.5.3	Gravarea cu laser	25
2.5.4	Marcarea directă prin punctare sau frezare	26
2.6	Evaluarea tehnicilor de marcarea și a materialelor folosite	26
2.7	Standard pentru codurile industriale	28
2.8	Parametrii folosiți în verificare	29
3	Metode de recunoaștere a codului Data Matrix	31
3.1	Achiziția codului	31
3.2	Localizarea codului	31
3.3	Scanarea codului	32
3.4	Concluzii	33
4	Localizarea codurilor Data Matrix	34
4.1	Sisteme de localizare semi-automată	34
4.1.1	Prezentarea sistemului de localizare semi-automată	34
4.1.2	Interfața video	35
4.1.3	Localizarea amprentei Data Matrix	35
4.1.3.1	Binarizarea imaginii cu metoda lui Otsu cu prag dinamic-adaptiv	36
4.1.3.2	Recunoașterea formei amprentei Data Matrix folosind Morfologie matematică	38
4.1.4	Estimarea distanței dintre module și a suprafeței acestora folosind histograma	43

4.1.5	Selecția modulelor corecte	44
4.1.6	Calculul orientării amprentei folosind transformarea Hough	45
4.1.6.1	Transformarea Hough	45
4.1.6.2	Implementarea transformării Hough pentru codurile Data Matrix	47
4.1.7	Recunoașterea amprentei codului Data Matrix	48
4.1.7.1	Detecția colțurilor amprentei codului Data Matrix	48
4.1.7.2	Recunoașterea amprentei de identificare	48
4.1.7.3	Amprenta de sincronizare	49
4.1.8	Concluzii	50
4.2	Sisteme de localizare automată	52
4.2.1	Achiziția și etichetarea imaginii	53
4.2.2	Extragerea informațiilor amprentei codului Data Matrix folosind Algoritmul cu Translație la Medie	54
4.2.2.1	Algoritmul cu Translație la Medie	54
4.2.2.2	Implementarea Algoritmului cu Translație la Medie pentru calcularea orientării amprentei codului Data Matrix	55
4.2.2.3	Implementarea Algoritmului cu Translație la Medie pentru calcularea distanței dintre modulele amprentei și suprafața acestora	59
4.2.3	Recunoașterea Amprentei „L” a Codului Data Matrix	61
4.2.4	Concluzii	66
5	Comparația metodelor de localizare	71
5.1	Experiment	71
5.2	Rezultate	74
6	Scanarea Codului Data Matrix industrial	77
6.1	Suprafețe plane	77
6.1.1	Extragerea numărului de rânduri și de coloane ale amprentei codului Data Matrix	77
6.1.2	Scanarea și detecția modulelor codului Data Matrix	78
6.1.3	Concluzii	80
6.2	Suprafețe neplane	82
6.2.1	Scanarea marginilor amprentei și extragerea punctelor de control	82
6.2.2	Suprafețe curbe	85
6.2.3	Estimarea poziției modulelor amprentei cu Matricea de transformare geometrică	87
6.2.4	Detecția modulelor codului Data Matrix	90
6.2.5	Concluzii	91
7	Comparația metodelor de scanare	93
7.1	Experiment	93
7.1.1	Experiment realizat pe imagini reale	94
7.1.2	Experiment realizat pe imagini generate sintetic	97
7.2	Rezultate	99
8	Concluzii generale și contribuții	102
8.1	Concluzii	102
8.2	Contribuții	103
8.3	Direcții viitoare de cercetare	104

Listă de figuri

1.1	Etapele procesului de recunoaștere a codului Data Matrix industrial . . .	12
2.1	CodaBar	16
2.2	UPC-A, UPC-E, EAN, EAN-14	16
2.3	PostNet	17
2.4	PDF417	18
2.5	RSS Extins	18
2.6	MaxiCode	18
2.7	Aztec	19
2.8	Code 39, primul cod alfanumeric	19
2.9	ITF	19
2.10	PDF417, cod bidimensional	19
2.11	Codul Data Matrix, cod bidimensional	20
2.12	DMC - (a) Amprenta de identificare, (b) Amprenta de sincronizare, (c) Informația codată, (d) Zona de liniște	21
2.13	Cod Data Matrix pătratic și dreptunghiular	22
2.14	Cod Data Matrix imprimat cu cerneală	25
2.15	Cod Data Matrix marcat cu laser	25
2.16	Cod Data Matrix punctat	26
2.17	Imaginea unui Cod Data Matrix	27
2.18	Histograma imaginii DMC standard din Fig. 2.17(a)	27
2.19	Histograma imaginii DMC din Fig. 2.17	28
4.1	Schema bloc a sistemului de localizare	34
4.2	Procesul de localizare a amprentei Data Matrix	36
4.3	Niveluri de binarizare locale	37
4.4	Determinarea pragului de binarizare	38
4.5	Dilatarea imaginii Data Matrix	39
4.6	Erodarea imaginii Data Matrix	39
4.7	Închiderea imaginii codului Data Matrix (DMC)	40
4.8	Axa mare și axa mică a unui obiect din imagine	41
4.9	Calcularea unghiului dintre vectori	41
4.10	Cod Data Matrix industrial	42
4.11	Regiunea de interes - ROI	42
4.12	Calculul distanței dintre module	43
4.13	Histograma distanțelor	44
4.14	Histograma suprafețelor	44
4.15	Selecția modulelor corecte	45
4.16	r, θ - parametrii liniei	46
4.17	Reprezentarea acumulatorul transformării Hough	46
4.18	Orientarea amprentei codului Data Matrix	47
4.19	Transformarea Hough a imaginii codului Data Matrix	47
4.20	Detecția colțurilor amprentei codului Data Matrix	48
4.21	Recunoașterea amprentei codului Data Matrix	49
4.22	Cupru lucios, dimensiune amprentă $2 \times 2cm^2$	50
4.23	Aluminiu lucios, dimensiune amprentă $1,6 \times 1,6cm^2$	51
4.24	Fier acoperit cu pete de rugină, dimensiune amprentă $1,6 \times 1,6cm^2$	51

4.25	Metal vopsit, dimensiune amprentă $2 \times 2cm^2$	51
4.26	Sistem de localizare automată	52
4.27	Vecinătățile unui pixel	53
4.28	Vecinătăți de ordin 4 și 8	54
4.29	Etichetarea modulelor codului Data Matrix	54
4.30	Algoritmul cu Translație la Medie - principiu	55
4.31	Imaginea DMC binarizată	56
4.32	Algoritmul cu Translație la Medie	56
4.33	Funcția Nucleu	57
4.34	ATM - Iterația 1	57
4.35	ATM - Iterația 2	58
4.36	ATM - Iterația 3	58
4.37	ATM - Iterația 4	58
4.38	ATM - Iterația 5	59
4.39	Calcularea distanței dintre module	59
4.40	Algoritmul cu Translație la Medie implementat pentru calculul distanței dintre module	60
4.41	Algoritmul cu Translație la Medie implementat pentru calculul suprafeței modulelor	60
4.42	Viziune în spațiul 3-D a modulelor codului Data Matrix	61
4.43	Șablonul amprentei de identificare	63
4.44	Șablonul amprentei de sincronizare	64
4.45	Selecția amprentei DMC	65
4.46	Aluminiu, dimensiune amprentă $2 \times 2cm^2$	67
4.47	Cupru, dimensiune amprentă $2 \times 2cm^2$	68
4.48	Fier, dimensiune amprentă $1,2 \times 1,2cm^2$	69
4.49	Plastic, dimensiune amprentă $2 \times 2cm^2$	70
5.1	Descrierea experimentului	71
5.2	Abaterea maximă unui modul	72
5.3	Curba de repartiție Gauss a densității de probabilitate a erorii în spațiul imagine	72
5.4	Amprenta DMC generată sintetic, mărimea = 26×26 , $d = 11px$, $\alpha = 78^\circ$	73
5.5	Eroarea medie	76
6.1	Extragerea numărului de rânduri ale codului Data Matrix	78
6.2	Scanarea amprentei de identificare a codului Data Matrix	78
6.3	Ajustarea ferestrei de scanare	79
6.4	Procesul de scanare a codului Data Matrix	80
6.5	Cupru lucios, dimensiune amprentă $2 \times 2cm^2$	80
6.6	Aluminiu lucios, dimensiune amprentă $1,2 \times 1,2cm^2$	81
6.7	Fier acoperit cu pete de rugină, dimensiune amprentă $1,2 \times 1,2cm^2$	81
6.8	Metal vopsit, dimensiune amprentă $2 \times 2cm^2$	82
6.9	Scanarea modulelor marginilor amprentei codului Data Matrix	83
6.10	Ferastra de scanare ce conține două obiecte	84
6.11	Selectarea punctelor de control	84
6.12	Demonstrarea clasificării suprafețelor	85
6.13	Codul Data Matrix inscripționat pe un cilindru	86
6.14	Proiecția unui obiect în imagine	88
6.15	Scanare pe baza Matricei de transformare	90
6.16	Ex.1 - Cod Data Matrix cu erori de perspectivă	91

6.17 Ex.2 - Cod Data Matrix marcat pe suprafață cilindrică	92
6.18 Ex.3 - Cod Data Matrix marcat pe suprafață sferică	92
7.1 Descrierea experimentului	93
7.2 Aluminu-suprafață plană, dimensiune amprentă $1,6 \times 1,6 \text{ cm}^2$	95
7.3 Fier-suprafață plană, dimensiune amprentă $1,8 \times 1,8 \text{ cm}^2$	95
7.4 Cilindru-aluminu, diametrul = 10 cm, dimensiune amprentă $0,8 \times 0,8 \text{ cm}^2$	96
7.5 Cilindru-aluminu, diametrul = 5 cm, dimensiune amprentă $2 \times 2 \text{ cm}^2$	97
7.6 Cilindru-cupru, diametrul = 3,5 cm, dimensiune amprentă $1,8 \times 1,8 \text{ cm}^2$	97
7.7 Sferă-cupru, diametrul = 4,5 cm, dimensiune amprentă $2 \times 2 \text{ cm}^2$	98
7.8 Amprentă DMC cilindrică, mărime = 26×26 , $d = 19 \mu\text{m}$, $\alpha = 86^\circ$. Partea stângă scanare vectorială, partea dreaptă scanare matricială	100
7.9 Amprentă DMC sferică, Mărime = 26×26 , $d = 22 \mu\text{m}$, $\alpha = 73^\circ$. Partea stângă scanare vectorială, partea dreaptă scanare matricială	101

Listă de tabele

2.1	Clasificarea codurilor în funcție de eroare	20
2.2	Caracteristicile codului Data Matrix	22
2.3	Mărimea simbolului și numărului de regiuni de date	23
5.1	Experiment 1, Segment Parzen = 100 exemplare, Acumulator Hough = 90 eșantioane	75
5.2	Experiment 2, Segment Parzen = 1000 exemplare, Acumulator Hough = 900 eșantioane	75
5.3	Exp. 3, Segment Parzen = 10000 exemplare, Acumulator Hough = 9000 eșantioane	75
6.1	Clasificarea suprafețelor	85
7.1	Eroarea medie pătratică în cazul celor două metode de scanare	94
7.2	Eroarea medie pătratică în cazul celor două metode de scanare, pentru amprente DMC cilindrice (Fig. 7.8)	98
7.3	Eroarea medie pătratică în cazul celor două metode de scanare, pentru amprente DMC sferice (Fig. 7.9)	99

Capitolul 1

Introducere

Capitolul introductiv al tezei face o prezentare a problematicii achiziției codurilor Data Matrix în mediul industrial și plasează în context estimarea neparametrică robustă, problemă asupra căreia se concentrează o parte a contribuțiilor tezei. Primul paragraf prezintă considerații generale privind metodele de achiziție a codurilor Data Matrix. În paragrafele următoare sunt prezentate structura tezei pe capitole și lista de lucrări publicate pe parcursul activității de cercetare.

1.1 Motivație

Codurile Data Matrix au devenit tot mai comune prin tipărirea lor pe articole media, cum ar fi etichete și scrisori. Codul poate fi citit rapid de către un cititor de bare sau o cameră video, astfel permițând extragerea informațiilor obiectului marcat.

În scopuri de inginerie industrială, codurile Data Matrix pot fi marcate direct pe componente, prin diferite metode, cele mai comune sunt: cu jet de cerneală, prin punctare, marcarea cu laser, gravură și erodare chimică electrolică. Aceste metode oferă o marcă permanentă, ce durează cât viața componentei.

Codurile Data Matrix standard pot fi citite cu ajutorul scannerelor 2D, utilizând camere video de rezoluție mare sau în ultima vreme chiar și cu telefoanele mobile. Se pune problema citirii codurilor Data Matrix, ce sunt folosite în industrie marcate direct pe material, unde condițiile de mediu sunt nefavorabile.

În cazul codurilor Data Matrix standard, amprenta este tipărită pe o etichetă din hârtie, cu negru pe alb unde contrastul este foarte mare, iar citirea nu este o problemă. În cazul codurilor Data Matrix industriale, materialul nu este o etichetă de culoare albă, iar modulele nu sunt celule pătratice. Aici materialul poate fi de orice culoare, cu texturi diferite sau cu deteriorări, iar modulele sunt cavități mici în suprafața materialului. De cele mai multe ori, imaginile nu pot fi captate ortogonal, sau codurile sunt marcate pe suprafețe curbe. Acest lucru, combinat cu un contrast mic, ca urmare a diverselor materiale pe care sunt marcate și cu perturbații tipice în mediul industrial cum ar fi zgârieturi, rugină, picături de ulei, face localizarea și decodarea o sarcină dificilă.

În literatură s-au implementat mai multe metode pentru achiziția și recunoașterea codului, dar cele mai multe sunt implementate pentru codurile standard. În cazul codurilor folosite în mediul industrial aceste metode nu pot fi utilizate. De aceea, în această teză se propune o soluție completă de achiziție, localizare și scanare a codului ce poate confrunța condițiile nefavorabile create de mediul industrial.

Teza de doctorat își propune să studieze în ce măsură metodele de estimare neparametrică pot fi avantajoase în raport cu metodele utilizate curent în achiziția și recunoașterea codurilor Data Matrix. Se implementază aceste metode și se compară rezultatele lor cu tehnica de estimare robustă pe baza Algoritmului cu Translație la Medie (Mean Shift). Se determină experimental că metodele propuse în cadrul tezei de doctorat sunt mai exacte și mai fiabile.

1.2 Prezentare generală a etapelor procesului de recunoaștere

În teza de față este propusă o metodă de recunoaștere și scanare a codurilor Data Matrix marcate direct pe material, metoda fiind implementată practic. Fig. 1.1 indică etapele procesului de recunoaștere, în același timp, prezentând și structura tezei de doctorat.

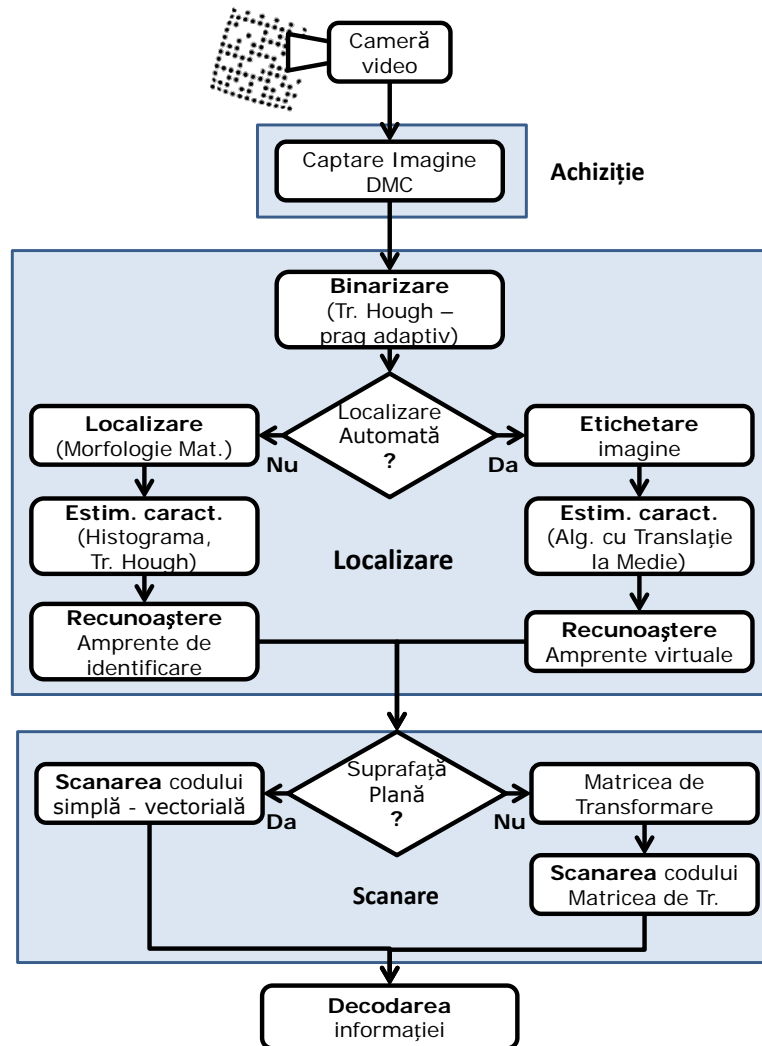


Figura 1.1: Etapele procesului de recunoaștere a codului Data Matrix industrial

Imaginea codului Data Matrix este captată de camera video și trimisă la procesul de localizare. În acest proces se binarizează imaginea cu metoda lui Otsu cu prag adaptiv și în funcție de modul de utilizare a programului, automat sau semi-automat, imaginea este prelucrată mai departe.

În cazul modului de lucru semi-automat, amprenta codului Data Matrix este localizată pe baza Morfologiei matematice prin recunoașterea formei pătratice a

amprente și pe baza informațiilor despre mărimea amprente și a caracteristicilor camerei video folosite. Având regiunea de interes, pe baza histogramei și a Transformării Hough, se estimează distanța dintre module, suprafața acestora și orientarea amprente codului Data Matrix. Pe baza acestor informații se recunoaște amprenta de localizare și cea de sincronizare a codului Data Matrix.

În modul automat, imaginea este etichetată, se estimează distanța dintre module, suprafața acestora și orientarea amprente codului pe baza Algoritmului cu Translație la Medie (Mean Shift). Pe baza acestor informații se construiește o amprentă virtuală de identificare și de sincronizare. Prin încercarea succesivă de potrivire a acestora cu modulele codului se recunoaște amprenta de identificare reală a codului Data Matrix.

Pentru scanarea codului, metoda propusă în teza de față oferă două soluții. Prima soluție este destinată amprențelor marcate pe materiale plane. În acest caz, scanarea este efectuată pe baza distanței dintre module și pe baza orientării amprente codului. Dacă se dorește scanarea unui cod marcat pe o suprafață curbă, a doua metodă de scanare pe baza Matricei de transformare oferă o soluție precisă independentă de traiectoria modulelor amprente codului.

În urma scanării amprente este creată o listă cu coordonatele modulelor codului Data Matrix. Aceasta este trimisă mai departe la blocul de decodare și interpretare a informației.

Experimentele sunt realizate pe modele reale, algoritmi prezentați în această teză se confruntă cu perturbațiile din lumea reală. Această metodă oferă 3 avantaje:

- robustețe. Codul Data Matrix poate fi localizat și în cazul imaginilor cu zgomot;
- precizie. Algoritmii propuși oferă acuratețe în recunoașterea, identificarea amprențelor și scanarea modulelor;
- implementare practică. Metoda este implementată practic în mediul de programare Matlab, fiind testată pe diverse coduri înscrisurate.

Rezultatele teoretice și practice arată că metoda propusă oferă performanțe bune pentru diferite condiții de iluminare, de material, de poziție sau alte perturbații externe. Această aplicație este realizată în întregime și are soluții pentru problemele ce se pot întâlni la captarea și prelucrarea codului Data Matrix industrial.

1.3 Structura tezei

Teza de doctorat este structurată în opt capitole, la care se adaugă bibliografia corespunzătoare.

- În Capitolul 1, **Introducere**, se prezintă o privire de ansamblu a achiziției codurilor Data Matrix folosite în mediu industrial. Sunt introduse pe scurt etapele acestei tehnici și problemele care le ridică fiecare etapă. Tot în cadrul acestui capitol, sunt prezentate motivele care au condus la tratarea acestei teme de cercetare ce face obiectivul tezei de doctorat.
- În Capitolul 2, **Codul Data Matrix**, se face o scurtă prezentare a evoluției istorice a codurilor, de la apariția primului cod de bare la codurile Data Matrix. Tot în acest capitol se prezintă structura codului Data Matrix cu caracteristicile lui și tehnicile de înscrisurare folosite în prezent.
- În Capitolul 3, **Metode de recunoaștere a codului Data Matrix**, se prezintă stadiul actual al cercetării în problema recunoașterii codurilor Data Matrix

industriale bazată pe tehnici robuste de estimare. Se prezintă o clasificare a metodelor de recunoaștere și scanare existente și etapele care duc la extragerea informației. De asemenea, se menționează tipurile de corespondență și problemele pe care acestea le ridică, împreună cu caracteristicile lor.

- În Capitolul 4, **Localizarea codurilor Data Matrix industriale**, sunt prezentate metodele propuse privind localizarea codurilor Data Matrix. Din necesitatea unei estimări a parametrilor amprentei Data Matrix, pe baza informațiilor cunoscute despre aceasta, s-au elaborat două soluții de achiziție și recunoaștere. În ambele soluții, imaginea este binarizată pe baza unui prag dinamic adaptiv, calculat cu metoda lui Otsu. Prima soluție utilizează histograma pentru estimarea distanței dintre module și a suprafeței acestora, iar pentru estimarea orientării, se folosește Transformarea Hough. Pentru localizarea amprentei codului se folosește Morfologia matematică ce ajută la recunoașterea formei acesteia. A doua soluție utilizează Algorimul cu Translație la Medie pentru estimarea distanței dintre module, a suprafeței modulelor și orientarea amprentei codului. Localizarea amprentei codului se face prin potrivirea obiectelor din imagine cu o amprentă virtuală, construită pe baza parametrilor estimați.
- În Capitolul 5, **Comparația metodelor de localizare**, se testează eficacitatea metodelor propuse și robustețea acestora, prezentându-se rezultatele obținute. Cu ajutorul unui generator de amprente Data Matrix se construiesc amprente orientate în unghiuri diferite. Se estimează, pe rând cu cele două metode, unghiul amprentelor și se compară rezultatele. Se urmărește creșterea acurateții și a eficienței estimării unghiului amprentei codului Data Matrix industrial. Experimentele au fost realizate cu ajutorul unei aplicații implementată în mediul de programare Matlab.
- În Capitolul 6, **Scanarea codurilor Data Matrix**, se prezintă tipurile de suprafețe pe care amprenta Data Matrix poate fi marcată. În funcție de acestea se dezvoltă două metode de scanare ale amprentei codului. Prima metodă efectuează scanarea modulelor în mod vectorial pe baza orientării amprentei și a distanței dintre module. Metoda a doua realizează scanarea amprentei codului pe baza unei grile de coordonate ce estimează poziția modulelor codului. Această grilă de coordonate este construită cu ajutorul Matricei de transformare geometrică. Se recomandă folosirea primei metode pentru scanarea suprafețelor plane, iar celei de-a doua pentru suprafețele curbe.
- În Capitolul 7, **Comparația metodelor de scanare**, se pun față în față cele două metode de scanare propuse în Capitolul 6, prin scanarea succesivă a mai multor tipuri de amprente. În urma experimentelor, se compară rezultatele și se discută eficiența și precizia folosirii celor două metode propuse. Experimentele au fost realizate cu ajutorul unei aplicații implementată în mediul de programare Matlab.
- Capitolul 8, **Concluzii generale și contribuții**, conține principalele concluzii rezultate din teza de doctorat și contribuțiile originale ale doctorandului, raportate în această teză.

1.4 Lucrări publicate

Soluțiile și metodele noi dezvoltate în cadrul acestei teze de doctorat au fost publicate în următoarele articole:

1. Ion-Cosmin Diță, Marius Oteșteanu, Franz Quint, „Scanning of Industrial Data Matrix Codes in non Orthogonal View Angle Cases”, 6th Euro American Conference on Telematics and Information Systems EATIS 2012, Valencia, Spania, 23 - 25 mai 2012, acceptat pentru publicare, indexat IEEE Xplore.
2. Ion-Cosmin Diță, Franz Quint, Marius Oteșteanu, „A Scanning Method for Industrial Data Matrix Codes marked on Cylindrical Surfaces”, 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems, Brno, Cehia, 23 - 25 mai 2012, acceptat pentru publicare, indexat IEEE Xplore.
3. Ion-Cosmin Diță, Vasile Gui, Franz Quint, Marius Oteșteanu, „Comparison of Hough Transform and Mean Shift Algorithm for Estimation of the Orientation Angle of Industrial Data Matrix Codes”, ICDIP 2012 - International Conference on Digital Image Processing, Veneția, Italia, 11 - 13 aprilie, 2012, acceptat pentru publicare, indexat ISI Proceedings.
4. Ion-Cosmin Diță, Marius Oteșteanu, Franz Quint, Nagaraju Annamdevula, „A Scanning Method for Industrial DMC Marked on Spherical Surfaces”, 11th WSEAS International Conference on TELECOMMUNICATIONS and INFORMATICS (TELE - INFO 2012), vol. 11, Saint Malo & Mont Saint-Michel, Franța, 2 - 4 aprilie 2012, acceptat pentru publicare, indexat ISI Proceedings.
5. Ion-Cosmin Diță, Vasile Gui, Marius Oteșteanu, Franz Quint, „Using Mean Shift Algorithm in the Recognition of Industrial Data Matrix Codes”, 11th WSEAS International Conference on SIGNAL PROCESSING (SIP '12), Saint Malo & Mont Saint-Michel, Franța, 2 - 4 aprilie 2012, acceptat pentru publicare, indexat ISI Proceedings.
6. Ion-Cosmin Diță, Marius Oteșteanu, Franz Quint, „Data Matrix Code - A Reliable Optical Identification of Microelectronic Components”, 2011 IEEE 17th International Symposium for Design and Technology in Electronic Packaging (SIITME), vol. 17, pp. 39 - 44, Timișoara, 20 - 23 octombrie 2011, indexat ISI Proceedings și IEEE Xplore.
7. Ion-Cosmin Diță, Marius Oteșteanu, Franz Quint, „Scanning Industrial Data Matrix Codes”, 19th Telecommunications Forum (TELFOR 2011) pp. 1211 - 1214, Belgrad, Serbia, 22-24 noiembrie 2011, indexat IEEE Xplore.
8. Ion-Cosmin Diță, Marius Oteșteanu, Franz Quint, „A Robust Localization Method for Industrial Data Matrix Code”, Scientific Bulletin of the „Politehnica” University, Timișoara, Tom 56(70), Fascicola 2, no. ISSN 1583-3380, pp. 12 - 17, septembrie 2011.
9. Ion-Cosmin Diță, Marius Oteșteanu, „Factors that Influence the Image Acquisition of Direct Marking Data Matrix Code”, 17th Telecommunications forum TELFOR 2009, no. ISBN 978-86-7466-375-2, pp 923 - 926, Belgrad, Serbia, 24 - 26 noiembrie 2009.
10. Ion-Cosmin Diță, „Image Acquisition of Direct Marking Data Matrix Code”, Sesiunea de comunicări științifice „Doctor ETc 2009”, no. ISSN 2066-883X, pp. 111-116, Timișoara, 24 - 25 septembrie 2009.

Capitolul 2

Codul Data Matrix

În acest capitol se face o scurtă prezentare a evoluției istorice a codurilor, de la apariția primului cod de bare la codurile Data Matrix. De asemenea, se prezintă structura codului Data Matrix cu caracteristicile lui și tehnicile de inscripționare folosite în prezent.

2.1 Codurile unidimensionale

Primul sistem de coduri de bare a fost brevetat în octombrie 1949 de Norman Woodland și Bernard Silver. Simbolul a fost sub formă de „ochi de taur”, compus dintr-o serie de cercuri concentrice. Ei au depus o cerere de brevet pentru o metodă de clasificare a elementelor pe baza unui răspuns optic la o amprentă formată din linii [1].

La începutul anilor 1960 a fost comercializat primul sistem de scanare de coduri de bare și a fost posibilă citirea unui set de bare de mai multe culori: roșie, albastră, albă și neagră. Acest avantaj a condus la utilizarea codurilor de bare în industria transporturilor și în supermarketuri. La sfârșitul aceluși deceniu a apărut tehnologia laser. Primul cititor laser funcționa pe baza radiațiilor de Helium-Neon, iar prețul său a fost aproximativ 10.000 de dolari. În zilele noastre, un scanner similar ar costa mai puțin de 2.000 de dolari [2].

În 1970 au început să se dezvolte o mulțime de aplicații industriale de gestionare a informațiilor. Primul cod de bare standard este CodaBar, folosit cu succes în industria automobilelor, Fig. 2.1 [3].



Figura 2.1: CodaBar

În aprilie 1973, a fost standardizat codul U.P.C. (Universal Product Code), devenind versiunea cea mai populară și mai importantă în tehnologia identificării produselor (Fig. 2.2) [4]. Acest sistem a permis un mod mai ușor și mai rapid de gestionare a achiziției și livrării produselor în magazinele mari. În 1976, Europa a făcut o versiune a U.P.C., numită EAN (European Article Number) [5].



Figura 2.2: UPC-A, UPC-E, EAN, EAN-14

În 1974, Davis Allais, împreună cu compania Intermec, au conceput codul numit Code 39, primul cod care poate să conțină numere și litere. În anii optzeci, Serviciul Poștal al Statelor Unite ale Americii a început să utilizeze codul PostNet, Fig. 2.3.

**Figura 2.3:** PostNet

În 1987 Allais și Intermec, au dezvoltat primul cod bidimensional, numit Cod 49. În următorul deceniu este publicat caietul de sarcini ANS X3 0.182 pentru reglementarea calității imprimării codurilor de bare liniare [6].

2.2 Codurile bidimensionale

Codurile de bare convenționale, cum sunt Codurile Universale pentru Mărfuri (UPC), au câștigat teren în aplicații, variind de la casă și controlul inventarului în vânzările cu amănuntul la urmărirea cablajelor cu circuite electronice în industria electronică [7]. Ar fi posibilă crearea unui cod de bare liniar cu mai mult de 20 de caractere, dar ar fi un cod foarte lung, imposibil de citit de un scanner standard. Pentru a spori conținutul simbolului și pentru a stoca informații în spații cât mai mici, companiile au dezvoltat alternativa codurilor bidimensionale. Informațiile sunt codificate de-a lungul și de-a latul simbolului, fiind posibilă stocarea unui set de caractere [1].

Avantajul principal al folosirii codurilor bidimensionale este că simbolul poate conține o cantitate mare de caractere ce pot fi citite rapid și fiabil, fără a fi nevoie de a folosi o bază de date pentru a stoca aceste informații. Codurile de bare liniare, în general, funcționează ca un index de a găsi un registru într-o bază de date înregistrată (de exemplu, numărul unui produs în supermarket referitor, la un preț, etc). În timp ce codurile bidimensionale pot face aceeași funcție cu ajutorul unui spațiu mult mai mic sau chiar să funcționeze ca o bază de date în sine, asigurând o portabilitate în ansamblu pentru elementele etichetate cu coduri 2D.

Securitatea implementată în codurile bidimensionale le asigură în totalitate protecția împotriva atacurilor externe. Pentru a deteriora informația dintr-un cod de bare liniar este suficient de a adăuga o altă bară, la începutul sau la sfârșitul simbolului, sau de a desena o linie paralelă cu barele, în orice loc în interiorul codului. Codurile bidimensionale pot fi construite cu o mulțime de grade redundante, dublând informația în întregime sau doar datele vitale. Redundanța crește dimensiunile simbolului, dar securitatea conținutului este, de asemenea, puternic mărită. În plus, codurile bidimensionale sunt mult mai rezistente la distrugeri decât cele liniare, datorită formulelor de corecție a erorii, așa numite ECC (Error-correcting code). Corecția erorii codului constă în adăugarea de biți suplimentari de paritate, de detecție a erorii, pentru a fi corectată de dispozitiv în timp ce datele sunt citite. Algoritmii ECC corectează datele la nivel de bit. Codurile bidimensionale sunt citite cu ajutorul a două tipuri de scanere:

- Cititoarele de amprente scanează codul pe baza unui fascicul laser ce se deplasează de la stânga la dreapta și de sus în jos. Această metodă este total depășită datorită evoluției tehnologiei CCD.
- Cititoarele CCD conțin o matrice bidimensională fotosensibilă, ce scanează imaginea în ansamblu.

Cele mai utilizate coduri 2D, sunt: PDF 417, Data Matrix și MaxiCode [8]. Drepturile de proprietate pentru aceste coduri sunt de domeniu public, fără a se plăti drepturi de autor pentru această tehnologie. Codurile bidimensionale pot fi reprezentate cu matrici sau simboluri aranjate după diferite reguli.

2.2.1 Codurile suprapuse

Codurile de bare suprapuse sunt pur și simplu un set de coduri de bare liniare așezate unul peste celălalt. Codul PDF417 face parte din această categorie, fiind un cod de lungime variabilă, ce poate reprezenta orice număr sau caracter. Fiecare caracter este format din 4 bare și 4 spații într-o structură de 17 module. Numele simbolului este derivat din formatul codului. PDF înseamnă Fișier de Date Portabile și 417 se referă la structura modulelor, Fig 2.4 [9].



Figura 2.4: PDF417

Recent, sunt date noi cerințe pentru informațiile conținute în codurile comerciale UPC și EAN . Mai multe informații sunt codate în același spațiu. Astfel a fost creat codul RSS (Reduced Space Symbology), Fig 2.5 [8]. RSS este una dintre cele mai compacte simbologii până în prezent. Această familie are trei versiuni diferite: RSS-14, RSS Limitat și RSS Extins .



Figura 2.5: RSS Extins

2.2.2 Codurile matriciale

Codurile matriciale sunt realizate din amprente formate din celule, ce pot fi pătratice, hexagonale sau rotunde. Codurile matriciale oferă o densitate mai mare de date decât codurile suprapuse [1].

MaxiCode este un cod matricial de dimensiune fixă, cu 866 elemente hexagonale aranjate în 33 de rânduri în jurul unei amprente centrale de localizare, Fig. 2.6. Acest cod este folosit de UPS pentru clasificarea la viteză mare a scrisorilor. Codul poate fi citit foarte rapid, în toate direcțiile [4].

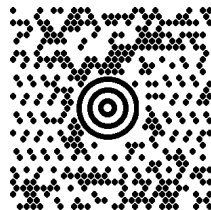


Figura 2.6: MaxiCode

Codul Aztec este un alt cod matricial de densitate mare [10]. Simbolurile sunt plasate în jurul unui pătrat aflat în centrul amprentei. Acesta poate fi citit indiferent de orientarea amprentei. Codul are mecanisme de corecție a erorilor configurabile diferit. În această clasificare se găsește, de asemenea, Codul Data Matrix care va fi descris în Capitolul 2.4 în mod mai detaliat.

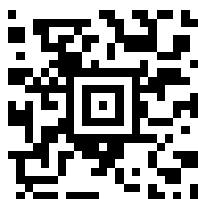


Figura 2.7: Aztec

2.3 Compararea codurilor

Înainte de a descrie pe larg codul Data Matrix, prezentăm un studiu al Universității din Ohio care justifică fiabilitatea codului Data Matrix. Toate codurile reprezentate în Fig. 2.8 - 2.11 au codificat aceeași informație: „1234567891234567890”, cu aceeași lățime de bare. Din aceste imagini este evidentă economia de spațiu în cazul simbolurilor 2D.



Figura 2.8: Code 39, primul cod alfanumeric



Figura 2.9: ITF



Figura 2.10: PDF417, cod bidimensional

La Universitatea din Ohio, au fost realizate aplicații reale pentru a studia exactitatea celor mai utilizate coduri. Acest studiu [1] a aratat că, în cazul codurilor de bare, cele mai slabe rezultate în ceea ce privește acuratețea au fost întâlnite la utilizarea codurilor UPC. În cel mai rău caz, codul prezintă o cotă de eroare la fiecare 394.000



Figura 2.11: Codul Data Matrix, cod bidimensional

de caractere. Codurile cu cele mai bune rezultate au fost codul Data Matrix și codul PDF417, cu o eroare la fiecare 10,5 milioane de caractere [1]. În tabelul de mai jos sunt prezentate toate rezultatele:

Tabela 2.1: Clasificarea codurilor în funcție de eroare

Simbol	Cel mai rău caz	Cel mai bun caz
DMC	o eroare la fiecare 10,5 mil	o eroare la fiecare 612,9 mil
PDF417	o eroare la fiecare 10,5 mil	o eroare la fiecare 612,4 mil
Code 128	o eroare la fiecare 2,8 mil	o eroare la fiecare 37 mil
Code 39	o eroare la fiecare 1,7 mil	o eroare la fiecare 4,5 mil
UPC	o eroare la 394.000 caract.	o eroare la 800.000 caract.

2.4 Codul Data Matrix ECC200

Codul Data Matrix a fost dezvoltat în 1989 de International Data Matrix Inc. Versiunea publică este ECC200, dezvoltată, de asemenea, de International Data Matrix în 1995. Codul Data Matrix este probabil cel mai cunoscut membru al familiei de 20 de coduri. A fost mai întâi conceput pentru imprimarea pe hârtie și datorită succesului său a fost preluat în industrie și în media.

Pornind de la dorința de a afla cât mai multe informații despre produsele din supermarketuri, afișele de pe stradă, reclamele TV și despre tot ce ne înconjoară și ne afectează viața, există posibilitatea stocării tuturor acestor informații într-o amprentă de dimensiune foarte mică numită Data Matrix. Încă de la apariția acestor coduri, marile companii au avut ideea de a le integra în produse, având astfel o cale mai ușoară și sigură de monitorizare și transmitere a informațiilor. În [11] se propune utilizarea acestora pentru a accesa conținutul multimedia prin intermediul web cu ajutorul unui telefon mobil. O etichetă poate fi tipărită pe articolele de publicitate (ziare, reviste, panouri), utilizatorii pot scana și achiziționa imaginea acesteia cu ajutorul unui telefon cu cameră. Software-ul client decodifică informațiile (de obicei, URL-ul unui site), se conectează automat la site-ul web și afișează sau descarcă conținutul multimedia (audio, video) pe telefonul mobil. În [12] se propune o metodă de generare rapidă a codului 2D pentru receptoare DTV. Această metodă intenționează să genereze o amprentă automată cu informații despre conținutul programului TV. Citind informațiile conținutului programului TV, receptorul DTV poate să blocheze sau să permită acest program, iar privitorul poate să acceseze mai multe informații folosind telefonul mobil.

Folosirea lor în comerțul mobil [4] sau în crearea legăturilor între hărțile mobile și o bază de date web a hărților [13] sau la inserarea de watermark-uri invizibile în imagini [14] sunt doar câteva exemple de utilizare a codurilor Data Matrix.

Codul de bare 2-D, fiind clasificat ca și codul cu cea mai mică dimensiune din lume, se aplică pe scară largă și în domeniul componentelor electronice și mecanice. Tehnologia de recunoaștere a codurilor de bare 2-D, prezintă o mare valoare comercială în prezent, fiind o cerere mare pentru cercetare.

2.4.1 Structura generală

Codul Data Matrix ECC 200 este compus din două părți separate (Fig. 2.12): amprenta de indentificare, care este folosită de cititor pentru localizarea simbolului, precum și datele codificate în sine [15].

Amprenta de indentificare este definită de forma pătratică sau dreptunghiulară, mărimea, distanța dintre module și numărul de rânduri și coloane din codul Data Matrix. Amprenta solidă întunecată este numită „Amprenta L de indentificare”. Aceasta este utilizată în principal pentru a determina dimensiunea și orientarea codului (Fig. 2.12(a)). Celelalte două părți ale amprentei sunt formate din elemente alternativ luminoase-întunecate, cunoscute sub numele de „Amprenta de sincronizare”. Aceasta definește structura de bază a simbolului și poate, de asemenea, determina mărimea și distorsiunile codului (Fig. 2.12(b)). Datele sunt codificate într-o matrice în interiorul amprentei de indentificare a codului. Aceasta este o traducere în simboluri Data Matrix de caractere binare (numerice sau alfanumerice), Fig. 2.12(c).

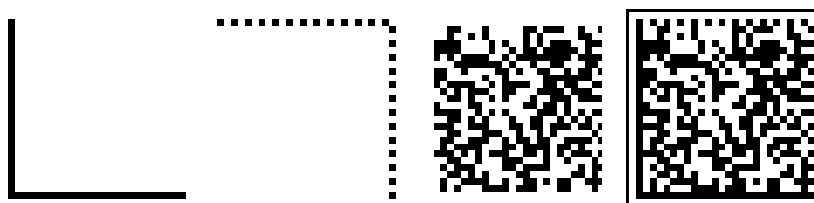


Figura 2.12: DMC - (a) Amprenta de indentificare, (b) Amprenta de sincronizare, (c) Informația codată, (d) Zona de liniște

La fel ca și codurile de bare liniare (1D), codul Data Matrix are obligatoriu o zonă de liniște. Aceasta este o zonă luminoasă în jurul codului, care nu trebuie să conțină niciun element grafic ce poate perturba citirea codului. Are o lățime constantă egală cu distanța dintre module, aflată pe fiecare din cele 4 laturi ale simbolului.

Fiecare cod Data Matrix este format dintr-un număr de rânduri și coloane. În versiunea ECC 200, numărul de rânduri și de coloane este întotdeauna un număr par. Prin urmare, codul Data Matrix ECC 200 are întotdeauna un modul luminos, în colțul din dreapta sus, Fig. 2.12. Evident, acest colț poate fi întunecat, dacă codul Data Matrix este tipărit în negativ (culori complementare).

2.4.2 Caracteristici tehnice

2.4.2.1 Prezentarea formei amprentei

La punerea în aplicare a codului Data Matrix, trebuie aleasă forma amprentei pe baza spațiului disponibil pe tipul de produs, cantitatea de informație pentru codare, procesul de imprimare, etc. Este posibilă codificarea aceleiași informații în două forme de amprente de cod Data Matrix:

Forma pătratică este cea mai frecvent utilizată deoarece permite codarea unei cantități mai mari de date în conformitate cu ISO / IEC 16022 (*Tehnici de captare de date și de indentificare automată - Specificațiile simbolului codului Data Matrix*) [16]. Cu toate acestea, forma dreptunghiulară poate fi selectată pentru a îndeplini constrângerile de viteză de imprimare pe linia de producție.

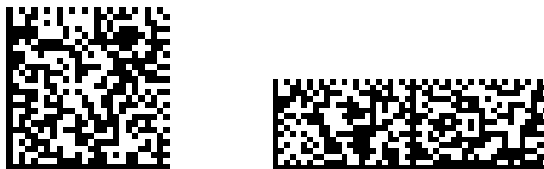


Figura 2.13: Cod Data Matrix pătratic și dreptunghiular

2.4.2.2 Mărimi și capacități de stocare

Codul Data Matrix este capabil să stocheze date de lungime variabilă. Prin urmare, dimensiunea rezultată a codului variază în funcție de cantitatea de date de informație stocată. În consecință, această secțiune poate estima aproximativ dimensiunea unui cod Data Matrix bazat pe acest parametru. Tabelul 2.2 este extras din ISO / IEC 16022 (a se vedea tabelul A.2 din *Data Matrix ECC 200 Symbol Attributes*), acesta oferă un ghid util pentru estimarea dimensiunii codului [16].

Tabela 2.2: Caracteristicile codului Data Matrix

Mărime amprentă	Regiuni de date		Cuvinte cod		Cap. maximă de date		Cuv. cod corecție	Cuv. cod corectabile
	Mărime	Nr.	Data	Er.	Num.	α Num	%	Er./Șters
10×10	8×8	1	3	5	6	3	62.5	2/0
12×12	10×10	1	5	7	10	6	58.3	3/0
14×14	12×12	1	8	10	16	10	55.6	5/7
16×16	14×14	1	12	12	24	16	50	6/9
18×18	16×16	1	18	14	36	25	43.8	7/11
20×20	18×18	1	22	18	44	31	45	9/15
22×22	20×20	1	30	20	60	43	40	10/17
24×24	22×22	1	36	24	72	52	40	12/21
26×26	24×24	1	44	28	88	64	38.9	14/25
32×32	14×14	4	62	36	124	91	36.7	18/33
36×36	16×16	4	86	42	172	127	32.8	21/39
40×40	18×18	4	114	48	228	169	29.6	24/45
44×44	20×20	4	144	56	288	214	28	28/53
48×48	22×22	4	174	68	348	259	28.1	34/65
52×52	24×24	4	204	84	408	304	29.2	42/78
64×64	14×14	16	280	112	560	418	28.6	56/106
72×72	16×16	16	368	144	736	550	28.1	72/132
80×80	18×18	16	456	192	912	682	29.6	96/180
88×88	20×20	16	576	224	1152	862	28	112/212
96×96	22×22	16	696	272	1392	1042	28.1	136/260
104×104	24×24	16	816	336	1632	1222	29.2	168/318
120×120	18×18	36	1050	408	2100	1573	28	204/390
132×132	20×20	36	1304	496	2608	1954	27.6	248/472
144×144	22×22	36	1558	620	3116	2335	28.5	310/590

Mărimile prevăzute mai sus sunt prezentate în termeni de număr de rânduri și de coloane. Pentru codul Data Matrix ECC 200 de formă pătratică, numărul de rânduri

și de coloane poate varia între 10 și 144, furnizând 24 de mărimi de simboluri. În contrast pentru codurile Data Matrix de formă dreptunghiulară numărul de rânduri este cuprins între 8 și 16 și numărul de coloane între 18 și 48. Codul Data Matrix dreptunghiular permite șase mărimi (forma pătratică are 24), iar utilizarea sa este mai puțin răspândită decât amprenta pătratică.

Când este printat un cod Data Matrix, dimensiunea imaginii depinde de următorii factori:

- cantitatea și formatul (numeric sau alfanumeric) informației codate. Numerele și caracterele sunt codate sub formă de biți, reprezentați de module întunecoase sau luminoase de dimensiuni identice. Cu cât este necesară o cantitate mai mare de biți, va fi folosit un simbolul mai mare;
- distanța dintre module;
- alegerea formei simbolului: pătratică sau dreptunghiulară.

Tabelul 2.2 arată valoarea maximă a informației care poate fi codificată în amprenta Codului Data Matrix. Codul Data Matrix poate codifica până la:

- 2,335 de caractere alfanumerice,
- 3,116 numere.

Acest maxim se referă la simbolurile de formă pătratică alcătuite din 144 de rânduri și 144 de coloane, împărțite în 36 de regiuni de date de 22 de rânduri și 22 de coloane fiecare.

2.4.2.3 Regiunea datelor

Amprenta codului Data Matrix, este compusă din mai multe zone de date (regiuni de date), care împreună codifică informația. Tabelul 2.3 prezintă un extras din ISO / IEC 16022 [7], ce oferă detalii despre modul în care sunt compuse regiunile de date. De exemplu, o amprentă formată din 32 de rânduri și 32 de coloane, include 4 sub-matrici de 14 rânduri și 14 coloane. Numărul și dimensiunea „sub-matricelor”, în cadrul simbolului Data Matrix, sunt afișate în coloana „Regiunile de date”, Tabelul 2.3.

Tabela 2.3: Mărimea simbolului și numărului de regiuni de date

Mărimea simbolului		Regiunile de date	
Rânduri	Coloane	Dimensiune	Nr. regiuni
24	24	22× 22	1
26	26	24× 24	1
32	32	14× 14	4
36	36	16× 16	4

2.4.2.4 Metode de corecție a erorilor

Există mai multe metode de detecție a erorilor. Un exemplu este check-digit [17] folosit de multe coduri de bare liniare, care utilizează un algoritm pentru a calcula ultima cifră a numărului codificat. Check-digit poate confirma dacă șirul de date este

codat corect, în conformitate cu algoritmul specificat. Însă, în cazul unei greșeli, nu poate indica unde a fost făcută greșeala.

O altă modalitate este de a repeta datele codificate în amprență, ceea ce va ajuta la obținerea unei citiri cu succes chiar dacă amprența este deteriorată. Aceasta se numește redundanță.

Codificarea datelor într-un simbol Data Matrix se poate face folosind niveluri multiple de securitate. Structura bidimensională permite codarea datelor și a mecanismelor de corecție a erorilor. Aceste mecanisme permit scannerului să reconstruiască o parte din informații, în cazul unui Cod Data Matrix deteriorat sau dificil de citit. În standardul Data Matrix ISO / IEC 16022 [16] sunt descrise mai multe niveluri de securitate [16]. Fiecare din tipurile de Coduri Data Matrix: ECC 000, ECC 050, ECC 080, ECC 100 și ECC 140 are o metodă de detecție și de corecție a erorilor.

2.4.2.5 Algoritmul Reed-Solomon pentru corecția erorilor

Data Matrix ECC 200 are configurația Codului Data Matrix, dar cu utilizarea algoritmului Reed-Solomon pentru corecția erorilor. Această caracteristică permite, într-o anumită măsură, localizarea erorilor și, în cazul în care este posibil, corectarea lor. Codurile Reed-Solomon (RS) sunt un tip de coduri pentru canalul de codare ce au fost introduse de către Irving S. Reed și Gustave Solomon, în anul 1960 [18]. Codarea canalului adaugă o parte redundantă la informația originală bazat pe un sistem specific de obținere a unor cuvinte cod. Un decodor adecvat ce primește cuvintele cod este capabil să corecteze o anumită parte din erori ce pot apărea în timpul transmisiei. Corecția erorii este posibilă numai datorită părții de cod redundantă. Codurile RS sunt coduri ciclice nonbinare, ceea ce înseamnă că simbolurile de cod sunt luate dintr-un câmp $GF(q) = GF(2^m)$. Astfel, fiecare simbol este format din m biți. Acestea sunt foarte utile pentru corecția erorilor și pot fi găsite în zilele noastre într-o mulțime de aplicații, cum ar fi compact discuri (CD-uri), sisteme de stocare în masă sau în sistemele de telecomunicații.

2.5 Technici de inscripționare

Această secțiune prezintă tehnologiile și procesele folosite la inscripționarea codurilor Data Matrix. Se sintetizează punctele lor forte și punctele lor slabe individuale pentru diverse aplicații. Tehnologiile cele mai potrivite pentru marcarea codului Data Matrix sunt următoarele:

- transfer termic
- cerneală
- laser
- marcarea directă prin frezare sau punctare prin lovire

Alegerea metodei corecte se face pe baza materialului suport și a cerințelor comerciale.

2.5.1 Transferul termic

Imprimarea prin transfer termic este, probabil, tehnologia cea mai utilizată pe scară largă pentru imprimarea etichetelor codurilor de bare. Tehnologia de imprimare este pe bază de căldură, codurile fiind transmise pe o panglică (o bandă acoperită cu cerneală special concepută), care apoi transferă imaginea pe etichetă. Pot fi imprimate

coduri de bare de foarte bună calitate, atunci când materialul supus etichetării și panglica de imprimare sunt pe deplin compatibile. Rezoluția de imprimare pentru imprimantele cu transfer termic este cuprinsă între 100 și 600 dpi (dots-per-inch).

2.5.2 Imprimarea cu cerneală

Imprimarea cu jet de cerneală este un proces ce nu necesită contact între imprimantă și etichetă. Tehnologia funcționează prin pulverizarea de picături mici de cerneală pe substrat pentru a crea amprenta Data Matrix (Fig. 2.14).



Figura 2.14: Cod Data Matrix imprimat cu cerneală

Imprimarea cu jet de cerneală imprimă modulele de obicei într-o formă neregulată. Acest lucru este cauzat de absorbția materialului și de forma neregulată a punctelor unice. Ampretele de bună calitate sunt posibile, atunci când se imprimă pe un suport adecvat, folosind o imprimantă de înaltă rezoluție și uscarea rapidă a cernelei. De asemenea, în cazul în care imprimanta cu jet de cerneală nu este ținută în parametrii de funcționare recomandați de către producător, acest lucru poate cauza probleme de calitate.

2.5.3 Gravarea cu laser

Gravarea cu laser folosește raze laser controlate cu precizie pentru gravarea sau marcarea codului direct pe material (Fig. 2.15). Aceasta are nevoie de un calculator ce comandă o serie de oglinzi și lentile pentru focalizarea și controlul fascicolului laser. Procesul permite ca un produs să fie marcat direct și permanent, dar este folosit numai pentru materiale ce permit marcarea cu laser fără a fi distruse.

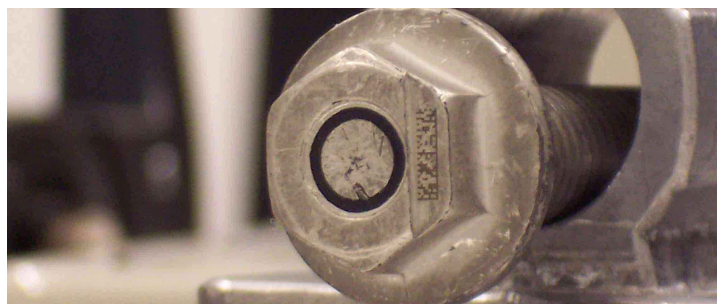


Figura 2.15: Cod Data Matrix marcat cu laser

2.5.4 Marcarea directă prin punctare sau frezare

Această tehnologie este folosită pentru marcarea în mod direct a materialului și se folosește în special pentru materialele solide (metale, plastic, lemn, etc ...). Aceasta poate fi folosită pentru marcarea mai multor informații: text, date, logo, etc, precum și pentru amprenta Data Matrix. Un cap mic, în mod normal dintr-un material foarte dur este controlat de calculator pentru a face o serie definită de adâncituri identice în suprafața materialului. Adâncimea amprentei poate fi cu precizie controlată pentru a se asigura că toate elementele sunt identice. Această tehnică de punctare este folosită în special la marcarea codurilor Data Matrix (Fig. 2.16).



Figura 2.16: Cod Data Matrix punctat

2.6 Evaluarea tehnicilor de marcarea și a materialelor folosite

Utilizarea codurilor Data Matrix în mediul industrial este în continuă creștere. Cu toate acestea producătorii au avut alte idei. Etichetele pot fi șterse sau pot să cadă, astfel companiile mari au început aplicarea codului direct pe produsul ce urmează a fi marcat și identificat. Marcarea pieselor ce se realizează prin metode precum gravare cu laser sau punctare prin lovire produce de multe ori coduri cu contrast redus sau cu modulele marcate slab calitativ. Citirea unor astfel de coduri este o provocare în prezent. Progresele în tehnicile de iluminare și de procesare video au permis dezvoltarea de soluții pentru recunoașterea codului Data Matrix, dar încă nu suficient de performante.

Primul pas în asigurarea unei rate mari de citire este de a ne asigura că procesul de marcarea inscripționează un cod bun. Deși acest lucru sună simplu în principiu, până în prezent verificarea calității codului a fost atât problematică, cât și frustrantă. Codurile pe care optic le putem descifra cu ușurință pot fi de multe ori greu de recunoscut pentru un cititor. Recent aprobat AIM (Association for Automatic Identification and Mobility) - DPM (Direct Part Mark), ghid de calitate, este destinat să se ocupe de varietatea tehnicilor de marcarea și de materialele utilizate în aplicațiile de marcarea directă [19].

Fiecare tehnică abordează anumite aplicații, în funcție de durata de viață a piesei, compoziția materialului, potrivit uzurii provocate de mediu și de volumul de producție. Textura suprafeței, la fel ca și cantitatea de date pe care amprenta trebuie să o stocheze, spațiul disponibil pentru imprimare și locul de amplasare al amprentei influențează alegerea tehnicii de marcarea.

În Fig 2.17 (a), este prezentată imaginea originală a unui cod Data Matrix standard tipărit pe hârtie, iar în Fig 2.17 (b), este prezentată imaginea originală unui cod Data Matrix marcat direct pe material cu metoda laser. În aceste figuri este de dorit să se

vădă contrastul slab între fundalul materialului și modulele codului, în cazul codurilor industriale.

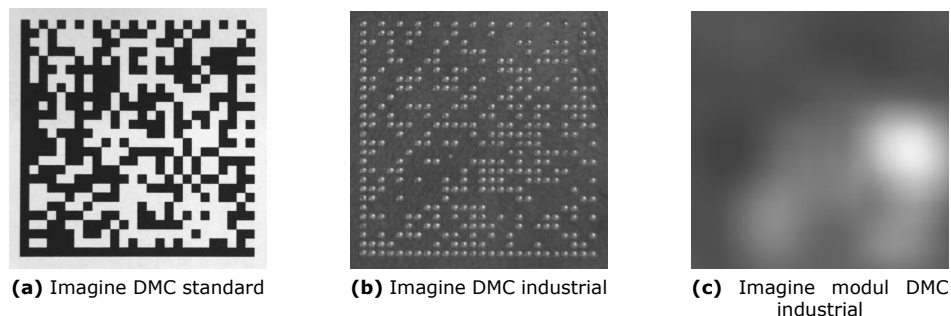


Figura 2.17: Imaginea unui Cod Data Matrix

În Fig. 2.17 (c) este prezentată trecerea de la o zonă liberă la un modul marcat. Se poate vedea că există o diferență de contrast, dar nu foarte mare, iar muchia modulului nu este bine definită.

În Fig. 2.18 este reprezentată histograma imaginii codului Data Matrix standard din Fig. 2.17(a), iar în Fig. 2.19 (b) este reprezentată histograma imaginii Codului Data Matrix din Fig. 2.17 (b). Histograma afișează nivelurile de gri de la 0 (negru) la 255 (alb), a tuturor pixelilor imaginii amprente codului. Histograma unui cod bine marcat ar trebui să arate ca două vârfuri distincte și bine separate. Pentru a decoda informația ce este purtată de amprentă valorile nivelurilor de gri ale modulelor trebuie decise să fie 1 sau 0, în funcție de ce parte a pragului de binarizare se situează în histogramă. Nivelurile de gri, ce sunt apropiate de pragul de binarizare, sunt mai dificil de decise unde să fie plasate. Pe baza histogramei din Fig. 2.18 este ușor de estimat pragul de binarizare, dar pe baza histogramei din Fig. 2.19 este mai greu de apreciat poziția pragului de binarizare.

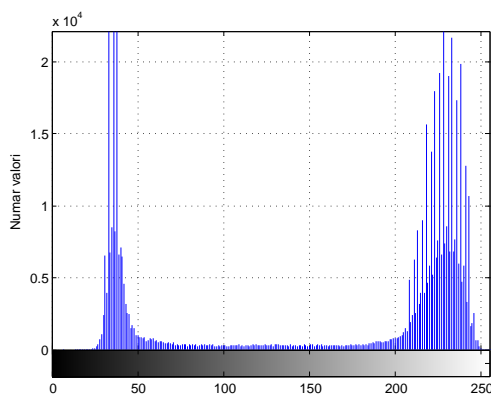


Figura 2.18: Histograma imaginii DMC standard din Fig. 2.17(a)

Pentru a vedea poziția nivelurilor de gri a modulelor amprente în histograma valorilor extragem o zonă de imagine în care se face trecerea de la un modul la un spațiu liber. Pe baza acestei probe, se trasează histograma din Fig. 2.19 (b).

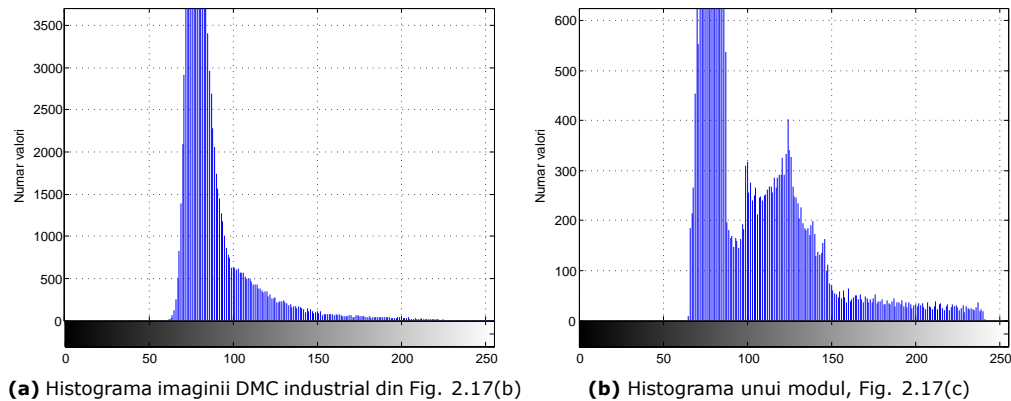


Figura 2.19: Histograma imaginii DMC din Fig. 2.17

Analizând histograma am putea delimita trecerea de la un modul la un spațiu liber și să alegem poziția pragului de binarizare a imaginii codului Data Matrix.

Pentru îmbunătățirea contrastului dintre module și fundal, se poate regla sursa de lumină într-un unghi cu suprafața amprentei. În lucrarea [20] am realizat un studiu despre unghiul potrivit, creat de sursa de lumină și suprafață, pentru o bună achiziție a imaginii. În urma evaluării mai multor imagini am ajuns la concluzia că cel mai bun contrast este realizat pentru materialele din aluminiu, iar unghiul optim pentru captarea imaginii este de 45° . Tot în urma acestui studiu, imaginile cu cel mai slab contrast sunt imaginile codurilor Data Matrix marcate pe materiale din plastic. Se poate concluziona că proprietatea de reflexie a luminii joacă un rol important în achiziția imaginii codului Data Matrix.

În toate cazurile de marcare, producătorii ar trebui să lase o zonă clară, de liniște, în jurul amprentei ori de câte ori este posibil, astfel încât codul să fie ușor de localizat pe produs. Bineînțeles, amprenta trebuie să fie curată și să nu fie afectată de zgomot, sau alte obiecte ce ar putea să intre în contact cu codul.

În funcție de tehnicile de marcare a amprentelor, imaginile codurilor pot varia mult de la o situație la alta. În plus, față de metoda de marcare selectată, piesele au diferite culori sau forme și pot fi realizate din diferite materiale. Suprafețele pot fi netede și lucioase, cu dungi sau textură.

Toate situațiile prezentate mai sus, din păcate, nu pot fi evitate în industrie. Metoda de scanare trebuie să fie fiabilă și să furnizeze rezultate corecte, pentru orice situație.

2.7 Standard pentru codurile industriale

Până în prezent industria codului Data Matrix standard și utilizatorii codurilor marcate direct pe material (DPM) au avut posibilități limitate pentru determinarea calității amprentei. ISO / IEC 16022 a definit modul de imprimare (sau marcare) a unui cod Data Matrix (structura codului, forma simbolului, algoritmi de decodare, ș.a.m.d.) [16]. Deși ISO / IEC 16022 a inclus inițial câțiva indicatori de calitate, autorii lui nu au intenționat niciodată să abordeze verificarea lui. Această idee a fost lansată de ISO / IEC 15415, care a apărut aproximativ cinci ani mai târziu [21].

Standardul ISO / IEC 15415 cere calibrarea cititorului înainte de utilizare, pe baza unui card alb destinat determinării expunerii la care se cunosc valorile de alb

(Certificatul de calibrare NIST) [22]. Calibrarea implică ajustarea setărilor sistemului video (expunerea camerei video sau amplificarea, de exemplu), astfel încât valorile albe observate de pe cardul de calibrare să corespundă cu valorile cunoscute. Odată ce calibrarea este efectuată, aceste setări nu se schimbă niciodată, indiferent de metoda de marcăre, materialele, sau caracteristicile suprafeței. Din nou, o astfel de cerință poate produce imagini acceptabile pentru etichetele din hârtie. Setările fiind fixe pentru piesele marcate direct, cu toate acestea, produc imagini care, în cele mai multe cazuri sunt, fie subexpuse, fie supraexpuse, în funcție de metoda de marcăre și proprietatea de reflexie a componentei.

O problemă comună ce apare la codurile marcate direct (DPM) (cum ar fi un cod gravat pe o bucată de metal), atunci când se aplică configurația unică de iluminare a standardului ISO / IEC 15415, este că imaginea arată mai mult gri pe negru decât alb pe negru. Vârfurile rezultate ale histogrammei sunt mult mai puțin distincte.

Un criteriu de evaluare standard este contrastul amprentei, răspândirea valorilor histogrammei între cel mai mic nivel („negru”) și cel mai mare nivel („alb”). Utilizarea setărilor fixe pentru amprentele marcate direct, deplasează „albul” din imagine mult în jos în scara de măsurare, producând valori scăzute ale contrastului. Avem nevoie de un standard pentru rutina de auto-expunere pentru optimizarea luminii reflectate de materialele descrise mai sus.

Chiar și cu o imagine optimă, se conturează o altă problemă la analiza histogrammei imaginii. Deoarece codurile sunt marcate prin procese independente, în general, histograma codurilor reale DPM nu prezintă o distribuție egală sau simetrică între pixelii ce alcătuiesc prim-planul și fundalul.

ISO / IEC 15415 realizează aceasta printr-o metodă simplă bazată pe punctul de mijloc dintre valoarea cea mai întunecată a histogrammei (reflexia minimă) și valoarea ei cea mai luminoasă (reflexie maximă). Desigur, această metodă ar oferi pragul de binarizare corect doar în cazul în care distribuția celor două vârfuri este identică, caz care nu se întâmplă niciodată, nici măcar pentru etichetele din hârtie. Această problemă este mult accentuată dacă se includ toți pixelii din imagine și nu doar pixelii din regiunea de interes (ROI), unde se află amprenta. În mod ideal, imaginea bună a unui cod Data Matrix conține numai trei tipuri de distribuții de valori de gri: prim-plan, fundal, și margini. „Marginile” separă prim-planul de fundal, poate fi și reflexia luminii produsă de modulele marcate prin frezare.

Un pas suplimentar de procesare a imaginii ar consta în aplicarea unui filtru trece jos, astfel s-ar mai putea împrăștia zgomotul din imagine. Dar în același timp s-ar pierde din detaliile imaginii și, în mod natural, și din informație.

Toate aceste abordări ale problemelor de achiziție de imagine încă nu au dus la o definiție finală a soluției optime.

2.8 Parametrii folosiți în verificare

Dacă avem o imagine optimă, o histogramă bine formată și un prag global optim de binarizare, putem începe în cele din urmă analiza diferitelor aspecte ale amprentei. ISO / IEC 15415 a făcut un salt semnificativ în direcția corectă, prin impunerea unei liste de parametri de verificare. Mai jos se dă o scurtă descriere a parametrilor utilizați în măsurarea amprentelor Data Matrix. Multe dintre aceste măsurări folosesc conceptul de grilă ideală. Aceasta este o matrice de segmente de linii plasate la distanțe egale, formată pe baza celor patru colțuri ale amprentei, divizând întreaga matrice la numărul de linii și coloane. Acești parametri sunt:

- unghiul de distorsiuni - abaterea unghiului creat de cele două margini solide ale codului Data Matrix față de 90°;

- neuniformitatea axială - diferența dintre înălțime și lățime cu privire la rândurile și coloanele amprente;
- contrastul modulelor - contrastul creat de populația modulelor pline și cea a modulelor goale;
- dimensiunea modulului - lățimea amprente împărțită la numărul de coloane sau de rânduri ale amprente;
- abaterea centrelor modulelor - diferența liniară de la centrul celulei, comparativ cu centrul rețelei ideale;
- abaterea dimensiunii modulelor - diferența între dimensiunile aparente ale fiecărui element din amprenta codului Data Matrix;
- neuniformitatea grilei - diferența dintre grila măsurată în raport cu grila ideală formată din cele patru colțuri ale Codului Data Matrix;
- dimensiunea matricei - dimensiunea totală a amprente măsurată liniar pe lățime sau înălțime;
- reflexia minimă - reflexia cea mai scăzută a oricărei probe din amprenta codului Data Matrix;
- modularea - în ISO / IEC 15415 este o măsură de uniformitate a culorii zonelor întunecate și zonele luminoase ale codului Data Matrix;
- factorul de umplere - procentul de completare a grilei ideale;
- dimensiunea nominală a modulelor - mărimea X a unui modul din amprentă;
- ovalitatea modulelor - diferența dintre axa mare și axa mică a unui modul;
- diferența de mărime în urma marcării sau imprimării - diferența pozitivă sau negativă a modulelor marcate sau imprimate față de grila ideală.

Cu toate acestea, inflexibilitatea condițiilor din jur au redus utilitatea lor. Metodele de calcul pentru acești parametrii oferă, de asemenea, limitări care ar putea crea probleme atât pentru codurile marcate direct cât și pentru cele printate pe hârtie.

Capitolul 3

Metode de recunoaștere a codului Data Matrix

3.1 Achiziția codului

Localizarea unui cod de bare 2-D, rapid și precis într-o imagine cu fundal complex, iluminare slabă sau alte condiții nedorite este crucială pentru procesul de recunoaștere. În subcapitolul 2.6 am arătat că iluminarea joacă un rol important în procesul de captare a imaginii. Prin studiu, am demonstrat că folosirea unei surse de lumină orientată la un unghi de 45° cu suprafața amprenteii oferă cea mai bună histogramă a imaginii [23]. Acesta este un avantaj în procesul de prelucrare dar, în același timp, un dezavantaj pentru binarizarea imaginii. Sursa de lumină fiind poziționată lateral cu piesa ce conține codul Data Matrix marcat direct pe material, o parte din imagine va fi supra-expusă, iar cealaltă va fi întunecată. De această problemă s-a lovit și Huijuan Yang în [24] unde dorește să achiziționeze o imagine cu cod PDF417, tipărit pe o etichetă. Calitatea slabă a imaginilor din cauza zgomotului, iluminarea neuniformă face ca sarcina de binarizare să fie dificilă. Tehnicile globale de binarizare utilizate, de cele mai multe abordări nu funcționează bine pentru imaginile achiziționate cu coduri de bare în condiții necontrolate. Huijuan Yang propune binarizarea imaginii cu un prag dinamic adaptiv, astfel rezultatele imaginii fiind satisfăcătoare. Metodele de binarizare sunt multe, în funcție de diferite criterii, dar în cazul achiziției codurilor Data Matrix, dorim să avem în același timp viteză și calitate mare la binarizare. În [25] se întâlnește problema prezentată mai sus chiar la imaginile codurilor Data Matrix marcate direct pe material. Pe baza analizei Wavelet a imaginii, se estimează distribuția luminii în imagine, astfel această variație se elimină direct din imaginea sursă în conformitate cu formarea imaginii model. În urma acesteia, imaginea este binarizată folosind metoda lui Otsu cu un prag global de binarizare [26]. Metoda prezentată mai sus face, practic, un contrast automat adaptiv al imaginii. Dar este posibil ca valori ale pixelilor aflate apropiat de extremele scării de gri să fie anulate sau saturate. Astfel, în imagine există o pierdere de informație, neacceptabilă în cazul codurilor când dorim să extragem toată informația codată. În [27] pentru scanarea și foto-copiarea documentelor, se combină binarizarea cu prag dinamic adaptiv, pragul fiind calculat cu metoda lui Otsu.

3.2 Localizarea codului

Pentru localizarea amprenteii codului Data Matrix și extragerea colțurilor și a amprentelor „L”, au fost propuși mai mulți algoritmi de localizare. Hu Donghong în [28] a propus un algoritm bazat pe transformarea Radon [29], care în general localizează codul Data Matrix după forma „L” a amprenteii de identificare și după amprenta de sincronizare. Acest algoritm are o precizie mare și funcționează bine pentru codurile Data Matrix standard de densitate mare, dar consumă foarte mult timp și nu este potrivit de a fi aplicat în timp real.

Zhao Chenguang în [30] a propus un algoritm de localizare bazat pe transformarea Hough [31]. Acest algoritm consumă foarte mult timp și spațiu de lucru. Deși se poate reduce consumul de timp al transformării Hough la o secundă, ce ar fi mai rău este că acest algoritm are o precizie scăzută pentru fundalul complex.

Cai Wenting [32] a discutat despre metoda de poziționare a codurilor Data Matrix

pe baza algoritmului Convex [33], care determină cele 3 colțuri ale amprentei de identificare „L”, conform punctelor de pe marginea codului. Acest algoritm este simplu și rapid, dar cere ca fundalul să fie curat iar codul să nu aibă zgomot și să fie complet. Există alți algoritmi de localizare [34] [35], dar sunt potriviți numai pentru situații specifice, cum ar fi fundal simplu, condiții bune de iluminare, etc. În realitate, imaginile codurilor Data Matrix întotdeauna însoțite de fundal complex sau calitatea codurilor este redusă. În aceste condiții nedorite, cei mai mulți dintre algoritmi menționați mai sus nu funcționează în mod eficient sau cererea de putere de procesare și de spațiu de stocare nu poate satisface aplicațiile în timp real. Problema de localizare a codurilor Data Matrix nu este deplin rezolvată.

În [36] se propune un algoritm de localizare a amprentei codului Data Matrix bazat pe detecția amprentei „L” de identificare și cea de sincronizare. Detecția acestora este realizată în principiu pe detecția și combinarea liniilor folosind algoritmul LSD [37]. Eficiența și viteza metodei sunt demonstrate prin experimente, dar această metodă se folosește pentru detecția codurilor cu amprente formate din module pătratice, ceea ce nu ar funcționa în cazul nostru.

În [38] se propune un algoritm eficient pentru extragerea codului de bare 2D dintr-o imagine preluată de o cameră video aflată în mișcare. Algoritmul binarizează imaginea după care o dilată, iar pe baza tehnicilor de detecție de muchii [39], extrage forma acestei amprente.

Atât pentru codurile Data Matrix standard, cât și pentru cele industriale, estimarea orientării amprentei este foarte importantă pentru detecția amprentelor de identificare și de sincronizare și pentru scanarea codului. În literatură sunt implementate câteva metode pentru detecția unghiului amprentei codului Data Matrix. În [40] este prezentată o metodă de localizare și de estimare a orientării amprentei codului Data Matrix standard folosind transformarea Hough. După ce amprenta a fost localizată, sunt extrase muchiile amprentei codului. Pe baza acestora și folosind transformarea Hough se extrag liniile de pe muchii și se calculează unghiul muchiilor amprentei. Această metodă este folosită în [41] pentru codurile Aztec, PDF417 și Data Matrix standard. Cei mai mulți algoritmi pentru localizarea codului Data Matrix, în literatura de specialitate detectează mai întâi liniile formate de amprentă, folosind transformarea Hough, cu scopul de a obține poziția și unghiul de rotație a amprentei „L”.

Știm din [42] [43] că Algoritmul cu Translație la Medie (Mean Shift) este o tehnică non-parametrică pentru analiza caracteristicilor spațiului, numită tehnică de căutare a maximumului. Acesta localizează maximumul unei funcții de densitate dată de un set de date discrete. Această metodă este folosită și în alinierea 2D a imaginilor [44]. În acest caz, putem spune că algoritmul propus oferă rezultate satisfăcătoare. Scopul acestei analize este de a evidenția metoda cea mai potrivită pentru detecția orientării amprentei.

3.3 Scanarea codului

Cunoscând regiunea de interes, se poate scana amprenta codului pentru extragerea informației dorite. În cazul scanării modulelor amprentei codului industrial, metodele de scanare folosite pentru codurile Data Matrix standard nu mai pot fi folosite sau comparate. În [45] este prezentată o metodă simplă de scanare a codului Data Matrix punctat.

În multe cazuri, din cauza sistemului optic al camerei video și din cauza spațiilor inaccesibile este dificil de a achiziționa o imagine perfect ortogonală a codului Data Matrix industrial. În [40] se folosește transformarea inversă de perspectivă pentru a reface imaginea amprentei codului. Această operație implică metoda de interpolare ce

acționează direct asupra calității imaginii. În cazul în care o imagine este achiziționată slab calitativ, această metodă poate diminua și mai mult calitatea imaginii.

De exemplu, în această direcție s-au mai făcut cercetări după cum urmează: K. Vijayan Asari [46] a folosit un model matematic pentru a descrie omogenizarea imaginii distorsionată a spațiului în spațiul imaginii corecte. Liu Tangyou [47] și Junhee Park [48] au stabilit un model de distorsionare pentru a corecta imaginea deformată pe baza rețelei standard a imaginii, care a corectat imaginile în aceleași condiții în funcție de parametrii de deformare a rețelei standard. Pentru corectarea distorsionării codului de bare bi-dimensional, Eisaku Ohbuchi [6] și Ming Anlong [7] au utilizat transformarea inversă a perspectivei pentru a corecta eroarea modulelor din cadrul codului QR.

Atunci când amprenta codului Data Matrix este marcată pe o suprafață cilindrică imaginea va fi distorsionată și ar duce la eșec în identificarea acesteia. În [49] autorii propun o metoda numită „rețea elipsă” pentru corectarea imaginii deformată a Codului Data Matrix standard, în scopul de a realiza scanarea și decodarea.

3.4 Concluzii

Metodele tradiționale de recunoaștere și scanare implementate pentru amprentele codurilor Data Matrix standard sunt nepotrivite în cazul de față. Cercetările în această direcție au dus la dezvoltarea unei aplicații de recunoaștere a codurilor destinată în special codurilor Data Matrix industriale.

Pentru localizarea codului Data Matrix, știind că amprenta acestuia este de formă pătratică putem să o vedem ca un pătrat solid, metodă descrisă în lucrarea [50]. Metoda implementată este eficientă, dar necesită a fi parametrizată în funcție de camera video folosită și de mărimea amprentei codului Data Matrix. În [51] am implementat o nouă metodă de localizare a amprentei codului Data Matrix industrial fără a se cere informații despre amprentă sau despre sistemul video folosit. În acest caz, algoritmul folosește informații despre structura amprentei de indentificare a codului.

Pentru estimarea orientării amprentei codului Data Matrix industrial, în [51] am implementat Algoritmul cu Translație la Medie, demonstrându-se eficacitatea lui în estimarea caracteristicilor amprentei.

Pentru scanarea codului, pornind de la transformarea geometrică inversă, dar dorind să nu afectăm imaginea, se creează o grilă de coordonate cu același defect de perspectivă sau cu aceeași curbura. Pe baza acestei metode prezentată detaliat în lucrările [52] [53] [54], se efectuează scanarea modulelor amprentei codului Data Matrix.

Toate aceste metode folosite până în prezent pentru achiziția, localizarea și scanarea codurilor Data Matrix standard au fost punctele de start pentru implementarea de metode noi, aplicabile codurilor industriale. Aceste noi metode sunt implementate și prezentate detaliat în această teză, ele fiind comparate cu metodele folosite pentru codurile Data Matrix standard.

Capitolul 4

Localizarea codurilor Data Matrix

Acest capitol propune două soluții de localizare pentru codurile Data Matrix marcate pe diferite tipuri de materiale și cu orientări diferite. Sistemele de localizare pentru codurile Data Matrix Industriale pot fi semi-automate sau automate. Sistemele de localizare semi-automată pot fi folosite în industrie, unde operatorul uman cunoaște informații despre coduri și are posibilitatea de a configura sistemul de scanare și recunoaștere. Sistemele de localizare automată pot fi folosite pentru diverse aplicații gen cititoare portabile sau telefoane inteligente unde nu se cunosc informații despre cod.

4.1 Sisteme de localizare semi-automată

Știind dimensiunea reală a amprentei Data Matrix și folosind parametrii camerei video a sistemului de localizare, metoda dezvoltată poate localiza poziția exactă și orientarea amprentei în imagine. Se folosește o metodă de binarizare cu prag adaptiv pentru ca sistemul să nu depindă de variațiile de iluminare sau de fundal neuniform [55]. Datorită faptului că amprenta codului Data Matrix este compusă numai din puncte (module), este dificil să se recunoască poziția ei în imagine [17]. Pentru a trece peste acest lucru, se folosesc operatorii morfologici pentru a transforma amprenta într-un pătrat solid.

4.1.1 Prezentarea sistemului de localizare semi-automată

Sistemul de localizare video are în componența sa o cameră video, sistemul de iluminare și un sistem hardware în care se rulează programul de localizare și prelucrare video. Programul de localizare permite parametrizarea sistemului de localizare cu caracteristicile camerei video folosite, de exemplu, distanța focală, rezoluția, dimensiunea senzorului, precum și caracteristicile codului Data Matrix, cum ar fi mărimea lui în spațiul real. Camera video este conectată la computer. *Blocul de Localizare* procesează în timp real, cadrele furnizate de camera video determinând poziția și orientarea amprentei Data Matrix. Acest bloc este conectat la *Interfața video* și *Blocul de scanare*, așa cum este prezentat în Figura 4.1. *Sursa de lumină* este montată pe carcasa camerei video, astfel încât lumina cade pe suprafața codului sub un anumit unghi, diferit de 90° . În [20], [23] se explică considerațiile, după care se alege o gamă de unghiuri potrivită pentru procesul de recunoaștere. În continuare se prezintă pe scurt Blocul *Interfeței video*.

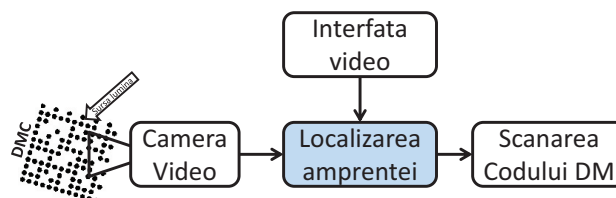


Figura 4.1: Schema bloc a sistemului de localizare

4.1.2 Interfața video

Interfața video este legătura directă dintre utilizator și sistemul de localizare, afișând rezultatul scanării codului Data Matrix. Acest bloc, de asemenea, permite parametrizarea sistemului de localizare pentru diferite tipuri de camere video și diferite dimensiuni ale amprentelor codului Data Matrix. Deoarece acest sistem este utilizat în mediul industrial putem selecta câteva caracteristici ale camerei video folosite cum ar fi: *dimensiunea sensorului, rezoluția sensorului, distanța focală a obiectivului utilizat* și informații despre amprentă (*dimensiunea amprentei și distanța aproximativă dintre cod și camera video*). Folosind aceste informații se poate calcula dimensiunea aproximativă în pixeli a amprentei codului Data Matrix [56] [57]. Acest calcul este o estimare pentru dimensiunea amprentei, ajutând să se restrângă aria de căutare a codului Data Matrix. Folosind ecuațiile (4.1) - (4.4), se poate calcula lungimea și lățimea, în pixeli, a amprentei Data Matrix în imagine astfel:

$$B = b \cdot \frac{G}{g}, \quad (4.1)$$

$$L_u = B \cdot w, \quad (4.2)$$

$$L_a = B \cdot h, \quad (4.3)$$

$$I = L_u \cdot L_a, \quad (4.4)$$

unde:

G este lungimea sau lățimea amprentei codului Data Matrix (cm),

B este lungimea sau lățimea proiecției codului Data Matrix pe senzorul de imagine (cm),

g este distanța dintre cod și camera video (cm),

b este distanța între centrul optic al obiectivului și senzor (la focalizare pe infinit corespunde cu distanța focală) (cm),

s este lungimea diagonalei senzorului de imagine (cm),

w este rezoluția senzorului pe verticală (pixeli/cm),

h este rezoluția senzorului pe orizontală (pixeli/cm).

I este aria în pixeli.

4.1.3 Localizarea amprentei Data Matrix

Etapa de pre-procesare este folosită pentru localizarea amprentei codului Data Matrix, fără a avea interes în detaliile imaginii. Pentru a identifica poziția corectă a regiunii de interes (*ROI*), se utilizează informații cu privire la forma amprentei codului Data Matrix [58].

Dacă se urmărește schema funcțională a sistemului de localizare (Fig. 4.2), se poate observa că imaginea este captată și sub-eșantionată folosind un raport de sub-eșantionare. În cazul în care este ales un raport mare de sub-eșantionare, imaginea este micșorată mult, viteza de pre-procesare fiind ridicată. Acesta este un obiectiv pentru sistemul de localizare, dar, pe de altă parte, caracteristicile obiectului

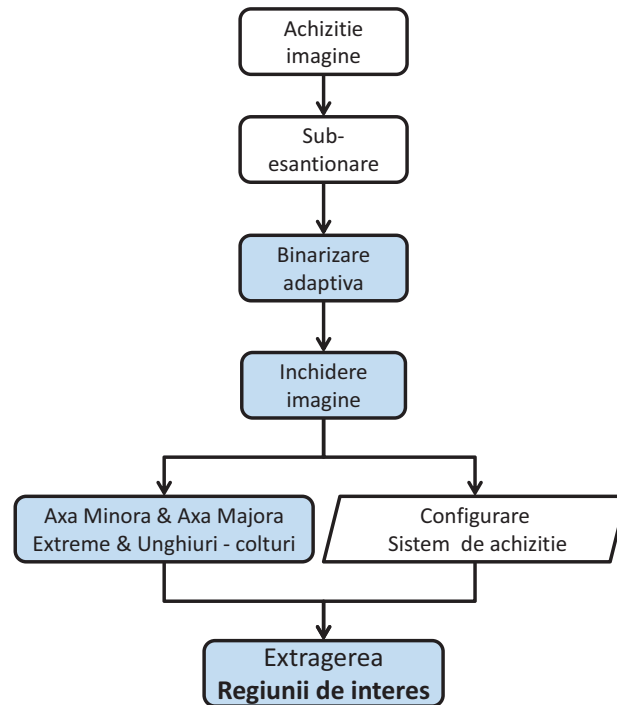


Figura 4.2: Procesul de localizare a amprentei Data Matrix

sunt reduse considerabil. Din această cauză, raportul de sub-eșantionare este ales manual de către operator, în funcție de dimensiunea amprentei Data Matrix în spațiul real.

4.1.3.1 Binarizarea imaginii cu metoda lui Otsu cu prag dinamic-adaptiv

Discriminarea cu prag este una din cele mai des utilizate metode de segmentare a imaginilor [59]. Datorită simplității ei, această metodă poate fi implementată în timp real. Comparând nivelurile de gri la fiecare locație din imagine cu un prag unic se generează o imagine binară, conform relației (4.9). Metoda permite separarea obiectelor luminoase așezate pe un fundal întunecat sau invers. Problema principală ce apare la discriminarea cu prag este selecția corectă a pragului de binarizare th_x . Procedeu uzual este de a folosi în acest scop histograma imaginii [60]. Dacă pragul este ales corespunzător și imaginea are o histogramă bimodală (cu două maxime locale: unul pentru fundal și unul pentru obiecte), se va obține o imagine binară cu valori de 1 pentru obiecte și 0 pentru fundal [61]. Binarizarea poate fi:

- cu prag global, cu un singur prag t pentru întreaga imagine f , $t = t(f)$,
- cu prag dinamic depinzând și de coordonatele x, y , $t(x, y) = t(f, f_i)$.

Iluminarea neuniformă combinată eventual cu alți factori (neomogenitatea obiectului și a fundalului de exemplu) duce la segmentarea incorectă a obiectului de fundal pentru orice valoare a pragului aleasă.

O soluție poate fi segmentarea imaginii folosind un prag dinamic adaptiv [62] [55]. În funcție de mărimea amprentei estimată în pixeli, imaginea gri este împărțită în

regiuni. În lucrarea prezentă s-a potrivit a fi avantajos ca mărimea regiunilor să fie aproximativ egală cu mărimea amprentei, Fig. 4.3. Folosind metoda lui Otsu, pentru fiecare regiune se calculează un prag de binarizare local [26]. Metoda lui Otsu caută pragul ce minimizează varianța între două clase de pixeli, ce este definită ca suma ponderată a varianțelor a celor două clase:

$$\sigma_W^2(t) = W_b(t) \cdot \sigma_b^2(t) + W_f(t) \cdot \sigma_f^2(t), \quad (4.5)$$

unde W_b, W_f denotă probabilitățile a două clase separate printr-un prag t , și σ_b^2, σ_f^2 indică varianțele acestor clase. Otsu a demonstrat că reducerea la minim a varianței între clase este același lucru cu maximizarea varianțelor extra-claselor:

$$\sigma_B^2(t) = \sigma^2 - \sigma_W^2(t) = W_b(t) \cdot W_f(t) \cdot (\mu_b(t) - \mu_f(t))^2, \quad (4.6)$$

ce este exprimată în termenii probabilităților claselor W_b, W_f și mediile claselor μ_b, μ_f , care la rândul lor pot fi actualizate iterativ. Se maximizează relația (4.6), pentru a obține pragul de binarizare dorit. Procedura metodei lui Otsu poate fi descrisă după cum urmează [27]:

- calculează histograma și estimează din aceasta probabilitățile fiecărei valori a intensității pixelilor,
- setează valorile inițiale ale $W_b(0), W_f(0)$ și $\mu_b(0), \mu_f(0)$,
- caută în buclă pentru toate pragurile de binarizare posibile t ,
- actualizează $W_b(t), W_f(t)$ și $\mu_b(t), \mu_f(t)$,
- calculează $\sigma_B^2(t)$ și alege pragul t^* corespunzător maximului $\sigma_B^2(t)$.

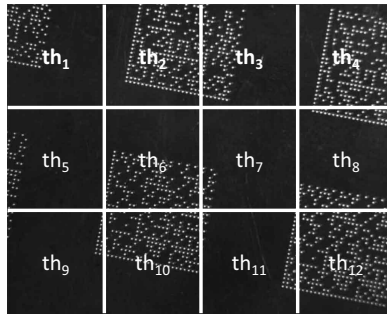


Figura 4.3: Niveluri de binarizare locale

Prin folosirea tuturor nivelurilor pragurilor este creată o matrice de praguri locale, relația (4.7).

$$Th = \begin{pmatrix} th_1 & th_2 & th_3 & th_4 \\ th_5 & th_6 & th_7 & th_8 \\ th_9 & th_{10} & th_{11} & th_{12} \end{pmatrix} \xrightarrow{\text{interpolare}} \begin{pmatrix} th_1 & \dots & th_w \\ \vdots & \dots & \vdots \\ th_{(w-1) \cdot h} & \dots & th_{w \cdot h} \end{pmatrix} \quad (4.7)$$

Utilizând metoda de interpolare biliniară, matricea pragurilor locale este extinsă la o matrice de praguri de mărimea rezoluției imaginii [63]. O valoare interpolată este

calculată pe baza mediei ponderate a patru valori vecine, situate pe grila eșantionată a matricei de intrare, utilizând relația următoare:

$$f_{(x,y)} = (1-\alpha) \cdot (1-\beta) \cdot f_{(|x|,|y|)} + \alpha \cdot (1-\beta) \cdot f_{(|x|+1,|y|)} + (1-\alpha) \cdot \beta \cdot f_{(|x|,|y|+1)} + \alpha \cdot \beta \cdot f_{(|x|+1,|y|+1)}, \quad (4.8)$$

unde α și β sunt partea fracționară a coordonatelor x și y .

Procedura de interpolare biliniară pentru determinarea pragului dinamic de binarizare este ilustrată în Fig. 4.4 [64]. Interpretarea grafică se bazează pe faptul că interpolarea liniară 2D se poate obține prin interpolări 1D succesive pe direcțiile axelor de coordonate [61]. Prin procesul de mediere se obține o binarizare mai netedă (Fig. 4.4).

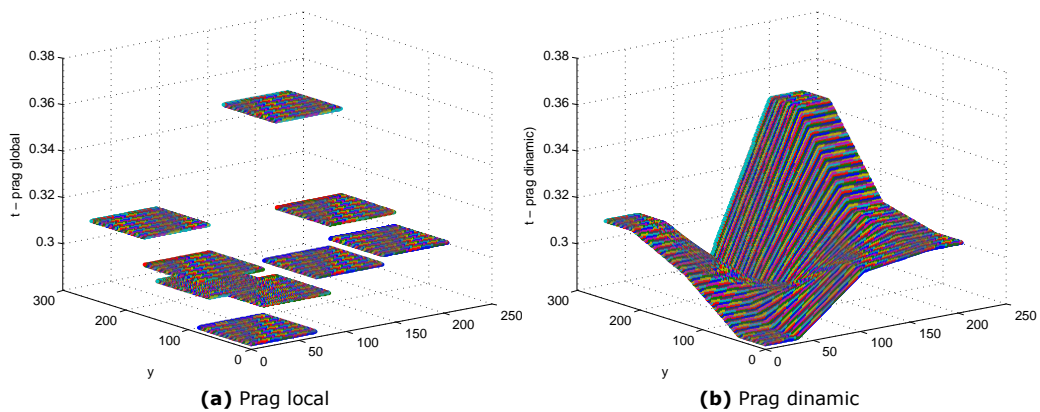


Figura 4.4: Determinarea pragului de binarizare

Fiecare pixel din imaginea gri este binarizat folosind un prag dinamic de binarizare preluat din matricea extinsă a pragurilor, așa cum este prezentat în cazurile următoare [63].

$$BW = \begin{cases} 1, & \text{pentru } \text{Img} \geq th_{xy} \\ 0, & \text{pentru } \text{Img} < th_{xy} \end{cases} \quad (4.9)$$

Folosind metoda de binarizare cu prag dinamic adaptiv se crează imaginea binară a imaginii codului Data Matrix achiziționată.

Pentru etapele următoare de prelucrare a imaginii este de dorit ca fundalul să fie negru (nivel 0) și planul principal să fie alb (nivelul 1). Din moment ce fiecare amprentă Data Matrix este înconjurată de o zonă curată, în imagine (Regiunea de Interes) există mai mult fundal decât prim-plan. Astfel, prin numărarea valorilor zerourilor și cele de unu din Regiunea de Interes (ROI), se poate afla nivelul actual al fundalului și dacă este necesar, se negativează imaginea pentru a avea un fundal de nivel zero [63].

4.1.3.2 Recunoașterea formei amprentei Data Matrix folosind Morfologie matematică

În imagine mai pot exista: zgomot, pete de lumină sau alte obiecte neapartinătoare codului Data Matrix. Deoarece obiectul căutat este compus numai din module

(puncte), este dificil să se identifice poziția acestuia în imagine. Pentru a recunoaște amprenta Codului Data Matrix, forma ei ar trebui să fie văzută ca un pătrat solid, nu ca o matrice de puncte (module).

Dilatarea și erodarea sunt operații morfologice elementare ce oferă metode de prelucrare de mare eficacitate în regularizarea și analiza formelor imaginilor binare [65]. În cazul imaginilor codurilor Data Matrix, folosind operația morfologică de dilatare, imaginea este dilatată în scopul de a umple spațiile goale dintre module (Fig. 4.5). Elementul structurator poate fi considerat un disc cu raza egală cu distanța dintre module. Această valoare este aleasă în funcție de distanța între module, care poate fi estimată prin împărțirea mărimii amprentei în pixeli a codului la numărul de module.

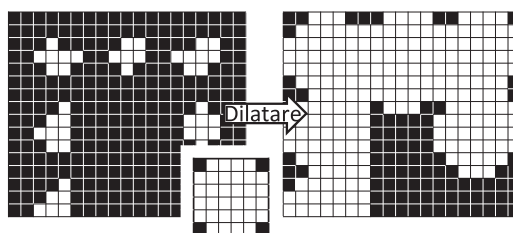


Figura 4.5: Dilatarea imaginii Data Matrix

Procesul de dilatare se efectuează prin deplasarea pe imaginea A a elementului structurator B în manieră asemănătoare convoluției [66]. În cazul în care un pixel din elementului structurator B coincide cu un pixel „alb” din imagine, se schimbă originea elementului structurator în „alb” [63]. Fie E un spațiu Euclidian sau o rețea de valori întregi, A o imagine binară în spațiul E , și B un element structurator. Dilatarea unui set A prin elementul structurator B este definită prin ecuația:

$$A \oplus B = \{z \in E | (B^S)_z \cap A \neq \emptyset\}, \quad (4.10)$$

unde B^S denotă simetricul lui B , fiind:

$$B^S = \{x \in E | -x \in B\}.$$

Efectul generat de operația de dilatare este de a extinde obiectele în imagine. Obiectele din imagine, cu distanța între ele mai mică decât raza elementului structurator, sunt conectate între ele, construind forme solide, în cazul codului Data Matrix un pătrat solid. În imaginea dilatată obiectele sunt extinse de jur împrejur cu valoarea distanței dintre module.

Prin utilizarea operației de erodare, obiectele sunt redimensionate la dimensiunea lor inițială.

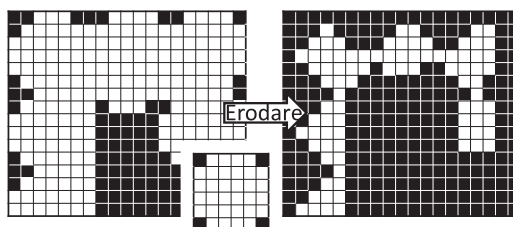


Figura 4.6: Erodarea imaginii Data Matrix

Procesul de erodare este similar cu cel de dilatare, dar pixelii sunt schimbați în „negru” nu în „alb”. Ca și înainte, elementul structurator este deplasat peste imagine. Dacă cel puțin unul dintre pixelii din elementul structurator se suprapune cu un pixel „negru” din imagine, atunci pixelul „alb” din imagine care coincide cu centrul elementului structurator este schimbat în „negru” [63]. Dacă toți pixelii din elementul structurator se suprapun cu pixeli „albi” în imagine, atunci pixelul care coincide cu originea elementului structurator nu este schimbat, iar elementul structurator se deplasează la următorul pixel. Erodera unei mulțimi A prin elementul structurator B este definită prin:

$$A \ominus B = \{z \in E | B_z \subseteq A\}, \quad (4.11)$$

unde B_z este translatatul lui B prin vectorul z ,

$$B_z = \{b + z | b \in B\}, \forall z \in E.$$

Efectul generat de operația de erodare este de a subția obiectele din imagine. Cele două operații aplicate împreună se numesc închiderea imaginii (Fig. 4.7).

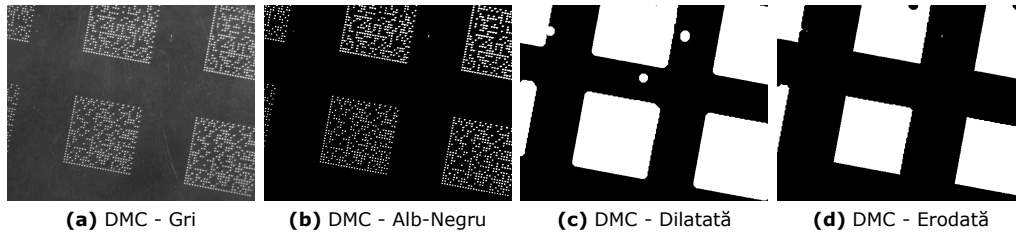


Figura 4.7: Închiderea imaginii codului Data Matrix (DMC)

Imaginea $f(x, y)$ poate fi modelată prin densitatea de probabilitate a unei variabile aleatoare bidimensionale, pentru care este uzuală caracterizarea statistică prin momente [67], iar $f(x, y)$ ia valorile 0 și 1 corespunzător cu „negru” și „alb”. Momentul de ordinul $p + q$ este dat în imagini discrete de relația:

$$m_{pq} = \sum \sum x^p \cdot y^q \cdot f(x, y). \quad (4.12)$$

Centroidul sau centrul de greutate al regiunii este punctul de coordonate:

$$\begin{cases} x_c = \frac{\mu_{10}}{\mu_{00}} \\ y_c = \frac{\mu_{01}}{\mu_{00}} \end{cases}, \quad (4.13)$$

cu ajutorul centroidului, se exprimă momentele centrate.

Pentru imaginile numerice, momentul central de ordinul $p + q$ este:

$$\mu_{pq} = \sum \sum (x - x_c)^p \cdot (y - y_c)^q \cdot f(x, y). \quad (4.14)$$

Pentru imagini binare m_{00} este aria regiunii. În cazul imaginii codului Data Matrix, se pot calcula axa mare și axa mică pentru fiecare obiect (Fig. 4.8), folosind ecuațiile (4.15) - (4.16) [68].

Unghiul α al axelor mari și mici și deci orientarea obiectului se calculează cu:

$$\alpha = \frac{1}{2} \cdot \arctan \frac{2\mu_{11}}{\mu_{20} - \mu_{02}}, \quad (4.15)$$

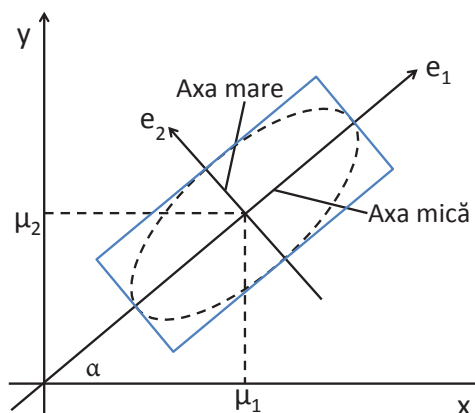


Figura 4.8: Axa mare și axa mică a unui obiect din imagine

iar ecuația axei mari este:

$$y = (x - x_c) \tan \alpha + y_c. \quad (4.16)$$

Pentru fiecare obiect din imagine se iau coordonatele maxime și minime ale pixelilor aparținători și se extrag cele patru extreme. Prin aceste puncte se trasează vectori imaginari, obținându-se unghiurile create de acești vectori cu axele principale ale imaginii. Intersectând doi câte doi vectori se calculează unghiurile dintre aceștia, folosind ecuațiile 4.17 - 4.20. Acest lucru se poate vedea în Fig. 4.9.

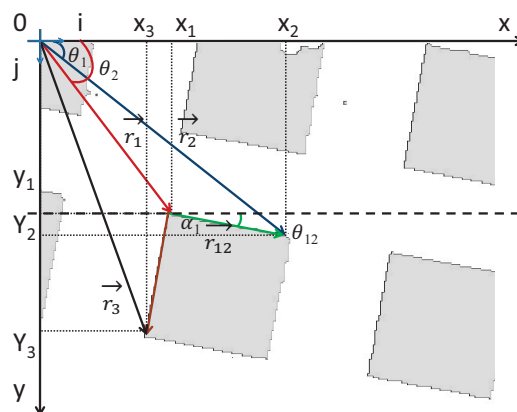


Figura 4.9: Calcularea unghiului dintre vectori

$$\vec{r}_1 = x_1 \cdot \vec{i} + y_1 \cdot \vec{j}, \quad (4.17)$$

$$\vec{r}_2 = x_2 \cdot \vec{i} + y_2 \cdot \vec{j}, \quad (4.18)$$

$$\vec{r}_{12} = \vec{r}_2 - \vec{r}_1, \quad (4.19)$$

$$\cos \alpha_1 = \frac{\vec{r}_1 \cdot \vec{r}_2}{|\vec{r}_1| \cdot |\vec{r}_2|}. \quad (4.20)$$

Se știe că amprenta codului Data Matrix este definită de un pătrat și, de asemenea, se știe mărimea estimativă a amprentei. Obiectul căutat în imagine trebuie să îndeplinească următoarele condiții:

- toate laturile amprentei trebuie să fie egale,
- laturile adiacente trebuie să fie ortogonale,
- mărimea amprentei trebuie să fie aproximativ egală cu valoarea estimată.

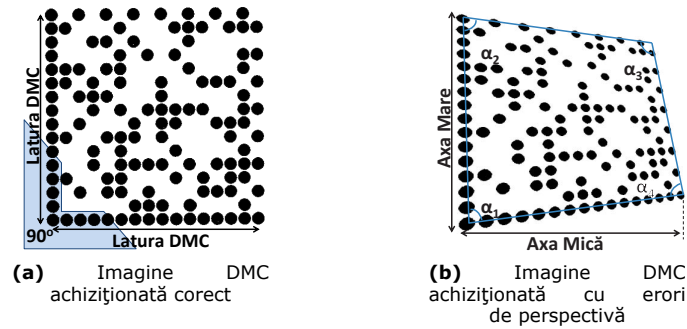


Figura 4.10: Cod Data Matrix industrial

Se caută în imagine doar obiectele ce îndeplinesc condițiile de a fi un pătrat. Din cauza erorilor de perspectivă [69], proiecția codul Data Matrix pe senzorul de imagine poate să nu fie un pătrat. Ar putea fi un patrulater convex, ca în Fig. 4.10 (b), și pentru a depăși această lucră, se folosește o toleranță pentru unghiuri și laturi. Prin intersectarea seturilor de caracteristici se obține regiunea de interes (Fig. 4.11), ce îndeplinește condiția impusă de a fi asemănătoare cu codul Data Matrix căutat.

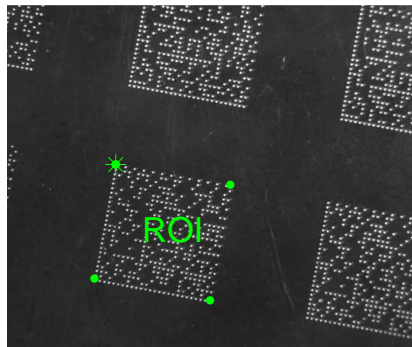


Figura 4.11: Regiunea de interes - ROI

4.1.4 Estimarea distanței dintre module și a suprafeței acestora folosind histograma

Histograma este una din cele mai vechi și mai simple modalități de reprezentare statistică a unui set de date X_i . Aceasta constă în găsirea probabilităților asociate fiecărei valori din domeniul de definiție a variabilei X_i . Pentru obținerea estimatorului trebuie în primul rând împărțit domeniul funcției în intervale egale, după care se numără datele din fiecare interval [70].

Pentru un interval de dimensiune h estimatorul histogramei este de forma următoare:

$$\hat{f}(x) = \frac{1}{nh} \cdot n_p, \quad (4.21)$$

n_p reprezentând numărul de puncte din interval.

Factorii care influențează forma histogramei sunt atât punctul de origine a intervalului, cât și dimensiunea intervalului. Controlul netezimii curbei de histogramă este dat de dimensiunea intervalului. Această dependență a formei histogramei de cei doi factori este unul din dezavantajele folosirii ei.

Având regiunea de interes, aceasta este proiectată în imaginea inițială pentru a continua cu calculul distanței dintre module. Pentru aceasta, se calculează în imaginea binarizată, și nu în cea dilatată, centrul de greutate al tuturor obiectelor care corespund, de obicei modulelor. Folosind expresia (4.22) sunt calculate distanțele Euclidiene dintre centrul fiecărui modul și patru vecini ai săi, Fig. 4.12.

$$d = \sqrt{(x_M - x_V)^2 + (y_M - y_V)^2}. \quad (4.22)$$

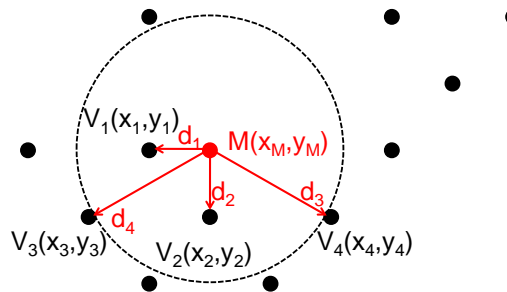


Figura 4.12: Calculul distanței dintre module

În Fig. 4.13 (a) este redat un exemplu în care histograma distanțelor este obținută prin divizarea intervalului de valori în 500 subintervale (coloane). Se remarcă faptul că histograma are un aspect neregulat, datorită numărului finit de eşantioane, în combinație cu efectele zgometului din imagine.

În mod evident, histograma din Fig. 4.13 (a) conține o mulțime de maxime (moduri) locale. Pentru reducerea numărului de moduri se măresc intervalele, respectiv se reduc numărul de coloane ale histogramei. În Fig. 4.13 (b) este dat un exemplu de histogramă cu 50 de coloane pentru distanțe.

Prin alegerea maximului histogramei distanțelor (Fig. 4.13), s-a implementat o metodă robustă pentru estimarea distanței dintre module.

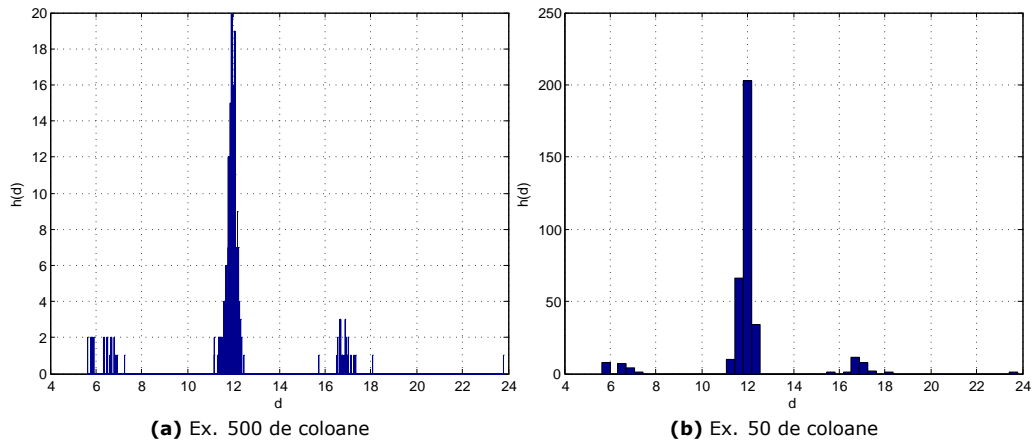


Figura 4.13: Histograma distanțelor

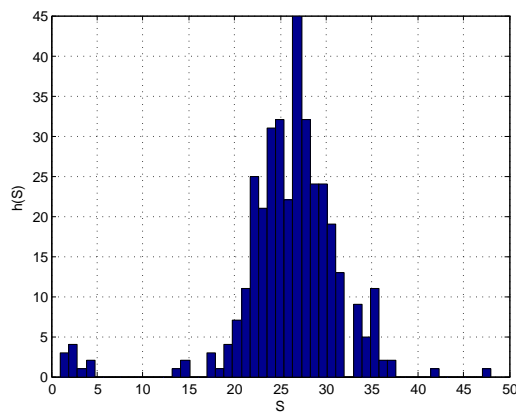


Figura 4.14: Histograma suprafețelor

Pentru estimarea suprafeței modulelor codului Data Matrix, se calculează aria fiecărui obiect din imagine și se scriu valorile într-o matrice a suprafețelor. Pentru calcularea suprafeței unui obiect, se numără toți pixelii învecinați ce au aceeași etichetă. Acest lucru este prezentat detaliat în Capitolul 4.2.1. Se creează histograma suprafețelor obiectelor iar vârful histogramei reprezintă suprafața majorității obiectelor din imagine (Fig. 4.14).

4.1.5 Selecția modulelor corecte

Selecția modulelor amprentei codului Data Matrix are scopul de a elimina zgomotul și obiectele ce nu fac parte din codul Data Matrix. Această selecție se efectuează în două etape:

- se elimină obiectele din imagine care nu au vecini la distanța estimată,
- se elimină obiectele din imagine care nu au suprafața estimată.

Se scanează imaginea binarizată și se construiește o matrice cu coordonatele tuturor obiectelor din imagine. Cunoșcând distanța dintre module, se începe calcularea distanței de la centrul de greutate al fiecărui modul la cel mai apropiat vecin din imagine. În cazul în care obiectul real are un vecin la distanța estimată, înseamnă că acel obiect este un modul și aparține amprentei codului Data Matrix. Se creează o matrice cu coordonatele obiectelor selectate pe baza distanței dintre module.

Pentru selecția pe baza suprafeței estimate, se scanează în imaginea binară numai obiectele care sunt marcate în matricea modulelor selectate pe baza distanței. Având aceste informații, se calculează varianța suprafețelor obiectelor din imagine. Se selectează numai obiectele care au varianța maximă a suprafețelor din imagine, dar și obiectele cu o varianță mai mică și cu o varianță mai mare cu o treaptă. Toate coordonatele sunt scrise într-o matrice nouă, care reprezintă lista modulelor codului Data Matrix.

Această operație de selecție poate elimina, de asemenea, unele module valide din imagine, dar chiar dacă se întâmplă acest lucru, scopul este de a găsi colțurile amprentei Data Matrix și de a se cunoaște dacă regiunea de interes detectată este cea reală.

În Fig. 4.15, este prezentat rezultatul procesului de selecție pentru amprenta codului Data Matrix.

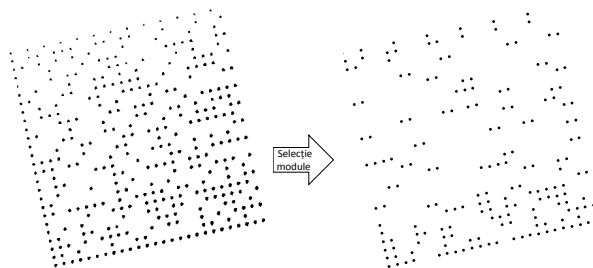


Figura 4.15: Selecția modulelor corecte

4.1.6 Calculul orientării amprentei folosind transformarea Hough

4.1.6.1 Transformarea Hough

Transformarea Hough este o tehnică de extragere de caracteristici folosită în analiza și prelucrarea imaginilor [71]. Scopul tehnicii este de a găsi exemplele imperfecte ale obiectelor într-o anumită clasă a formelor dată de o procedură de votare. Această procedură de votare este efectuată într-un spațiu de parametri, în care obiectele candidate sunt obținute ca și maxime locale într-un spațiu numit acumulator, ce este în mod explicit construit de către algoritm, pentru calcularea transformării Hough.

Cazul cel mai simplu al transformării Hough este transformarea liniară pentru detecția liniilor drepte [71]. În spațiul imagine, linia dreaptă poate fi descrisă de ecuația: $y = mx + n$ și poate fi afișată grafic pentru fiecare pereche de puncte (x, y) din imagine. În transformarea Hough, o idee principală este de a considera caracteristicile liniei drepte nu ca puncte din imagine, $(x_1, y_1), (x_2, y_2)$, ci ca parametrii unei drepte, (m, n) , panta și ordonata în origine. Bazat pe acest fapt, linia dreaptă $y = mx + n$ poate fi reprezentată ca un punct (m, n) în spațiul parametrilor. Cu toate acestea, ne confruntăm cu problema că liniile verticale dau naștere de valori infinite ale parametrului m . Din motive de calcul, este recomandată folosirea unei perechi de parametri diferită, notată cu r și θ , pentru liniile din transformarea Hough.

Parametrul r reprezintă distanța dintre linie și origine, adică lungimea perpendicularei din origine pe linie, în timp ce θ este unghiul acesteia. Folosind această parametrizare, ecuația liniei poate fi scrisă sub forma: [71]

$$y = -\frac{\cos\theta}{\sin\theta} \cdot x + \frac{r}{\sin\theta}, \quad (4.23)$$

ce poate fi rearanjată astfel: $r = x \cdot \cos\theta + y \cdot \sin\theta$.

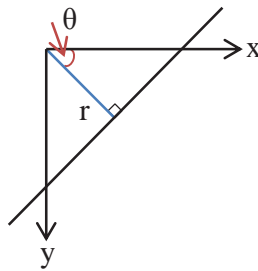


Figura 4.16: r, θ - parametrii liniei

Algoritmul transformării Hough folosește o matrice, numită acumulator, pentru a detecta existența unei linii $r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$. Dimensiunea acumulatorului este egală cu numărul de parametri necunoscuți ai problemei transformării Hough. Problema liniară a transformării Hough are doi parametri necunoscuți: r și θ . Cele două dimensiuni ale matricii acumulator ar corespunde cu valorile cuantificate pentru r și θ (Fig. 4.17). Pentru fiecare pixel și vecinătatea lui, algoritmul transformării Hough determină dacă există suficiente dovezi de o limită, pentru acel pixel. Dacă este așa, se vor calcula parametrii acelei linii, și apoi, se verifică în coloanele acumulatorului dacă parametrul se încadrează în aceștia, și crește valoarea acelei coloane. Găsind coloanele cu valorile cele mai ridicate, de obicei prin căutarea maximelor locale în spațiul acumulator, pot fi extrase liniile cele mai probabile și pot fi citite definițiile lor geometrice (a se vedea vârful acumulatorului încercuit cu roșu în Fig. 4.17).

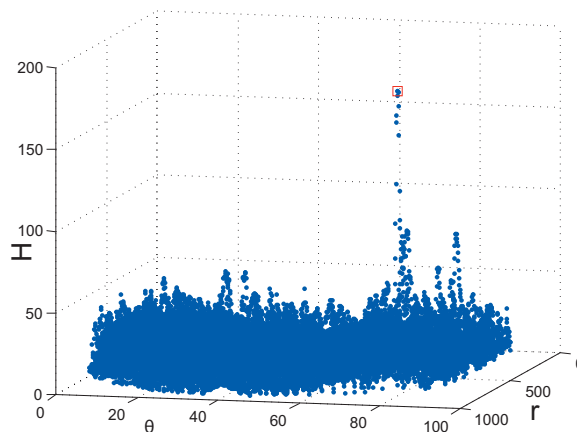


Figura 4.17: Reprezentarea acumulator transformării Hough

Cea mai simplă cale de a găsi aceste vârfuri este aplicarea unor praguri. Diferite

tehnici pot duce la rezultate mai bune în diferite circumstanțe - stabilirea liniilor constatate, precum și numărul lor. Având în vedere că liniile de întoarcere nu conțin nicio informație legată de lungime, este necesar de a afla care părți ale imaginii se potrivesc cu ce linii.

4.1.6.2 Implementarea transformării Hough pentru codurile Data Matrix

Să considerăm imaginea modulelor validate a unui cod Data Matrix. Module validate sunt considerate fără erori. Pentru calculul orientării amprentei codului Data Matrix, se dorește ca datele de intrare ale algoritmului să fie cât mai precise.

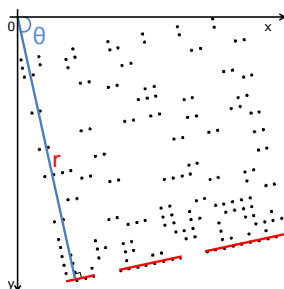


Figura 4.18: Orientarea amprentei codului Data Matrix

Pentru fiecare set de date, sunt reprezentate grafic o serie de linii trecând prin acestea, toate la unghiuri diferite. În Fig. 4.18 sunt reprezentate doar 3 linii. Se pot observa unghiul θ și lungimea r a perpendicularei din originea imaginii pe una din drepte. Pentru fiecare set de două puncte se calculează r și θ , datele fiind scrise într-o matrice numită acumulator, așa cum este arătat în Fig. 4.17. În Fig. 4.19 este prezentată transformarea Hough pentru imaginea codului Data Matrix. Din figură se pot extrage unghiul și distanța liniei corespunzătoare vârfului acumulatorului (încercuit cu alb).

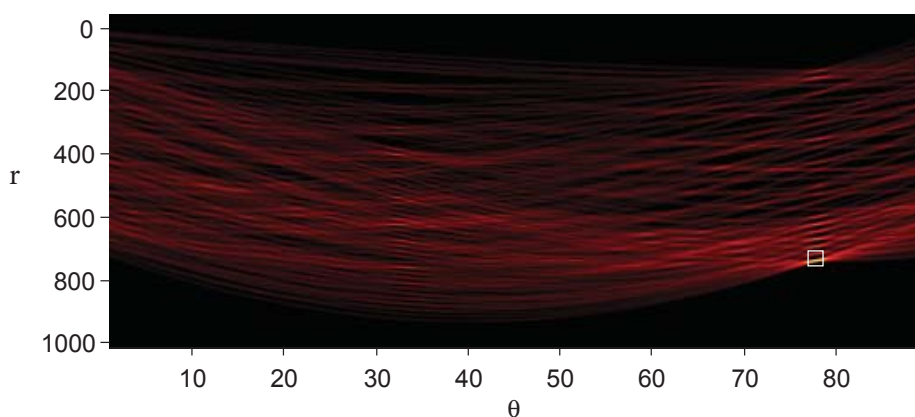


Figura 4.19: Transformarea Hough a imaginii codului Data Matrix

Lungimea și orientarea liniei sunt detectate căutând maximum matricii acumulator (Fig. 4.17). Se folosește ca început de căutare unghiul aproximativ obținut cu ajutorul ecuațiilor (4.15, 4.16).

4.1.7 Recunoașterea amprentei codului Data Matrix

Pentru recunoașterea amprentei codului Data Matrix, trebuie să folosim datele tehnice ale codului. Se cunoaște că amprenta este de formă pătratică și este formată dintr-o amprentă de identificare și una de sincronizare [16]. Pentru recunoașterea ei, mai întâi se detectează colțurile amprentei după care se recunosc cele două amprente. În continuare se prezintă detaliat aceste etape de recunoaștere.

4.1.7.1 Detecția colțurilor amprentei codului Data Matrix

Identificarea colțurilor amprentei codului Data Matrix este foarte importantă pentru identificarea amprentei de identificare și a celei de sincronizare. Colțurile amprentei codului Data Matrix sunt detectate pe baza matricei modulelor valabile și în funcție de distanța dintre module și de orientarea amprentei. Mai întâi se extrag coordonatele minime și maxime din matricea modulelor valabile. În acest fel se creează punctele de start pentru scanarea colțurilor amprentei, așa cum este prezentat în Fig. 4.20 (stânga).

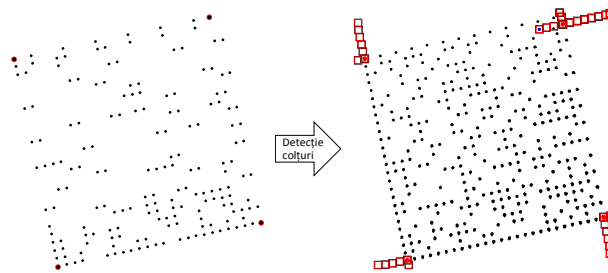


Figura 4.20: Detecția colțurilor amprentei codului Data Matrix

Cunoscând punctele extreme, distanța dintre module și unghiul orientării amprentei, se scanează în imagine în două direcții ortogonale pentru găsierea fiecărui colț extrem real, Fig. 4.20 (dreapta). Scanarea se realizează într-o fereastră de scanare din pas în pas, pasul fiind de mărimea distanței dintre module. Mai întâi, se scanează într-o direcție și în cazul în care fereastra de scanare părăsește imaginea, direcția de scanare este schimbată cu 90° către exterior. Din nou, se scanează în noua direcție până când fereastra părăsește imaginea. De fiecare dată când fereastra de scanare părăsește imaginea se schimbă direcția de scanare cu 90° către exterior. Ultimul modul detectat este un colț al amprentei codului Data Matrix. Această operație se repetă pentru toate colțurile codului Data Matrix.

4.1.7.2 Recunoașterea amprentei de identificare

Amprenta de identificare este compusă din două margini adiacente punctate în formă de „L” [16]. Pentru recunoașterea acestei amprente, folosind distanța dintre module și unghiul de orientare, sunt interogate dacă au vecini toate cele 4 colțuri. Se construiește o fereastră de scanare de dimensiune egală cu distanța dintre module. Pentru fiecare colț, folosind relațiile (4.24) - (4.25), se deplasează fereastra de scanare în două direcții ortogonale către exteriorul amprentei. Scanând în interiorul ferestrei se poate afla dacă colțul interogată are un vecin sau nu. Repetând această operație pentru toate colțurile, colțul ce are doi vecini este colțul principal al codului Data Matrix. Celelalte două colțuri cu câte un vecin sunt colțurile adiacente colțului principal

din amprenta de indentificare a codului Data Matrix. Dacă $C(x_0, y_0)$ este un colț, ce aparține codului Data Matrix, pozițiile următoare ale ferestrei de scanare sunt:

$$\begin{cases} x_1 = x_0 + d \cdot \cos(\alpha) \\ y_1 = y_0 + d \cdot \sin(\alpha) \end{cases}, \quad (4.24)$$

$$\begin{cases} x_2 = x_0 - d \cdot \sin(\alpha) \\ y_2 = y_0 + d \cdot \cos(\alpha) \end{cases}, \quad (4.25)$$

unde α este unghiul de orientare a amprentei, obținut după procedura descrisă în Capitolul 4.1.6.1, iar d este distanța dintre module.

În Fig. 4.21 (a) este prezentat procesul de scanare a colțurilor codului Data Matrix. Colțul principal are două module ortogonale învecinate, marcate cu roșu, celelalte colțuri au doar câte un modul vecin.

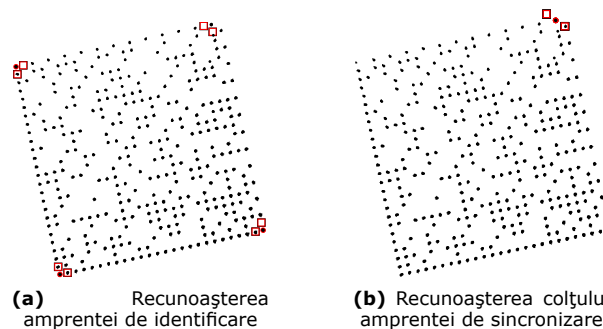


Figura 4.21: Recunoașterea amprentei codului Data Matrix

4.1.7.3 Amprenta de sincronizare

Pentru recunoașterea amprentei de sincronizare, folosind distanța dintre module și orientarea amprentei, colțul diagonal opus colțului principal al codului Data Matrix este interogată dacă are vecini, Fig. 4.21 (b). Din cauză că amprenta de sincronizare este compusă din module alternative, colțul amprentei de sincronizare are în componența sa două module. Pentru a găsi cel de-al doilea modul al colțului amprentei se scanează în interiorul unei ferestre de scanare în direcții opuse. Unghiul de scanare este unghiul creat de colțurile adiacente ale amprentei de indentificare, iar dimensiunea ferestrei este de mărimea distanței dintre module.

Dacă $C(x_0, y_0)$ este unul din colțurile amprentei de sincronizare ale codului Data Matrix, pozițiile următoare ale ferestrei de scanare sunt:

$$\begin{cases} x_1 = x_0 + 2 \cdot d \cdot \sin(\beta) \\ y_1 = y_0 + 2 \cdot d \cdot \sin(\beta + \frac{\pi}{2}) \end{cases}, \quad (4.26)$$

$$\begin{cases} x_2 = x_0 - \sqrt{2} \cdot d \cdot \sin(\beta) \\ y_2 = y_0 - \sqrt{2} \cdot d \cdot \sin(\beta + \frac{\pi}{2}) \end{cases}, \quad (4.27)$$

unde β este unghiul creat de colțurile adiacente ale amprentei de sincronizare, iar d este distanța dintre module.

4.1.8 Concluzii

Localizarea regiunii de interes (ROI) este o etapă importantă în funcționarea procesului de prelucrare a imaginii pentru recunoașterea codurilor Data Matrix. Prin utilizarea pragului dinamic-adaptiv pentru binarizarea imaginii este compensată neuniformitatea fundalului. Închiderea imaginii utilizând operatorii morfologici de dilatare și erodare ajută la localizarea poziției amprentei codului Data Matrix, astfel sistemul de localizare este mai precis. Datorită faptului că sistemului de localizare este utilizat pentru localizarea codurilor Data Matrix în mediu industrial, caracteristicile amprentei sunt aproximativ cunoscute. Astfel, se poate defini o formă a amprentei și în acest fel zona de căutare este redusă. Estimarea distanței dintre module și a suprafeței acestora, pe baza histogramei, este o metodă robustă ce poate fi implementată să ruleze în timp real. Transformarea Hough este la fel o metodă robustă pentru detecția orientării amprentei codului Data Matrix. Având aceste informații estimate și având informații despre colțurile amprentei codului Data Matrix, se pot detecta amprenta de identificare și cea de sincronizare.

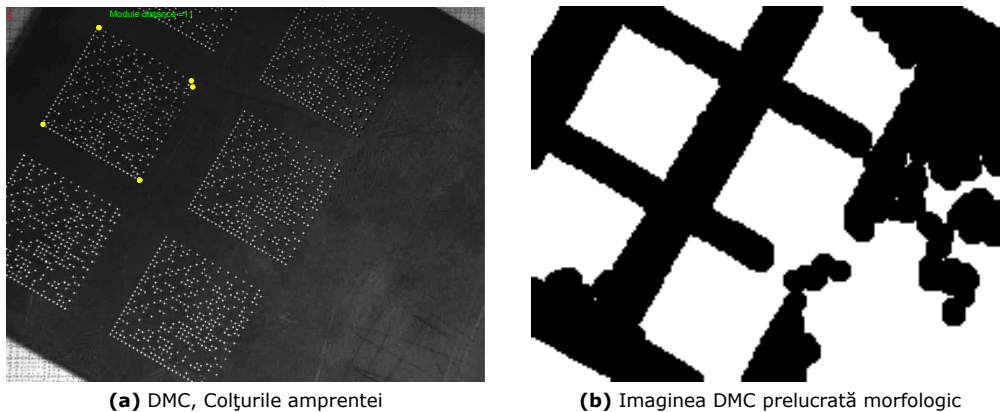


Figura 4.22: Cupru lucios, dimensiune amprentă $2 \times 2\text{cm}^2$

În figurile Fig. 4.22 - 4.25 se dau patru exemple de teste pentru diferite tipuri de coduri Data Matrix. În Fig. 4.22 codul este punctat pe cupru lucios, dimensiunea amprentei este de $2 \times 2\text{cm}^2$, iar în Fig. 4.23 codul este punctat pe aluminiu lucios, dimensiunea amprentei fiind de $1,6 \times 1,6\text{cm}^2$.

În Fig. 4.22 și 4.23 se poate vedea că dimensiunea reală a codului și numărul de module nu sunt importante pentru localizarea amprentei. Colțurile amprentei fiind localizate cu succes în ambele cazuri. În Fig. 4.24 codul este punctat pe fier acoperit cu pete de rugină, dimensiunea amprentei este de $1,6 \times 1,6\text{cm}^2$, iar în fig. 4.25 codul este punctat pe metal acoperit cu vopsea cu o anumită textură, dimensiunea amprentei fiind de $2 \times 2\text{cm}^2$.

În aceste două cazuri, se poate observa că fundalul acestor coduri nu este important în procesul de localizare în imagine. În ambele cazuri, petele de rugină și vopseaua nu afectează sistemul de localizare și identificare a celor două amprente. A se observa, în figurile prelucrate morfologic, eficiența procesului de binarizare cu prag adaptiv. Se vede ca fundalul este binarizat uniform. În urma operației de închidere morfologică codurile sunt transformate în forme pătratice.

Testele sunt executate într-un mediu cu o singură sursă de lumină montată la un unghi de 45° față de suprafața codului. Această etapă de pre-procesare a imaginii,



(a) DMC, Colțurile amprentei



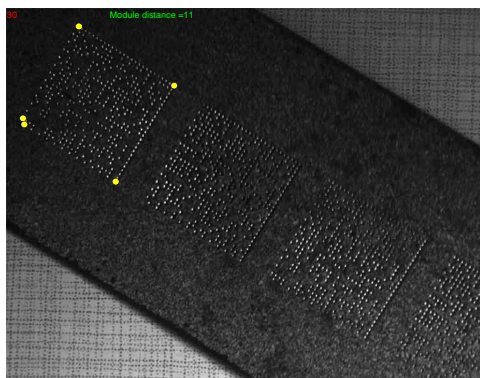
(b) Imaginea DMC prelucrată morfologic

Figura 4.23: Aluminu lucios, dimensiune amprentă $1,6 \times 1,6 \text{ cm}^2$ 

(a) DMC, Colțurile amprentei



(b) Imaginea DMC prelucrată morfologic

Figura 4.24: Fier acoperit cu pete de rugină, dimensiune amprentă $1,6 \times 1,6 \text{ cm}^2$ 

(a) DMC, Colțurile amprentei



(b) Imaginea DMC prelucrată morfologic

Figura 4.25: Metal vopsit, dimensiune amprentă $2 \times 2 \text{ cm}^2$

rulează în timp real, cu rezultate bune pentru materialele ce au proprietatea de a reflecta lumina. În cazurile în care au fost testate materiale plastice, este greu să se recunoască poziția în imagine a amprentei codului Data Matrix. În aceste cazuri, reflexia luminii este mult prea mică sau chiar nulă, mai ales pentru materialele de culoare albă.

4.2 Sisteme de localizare automată

Sistemele de localizare automată oferă utilizatorului flexibilitate și simplitate în instalare și utilizare. Acestea nu necesită configurarea lor cu date despre codurile Data Matrix ce urmează a fi scanate. Datorită acestei calități, sistemele de localizare automată pot scana și detecta mai multe coduri Data Matrix fără intervenția operatorului. Algoritmul utilizat pentru recunoaștere, folosește informații despre amprenta de identificare și de sincronizare a codului Data Matrix. În Fig. 4.26 este prezentată schema funcțională a sistemului de localizare automată.

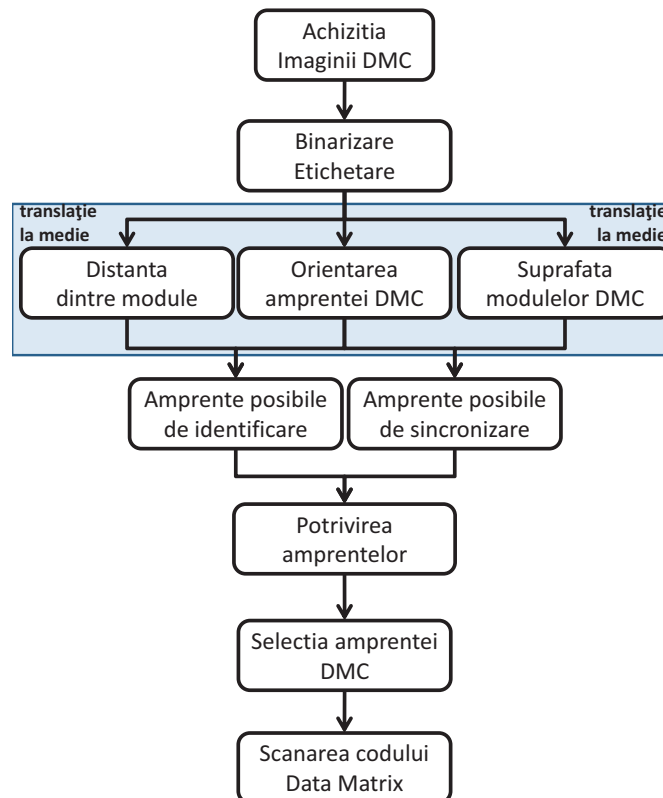


Figura 4.26: Sistem de localizare automată

Imaginea preluată de camera foto este binarizată și etichetată [72]. Aplicând Algoritmul cu Translație la Medie [42] asupra obiectelor etichetate din imagine, se calculează distanța dintre modulele amprentei, orientarea amprentei și suprafața modulelor codului Data Matrix. Având aceste informații se poate crea un șablon pentru amprenta de identificare și unul pentru amprenta de sincronizare. Pe baza

acestor șabloane se caută în structura obiectelor etichetate și se rețin obiectele, așa-zise module, ce se încadrează în șabloanele respective. Într-un cod Data Matrix, de dimensiune mare, pot exista mai multe seturi de module ce se încadrează în șabloanele prescrise. Pentru aceasta se potrivesc amprente de identificare cu cele de sincronizare și se selectează amprenta codului Data Matrix căutat. Cunoscând poziția amprente, se poate scana codul Data Matrix.

4.2.1 Achiziția și etichetarea imaginii

Imaginea furnizată de camera video este binarizată folosind metoda lui Otsu cu prag adaptiv (a se vedea în Capitolul 4.1.3.1) [26] [62]. În urma binarizării, în imagine există un set de obiecte. Un obiect este un grup de pixeli conectați între ei pe baza unei relații de conectivitate [73], fiecare obiect fiind etichetat cu o valoare diferită.

Un pixel p , cu coordonatele (x, y) , are patru vecini pe orizontală și verticală și patru vecini pe diagonală ale căror coordonate sunt date de ecuația (4.28) [61]. Apare o problemă la pixelii situați la marginile imaginii, aceștia având un set incomplet de vecini.

$$\begin{aligned} & (x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1), \\ & (x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1). \end{aligned} \quad (4.28)$$

Toți cei 4 vecinii sunt situați la o distanță de un pixel față de p . Următorul set de puncte sunt vecinii diagonali ai lui P , notați cu $V_D(p)$. Distanța Euclidiană până la vecinii diagonali este de $\sqrt{2}$ pixeli. Toate aceste 8 puncte formează împreună vecinatatea de ordin 8 a lui P , notată cu $V_8(p)$ (Fig 4.27).

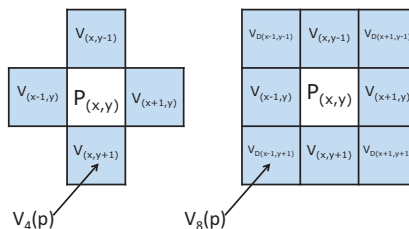


Figura 4.27: Vecinătățile unui pixel

Orice pixel din vecinatatea lui P este adiacent cu P . Doi pixeli p și q se spune că sunt conectați printr-o vecinătate de ordin 4 sau 8 în cazul în care există 4 sau 8 conexiuni între ei, formate în totalitate din pixeli prim-plan. Pentru orice pixel P , setul de pixeli conectați la acesta este numit vecinatatea lui p . Vecinatatea depinde de calea de conectivitate dintre pixeli. În Fig. 4.28 (a) este afișat un subset de pixeli al unei imagini. Prima imagine indică subsetul S și pe urma conectivitatea folosită $N_4(p)$ și $N_8(p)$.

În vecinatatea de ordin 4, pixelii care sunt încercuiți cu negru nu sunt conectați la orice alt „1”, deoarece în $N_4(p)$ adiacența diagonală nu este inclusă. Toți pixelii care sunt delimitați sunt conectați la alți pixeli în acea regiune.

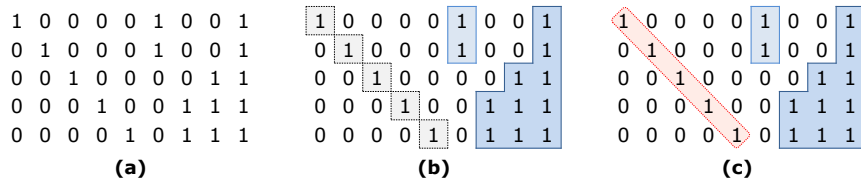


Figura 4.28: Vecinătăți de ordin 4 și 8

Atunci când o imagine este etichetată, pixelilor din fiecare vecinătate diferită li se alocă un număr întreg unic de la „1” la numărul total al componentelor conectate. Cu alte cuvinte, obiectul cu eticheta „1” este considerat ca fiind prima vecinătate de pixeli și cel cu eticheta „2” cea de-a doua și așa mai departe. În Fig. 4.29 este prezentat un exemplu de imagine cu obiectele etichetate.

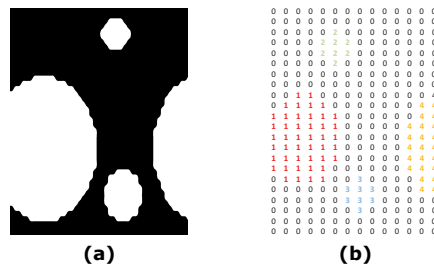


Figura 4.29: Etichetarea modulelor codului Data Matrix

4.2.2 Extragerea informațiilor amprentei codului Data Matrix folosind Algoritmul cu Translație la Medie

În acest subcapitol se prezintă o metodă de extragere a informațiilor obiectelor ce alcătuiesc imaginea binarizată unui cod Data Matrix. Se cunoaște că amprenta codului Data Matrix este alcătuită din module de suprafețe egale, situate între ele la distanță egală. Se mai știe că modulele trasează drepte în imagine două câte două, toate orientate cu același unghi. Deci, în imaginea binarizată există un grup de obiecte ce sunt așezate la distanțe egale între ele, suprafețele lor fiind de asemenea egale, iar orientările dreptelor trasate între module sunt egale. Pe baza pixelilor etichetați, folosind Algoritmul cu Translație la Medie se pot extrage informații despre o regiune de obiecte din imagine, îndeplinind condițiile enumerate mai sus.

4.2.2.1 Algoritmul cu Translație la Medie

Algoritmul cu Translație la Medie este o tehnică non-parametrică pentru analiza spațiului caracteristicilor, numită tehnică de căutare a maximumului [42] [74]. Algoritmul cu Translație la Medie (ATM) este un algoritm de tip gradient ascendent, pentru detecția modurilor funcției densitate de probabilitate, pe baza unui set de date finit. Acesta localizează maximumul unei funcții de densitate dată de un set de date discrete [43] [75]. Prin deplasarea iterativă și analizarea unei ferestre Parzen se pornește de la o estimare inițială x și se detectează maximele densității de puncte.

Se dă o funcție Nucleu $K(x_i - x)$ [76], această funcție determină ponderea unui set de puncte învecinate pentru reestimarea mediei. De obicei se folosește funcția Nucleu

Gaussiană cu privire la distanță, pentru estimarea actuală, $K(x_i - x) = e^{-||x_i - x||^2}$. Media ponderată a densității în fereastra determinată de K este:

$$y = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)}, \quad (4.29)$$

unde $N(x)$ este vecinătatea lui x , un set de puncte pentru care $K(x) \neq 0$. Algoritmul cu Translație la Medie acum inițializează $x \leftarrow m(x)$ și repetă estimarea până când $m(x)$ converge [42].

În Fig. 4.30 este dat un exemplu pentru determinarea modului funcției densitate de probabilitate unui set de date, fiecare dată este marcată cu ajutorul unui punct.

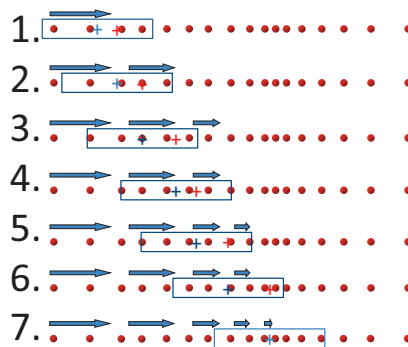


Figura 4.30: Algoritmul cu Translație la Medie - principiu

Dacă în spațiul caracteristicilor sunt prezente regiuni dense (sau grupuri), atunci acestea corespund majorității (maxim local) funcției densitate de probabilitate. Pentru fiecare set de date, Algoritmul cu Translație la Medie asociază acestea cu vârful din apropierea funcției densitate de probabilitate a seturilor de date. După fiecare iterație, putem considera că funcția se deplasează către o regiune mult mai densă a setului de date.

4.2.2.2 Implementarea Algoritmului cu Translație la Medie pentru calcularea orientării amprentei codului Data Matrix

În Fig. 4.31 este prezentată imaginea binarizată unui cod Data Matrix. Ne interesează calcularea orientării amprentei, folosind Algoritmul cu Translație la Medie.

Se pornește de la primul modul până la ultimul modul din imagine și se calculează unghiul dreptei creat de module, folosind relația:

$$\alpha_{ij} = \arctan \frac{x_j - x_i}{y_j - y_i}, \quad (4.30)$$

unde (x_i, y_i) și (x_j, y_j) sunt coordonatele modulelor ce definesc o dreaptă orientată cu unghiul α .

Se scrie o structură cu toate unghiurile calculate, unghiurile fiind ordonate crescător. Problemele ce se pun sunt: care este punctul de start al ferestrei Parzen și care este dimensiunea ferestrei.

Cea mai simplă formă non-parametrică de estimare a densității este histograma. Histograma împarte spațiul-probă într-un număr de fâșii egale și aproximează densitatea la centrul fiecărei fâșii de fracția punctelor datelor de formare care se

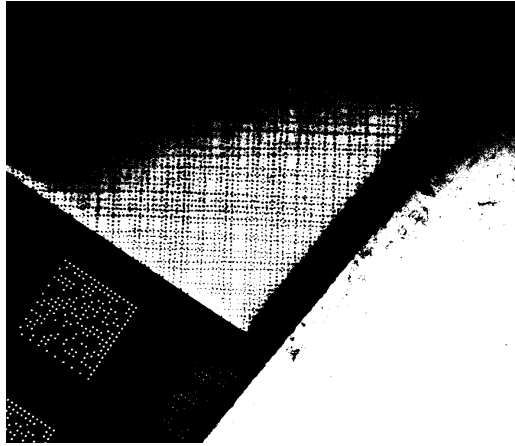


Figura 4.31: Imaginea DMC binarizată

încadrează în fâșia corespunzătoare. Histograma este o formă foarte simplă de estimare a densității, dar care are multe dezavantaje. În cazul nostru ne folosim de histogramă doar pentru a estima poziția unghiului căutat în structura unghiurilor (Fig 4.32(a)). Această poziție reprezintă punctul de start al ferestrei Parzen (centrul ferestrei), Fig 4.32(b). În fig 4.32(a) se observă că în histogramă sunt patru vârfuri. Două vârfuri apar datorită formei pătratice a amprentei codului Data Matrix, iar două apar datorită diagonalelor codului. În imagine poate să apară și zgomot sau alte obiecte neapartinătoare codului. Acestea pot crea un vârf în histograma imaginii binarizate. Pentru a preveni această situație se ia pe rând fiecare vârf din cele patru și se scanează pe histogramă după complementarul său. Este clar că poziția corectă a unghiului estimat este acea poziție care are și o poziție complementară în histograma unghiurilor. Având poziția, se centrează fereastra Parzen pe această poziție de start. Dimensiunea ferestrei nu se alege mai mare decât jumătate din domeniul pozițiilor unghiurilor. Se încearcă să nu se cuprindă mai mult de două vârfuri din histogramă, în această fereastră. În exemplul din Fig 4.32(b), dimensiunea este aleasă cât jumătatea spațiului ocupat de valorile unghiurilor.

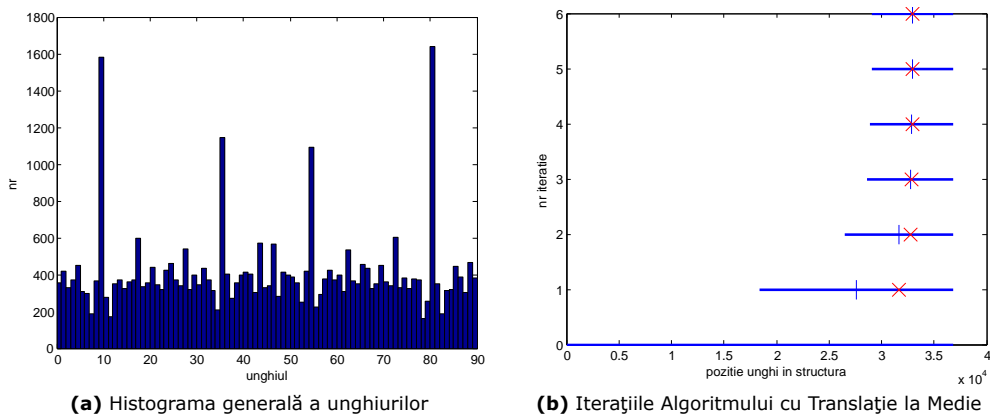


Figura 4.32: Algoritmul cu Translație la Medie

Expresia generală pentru estimarea non-parametrică a densității este:

$$P(x) \cong \frac{n_p}{n \cdot V}, \quad (4.31)$$

unde V este volumul ce înconjoară poziția x , n este numărul total de exemple, iar n_p este numărul de exemple din interiorul lui V [77].

Dacă presupunem că regiunea care înconjoară exemplele k este un Hiper cub cu laturi de lungime h , centrat în punctul x estimat, atunci volumul său este dat de $V = h^D$, unde D este numărul de dimensiuni, Figura 4.33.

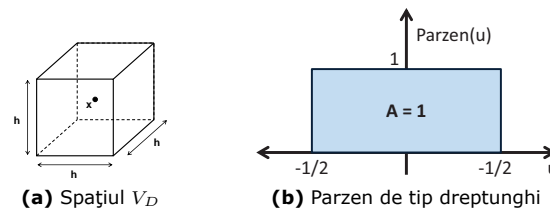


Figura 4.33: Funcția Nucleu

În cazul nostru, dimensiunea este $D = 1$, înseamnă că $V = \text{lungimea unui segment}$, iar $k = \text{Numărul valorilor cuprinse în segment}$ [74]. În continuare, pe baza fig. 4.32(b), se enumeră pașii Algoritmului cu Translație la Medie [78].

În primul pas se centrează segmentul Parzen în punctul estimat de histograma unghiurilor. Lungimea segmentului este egală cu jumătatea domeniului. Folosind relația (4.31), se calculează centrul densității punctelor de pe segment (Fig. 4.34(b)). Se deplasează centrul segmentului la poziția mediei calculate (Fig.4.32(b)).

În pasul 2 segmentul ar depăși domeniul unghiurilor. De aceea, dimensiunea lui este recalculată prin micșorarea lui până când se încadrează în domeniu. Se recalculează media cu noile valori din segment și lungimea lui actuală.

Se repetă pașii din Fig 4.35 - 4.38 până când centrul segmentului converge la medie. Algoritmul cu Translație la Medie nu necesită o parametrizare pre-procesivă, algoritmul adaptându-se după datele din spațiul valorilor supus estimării.

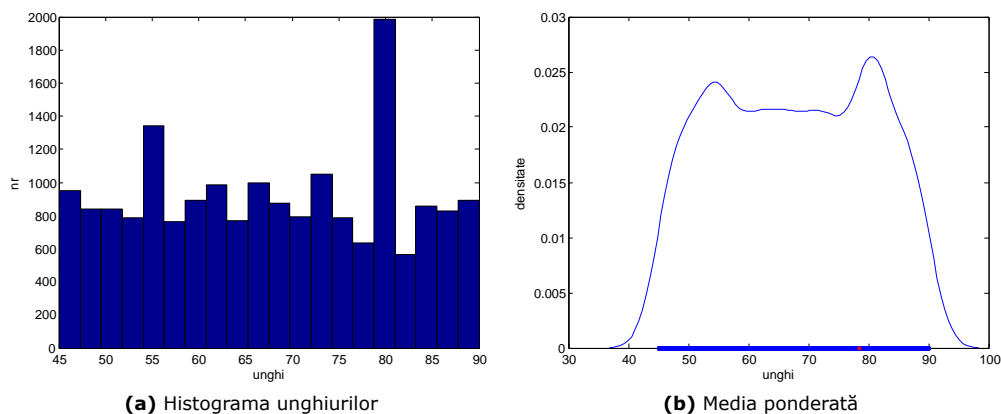


Figura 4.34: ATM - Iterația 1

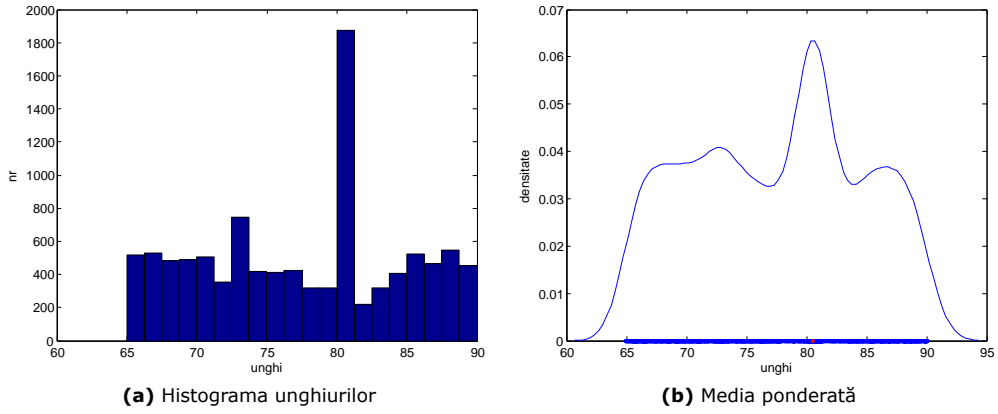


Figura 4.35: ATM - Iterația 2

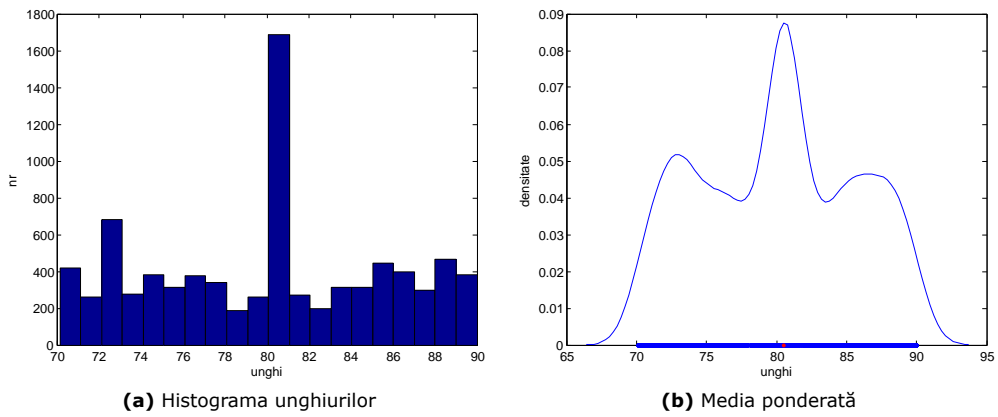


Figura 4.36: ATM - Iterația 3

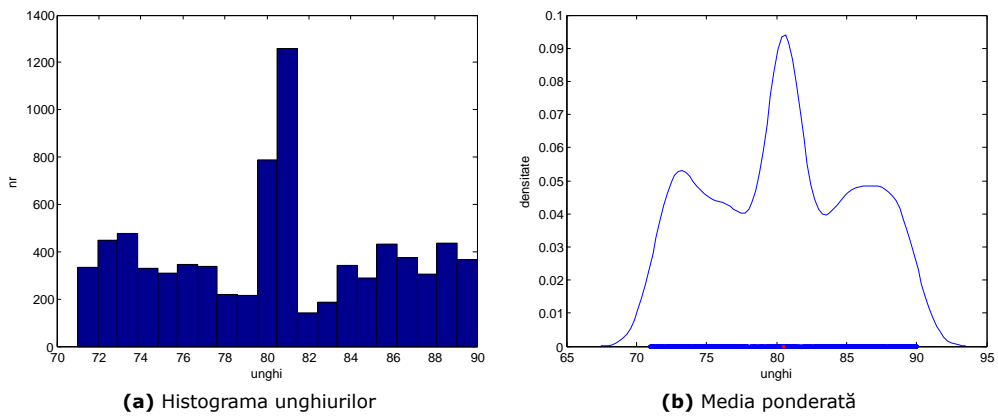


Figura 4.37: ATM - Iterația 4

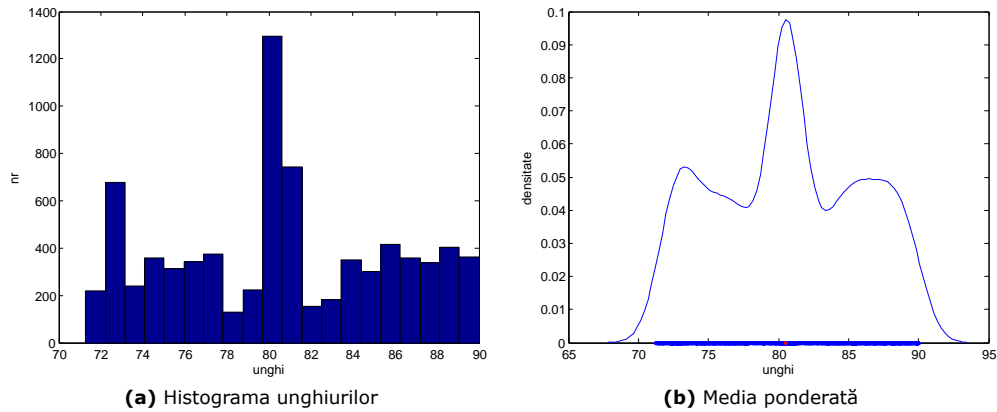


Figura 4.38: ATM - Iterația 5

4.2.2.3 Implementarea Algoritmului cu Translație la Medie pentru calcularea distanței dintre modulele amprentei și suprafața acestora

Pentru calcularea distanței dintre module și a suprafeței modulelor, se folosește metoda prezentată în Subcapitolul 4.2.2.1.

Pentru crearea structurii distanțelor se calculează distanța Euclidiană dintre fiecare obiect din imagine și 8 vecini apropiați lui, principiul fiind același ca cel prezentat în cadrul *Sistemelor de localizare semi-automată*, Capitolul 4.1 (Fig. 4.39).

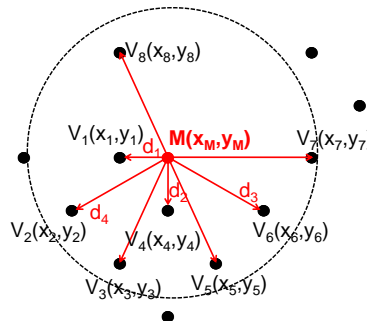


Figura 4.39: Calcularea distanței dintre module

În Fig. 4.40 este prezentată histograma distanțelor din imagine și derularea Algoritmului cu Translație la Medie, implementat pentru calculul distanței dintre modulele codului Data Matrix.

Pentru calcularea suprafeței comune a modulelor codului Data Matrix, pe baza imaginii etichetate, se număra pixelii din fiecare obiect și se crează o structură a obiectelor. Folosind Algoritmul cu Translație la Medie, se calculează poziția densității maxime a valorilor suprafețelor în structură. Valoarea acelei poziții fiind suprafața

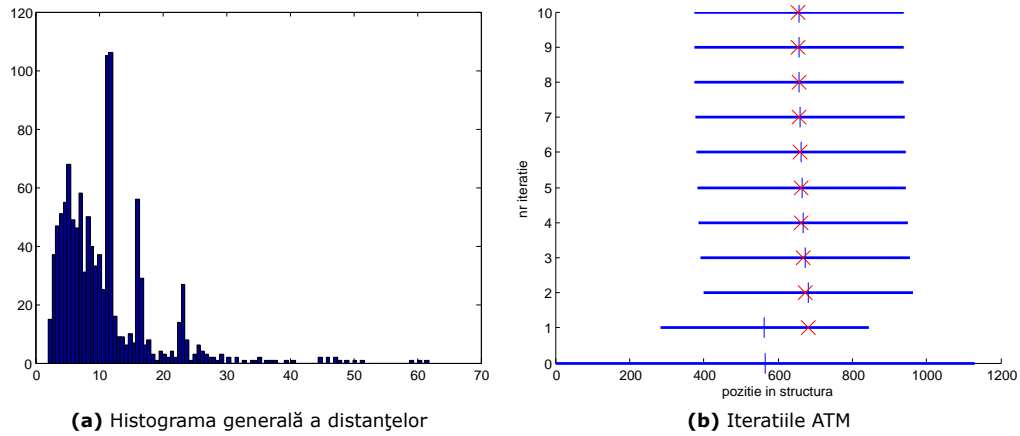


Figura 4.40: Algoritmul cu Translație la Medie implementat pentru calculul distanței dintre module

modulului codului Data Matrix. În Fig. 4.41 este prezentată histograma suprafețelor din imagine și derularea Algoritmului cu Translație la Medie, implementat pentru calcularea suprafeței modulelor codului Data Matrix.

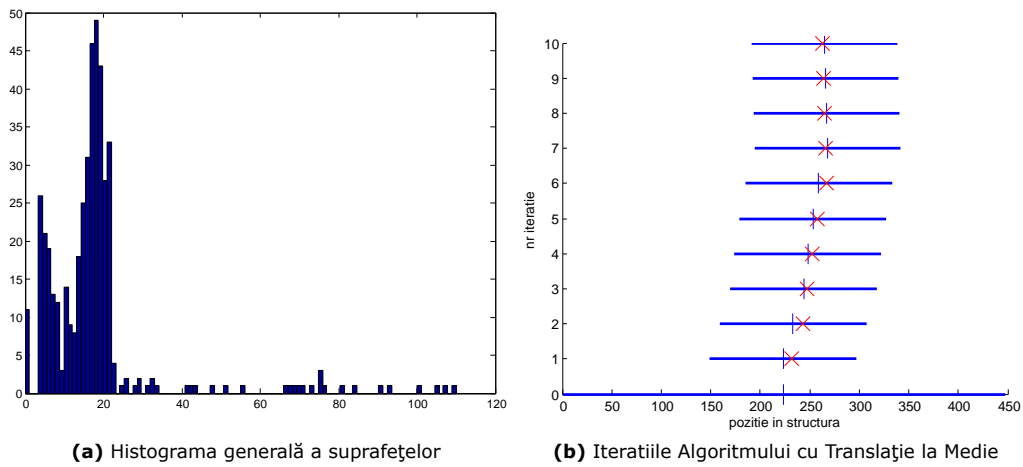


Figura 4.41: Algoritmul cu Translație la Medie implementat pentru calculul suprafeței modulelor

Pentru a înțelege mai bine metoda descrisă anterior, în Fig. 4.42 este prezentată poziția modulelor Data Matrix în spațiul 3-D. Cele 3 dimensiuni ale spațiului sunt date de suprafața, distanța și unghiul modulelor. În figură, punctele reprezentate cu roșu sunt modulele codului Data Matrix, cu albastru sunt redat toate celelate obiecte din imagine. A se observa că în urma Algoritmului cu Translație la Medie, modulele determinate formează un nucleu în figura de față. Prin această figură se mai poate remarca acuratețea metodei descrise anterior.

Avantajele și diferențele folosirii Algoritmului cu Translație la Medie față de histogramă sunt analizate și prezentate detaliat în Capitolul 5.

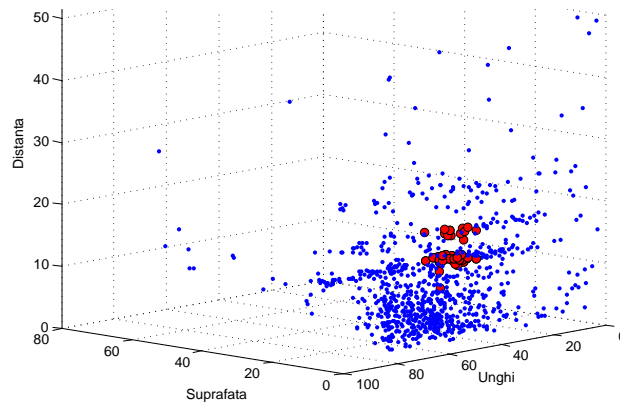


Figura 4.42: Viziune în spațiul 3-D a modulelor codului Data Matrix

4.2.3 Recunoașterea Amprentei „L” a Codului Data Matrix

Pentru recunoașterea amprentei codului Data Matrix, se pornește de la specificațiile ei, acelea că este formată din două margini adiacente marcate în totalitate cu module și două margini adiacente marcate numai în pozițiile pare. Se mai cunoaște că amprenta este de formă pătratică și, bineînțeles, se știe orientarea ei, distanța dintre module și suprafața modulelor.

Având aceste date de intrare, se construiește un șablon pentru amprenta de identificare și unul pentru cea de sincronizare. Știind că numărul de rânduri și coloane din amprenta codului Data Matrix, variază între 8x8 și 144x144, considerăm dimensiunea amprentei de identificare de 8x8. Folosind matricea de coordonate, relațiile (4.32 - 4.33), se construiește un șablon ce se suprapune peste structura de coordonate a obiectelor imaginii Data Matrix. Se caută obiectele ce se potrivesc în matricea șablon, definită prin ecuația (4.32).

$$L_{Ident} = \begin{pmatrix} L_{8,1} & - & - & - & - & - & - & - \\ L_{7,1} & - & - & - & - & - & - & - \\ L_{6,1} & - & - & - & - & - & - & - \\ L_{5,1} & - & - & - & - & - & - & - \\ L_{4,1} & - & - & - & - & - & - & - \\ L_{3,1} & - & - & - & - & - & - & - \\ L_{2,1} & - & - & - & - & - & - & - \\ L_{1,1} & L_{1,2} & L_{1,3} & L_{1,4} & L_{1,5} & L_{1,6} & L_{1,7} & L_{1,8} \end{pmatrix} \quad (4.32)$$

unde:

$$\begin{aligned}
L_{1,1} &= \begin{cases} x_0 \\ y_0 \end{cases} \\
L_{1,2} &= \begin{cases} x_0 + d \cos \alpha \\ y_0 + d \sin \alpha \end{cases} & L_{2,1} &= \begin{cases} x_0 - d \sin \alpha \\ y_0 + d \cos \alpha \end{cases} \\
L_{1,3} &= \begin{cases} x_0 + 2d \cdot \cos(\alpha) \\ y_0 + 2d \cdot \sin(\alpha) \end{cases} & L_{3,1} &= \begin{cases} x_0 - 2d \sin \alpha \\ y_0 + 2d \cos \alpha \end{cases} \\
L_{1,4} &= \begin{cases} x_0 + 3d \cdot \cos(\alpha) \\ y_0 + 3d \cdot \sin(\alpha) \end{cases} & L_{4,1} &= \begin{cases} x_0 - 3d \sin \alpha \\ y_0 + 3d \cos \alpha \end{cases} \\
L_{1,5} &= \begin{cases} x_0 + 4d \cdot \cos(\alpha) \\ y_0 + 4d \cdot \sin(\alpha) \end{cases} & L_{5,1} &= \begin{cases} x_0 - 4d \sin \alpha \\ y_0 + 4d \cos \alpha \end{cases} \\
L_{1,6} &= \begin{cases} x_0 + 5d \cdot \cos(\alpha) \\ y_0 + 5d \cdot \sin(\alpha) \end{cases} & L_{6,1} &= \begin{cases} x_0 - 5d \sin \alpha \\ y_0 + 5d \cos \alpha \end{cases} \\
L_{1,7} &= \begin{cases} x_0 + 6d \cdot \cos(\alpha) \\ y_0 + 6d \cdot \sin(\alpha) \end{cases} & L_{7,1} &= \begin{cases} x_0 - 6d \sin \alpha \\ y_0 + 6d \cos \alpha \end{cases} \\
L_{1,8} &= \begin{cases} x_0 + 7d \cdot \cos(\alpha) \\ y_0 + 7d \cdot \sin(\alpha) \end{cases} & L_{8,1} &= \begin{cases} x_0 - 7d \sin \alpha \\ y_0 + 7d \cos \alpha \end{cases}
\end{aligned} \tag{4.33}$$

Același lucru se realizează și pentru identificarea amprente de sincronizare. Se știe că amprenta are în componența sa module marcate alternativ. Folosind această informație, se construiește o matrice șablon de 4x4, relațiile (4.34 - 4.35) și se caută în structura coordonatelor obiectelor din imagine.

$$L_{Sinc} = \begin{vmatrix} L_{8,1} & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ L_{6,1} & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ L_{4,1} & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ L_{2,1} & - & - & - & - & - & - & - \\ - & L_{1,2} & - & L_{1,4} & - & L_{1,6} & - & L_{1,8} \end{vmatrix} \tag{4.34}$$

unde:

$$\begin{aligned}
L_{1,2} &= \begin{cases} x_0 \\ y_0 \end{cases} & L_{2,1} &= \begin{cases} x_0 - d(\cos(\alpha) + 2 \sin(\alpha)) \\ y_0 - d(\sin(\alpha) - 2 \cos(\alpha)) \end{cases} \\
L_{1,4} &= \begin{cases} x_0 + 2d \cos \alpha \\ y_0 + 2d \sin \alpha \end{cases} & L_{4,1} &= \begin{cases} x_0 - 2d(\cos(\alpha) + 2 \sin(\alpha)) \\ y_0 - 2d(\sin(\alpha) - 2 \cos(\alpha)) \end{cases} \\
L_{1,6} &= \begin{cases} x_0 + 4d \cos \alpha \\ y_0 + 4d \sin \alpha \end{cases} & L_{6,1} &= \begin{cases} x_0 - 4d(\cos \alpha + 2 \sin \alpha) \\ y_0 - 4d(\sin \alpha - 2 \cos \alpha) \end{cases} \\
L_{1,8} &= \begin{cases} x_0 + 6d \cos \alpha \\ y_0 + 6d \sin \alpha \end{cases} & L_{8,1} &= \begin{cases} x_0 - 6d(\cos \alpha + 2 \sin \alpha) \\ y_0 - 6d(\sin \alpha - 2 \cos \alpha) \end{cases}
\end{aligned} \tag{4.35}$$

Datorită formei pătratice a amprentei codului Data Matrix, pentru siguranța identificării amprentelor, se rotesc cele două matrici șablon cu $[90^\circ, 180^\circ, 270^\circ]$, (Fig. 4.43 - 4.44).

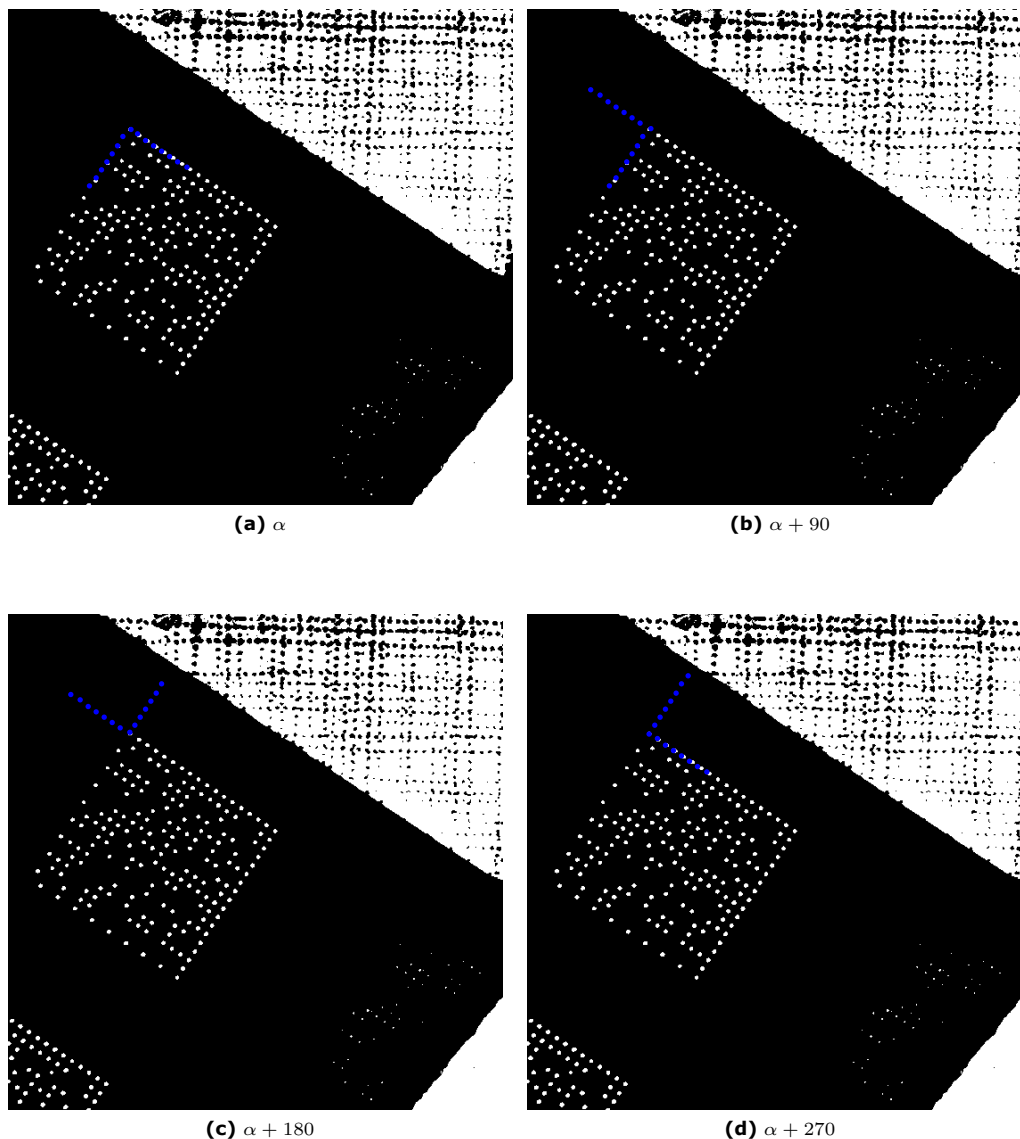


Figura 4.43: Șablonul amprentei de identificare

Coordonatele obiectelor ce îndeplinesc condițiile de încadrare simultană în șabloanele construite, se scriu într-o structură a amprentelor de identificare și de sincronizare posibile.

Având mai multe amprente posibile, de identificare și de sincronizare, se selectează amprentele ce au marginile două câte două situate la distanțe egale și colțurile lor orientate înspre exterior, adică orientări defazate cu 180° . În Fig. 4.45 (a, b) se

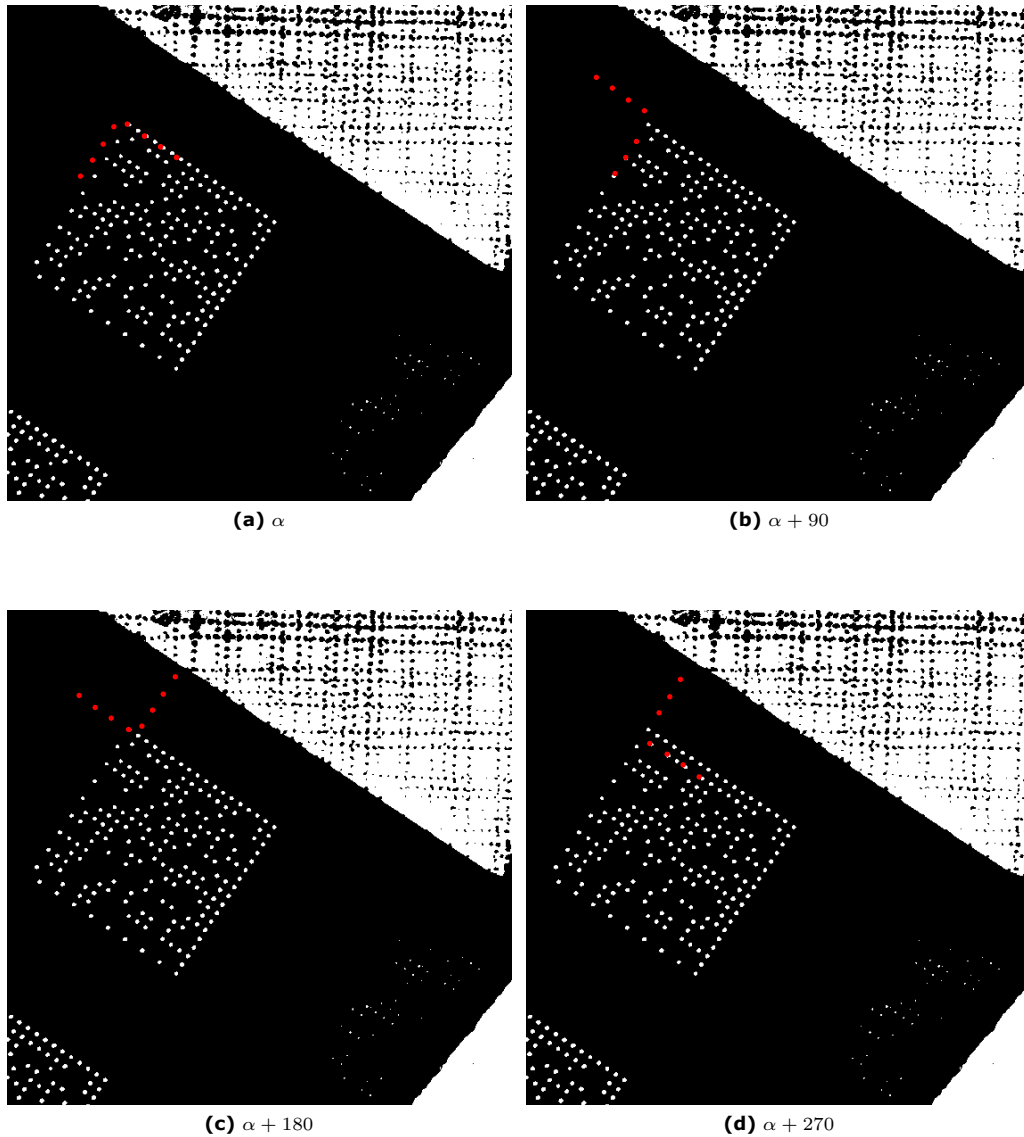


Figura 4.44: Șablonul amprentei de sincronizare

prezintă posibilitățile amprentelor de identificare și sincronizare, iar în Fig. 4.45 (c), sunt prezentate cele două amprente.

Ampretele detectate sunt de mărime 8x8 module, dar mărimea codului Data Matrix fiind încă necunoscută nu se cunosc celelalte colțuri ale amprentelor. Trasând linii imaginare peste cele 4 margini detectate, intersecția lor ne duce la descoperirea celorlalte două colțuri ale amprentei. Astfel se cunosc cele 4 colțuri și poziția lor în amprenta codului Data Matrix.

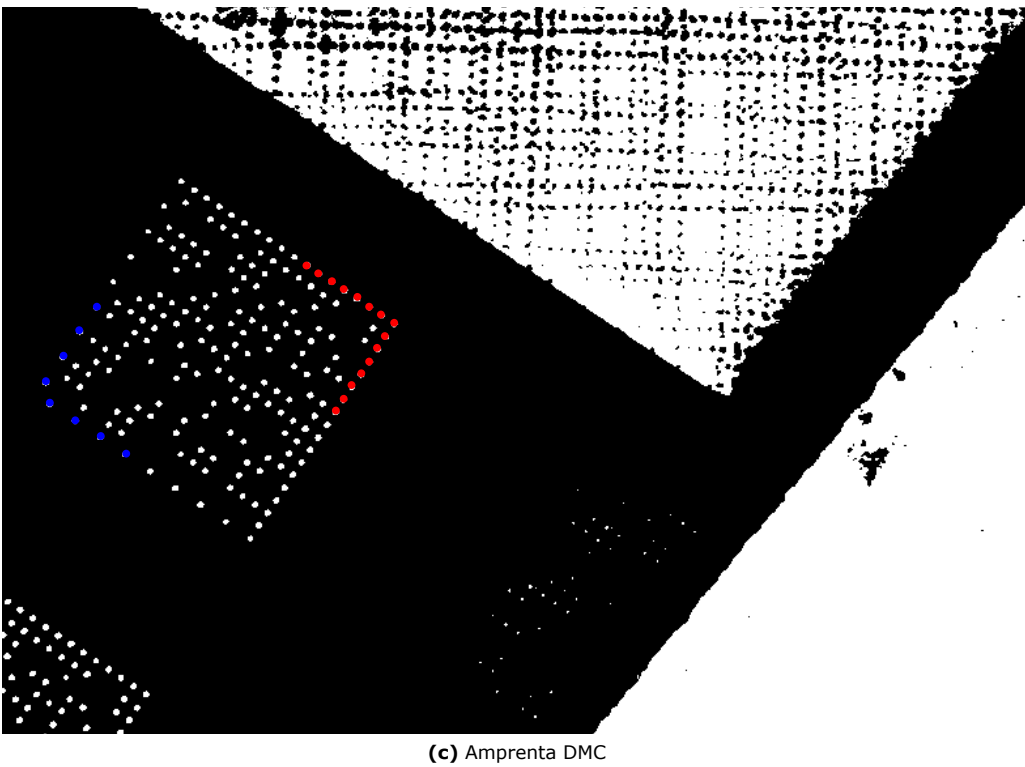
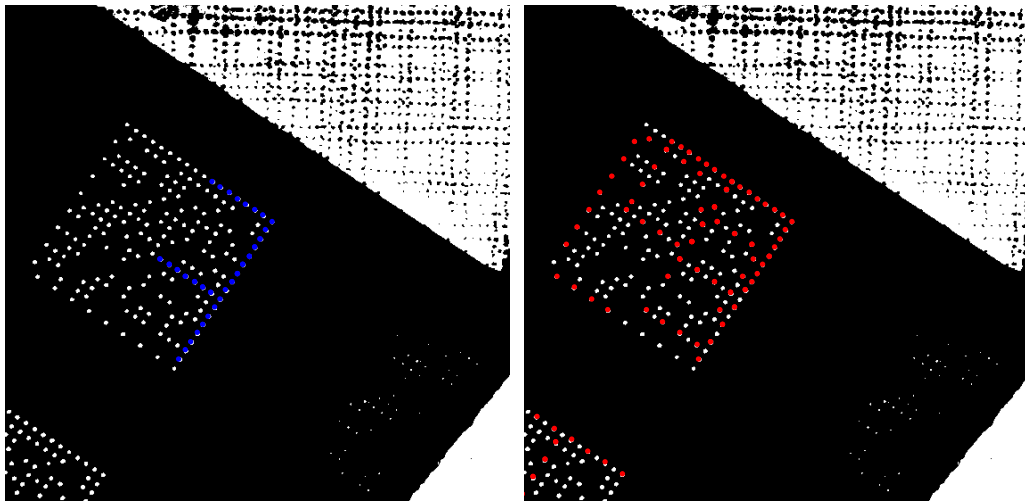


Figura 4.45: Selecția amprentei DMC

4.2.4 Concluzii

Acest subcapitol oferă o soluție automată de scanare a codurilor Data Matrix industriale marcate pe diferite tipuri de materiale, fără a se cunoaște informații despre amprenta acestora.

Construind o structură cu coordonatele obiectelor din imagine se folosește Algoritmul de Translație la Medie pentru estimarea orientării amprentei, distanței dintre module și suprafeței acestora. Algoritmul cu Translație la Medie nu necesită a fi parametrizat, acesta adaptându-se după datele din spațiul valorilor supus estimării.

Se mai cunoaște din datele tehnice ale amprentei codului Data Matrix că este compusă dintr-o amprentă de identificare și una de sincronizare, iar dimensiunea amprentelor variază de la 8x8 la 144x144 [16]. Folosind datele calculate despre module, se construiesc două șabloane pentru cele două amprente. Șablonul amprentei de identificare este format dintr-o matrice de dimensiunea 8x8, iar cel al amprentei de sincronizare este de dimensiunea 4x4. Se caută în structura imaginii centrele obiectelor ce se suprapun simultan peste cele două șabloane impuse. Ampretele de sincronizare și de identificare ce sunt diagonal opuse formează un pătrat ce reprezintă amprenta codului Data Matrix.

În Fig. 4.46 - 4.49, se prezintă câteva teste ale metodei propuse și rezultatele acestora. Se indică cu albastru modulele amprentei de identificare, iar cu roșu modulele amprentei de sincronizare. În Fig. 4.46 (a, b) - 4.49 (a, b), se poate vedea posibilitatea ca mai multe formațiuni de module să se potrivească în șabloanele construite, dar sunt selectate doar cele ce îndeplinesc condiția să formeze o amprentă Data Matrix (Fig. 4.46 (c) - 4.49 (c)). Testele sunt executate într-un mediu cu o singură sursă de lumină montată la un unghi de 45° cu suprafața codului Data Matrix.

Se observă ca s-au testat diferite tipuri de materiale, iar amprentele codurilor sunt de diferite dimensiuni și cu orientări diferite. Unele materiale au pe suprafața acestora pete de rugină sau vopsea. În toate figurile prezentate, amprenta codului Data Matrix este recunoscută cu precizie, dimensiunea, orientarea acesteia sau chiar materialul pe care este marcată neinfluențând sistemul de localizare.

Având poziția amprentei codului Data Matrix, se poate scana în interiorul acesteia pentru extragerea informației.

Metoda prezentată funcționează foarte bine pentru toate materialele ce au proprietatea de a reflecta lumina. Datorită capacității metodei de localizare a amprentei, este posibilă localizarea codurilor Data Matrix a căror imagine nu este achiziționată cu o cameră foto orientată perfect perpendicular pe suprafața amprentei codului. Această metodă oferă flexibilitate mare de utilizare, având capacitatea recunoașterii codurilor Data Matrix marcate pe suprafețe curbe.

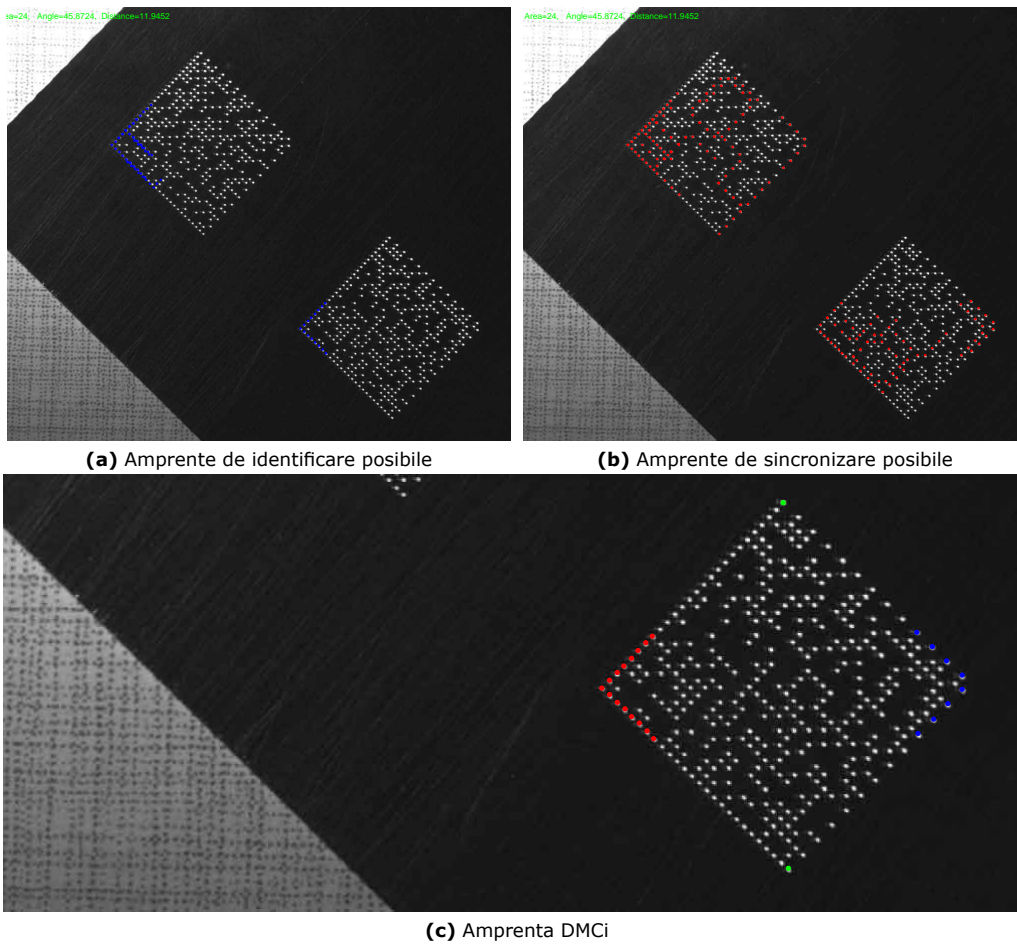


Figura 4.46: Aluminu, dimensiune amprentă $2 \times 2\text{cm}^2$

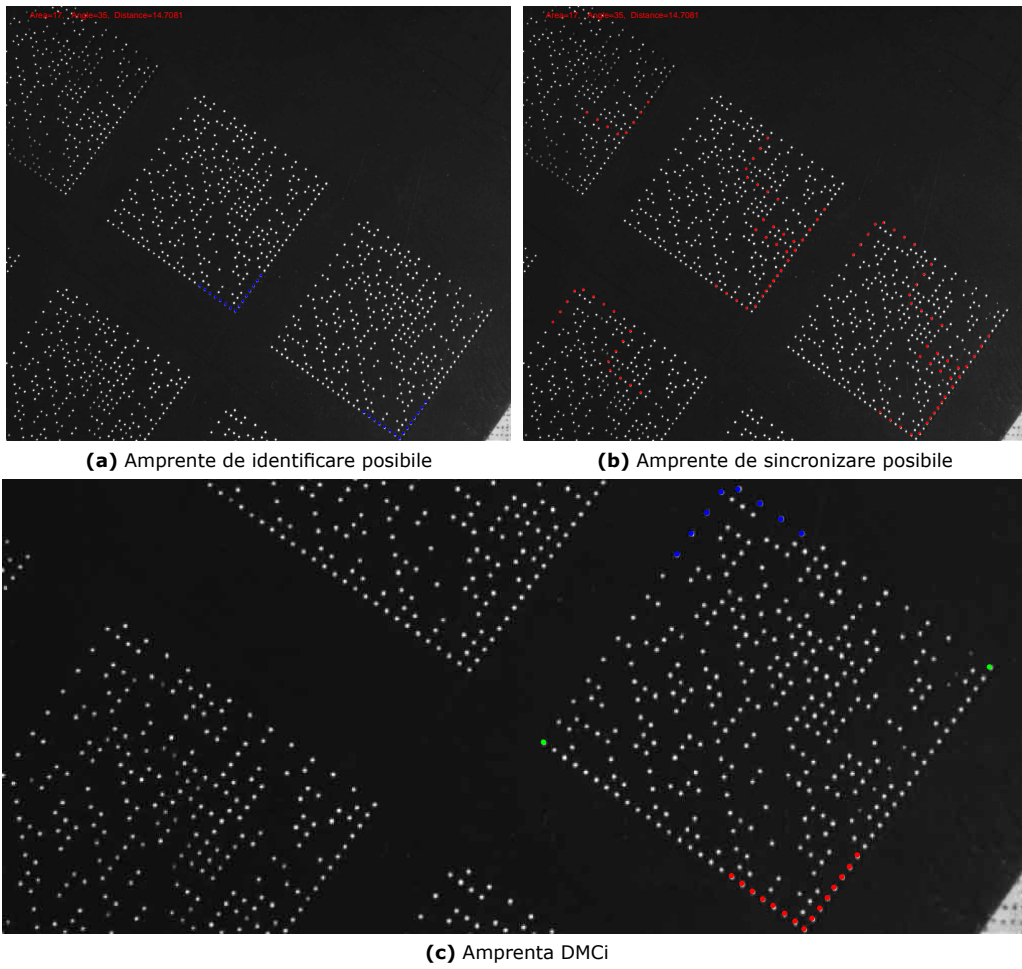


Figura 4.47: Cupru, dimensiune amprentă $2 \times 2\text{cm}^2$

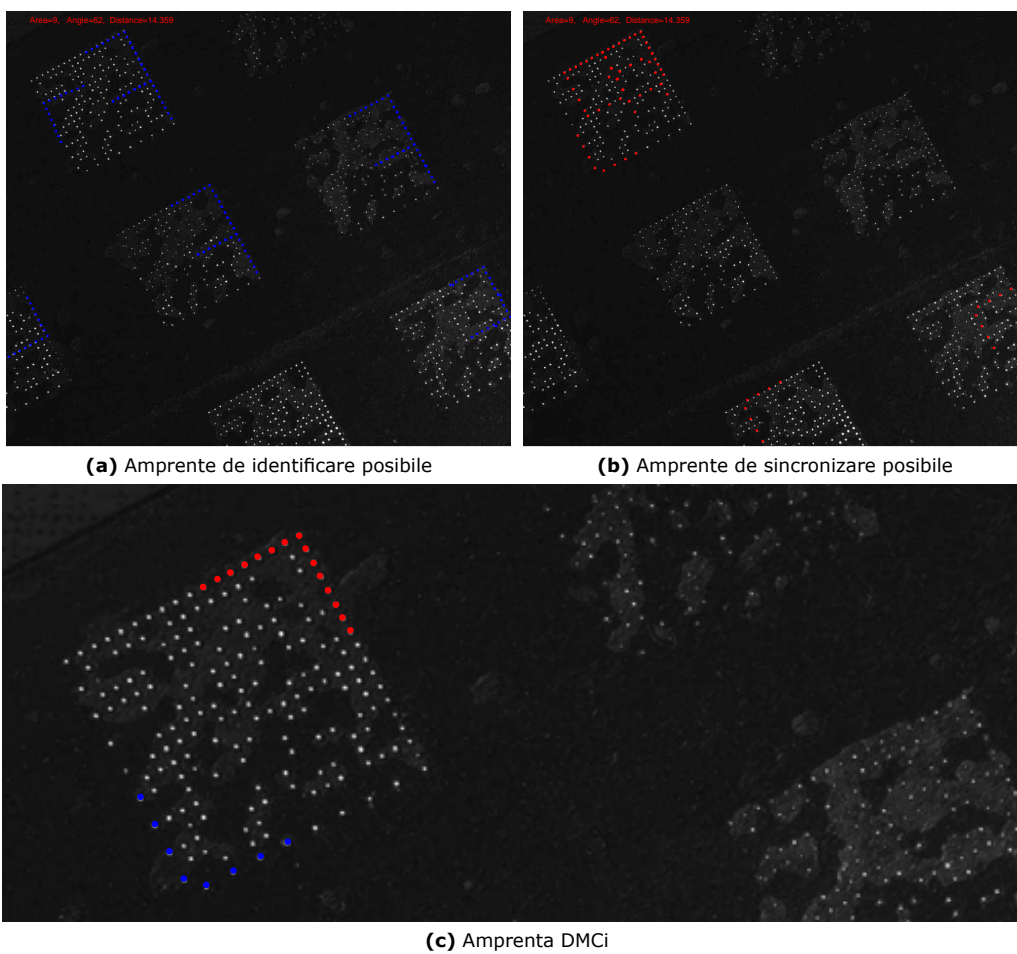


Figura 4.48: Fier, dimensiune amprentă $1,2 \times 1,2 \text{ cm}^2$

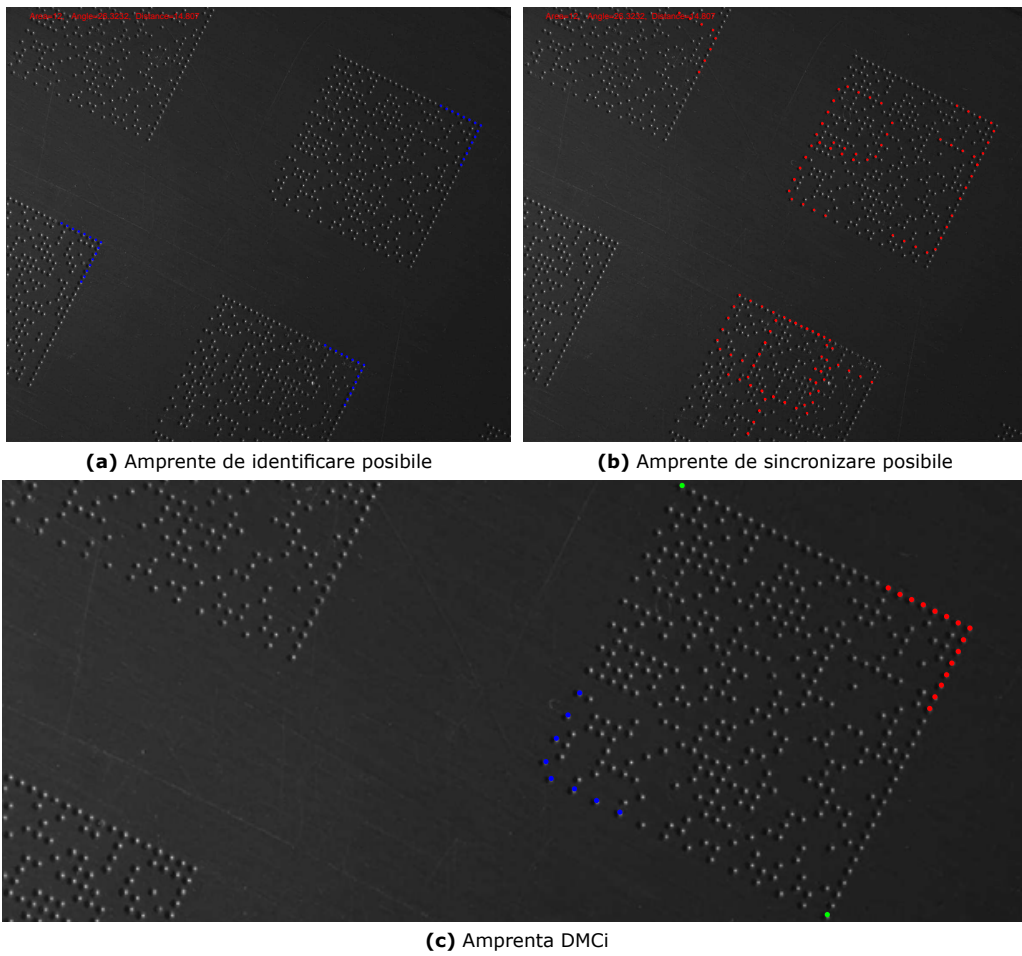


Figura 4.49: Plastic, dimensiune amprentă $2 \times 2\text{cm}^2$

Capitolul 5

Comparația metodelor de localizare

În acest capitol se compară două metode de estimare a unghiului orientării amprentei codurilor Data Matrix industriale. Prima metodă este bazată pe transformarea Hough iar cea de-a doua este bazată pe Algoritmul cu Translație la Medie [79] [42]. Scopul acestui capitol este de a sublinia punctele forte și cele slabe ale celor două metode prezentate în Capitolul 4.

5.1 Experiment

Pentru a putea compara cele două metode trebuie analizate mai multe amprente Data Matrix orientate în unghiuri diferite, pentru a avea suficiente date statistice. Acest lucru ar fi mai greu de realizat pe baza obiectelor reale marcate cu cod Data Matrix. În acest caz ar putea interveni și eroarea umană. De aceea, se dezvoltă o metodă de testare pe baza unor amprente Data Matrix generate sintetic.

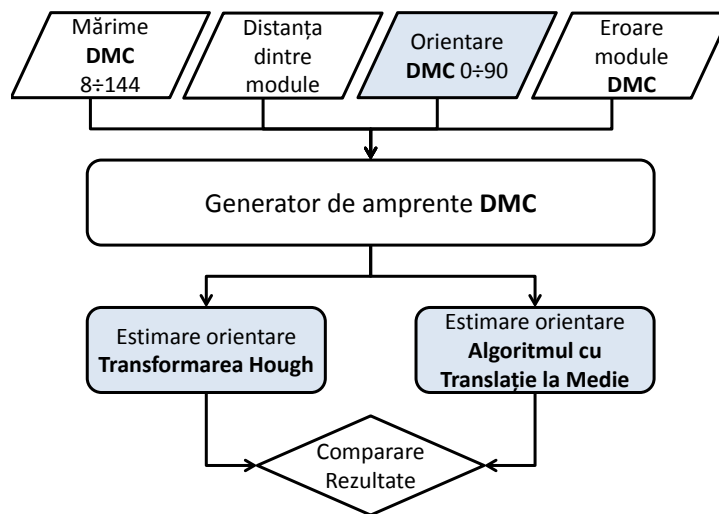


Figura 5.1: Descrierea experimentului

Cunoscând structura amprentei codului Data Matrix, în funcție de mărimea amprentei, distanța dintre module și unghiul acesteia se poate genera o imagine sintetică a acestei amprente (Fig. 5.1). Dacă acest proces este automatizat prin generarea succesivă a mai multor amprente la unghiuri diferite, atunci se pot testa simultan cele două metode propuse în Capitolul 4 și realizarea unei statistici a rezultatelor acestora. Când o imagine este achiziționată mai intervine o eroare. Această eroare poate fi provocată de sistemul optic al camerei foto, de convertorul analog-numeric al acesteia sau de factori externi: lumină, poziție cameră foto, ș.a.m.d. Imaginea generată cu generatorul de amprente Data Matrix propus în acest capitol ar fi o imagine perfectă, fără intervenția factorilor enumerați mai sus. De

aceea, pentru ca amprenta Data Matrix generată sintetic să fie cât mai apropiată de amprenta reală achiziționată cu o cameră foto, se introduce o eroare de poziție, „ σ ”. Această eroare, „ σ ”, deplasează centrele modulelor Data Matrix, în mod aleator, între două limite impuse manual.

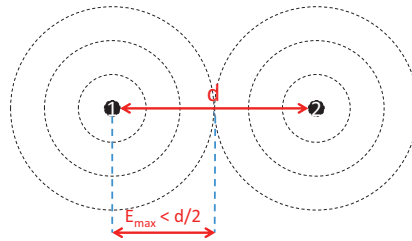


Figura 5.2: Abaterea maximă unui modul

Dacă considerăm două module vecine „1” și „2”, situate la o distanță „ d ” (Fig. 5.2), eroarea maximă, E_{max} , nu trebuie să fie mai mare decât jumătatea distanței dintre module.

Într-o dimensiune putem asocia această eroare cu distribuția normală Gauss, a cărei funcție de densitate de probabilitate se scrie [80]:

$$f(X_m) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(X_m - \bar{X})^2}{2\sigma^2}} \quad (5.1)$$

pentru: $-\infty < X_m - \bar{X} < \infty$, unde: \bar{X} este media aritmetică a datelor de intrare și σ este abaterea standard. În Fig. 5.3 se prezintă graficul funcției distribuției de probabilitate în două dimensiuni, ce poate exprima distribuția erorii în spațiul imagine.

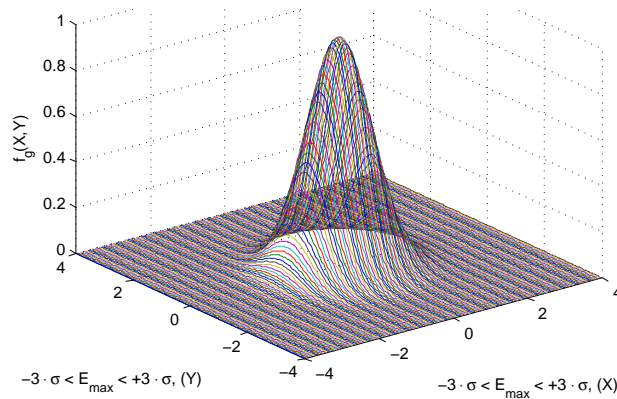


Figura 5.3: Curba de repartiție Gauss a densității de probabilitate a erorii în spațiul imagine

Având cunoscute aceste lucruri, putem genera amprenta codului Data Matrix. Se creează o rețea de puncte folosind relațiile: pentru rânduri:

$$\begin{cases} x_1 = (x_0 + d \cdot \cos \alpha) + E_\sigma \\ y_1 = (y_0 + d \cdot \sin \alpha) + E_\sigma \end{cases} \quad (5.2)$$

pentru coloane:

$$\begin{cases} x_2 = (x_0 - d \cdot \sin \alpha) + E_\sigma \\ y_2 = (y_0 + d \cdot \cos \alpha) + E_\sigma \end{cases}, \quad (5.3)$$

unde α este unghiul orientării amprentei, d este distanța dintre module și E_σ este eroarea poziției modulelor codului Data Matrix, o valoare aleatoare furnizată conform densității de probabilitate, din ecuația 5.1.

Parametrii densității de probabilitate sunt aleși pentru $\bar{x} = 0$, pentru că prin $(x_0 + d \cdot \cos \alpha)$ se alege deja poziția centrală și distribuția trebuie doar să furnizeze abaterea de la poziția corectă. Iar din considerentul $E_{max} < d/2$, în limitele $[\bar{x} - 3\sigma, \bar{x} + 3\sigma]$ sunt cuprinse 99,7% a tuturor valorilor, alegem $3\sigma = E_{max}$ și deci $E_{max} < d/6$.

În Fig. 5.4 este afișată imaginea codului Data Matrix originală și amprenta construită sintetic pentru diferite niveluri de eroare. A se observa că imaginea codului Data Matrix este tot mai distorsionată cu cât abaterea pătratică medie este mai mare.

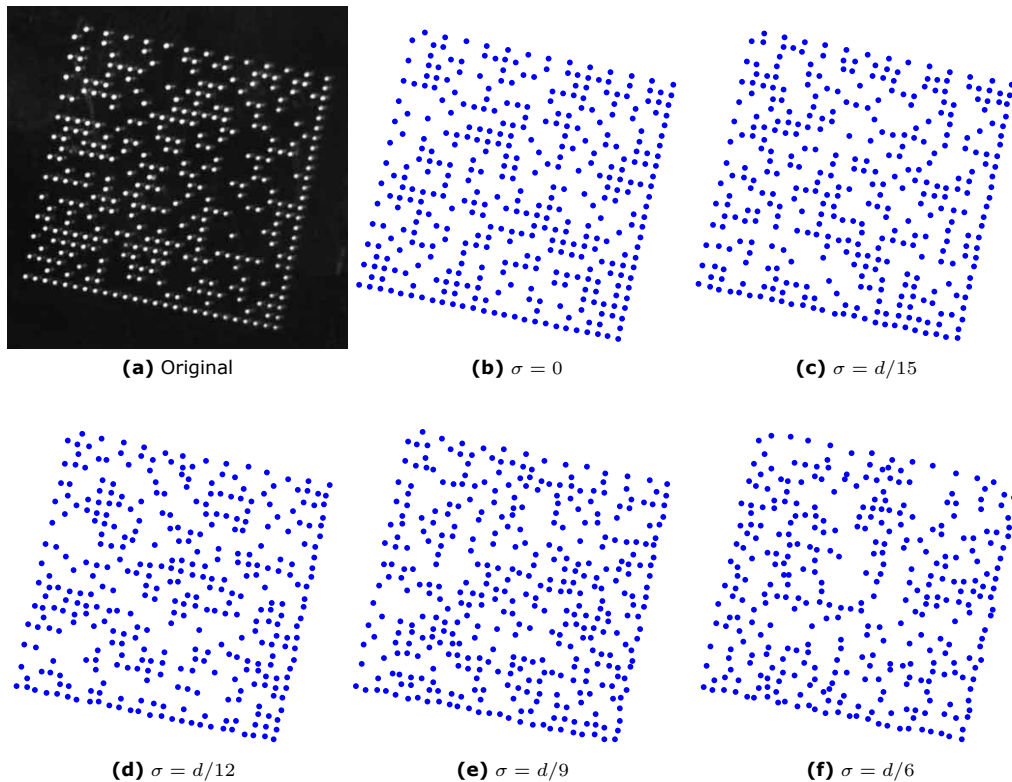


Figura 5.4: Amprenta DMC generată sintetic, mărimea = 26×26 , $d = 11px$, $\alpha = 78^\circ$

Pentru testarea celor două metode, acestea trebuie parametrizate în condiții egale, la scară optimă, astfel rezultatele pot fi comparate între ele. În continuare, prezentăm succint cele două metode și factorii care influențează în mod direct algoritmi propuși testării.

Transformarea Hough

Din Capitolul 4.1.6.1 știm că algoritmul transformării Hough folosește o matrice, numită acumulator, pentru a detecta existența unei linii $r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$. Dimensiunile matricii acumulator corespund cu valorile cuantificate pentru r și θ și coloanele de votare. Pentru fiecare pixel și vecinătatea lui, algoritmul transformării Hough determină dacă există suficiente dovezi pentru acel pixel. Parametrul de care depinde algoritmul transformării Hough este numărul coloanelor din acumulator. Cu cât acumulatorul are mai multe coloane, algoritmul este mai precis dar mai instabil. În schimb, cu cât acumulatorul transformării are mai puține coloane, algoritmul este mai stabil, dar nu atât de precis.

Algoritmul cu Translație la Medie

Din Capitolul 4.2.2.1 cunoaștem că Algoritmul cu Translație la Medie localizează maximum unei funcții de densitate dată de un set de date discrete. Prin deplasarea iterativă și analizarea unei fereaste Parzen, se pornește de la o estimare inițială x , și se detectează maximele densității de puncte [81]. Parametrii de care depinde Algoritmul cu Translație la Medie sunt dimensiunea ferestrei Parzen (în cazul nostru segment) și numărul de date care sunt cuprinse în această fereastră.

Având aceste cunoștințe despre metodele ce urmează a fi testate, ne propunem să alegem dimensiunea acumulatorului transformării Hough. Știm că plaja de unghiuri este de $0^\circ - 90^\circ$. Pentru a avea o precizie de 1° , construim acumulatorul din 90 de coloane, adică câte un unghi pentru fiecare colană din acumulator (90 de unghiuri). Dacă construim un acumulator de dimensiune mai mare, de exemplu 900 de coloane, atunci precizia transformării Hough este de $0,1^\circ$. Pentru un acumulator mai mare, de exemplu 9000 de coloane, precizia este de $0,01^\circ$ care deja este foarte bună. Generatorul de amprente Data Matrix generează amprente ale căror module sunt deviate cu σ . Pentru a vedea care este dimensiunea optimă pentru acumulator, testăm următoarele dimensiuni: [90, 900, 9000].

Pentru Algoritmul cu Translație la Medie, alegem segmentul „Parzen” jumătate din domeniul setului de date. În cazul în care toate datele din segment sunt selectate de algoritm, atunci precizia este maximă, dar timpul de lucru este foarte mare. Creșterea vitezei metodei se poate realiza numai prin selectarea unui număr mai mic din setul de date, dar în același timp scade și precizia. Pentru experiment alegem un număr de date de: 1%, 10% și 100% din numărul total de date ce cad în segmentul Parzen, ele fiind alese la distanțe egale. De asemenea, testăm metodele propuse pentru 3 tipuri de amprente Data Matrix, în funcție de abaterea modulelor $\sigma = [d/15, d/12, d/9, d/6]$. Algoritmii se testează în paralel, pentru fiecare situație realizându-se câte 100 de teste. Rezultatele testelor sunt reprezentate în Tabelele 5.1 - 5.3.

5.2 Rezultate

În cazul transformării Hough, cu cât abaterea modulelor este mai mare, indiferent de numărul de eşantioane din acumulator, crește și eroarea la detecția unghiului. În cazul Algoritmului cu Translație la Medie, abaterea modulelor codului Data Matrix influențează foarte puțin unghiul estimat. În Fig. 5.5 este reprezentată grafic eroarea medie pentru unghiurile estimate cu cele două metode.

Transformarea Hough este eficientă numai în cazul în care un număr mare de voturi intră în coloana corectă, astfel încât coloana poate fi ușor detectată din zgomotul de fundal. Acest lucru înseamnă că lățimea coloanelor nu trebuie să fie prea mică, altfel unele voturi vor cădea în coloanele vecine, reducându-se astfel vizibilitatea coloanei principale. Dacă analizăm Tabele 5.1 - 5.3 cu rezultatele experimentelor, observăm că,

Tabela 5.1: Experiment 1, Segment Parzen = 100 exemplare, Acumulator Hough = 90 eşantioane

Comparație între Transformarea Hough și Algoritmul cu Translație la Medie pentru estimarea unghiului amprentei DMC											
Serie teste	DMC (E_{max})	Eroare medie $ \alpha_{DMC} - \alpha_{Masurat} $		E_{min}		E_{max}		Estimări nereușite		Parametrizare	
		TH	ATM	TH	ATM	TH	ATM	TH	ATM	TH	ATM
1	d/15	0.9982	0.4644	0.0071	0.0045	15.92	10.02	1	0	90	100
2	d/12	9.0203	2.5161	0.1591	0.0116	22.20	20.94	9	0		
3	d/9	9.4872	2.6119	0.5588	0.0341	22.32	22.28	13	0		
4	d/6	9.3913	3.1246	0.4114	0.0459	22.02	19.44	18	0		

Tabela 5.2: Experiment 2, Segment Parzen = 1000 exemplare, Acumulator Hough = 900 eşantioane

Comparație între Transformarea Hough și Algoritmul cu Translație la Medie pentru estimarea unghiului amprentei DMC											
Serie teste	DMC (E_{max})	Eroare medie $ \alpha_{DMC} - \alpha_{Masurat} $		E_{min}		E_{max}		Estimări nereușite		Parametrizare	
		TH	ATM	TH	ATM	TH	ATM	TH	ATM	TH	ATM
1	d/15	1.1545	0.3666	0.0064	0.0044	17.80	4.750	0	0	900	1000
2	d/9	6.3865	2.7428	0.1235	0.0038	21.30	21.79	6	0		
3	d/3	8.6701	2.2995	0.1268	0.0290	22.39	21.16	13	0		
4	d/2	7.9894	3.3091	0.3313	0.0818	21.98	21.72	11	0		

Tabela 5.3: Exp. 3, Segment Parzen = 10000 exemplare, Acumulator Hough = 9000 eşantioane

Comparație între Transformarea Hough și Algoritmul cu Translație la Medie pentru estimarea unghiului amprentei DMC											
Serie teste	DMC (E_{max})	Eroare medie $ \alpha_{DMC} - \alpha_{Masurat} $		E_{min}		E_{max}		Estimări nereușite		Parametrizare	
		TH	ATM	TH	ATM	TH	ATM	TH	ATM	TH	ATM
1	d/15	0.8208	0.5652	0.0093	0.0056	22.40	8.488	0	0	9000	10000
2	d/9	6.6697	2.7433	0.1460	0.0156	22.19	21.95	8	0		
3	d/3	8.0336	2.5994	0.4943	0.0277	22.29	20.89	11	0		
4	d/2	9.5836	2.7005	0.8312	0.0013	22.44	19.45	10	0		

În cazul în care acumulatorul este format din 90 de coloane, algoritmul funcționează cel mai bine, numărul total de estimări nereușite fiind cel mai mic (Tabelul 5.1). Dacă urmărim pe graficul din Fig. 5.5, în partea de sus, vedem că în acest caz eroarea medie este cea mai mare pentru toate experimentele, ajungând până la $9,39^\circ$, în cazul în care abaterea modulelor este maximă. De asemenea, în cazul în care numărul de parametri este mare, numărul mediu de voturi exprimate într-o singură coloană este foarte scăzut. Acele coloane corespund la o figură reală în imagine și nu apar neapărat să aibă un număr mult mai mare de voturi decât vecinii lor. În experimentele

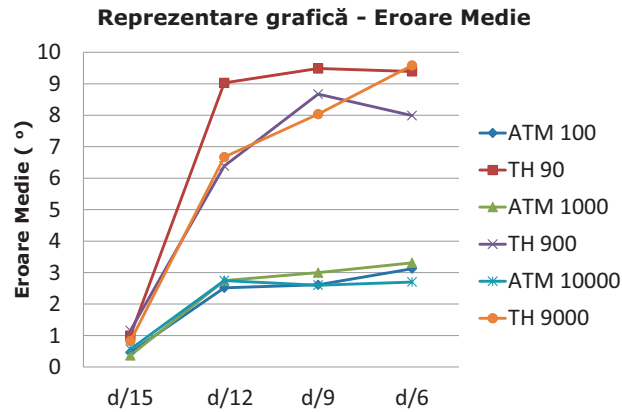


Figura 5.5: Eroarea medie

realizate, se poate observa că în cazul în care acumulatorul este foarte mare, la fel și abaterea modulelor, algoritmul devine instabil, astfel înregistrându-se mai multe estimări nereușite și chiar o eroare medie de maxim $9,58^\circ$ (Tabelul 5.2). Putem concluziona că estimarea pe baza transformării Hough este fiabilă în cazul în care dimensiunea acumulatorului este mică (90 de coloane) și abaterea modulelor este minimă ($E_{max} = d/15$), în acest caz înregistrându-se o eroare medie de maxim $0,99^\circ$ (Tabelul 5.1). În cele din urmă, eficiența transformării Hough este dependentă de calitatea datelor de intrare. Utilizarea transformării Hough pentru imaginile cu zgomot este o chestiune foarte delicată.

Algoritmul cu Translație la Medie este o tehnică neparametrică de grupare, care nu are nevoie de o cunoaștere prealabilă a numărului de grupuri și nu constrânge forma de grupuri. Este nevoie ca parametrul de lățime de bandă h , să fie reglat. Alegerea lățimii de bandă influențează rata de convergență și numărul de grupuri. Un h mare ar putea duce la clustering incorect și s-ar putea fuziona grupări distincte. Un h foarte scăzut ar putea conduce la grupuri prea multe. Algoritmul cu Translație la Medie ar putea să nu funcționeze bine la dimensiunile mai mari. La dimensiunile mai mari, numărul de maxime locale este destul de mare și ar putea converge către un optim local prea curând, dar, pentru că în cazul nostru setul de unghiuri este unidimensional, algoritmul funcționează cu rezultate foarte bune. În urma testelor realizate putem remarca independența algoritmului față de numărul de parametri din segmentul Parzen. Se obține un rezultat foarte bun în cazul în care abaterea modulelor este mică ($E_{max} = d/15$), eroarea medie fiind doar de $0,36^\circ$ (Tabelul 5.2). Bineînțeles, în realitate modulele au o distribuție aleatoare, dar și în aceste cazuri când E_{max} ajunge până la $d/6$, pentru orice număr de parametri, dacă vedem în graficul din Fig. 5.5, partea de jos, tragem concluzia că eroarea medie nu crește foarte mult, ajungând în cel mai rău caz la maxim $3,3^\circ$ (Tabelul 5.2), în acest caz mult mai mică decât cazul transformării Hough ($9,53^\circ$).

Fiind independent de alegerea lățimii de bandă, în cazul nostru Algoritmul cu Translație la Medie ne-a furnizat rezultate mai bune în estimarea unghiului orientării amprentei codului Data Matrix.

Capitolul 6

Scanarea Codului Data Matrix industrial

Codul Data Matrix industrial poate fi marcat pe diferite obiecte folosite în industria electronică, electrotehnică sau mecanică. Obiectele pot avea diferite forme, iar uneori găsirea unei zone plană pe suprafața obiectului pentru marcarea codului este imposibilă. În aceste cazuri codul Data Matrix se marchează pe suprafața neplană a obiectului. Din această cauză, în capitolul de față, se prezintă două metode de scanare a codului în funcție de suprafața pe care este marcat.

Prima metodă este destinată suprafețelor plane, scanarea fiind vectorială pe baza unghiului orientării și a distanței dintre modulele amprentei codului Data Matrix.

A doua metodă este dezvoltată pentru codurile marcate pe suprafețe cilindrice și sferice. În acest caz, scanarea se efectuează pe baza unei Matrice de transformare geometrică calculată pornind de la modulele amprentei de identificare și de sincronizare [69].

6.1 Suprafețe plane

Modulele codurilor Data Matrix ce sunt marcate pe suprafețe plane, sunt distribuite uniform în amprenta codului. Distanța dintre două module este aproximativ egală pentru toate modulele, iar unghiul creat de două module cu originea este aproximativ egal cu orientarea amprentei codului Data Matrix. Se scanează marginile amprentei codului Data Matrix, pentru identificarea numărului de rânduri și de coloane ale amprentei codului Data Matrix. Pe fundamentul acestor date, se scanează pe rând fiecare poziție din amprenta codului Data Matrix, construindu-se o matrice de coordonate ale centrelor modulelor codului Data Matrix.

6.1.1 Extragerea numărului de rânduri și de coloane ale amprentei codului Data Matrix

Scopul acestui bloc este de a verifica numărul rândurilor și al coloanelor codului Data Matrix [82]. Aceste numere sunt de fapt cunoscute prin împărțirea distanței dintre colțurile amprentei la distanța dintre module. Din cauza posibilei erori de perspectivă, distanța dintre module poate fi uneori neregulată. Pentru a avea precizie și siguranță cu privire la numărul de rânduri și de coloane, este recomandată efectuarea acestei operații de verificare.

Pentru a realiza acest lucru, se scanează amprenta de identificare, numărând modulele detectate, așa cum este prezentat în Fig. 6.1.

Se cunoaște orientarea amprentei și distanța dintre modulele amprentei. Utilizând aceste informații, mai întâi, algoritmul scanează o parte a amprentei de identificare, din originea amprentei spre exterior. Fiecare modul este scanat într-o fereastră de mărimea distanței dintre module. Pentru siguranța că niciun modul nu va fi ratat se scanează un pas în direcția opusă de scanare, pentru a verifica dacă originea este identificată corect. În continuare, procesul de scanare explorează în direcția inițială până ce fereastra de scanare părăsește regiunea de interes. Apoi, se oprește. Toate modulele detectate sunt numărate, ele reprezentând numărul de linii sau de coloane ale codului Data Matrix.

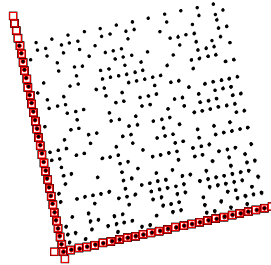


Figura 6.1: Extragerea numărului de rânduri ale codului Data Matrix

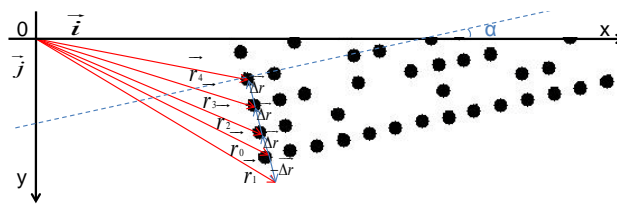


Figura 6.2: Scanarea ampretei de identificare a codului Data Matrix

În Fig. 6.2 este prezentată metoda de scanare a ampretei de identificare a codului Data Matrix.

Pornind din colțul principal $O(x_0, y_0)$, ce are vectorul:

$$\vec{r}_0 = x_0 \cdot \vec{i} + y_0 \cdot \vec{j}, \quad (6.1)$$

ferestra este deplasată în sens opus după vectorul:

$$\vec{\Delta r} = d \cdot \cos(\alpha) \cdot \vec{i} + d \cdot \sin(\alpha) \cdot \vec{j}, \quad (6.2)$$

se începe să se scaneze în direcția normală de scanare.

În aceeași manieră, cealaltă margine a ampretei este scanată, numărând modulele detectate. Știind că forma ampretei codului Data Matrix este pătratică, numărul de rânduri trebuie să fie egal cu numărul de coloane.

6.1.2 Scanarea și detecția modulelor codului Data Matrix

În acest bloc este scanat fiecare modul al ampretei codului Data Matrix, astfel, creându-se o matrice cu coordonate ce reprezintă modulele marcate („1”) și nemarcate („0”) din cod [50]. Scanarea modulelor pornește din colțul principal, pe baza distanței dintre module și a unghiului orientării ampretei codului Data Matrix. Folosind o fereastră, se caută în rânduri și în coloane fiecare modul marcat, creând o matrice de coordonate ale modulelor marcate și nemarcate. Dacă în fereastră există un modul, atunci elementul din matrice este un „1”. În cazul în care fereastră este goală, atunci elementul din matrice este un „0”. În cazul în care este detectat un modul, centrul ferestrei este ajustat la centrul de greutate al modulului. Dimensiunea ferestrei este egală cu distanța dintre module. Folosind această operație, acumularea de eroare este eliminată. Această operație se repetă până când toate modulele dintr-un rând sunt scanate. Scanarea continuă cu rândul următor până când toate rândurile sunt verificate. În Fig. 6.3 sunt prezentați pașii vectoriali de scanare a modulelor codului Data Matrix.

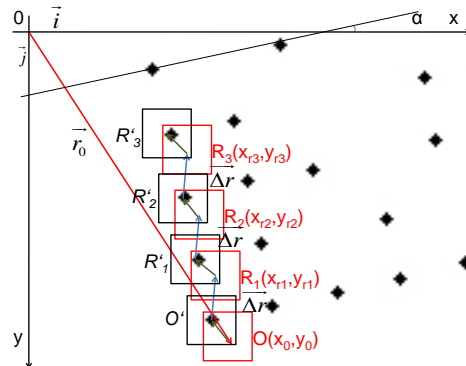


Figura 6.3: Ajustarea ferestrei de scanare

Se pornește din colțul principal $O(x_0, y_0)$, ce are vectorul:

$$\vec{r}_0 = x_0 \cdot \vec{i} + y_0 \cdot \vec{j}. \quad (6.3)$$

Utilizând orientarea amprenteii, α , și a distanței dintre module, d , fereastra este deplasată în coloană după vectorul:

$$\vec{\Delta r}_c = d \cdot \cos(\alpha) \cdot \vec{i} + d \cdot \sin(\alpha) \cdot \vec{j}, \quad (6.4)$$

poziția următoare $C_1(x_{c1}, y_{c1})$, ce are vectorul:

$$\vec{r}_{c1} = \vec{r}_0 + \vec{\Delta r}_c = x_0 \cdot \vec{i} + y_0 \cdot \vec{j} + d \cdot \cos(\alpha) \cdot \vec{i} + d \cdot \sin(\alpha) \cdot \vec{j}. \quad (6.5)$$

Predicția pentru C_1 este:

$$\begin{cases} x_1 = x_0 + d \cdot \cos(\alpha) \\ y_1 = y_0 + d \cdot \sin(\alpha) \end{cases}. \quad (6.6)$$

Dar coordonatele centrului de greutate ale modulului detectat sunt: $C'_1(x'_{c1}, y'_{c1})$, apoi trebuie ajustat centrul ferestrei la centrul de greutate al modulului:

$$\begin{cases} x_1 = x'_1 \\ y_1 = y'_1 \end{cases}, \quad (6.7)$$

ce reprezintă corecția ferestrei de scanare.

Această operație de scanare este aplicată în același mod pentru toate rândurile din codul Data Matrix, folosind vectorul:

$$\vec{\Delta r}_r = d \cdot \sin(\alpha) \cdot \vec{i} - d \cdot \cos(\alpha) \cdot \vec{j}. \quad (6.8)$$

După ce sunt scanate toate modulele, este scrisă o matrice cu coordonatele modulelor (Fig. 6.4).

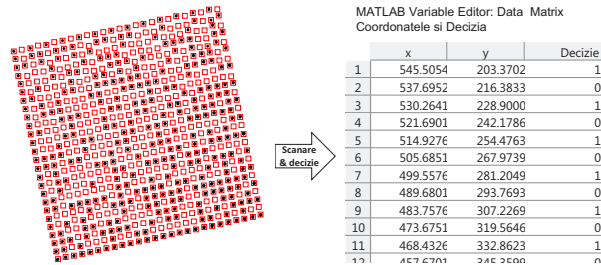


Figura 6.4: Procesul de scanare a codului Data Matrix

6.1.3 Concluzii

Pentru a concluziona această metodă de scanare prezentată, propunem să supunem scanării câteva tipuri de materiale marcate cu cod Data Matrix. Amprenta codului Data Matrix poate fi achiziționată și recunoscută cu una din cele două metode prezentate în Capitolul 4. Se iau ca exemple aceleași materiale inscripționate cu cod Data Matrix, testate în Capitolul 4.1.8.

În Fig. 6.5 - 6.8, se dau patru exemple de materiale marcate cu cod Data Matrix, la care se cunoaște regiunea de interes. În Fig. 6.5 codul este punctat pe cupru lucios, dimensiunea amprentei este de $2 \times 2 \text{ cm}^2$, iar în Fig. 6.6 codul este punctat pe aluminiu lucios, dimensiunea amprentei fiind de $1,2 \times 1,2 \text{ cm}^2$.

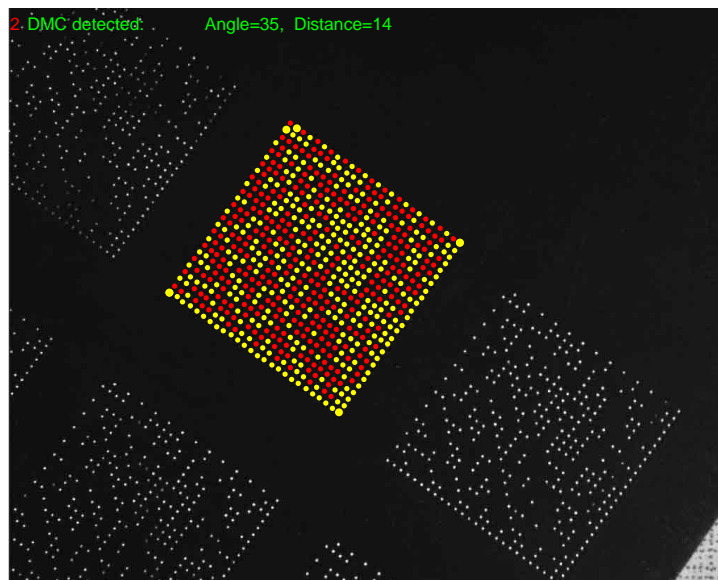


Figura 6.5: Cupru lucios, dimensiune amprentă $2 \times 2 \text{ cm}^2$

În Fig. 6.7 codul este punctat pe fier acoperit cu pete de rugină, dimensiunea amprentei este de $1,2 \times 1,2 \text{ cm}^2$, iar în Fig. 6.8 codul este punctat pe metal acoperit vopsea cu o anumită textură, dimensiunea amprentei fiind de $2 \times 2 \text{ cm}^2$.

În testele prezentate anterior, modulele sunt detectate corect, chiar și în cazurile în care obiectele au fundal neregulat sau între module se găsesc pete de rugină sau

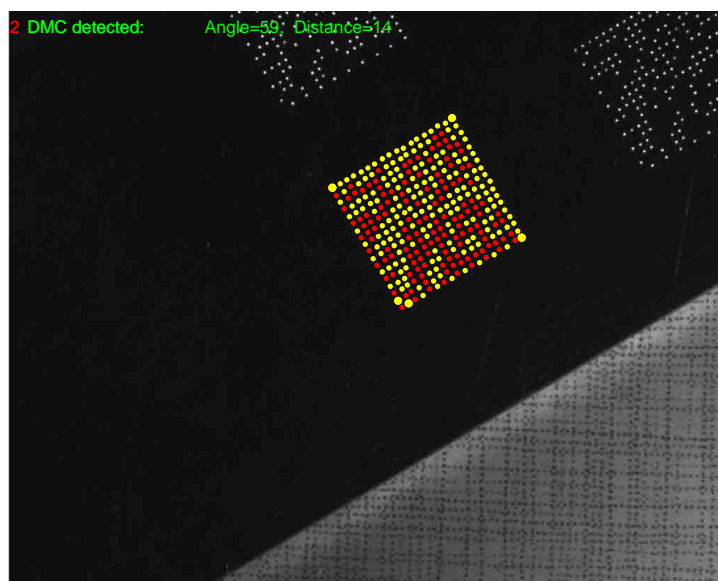


Figura 6.6: Aluminiu lucios, dimensiune amprentă $1,2 \times 1,2\text{cm}^2$

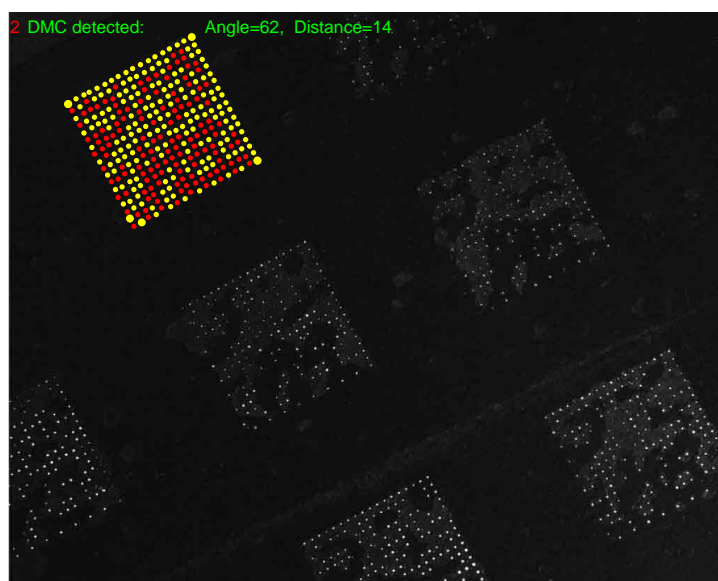


Figura 6.7: Fier acoperit cu pete de rugină, dimensiune amprentă $1,2 \times 1,2\text{cm}^2$

vopsea. Menționez că în figurile prezentate mai sus, cu galben sunt reprezentate modulele detectate ce indică valoarea „1”, iar cu roșu sunt reprezentate modulele nemarcate ce indică valoarea „0”. Aceste date sunt scrise în matricea binară echivalentă codului Data Matrix ce urmează a fi decodată. Singura cerință obligatorie pentru ca această etapă de scanare să funcționeze cu rezultate bune, este ca imaginea

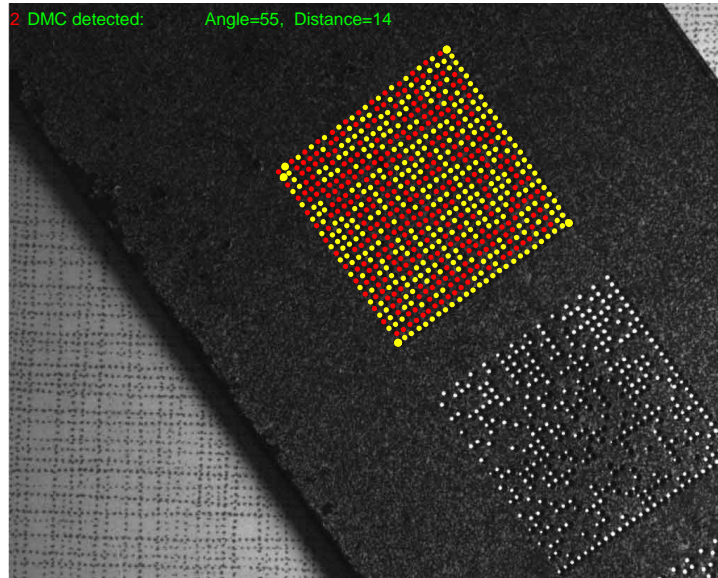


Figura 6.8: Metal vopsit, dimensiune amprentă $2 \times 2 \text{ cm}^2$

sau calitatea marcării codului să fie suficient de bună încât să se poate face distincție între modulele amprentei.

6.2 Suprafețe neplane

În multe cazuri, din cauza sistemului optic al camerei foto și din cauza spațiilor inaccesibile este dificil de a achiziționa o imagine perfect ortogonală a codului Data Matrix. Astfel, ne confruntăm cu situația de a nu se recunoaște codul sau de a nu se extrage corect informația dorită. Pentru codurile Data Matrix industriale marcate pe suprafețe plane, fără erori de perspectivă, procesul de scanare se efectuează pe baza unghiului orientării amprentei și a distanței dintre module, după procedura prezentată în Capitolul 6.1. Această metodă este simplă și eficientă pentru suprafețele plane, dar pentru cazurile în care imaginea codului Data Matrix are erori mari de perspectivă este inutilă. Pentru a depăși acest neajuns, vom descrie o metodă eficientă pentru scanarea codurilor Data Matrix industriale pentru situațiile speciale, cum ar fi erori de perspectivă sau suprafețe cilindrice și sferice.

Pe baza marginilor amprentei, se construiește o rețea imaginară de puncte, ce are aceeași mărime și orientare cu amprenta codului Data Matrix. Grila imaginară este proiectată peste imagine și în urma acesteia, imaginea codului Data Matrix este scanată, iar informația binară este extrasă.

6.2.1 Scanarea marginilor amprentei și extragerea punctelor de control

Marginile amprentei codului Data Matrix sunt scanate pentru găsirea poziției modulelor de-a lungul amprentei de identificare și a celei de sincronizare. În Fig. 6.9 este afișată o imagine binară cu fiecare obiect etichetat a unui cod Data Matrix. De asemenea sunt

denumite colțurile amprenteii. Colțurile A și B sunt folosite pentru găsirea orientării primei margini cu axa x (relația (6.9)):

$$\theta_1 = \text{atan} \left(\frac{y_B - y_A}{x_B - x_A} \right). \quad (6.9)$$

Este calculat punctul care se află la distanța d și unghiul θ_1 cu punctul A (relația (6.10)), d fiind o primă estimare a distanțelor dintre module.

$$\begin{cases} x = x_1 + d \cdot \cos(\theta_1) \\ y = y_1 + d \cdot \sin(\theta_1). \end{cases} \quad (6.10)$$

În cazul suprafețelor neplane distanța dintre module nu este constantă în imagine, dar distanța dintre modulele adiacente va fi aproape la fel. Deci, din acest motiv, de fiecare dată valoarea distanței trebuie să fie actualizată cu valoarea calculată pentru modulul recent, care deja este cunoscută.

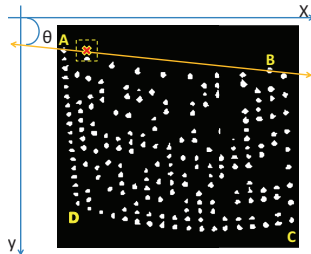


Figura 6.9: Scanarea modulelor marginilor amprenteii codului Data Matrix

De fiecare dată când este calculat un nou punct acesta este considerat ca și punctul curent. Mai întâi, este calculată valoarea unghiului folosind punctul curent și punctul B , fiind apoi actualizate. Începând cu punctul A , ca și punct curent, se estimează în mod repetat poziția modulului următor, ecuația (6.10).

Este construită o fereastră în jurul poziției approximate. Se caută în interiorul acesteia după obiectele dorite, calculând centrele acestora. Dimensiunea ferestrei este aceeași cu distanța dintre module. Numai un obiect are centrul de greutate dorit, restul trebuie eliminate. De exemplu, în Fig. 6.10 se consideră fereastra de scanare regiunea marcată cu un pătrățel. Fereastra conține două obiecte. Pentru a lua în considerare fiecare obiect, se calculează centrele de greutate ale acestora utilizând locațiile tuturor pixelilor aceluși obiect. Dacă L este eticheta unui obiect, se calculează centrul de greutate al obiectului L , folosind ecuația (6.11).

$$\begin{aligned} x_c &= \frac{\sum_{x \in L} x}{n}, \\ y_c &= \frac{\sum_{y \in L} y}{n}, \end{aligned} \quad (6.11)$$

unde x_c și y_c sunt coordonatele centrului de greutate ale acestui obiect.

În fereastra prezentată în Fig. 6.10 (stânga) există două obiecte: modulul și umbra acestuia. Doar un obiect poate fi cel adevărat, iar obiectele ce nu sunt necesare trebuie eliminate. În Fig. 6.10 (dreapta) centrozii a două obiecte sunt marcați cu 'X'.

Pentru a elimina obiectele inutile sunt calculate, de asemenea, unghiul creat de către centrozii cu punctul A și distanța dintre aceste două puncte la A . Având



Figura 6.10: Fereastra de scanare ce conține două obiecte

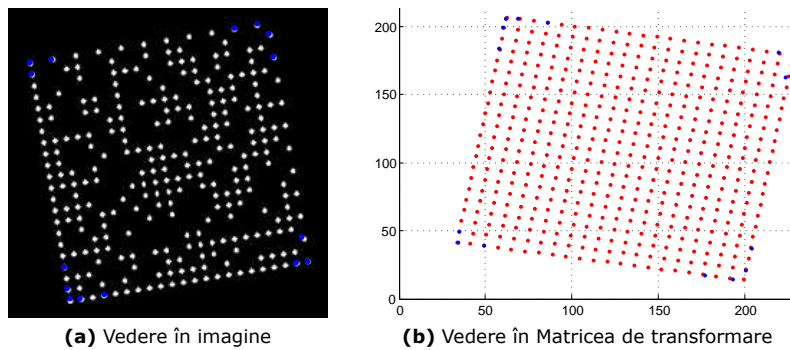


Figura 6.11: Selectarea punctelor de control

unghiurile formate de aceste două puncte cu A și distanța lor la A , aceste valori trebuie să îndeplinească câteva condiții pentru a fi punctul dorit. Dar, uneori, o umbră s-ar putea, de asemenea, să îndeplinească condițiile pentru punctul dorit. Pentru a elimina umbra modulului se ia în considerare ca modul dorit, obiectul cu suprafața cea mai mare. Având în vedere că umbrele au o suprafață mai mică decât modulul real, ele pot fi ușor eliminate. Prin eliminarea obiectelor inutile în fereastra de scanare rămâne un singur obiect ce este considerat modulul consecutiv luminos. Modulul curent este actualizat cu această valoare și procesul de scanare continuă.

Pentru calcularea unghiului, la început sunt folosite punctele A și B . După aceea unghiul este calculat folosind colțul A și modulul curent, valorile fiind actualizate după fiecare iterație. Pentru calculul distanței dintre module sunt utilizate pentru început ultimele două puncte de la marginea anterioară. Apoi colțul A și modulul curent sunt utilizate pentru actualizarea valorii după fiecare iterație.

Scanarea codului Data Matrix se realizează pe baza Matricii de transformare geometrice, iar pentru determinarea acesteia avem nevoie de puncte de control.

Selectarea punctelor de control (Fig. 6.11) este importantă deoarece procesul de transformare geometrică depinde de matricile ce sunt calculate pornind de la aceste valori. Punctele de control, reprezentate cu albastru, sunt alese din modulele de-a lungul marginilor codului Data Matrix. Experimental s-a selectat cincisprezece puncte de control din diferite părți ale codului. Cinci puncte fixe ce reprezintă colțurile codului Data Matrix, sunt folosite ca puncte de control. Modulele adiacente colțurilor sunt de asemenea, considerate puncte de control. Restul punctelor sunt alese astfel încât să fie distribuite uniform pe marginile amprentei codului Data Matrix.

Având aceste puncte de control și alte informații despre cod, cum ar fi dimensiunea reală și raza de curbură a amprentei, se poate construi grila estimată a modulelor codului Data Matrix. Această grilă estimată este o conexiune între poziția reală a modulelor codului și poziția lor în imagine.

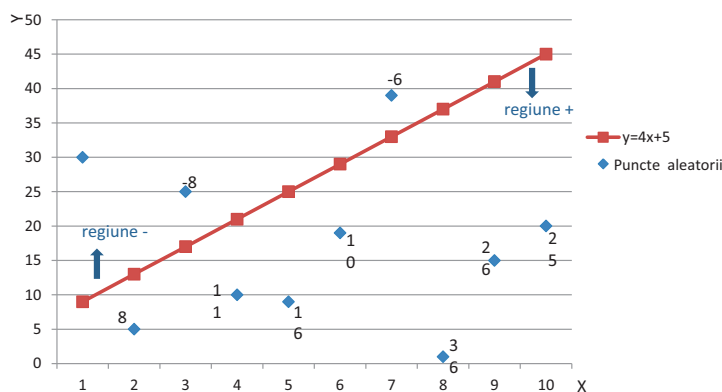


Figura 6.12: Demonstarea clasificării suprafețelor

6.2.2 Suprafețe curbe

Inițial, codurile Data Matrix standard (DMC) au fost dezvoltate pentru a fi imprimate cu culoare închisă pe suprafețe plane luminoase (de obicei imprimate pe hârtie) [15]. Cu toate acestea, tot mai multe coduri Data Matrix sunt folosite în mediul industrial pentru a identifica componentele într-un proces de fabricație [17]. Astfel, ele nu sunt doar tipărite, dar și gravate, arse cu laser, marcate direct pe diferite tipuri de suprafețe cum ar fi metal, plastic, etc [83]. Suprafețele pe care sunt marcate nu sunt numai plane, aspectul lor în imagine nu va mai fi o grilă dreptunghiulară.

În acest capitol este prezentată o metodă de localizare a modulelor codurilor Data Matrix industriale marcate pe suprafețe curbe. Pentru a face diferența între suprafețele cilindrice și cele sferice folosim orientarea marginilor amprentei codului Data Matrix. În cazul în care suprafața este cilindrică atunci, indiferent de poziția camerei foto, de fiecare dată în imagine două dintre marginile codului vor fi liniare și, în cazul în care camera foto este perpendiculară pe axa cilindrului, atunci toate cele patru laturi vor fi liniare. Cât despre suprafețele sferice, toate marginile sunt curbate. În plus laturile opuse sunt curbate în direcții opuse.

Se consideră „L”, o linie dreaptă cu panta m și interceptia c (Fig. 6.12). Punctele $P_1(x_1, y_1)$ și $P_2(x_2, y_2)$ sunt două puncte în planul XY. În cazul în care punctul P_1 se află pe dreaptă, atunci $L(P_1)$ va fi zero, de exemplu $y_1 = m \cdot x_1 + c$, iar dacă P_1 și P_2 se află de fiecare parte a dreptei, atunci valoarea va fi de semn opus. Folosind aceasta, poate fi determinat dacă laturile amprentei sunt curbate sau liniare.

Tabela 6.1: Clasificarea suprafețelor

Suprafața	lin 1	lin 2	lin 3	lin 4	lin
Plană	1	1	1	1	+4
Cilindrică	1	+2	1	+2	+6
	1	-2	1	-2	-2
	-2	1	-2	1	-2
Sferică	+2	1	+2	1	+6
	laturile opuse sunt orientate în direcții opuse				0

Se observă că, în acest fel, în imagine sunt definite două regiuni. Regiunea deasupra dreptei este regiunea negativă și regiunea de sub dreaptă este regiunea pozitivă. Dreapta L va fi dreapta care unește cele două colțuri ale codului Data Matrix și celelalte puncte reprezintă centrele modulelor de-a lungul marginii. Înlocuind cu aceste puncte în ecuația dreptei se poate constata dacă acestea se află pe linie, în regiunea negativă sau în cea pozitivă. Dacă un punct se află aproximativ pe dreaptă, acestuia i se dă valoarea „1”. În cazul în care punctele se află pe cealaltă parte a marginii, atunci i se atribuie valoarea „-2” sau „+2”, în funcție de semnul restant. Să presupunem că semnul restant al marginii a doua este pozitiv și deci, are valoarea „+2” și este același cu semnul marginii a patra. De asemenea, are aceeași valoare „2”. Întrucât prima și a treia margine sunt liniare, atunci acestea primesc valoare „1”. Suma dată de valorile marginilor este folosită pentru a clasifica suprafețele. Pentru suprafețele sferice, toate marginile sunt curbate, iar marginile opuse sunt curbate în direcții opuse una față de cealaltă. Aceasta indică faptul că suma totală va fi zero pentru suprafața sferică.

În cazul suprafețelor cilindrice, pentru a crea grila predicțiilor trebuie să știm lungimea codului, raza cilindrului și numărul rândurilor și al coloanelor codului Data Matrix. Mai întâi, se calculează grila în coordonate cilindrice, după care este convertită în coordonate carteziene.

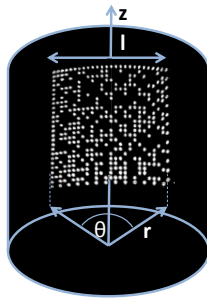


Figura 6.13: Codul Data Matrix inscripționat pe un cilindru

Este important să fie cunoscută orientarea amprentei codului Data Matrix față de axa cilindrului. Odata ce orientarea amprentei pe suprafață este cunoscută, se poate începe crearea grilei imaginare. În primul rând trebuie să fie determinat unghiul creat de cod cu axa cilindrului:

$$\theta = \frac{l}{r}, \quad (6.12)$$

unde θ este unghiul, l este lungimea (lungimea arcului) codului iar r este raza cilindrului.

Acesta trebuie împărțit în părți egale, pentru a se cuprinde cu o parte din unghi fiecare din rândurile sau coloanele codului. În cazul în care codul Data Matrix are n rânduri și n coloane, atunci θ este împărțit în $(n - 1)$ părți. Este nevoie de n sloturi pentru fiecare rând sau coloană. Așa că se calculează o matrice astfel încât să aibă n distanțe egale și diferența dintre valoarea maximă și cea minimă să fie egală cu valoarea θ . Dimensiunea pasului este $l/(r \cdot (n - 1))$ unde $l/(n - 1)$ este distanța dintre module și este reprezentată cu valoarea d . Considerăm sistemul de coordonate cilindrice, (r, θ, z) , unde r este raza cilindrului, θ indică unghiul poziției modulului iar h indică înălțimea. Astfel, există un rând sau o coloană la fiecare unghi în matrice:

$$\theta = \left(-\frac{x_{mod}}{2} + \frac{1}{2} \right) \cdot \frac{d}{r} \rightarrow \frac{d}{r} \rightarrow -\frac{x_{mod}}{2} \cdot \frac{d}{r}, \quad (6.13)$$

x_{mod} este numărul de rânduri din amprenta codului și y_{mod} este numărul de coloane din amprentă. A treia dimensiune depinde de lungimea codului și de numărul de module de-a lungul acelei dimensiuni.

După ce sunt cunoscute pozițiile modulelor în coordonatele cilindrice, acestea urmează să fie convertite în coordonate carteziene. Să presupunem (X, Y, Z) sunt coordonatele carteziene și (r, θ, z) sunt coordonate cilindrice, atunci:

$$\begin{aligned} X &= r * \cos(\theta), \\ Y &= r * \sin(\theta), \\ Z &= z. \end{aligned} \quad (6.14)$$

După ce grila este construită se calculează matricile pentru implementarea Matricii de transformare geometrice.

6.2.3 Estimarea poziției modulelor amprentei cu Matricea de transformare geometrică

Calibrarea camerei este una din problemele fundamentale în grafica computerizată [84]. Procesul de calibrare stabilește modelul de calcul pe baza informațiilor pixelilor cunoscuți în spațiul 2-D al imaginii pentru a obține informații geometrice de mare precizie ale obiectului 3-D. Corelația dintre locația spațială a obiectului și punctului corespondent în imagine este determinată de modelul geometric al camerei. Procesul de realizare a modelului parametric se numește etalonarea camerei [85].

Transformarea Direct Liniară (DLT) este una dintre cele mai populare metode de calibrare (etalonare) [86]. Esența DLT este de a obține parametrii camerei prin rezolvarea ecuațiilor liniare în mod direct. Cu Transformarea Direct Liniară, valorile inițiale ale parametrilor de orientare interiori nu sunt necesari, dar cel puțin 6 puncte ale căror coordonate în imagine și coordonatele reale corespondente trebuie să fie cunoscute.

Începem cu algoritmul liniar, simplu pentru determinarea lui H (Matricea de transformare), atunci când este dat un set de patru puncte corespondente dintr-un spațiu 2D către un alt spațiu 2D, $X_i \rightarrow X'_i$. Înainte de a explica Matricea de transformare să vedem cum un obiect este proiectat în imagine [85].

Considerăm O un obiect într-un spațiu bidimensional și punctul său corespondent în imagine este I . N este centrul obiectivului camerei (foto) folosite. Linia ce unește orice obiect din spațiul obiectului cu punctul corespunzător din imagine trece prin centrul lentilei camerei video. Deci, punctele I , N și O sunt coliniare [87]. În Fig. 6.14 sunt definite două cadre de referință, cadrul imaginii cu axele u și v și cadrul obiectului cu axele de referință X și Y . Coordonatele obiectului și punctul corespunzător din imagine sunt de asemenea indicate. Punctul N este numit centrul proiecției [87].

Se adaugă a treia dimensiune a spațiului obiectului pentru a stabili pe N în spațiu. Acum, să presupunem că poziția lui N în cadrul spațiului de referință a obiectului este $[x_0, y_0, z_0]$. Vectorul A , trasat din punctul O la N devine $[x - x_0, y - y_0, -z_0]$, deoarece coordonata a treia O este „zero”. După omogenizarea coordonatelor, coordonatele planului imagine și coordonatele obiect-spațiu sunt corelate folosind Matricea de transformare după cum se arată mai jos. Întrucât s-a considerat că spațiul obiectului, este bidimensional, coordonata „z” nu este folosită:

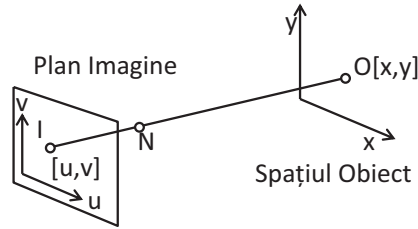


Figura 6.14: Proiecția unui obiect în imagine

$$\begin{bmatrix} ku \\ kv \\ k \end{bmatrix} = \begin{bmatrix} H_1 & H_2 & H_3 \\ H_4 & H_5 & H_6 \\ H_7 & H_8 & H_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (6.15)$$

în cazul în care, (x, y) sunt coordonatele obiectelor din spațiul real și (u, v) reprezintă punctele proiectate în imagine.

Transformarea este dată de ecuația $X'_i = H \cdot X_i$. Aceasta este o ecuație ce implică vectori omogeni, astfel vectorii: X'_i și $H \cdot X_i$ nu sunt egali, au aceeași direcție, însă diferă în mărime cu un factor de scară nenul. Această formulare permite o soluție liniară, simplă pentru H ce urmează să fie derivat [87].

În cazul în care rândul j^{th} din matricea H se notează cu h^{jT} , atunci putem scrie:

$$H \cdot X_i = \begin{bmatrix} h^{1T} \cdot X_i \\ h^{2T} \cdot X_i \\ h^{3T} \cdot X_i \end{bmatrix}, \quad (6.16)$$

scriind pe $X'_i = (x'_i, y'_i, w'_i)$, în mod explicit ca:

$$X'_i \cdot H X_i = \begin{bmatrix} y'_i h^{3T} X_i - w'_i h^{2T} X_i \\ w'_i h^{1T} X_i - x'_i h^{3T} X_i \\ x'_i h^{2T} X_i - y'_i h^{1T} X_i \end{bmatrix}. \quad (6.17)$$

Datorită $h^{jT} X_i = X_i^T h^j$ pentru $j = 1, 2, 3$, aceasta oferă un set de trei ecuații pentru intrările lui H , ce pot fi scrise sub forma:

$$\begin{bmatrix} 0^T & -w'_i X_i^T & y'_i X_i^T \\ w'_i X_i^T & 0^T & -x'_i X_i^T \\ -y'_i X_i^T & x'_i X_i^T & 0^T \end{bmatrix} \cdot \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0. \quad (6.18)$$

Acestea au forma $A_i h = 0$, unde A_i este o matrice de 3×9 , și h este un vector format din intrările matricei H , cu h_i al i^{th} element al lui h :

$$h = \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix}, H = \begin{bmatrix} h^1 & h^2 & h^3 \\ h^4 & h^5 & h^6 \\ h^7 & h^8 & h^9 \end{bmatrix} \quad (6.19)$$

Cu privire la aceste ecuații pot fi prezentate următoarele observații:

- Ecuația $A_i h = 0$, este o ecuație liniară cu necunoscuta h . Elementele matricii A_i sunt pătratice în coordonatele cunoscute ale punctelor.
- Deși există trei ecuații numai două dintre ele sunt liniar independente. Eliminând utilizarea ecuației a treia, fiecare punct oferă două ecuații pentru rezolvarea lui H .

- Ecuațiile rețin reprezentarea coordonatelor punctului $(x_i, y_i, w_i)^T$. Se poate alege $w_i' = 1$, ceea ce înseamnă că (x_i', y_i') sunt coordonatele măsurate în imagine. În caz contrar, poate fi ales ca și factor de scalare k , care impune unele modificări în coordonatele $(kx_i', ky_i', k)^T$. Deci X_i' devine $(kx_i', ky_i', k)^T$. Matricea cu X_i' extinsă este prezentată mai jos.

$$\begin{bmatrix} 0 & 0 & 0 & -w_1'x_1 & -w_1'y_1 & -w_1'w_1 & y_1'x_1 & y_1'y_1 & y_1'w_1 \\ w_1'x_1 & w_1'y_1 & w_1'w_1 & 0 & 0 & 0 & x_1'x_1 & x_1'y_1 & x_1'w_1 \\ 0 & 0 & 0 & -w_2'x_2 & -w_2'y_2 & -w_2'w_2 & y_2'x_2 & y_2'y_2 & y_2'w_2 \\ w_2'x_2 & w_2'y_2 & w_2'w_2 & 0 & 0 & 0 & x_2'x_2 & x_2'y_2 & x_2'w_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -w_9'x_9 & -w_9'y_9 & -w_9'w_9 & y_9'x_9 & y_9'y_9 & y_9'w_9 \\ w_9'x_9 & w_9'y_9 & w_9'w_9 & 0 & 0 & 0 & x_9'x_9 & x_9'y_9 & x_9'w_9 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ \vdots \\ h_9 \end{bmatrix} = [0]. \quad (6.20)$$

Pentru matricea H , ne presupunem că $h_9 = 1$ și mai rămân opt necunoscute, necesitând opt ecuații pentru găsirea soluției. Din cele patru puncte de corespondență cunoscute se obține un set de ecuații $Ah = 0$, unde A este matricea coeficienților ecuației construite din rândurile matricii A_i ce au contribuit la fiecare corespondență și h este vectorul intrărilor necunoscute ale lui H . Matricea A este prezentată mai jos, cu unele modificări. Se caută o soluție non-zero lui h , deoarece soluția evidentă $h = 0$ nu prezintă interes [69].

Pentru a avea o soluție mai bună pentru sistem, utilizăm un număr de ecuații mai mare decât necunoscutele. Având în vedere că punctele de corespondență nu sunt exacte, nu va exista nicio soluție exactă a sistemului supradeterminat. Prin urmare, aceste ecuații trebuie să fie rezolvate cu ajutorul altor tehnici. Matricile de mai jos reprezintă, de asemenea, același sistem ca și cel anterior - ecuația (6.21). Unde $(ku, kv, k) \rightarrow (x_i', y_i', w_i')$ sunt punctele din imagine și $(X, Y, 1) \rightarrow X_i^T(x_i, y_i, w_i)$ sunt coordonatele din spațiul real. O transformare de la 3-D la 2-D poate fi realizată folosind următoarele:

$$\begin{bmatrix} ku \\ kv \\ k \end{bmatrix} = \begin{bmatrix} H_1 & H_2 & H_3 & H_4 \\ H_5 & H_6 & H_7 & H_8 \\ H_9 & H_{10} & H_{11} & H_{12} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (6.21)$$

Din aceste matrici putem elimina variabila k prin înlocuire:

$$k = H_9x + H_{10}y + H_{11}z + H_{12}. \quad (6.22)$$

Atunci, avem două ecuații independente în H :

$$\begin{aligned} H_1x + H_2y + H_3z + H_4 + H_5(0) + H_6(0) + H_7(0) + H_8(0) + \\ + H_9ux + H_{10}uy + H_{11}uz + H_{12}u = 0. \end{aligned} \quad (6.23)$$

Cu diferite puncte de corespondență poate fi construită o matrice folosind ecuațiile de mai sus. Este de forma $A \cdot H = 0$:

$$\begin{bmatrix}
 x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 & -u_1 \\
 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1x_1 & -v_1y_1 & -v_1z_1 & -v_1 \\
 x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 & -u_2z_2 & -u_2 \\
 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v_2 \cdot x_1 & -v_2y_1 & -v_2z_1 & -v_2 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 x_n & y_n & z_n & 1 & 0 & 0 & 0 & -u_nx_n & -u_ny_n & -u_nz_n & -u_n & -u_n \\
 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -v_nx_n & -v_ny_n & -v_nz_n & -v_n
 \end{bmatrix}
 \begin{bmatrix}
 H_1 \\
 H_2 \\
 H_3 \\
 H_4 \\
 H_5 \\
 H_6 \\
 H_7 \\
 H_8 \\
 H_9 \\
 H_{10} \\
 H_{11} \\
 H_{12}
 \end{bmatrix}
 = [0]
 \quad (6.24)$$

Prin utilizarea metodei de descompunere a valorii singulare, se pot calcula valorile H . Aceste valori nu sunt exacte, deoarece punctele de corespondență sunt inexacte. În loc de a folosi aceste valori pentru Matricea de transformare, le folosim pentru calcularea valorii lui k . Eliminând variabila k din ecuații și rezolvând cele trei ecuații, putem găsi soluția lui H .

6.2.4 Detecția modulelor codului Data Matrix

Proiectând grila de estimare a poziției modulelor peste imagine pot fi localizate aproximativ toate modulele. Aceasta înseamnă că fiecare punct din imagine reprezintă centrul unui modul din cod. Există unele erori din cauza punctelor inexacte de corespondență ce au fost folosite. Din acest motiv, cu ajutorul unei ferestre de scanare, se caută în rândurile și coloanele amprente codului după fiecare modul, construind o matrice cu coordonatele modulelor marcate și nemarcate. Dimensiunea ferestrei de scanare este egală cu distanța dintre module. Centrul ferestrei urmărește grila de preziceri (prin Matricea de transformare) a codului Data Matrix [82].

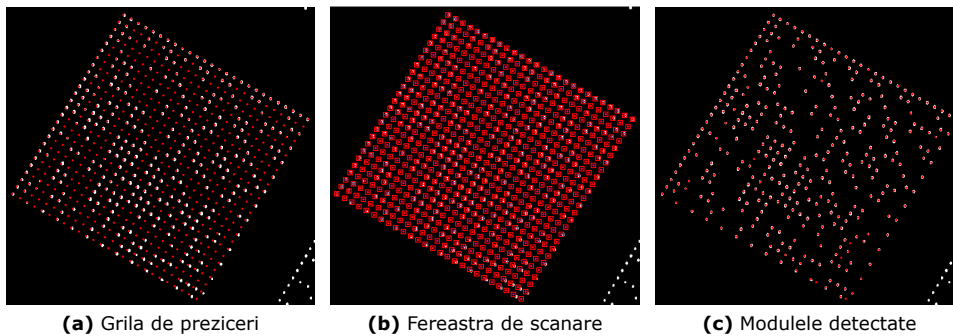


Figura 6.15: Scanare pe baza Matricei de transformare

În Fig. 6.15 (a) este reprezentată grila obținută cu Matricea de transformare iar în Fig. 6.15 (b) grila este proiectată peste imagine. Dacă observăm cu atenție, centrele modulelor sunt puțin deplasate față de grila estimată, pentru aceasta este necesară scanarea modulelor în interiorul ferestrei. În acest caz avem o toleranță de scanare, fiind detectate inclusiv modulele ce nu cad pe grilă.

6.2.5 Concluzii

Pentru a localiza modulele din interiorul amprentei codului Data Matrix este necesar să se scaneze în imagine. Dar programul de scanare nu știe cum și unde să efectueze scanarea. În acest sens, această soluție oferă scannerului locații aproximative cu poziția modulelor. Se scanează în jurul locației prezisă în cadrul unei ferestre de scanare pentru a găsi poziția exactă a modulelor. Acest algoritm funcționează dacă amprenta codului Data Matrix este rotită în planul ei, sau dacă punctul de vedere al camerei nu este perpendicular cu suprafața obiectului. În timp ce modulele nu sunt conectate între ele, pozițiile lor pot fi găsite, aceasta fiind o cerință necesară și obligatorie scanării.

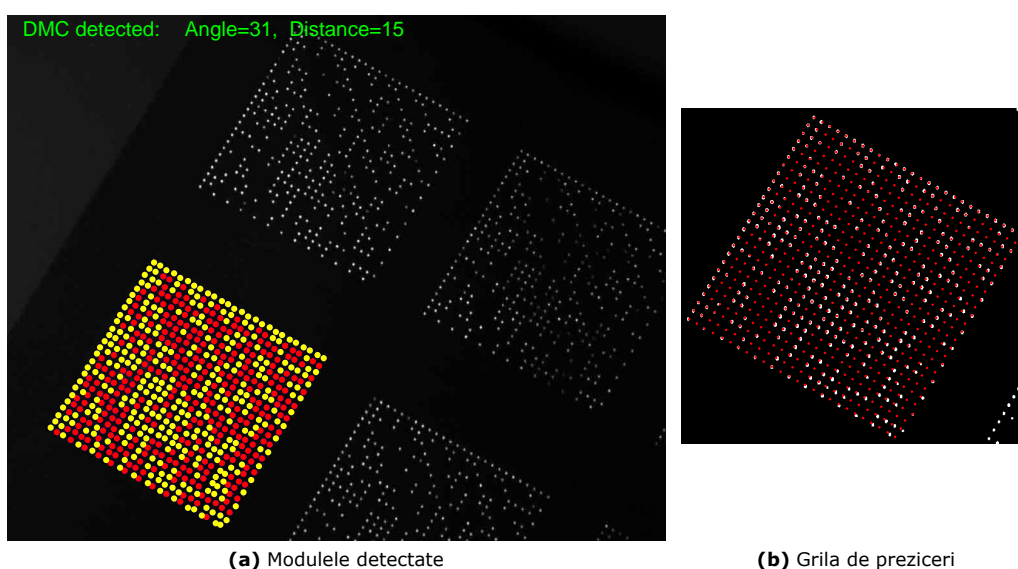


Figura 6.16: Ex.1 - Cod Data Matrix cu erori de perspectivă

În Fig. 6.16 este prezentată o situație în care imaginea codului Data Matrix are erori de perspectivă. În Fig. 6.17 - 6.18, sunt prezentate două situații în care codul Data Matrix este marcat pe suprafețe neplane, respectiv cilindrică și sferică. Transformarea Direct Liniară ia punctele extreme ale codului Data Matrix și calculează grila prezicerilor distribuției modulelor amprentei Data Matrix pe suprafața obiectului marcat. Informațiile sunt extrase pe baza acestei grile de preziceri, fiind analizate modulele ce se intersectează cu grila. În reprezentațiile grafice din acest capitol, grila de preziceri construită pe baza Matricii de transformare este reprezentată în Fig. 6.16(b) - 6.18 (b) cu roșu. În Fig. 6.16(a) - 6.18 (a), se pot observa modulele marcate și detectate, reprezentate cu roșu, iar modulele nemarcate sunt reprezentate grafic cu galben.

Folosind această metodă, nu imaginea este transformată geometric, ci este creată doar o grilă de poziții pe baza imaginii [88]. Avantajul este că imaginea nu este procesată, informațiile din imagine fiind nemodificate, de asemenea sistemul de procesare nu are nevoie de multă putere de calcul pentru scanare și procesare.

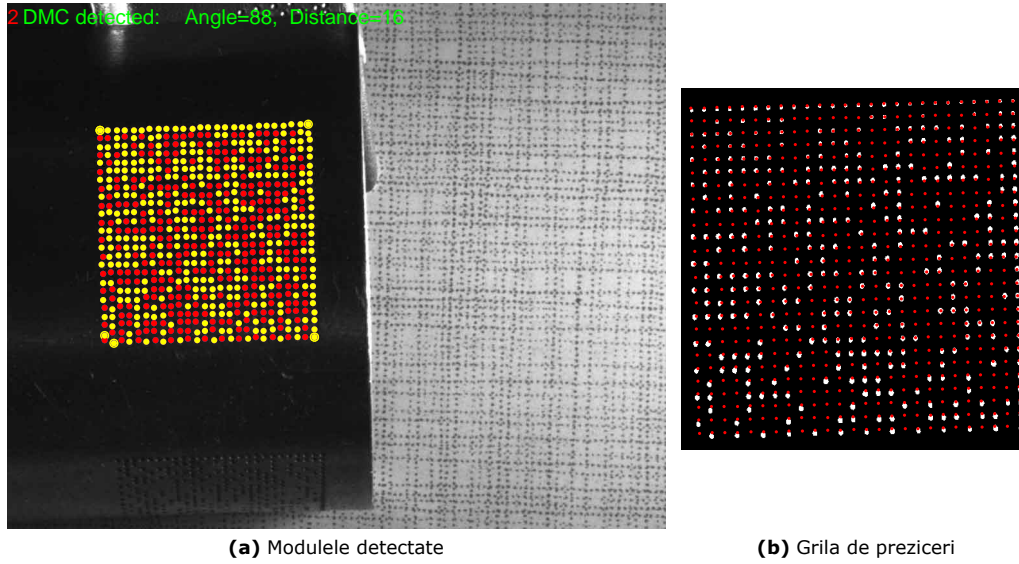


Figura 6.17: Ex.2 - Cod Data Matrix marcat pe suprafață cilindrică

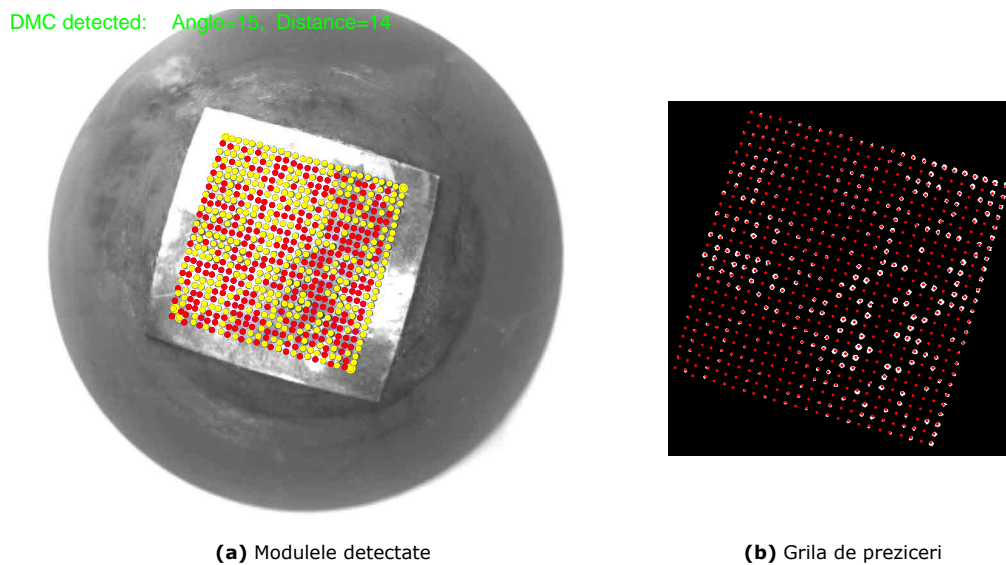


Figura 6.18: Ex.3 - Cod Data Matrix marcat pe suprafață sferică

Capitolul 7

Comparația metodelor de scanare

În acest capitol dorim să comparăm metodele de scanare a codului Data Matrix industrial, prezentate în Capitolul 6. Prima metodă este vectorială, bazată pe orientarea și distanța dintre module, iar cea de-a doua este bazată pe Matricea de transformare geometrică ce estimează pozițiile modulelor amprentei. Scopul acestui capitol este de a evidenția câștigul utilizării metodei de scanare, pe baza Matricei de transformare, implementată pentru suprafețele curbe (Capitolul 6).

7.1 Experiment

Pentru a putea compara cele două metode ar trebui analizate mai multe amprente Data Matrix orientate în unghiuri diferite. Propunem testarea celor două metode, atât pe materiale reale, cât și pe imagini generate de calculator. Cunoscând structura amprentei codului Data Matrix, în funcție de mărimea amprentei, de distanța dintre module și de unghiul acesteia, se poate genera o imagine sintetică cu cod Data Matrix. Metoda de construcție a amprentei sintetice este prezentată în Capitolul 5. Amprenta generată în Capitolul 5.1 este plană, dar dorim testarea amprentelor marcate pe materiale cilindrice și sferice. Pentru acest lucru se proiectează amprenta codului Data Matrix pe un cilindru sau pe o sferă [69]. Astfel, se pot testa simultan cele două metode propuse și să tragem concluziile corespunzătoare. În Fig. 7.1, este prezentată diagrama procesului de testare a celor două metode de scanare.

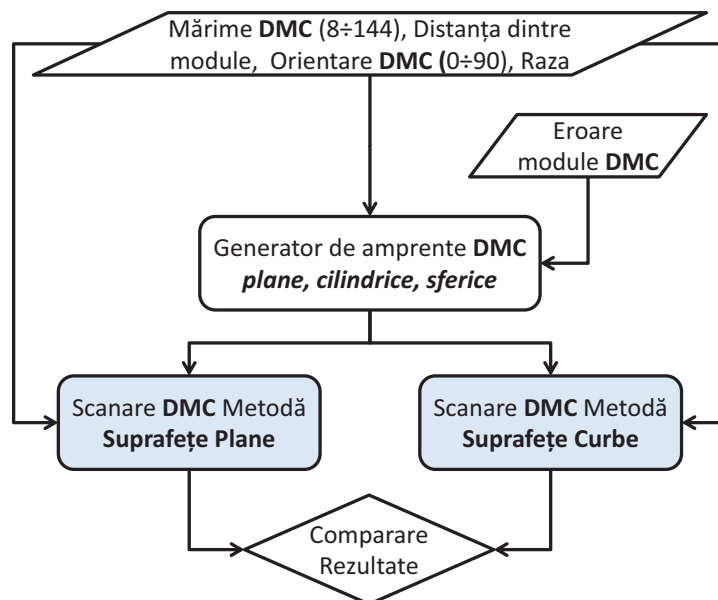


Figura 7.1: Descrierea experimentului

Pentru a înțelege mai bine experimentul ce se efectuează în acest capitol, prezentăm pe scurt cele două metode de scanare.

Scanarea suprafețelor plane

Pentru suprafețele plane, scanarea modulelor se realizează pe baza distanței dintre module și a unghiului orientării amprente codului Data Matrix. Utilizând o fereastră de scanare se analizează rândurile și coloanele codului Data Matrix, iar pe baza răspunsurilor se creează o matrice de coordonate ale modulelor marcate și nemarcate. Dacă în fereastră există un modul, atunci elementul din matrice este „1”. Dacă fereastra este goală, atunci elementul din matrice este „0”. În cazul în care este detectat un modul, centrul ferestrei este ajustat la centrul de greutate al modulului. Dimensiunea ferestrei este egală cu distanța dintre module. Această operație se repetă până când toate rândurile și coloanele sunt verificate.

Scanarea suprafețelor curbe

Pe baza colțurilor codului Data Matrix se creează o listă de puncte de corespondență între imagine și Matricea de transformare. Având acestea și cunoscând distanța dintre module, pe baza matricei de transformare se construiește o grilă cu presupusele module Data Matrix. Proiectând grila predicțiilor peste imagine, toate modulele pot fi localizate aproximativ. Există unele erori din cauza punctelor de corespondență inexacte folosite. Din acest motiv, folosind o fereastră de scanare se urmărește grila predicțiilor modulelor codului Data Matrix. Se verifică fiecare poziție și se creează o matrice cu coordonatele modulelor codului Data Matrix.

7.1.1 Experiment realizat pe imagini reale

Pentru a măsura performanțele celor două metode de scanare, ne propunem să calculăm abaterea medie pătratică dintre poziția modulelor de pe material și poziția modulelor detectate de cele două metode. Astfel, pentru compararea celor două metode de scanare pentru codurile marcate pe suprafețe plane, construim Tabelul 7.1, în care afișăm rezultatele experimentului.

Tabela 7.1: Eroarea medie pătratică în cazul celor două metode de scanare

Tip suprafață	Abatere - Scanare vect.	Abatere - Scanare matr.
Aluminiu - s. plană, Fig. 7.2	0,945	0,931
Fier - s. plană, Fig. 7.3	1,523	1,500
Aluminiu - s. cilindrică, Fig. 7.4	1,074	0,989
Aluminiu - s. cilindrică, Fig. 7.5	1,141	1,117
Cupru - s. cilindrică, Fig. 7.6	1,246	1,129
Cupru - s. sferică, Fig. 7.7	2,434	1,326

În Fig. 7.2 și Fig. 7.3 sunt prezentate două cazuri în care codul Data Matrix este marcat pe aluminiu lucios și pe fier vopsit. Cu roșu sunt reprezentate modulele detectate de cele două aplicații, iar cu albastru sunt reprezentate spațiile goale. Mai întâi testăm cele două metode de scanare pe suprafețe plane, după care le încercăm și pe suprafețe cilindrice și sferice. A se observa în cele două figuri, chiar dacă pe suprafața materialelor se prezintă zone de zgârieturi sau pete de rugină, ambele metode au detectat cu succes modulele codului Data Matrix. Prin rezultatele din Tabelul 7.1 se demonstrează funcționarea în mod egal a celor două metode de scanare în cazul codurilor Data Matrix marcate pe suprafețe plane din aluminiu, eroarea medie

pătratică fiind doar de 0,945, respectiv 0,931 pixeli. Pentru codurile marcate pe suprafețele metalice vopsite (Fig. 7.3), din tabelul rezultatelor se poate observa influența texturii materialului pentru cele două metode de scanare, abaterea medie fiind de 1,523 pixeli pentru metoda destinată suprafețelor plane, respectiv 1,500 pixeli pentru metoda destinată suprafețelor curbe.

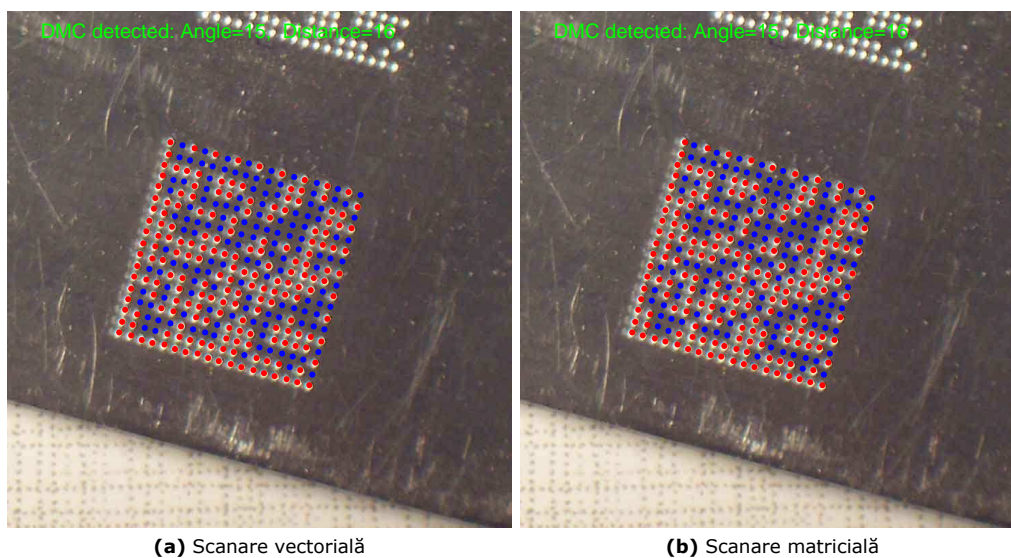


Figura 7.2: Aluminiu-suprafață plană, dimensiune amprentă $1,6 \times 1,6 \text{ cm}^2$

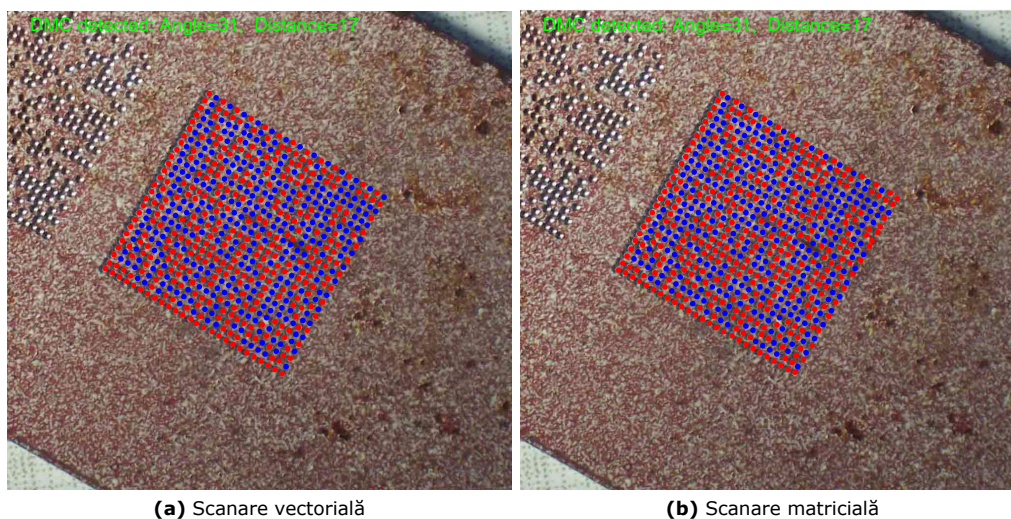


Figura 7.3: Fier-suprafață plană, dimensiune amprentă $1,8 \times 1,8 \text{ cm}^2$

În Fig. 7.4 codul Data Matrix este marcat pe un cilindru din aluminu de diametru 10 cm. A se observa în Fig. 7.4(a) că metoda de scanare vectorială este influențată

de curbura materialului. Curbura nu este foarte mare, dar suficientă cât să schimbe direcția de scanare a modulelor dintr-un rând. În Fig. 7.4(b) metoda de scanare pe baza Matricei de transformare a detectat pozițiile modulelor din cadrul amprentei codului Data Matrix. Acest lucru poate fi observat și în Tabelul 7.1, unde eroarea medie pătratică este mai mare în cazul metodei de scanare vectorială.

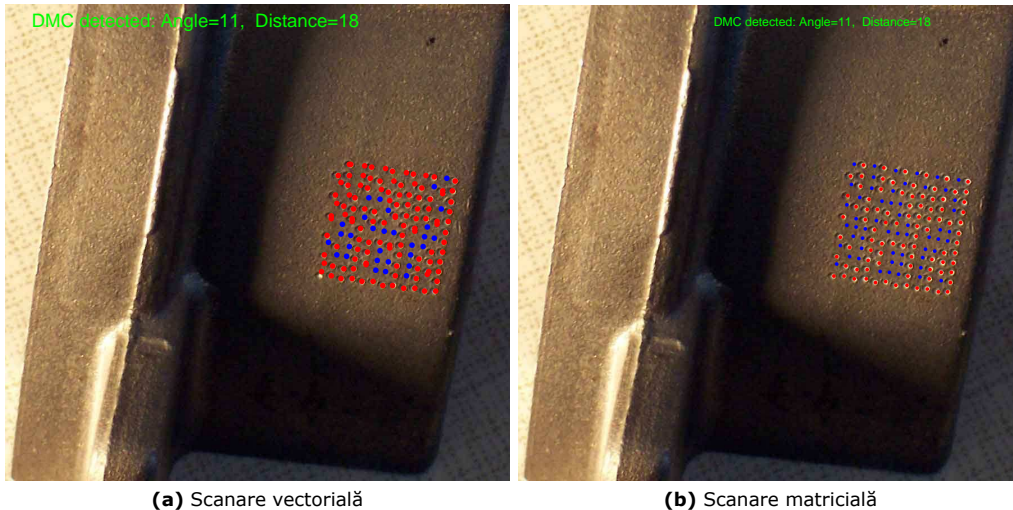


Figura 7.4: Cilindru-aluminiu, diametrul = 10 cm, dimensiune amprentă $0,8 \times 0,8 \text{ cm}^2$

În Fig. 7.5 se testează codul Data Matrix marcat pe cilindrul din aluminiu de diametru 5 cm, dimensiunea amprentei codului Data Matrix fiind de $2 \times 2 \text{ cm}^2$. Suprafața cilindrului nu este foarte curbată, iar scanarea codului Data Matrix putem spune că a fost efectuată cu succes în ambele cazuri. Dacă analizăm cu atenție Fig. 7.5(a), putem recunoaște deviația punctelor albastre ce reprezintă spațiile goale dintre modulele amprentei codului Data Matrix. Putem spune că și metoda destinată suprafețelor plane a detectat modulele codului Data Matrix, dar nu cu o precizie la fel de bună ca și cea a metodei destinată scanării amprentelor marcate pe suprafețe curbe, diferența de eroare fiind mică.

În Fig. 7.6 codul Data Matrix este marcat pe un cilindru de cupru de diametru 3,5 cm. În acest caz raza este mai mică decât în exemplul dat anterior. În cazul scanării cu metoda vectorială se poate observa o abatere a modulelor detectate față de poziția lor reală în imagine. În cazul metodei matriciale, modulele codului Data Matrix sunt detectate cu acuratețe.

În Fig. 7.7 testăm scanarea codului Data Matrix marcat pe o sferă din cupru de diametru 4,5 cm. În acest caz putem observa cu ușurință diferența dintre metodele de scanare. În cazul scanării amprentei codului Data Matrix cu metoda destinată suprafețelor plane, datorită curburii materialului, anumite module nu sunt detectate, iar poziția modulelor detectate de cele mai multe ori este inexactă. Metoda implementată pentru scanarea amprentelor Data Matrix marcate pe suprafețe curbe detectează poziția modulelor codului Data Matrix cu o eroare mai mică față de metoda destinată suprafețelor plane. Acest lucru reiese și din Tabelul 7.1, unde metoda de scanare destinată pentru suprafețele plane are o eroare mai mare de un pixel față de metoda destinată suprafețelor curbe.

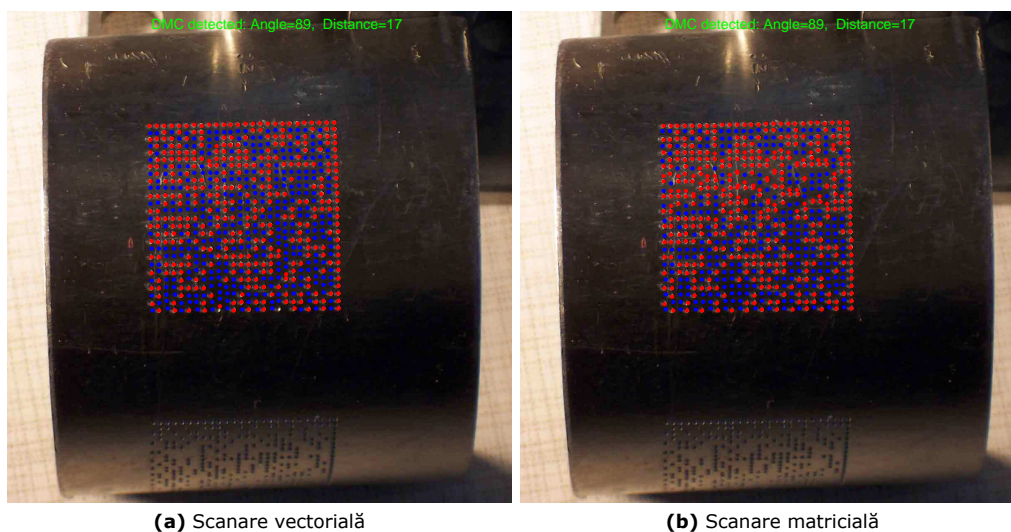


Figura 7.5: Cilindru-aluminiu, diametrul = 5 cm, dimensiune amprentă $2 \times 2 \text{cm}^2$

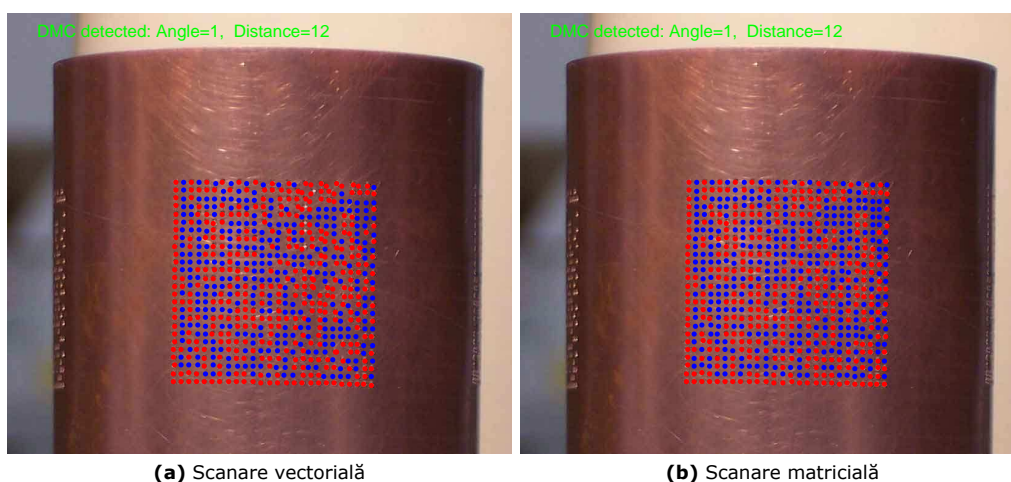


Figura 7.6: Cilindru-cupru, diametrul = 3,5 cm, dimensiune amprentă $1,8 \times 1,8 \text{cm}^2$

7.1.2 Experiment realizat pe imagini generate sintetic

Marcarea unui Cod Data Matrix pe o suprafață curbă de rază mică poate fi mai greu de realizat practic. De aceea, pentru a avea un rezultat mult mai precis asupra celor două metode de scanare, generăm o amprentă Data Matrix pe care o proiectăm peste o suprafață cilindrică și una sferică. Pornind de la generarea unei amprente plane se generează succesiv amprente cilindrice și sferice de raze diferite. Am considerat numărul de rânduri și de coloane de 26, iar distanța dintre module de 19 pixeli. În funcție de mărimea amprente în pixeli, generăm o suprafață geometrică curbă de rază variabilă, pornind de la $R = \infty$, la $R = n \cdot d$, unde $n =$ numărul de rânduri, iar $d =$ distanța dintre module. În Fig. 7.8 se testează metodele de scanare pentru o

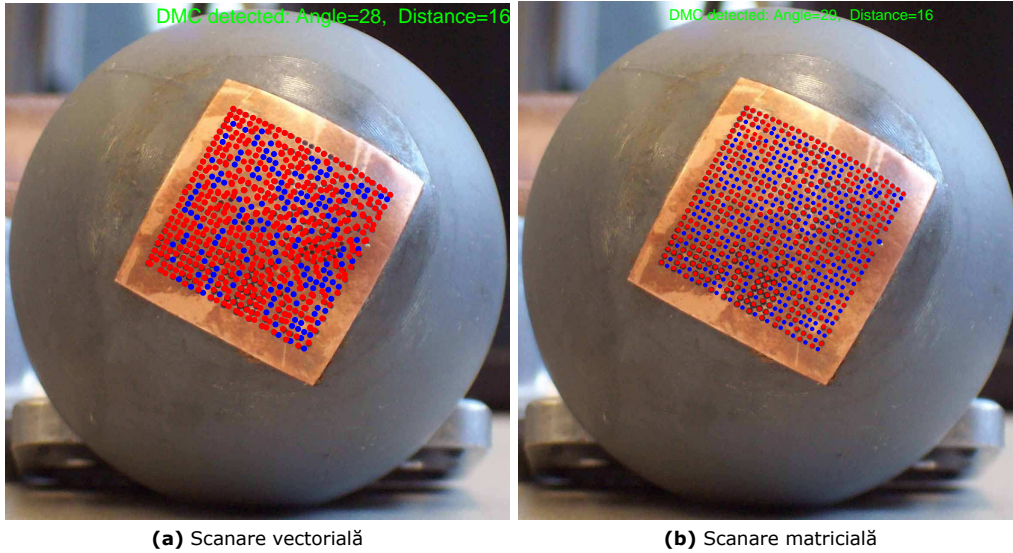


Figura 7.7: Sferă-cupru, diametrul = 4,5 cm, dimensiune amprentă $2 \times 2 \text{ cm}^2$

amprentă de formă cilindrică. În Fig. 7.8 (a), respectiv (b), sunt generate coduri Data Matrix pe suprafețe plane. În Tabelul 7.2 se pot observa rezultatele celor două metode de scanare. Dar cu cât suprafața devine mai curbă, pornind de la $R = 3 \cdot n \cdot d$ la $R = n \cdot d$, detecția modulelor cu metoda de scanare destinată suprafețelor plane nu mai este eficientă.

Tabela 7.2: Eroarea medie pătratică în cazul celor două metode de scanare, pentru amprentele DMC cilindrice (Fig. 7.8)

Curbură (raza)	Abatare - Scanare vect.	Abatare - Scanare matr.
$R = \infty$	0,090	0,850
$R = 3 \cdot n \cdot d$	0,863	0,859
$R = 2 \cdot n \cdot d$	1,917	0,911
$R = n \cdot d$	1,949	0,92

Se poate remarca din figuri că fereastra de scanare are o traiectorie dreaptă, iar în cazul când suprafața devine tot mai curbă, fereastra de scanare își păstrează direcția, dar părăsind rândul modulelor. Diferența dintre rezultatele celor două metode este afișată în Tabelul 7.2.

În cazul suprafețelor sferice, situația este aceeași, doar că amprenta codului Data Matrix are și rândurile și coloanele curbe. Succesiv sunt generate imagini ale codurilor Data Matrix proiectate pe suprafețe sferice, începând cu Fig. 7.9 (a) și (b) unde raza sferei este de $R = 4 \cdot n \cdot d$ Fig. 7.9 (g) și (h), raza sferei fiind de $R = n \cdot d$. Rezultatele acestor metode sunt scrise în Tabelul 7.9.

Se observă că în cazul suprafețelor foarte curbe, metoda de scanare destinată suprafețelor plane nu mai reușește să finalizeze procesul de scanare, scanarea fiind întreruptă. În Figurile 7.9 (e) și (g), punctele verzi reprezintă spațiile goale nesigure din amprentă. În Fig. 7.9 (e) eroarea medie pătratică este de 6,671 pixeli în cazul în

Tabela 7.3: Eroarea medie pătratică în cazul celor două metode de scanare, pentru amprentele DMC sferice (Fig. 7.9)

Curbură (raza)	Abatare - Scanare vect.	Abatare - Scanare matr.
$R = 4 \cdot n \cdot d$	0,711	0,690
$R = 3 \cdot n \cdot d$	0,540	0,528
$R = 2 \cdot n \cdot d$	6,671	0,462
$R = n \cdot d$	-	0,321

care amprenta este scanată cu metoda destinată suprafețelor plane, mult mai mare față de 0,462 pixeli, eroare măsurată la metoda de scanare destinată suprafețelor curbe. În Fig. 7.9 (g) unde raza este de $n \cdot d$, modulele detectate cu metoda vectorială sunt prea puține pentru a calcula eroarea. Dar totuși, și în acest caz, metoda destinată suprafețelor cilindrice, oferă un rezultat cu o eroare de doar 0,321 pixeli.

7.2 Rezultate

În acest capitol, am testat cele două metode de scanare pentru diferite tipuri de materiale și diferite suprafețe. Metoda de scanare vectorială, pe baza orientării amprentei și distanței dintre module, este eficientă pentru codurile marcate pe suprafețe plane, dar în cazul scanării codurilor marcate pe suprafețele curbe este ineficientă. De aceea, s-a dezvoltat metoda matricială care, pe baza unei predicții a poziției modulelor, scanează amprenta codului Data Matrix. Comparând cele două metode, din Tabelele 7.1 - 7.3 reiese că performanțele sunt mai bune în cazul metodei de scanare destinată suprafețelor cilindrice. În toate cazurile, eroarea fiind mai mică de un pixel pentru metoda scanării matriciale. Timpul de scanare este mai mic în cazul metodei implementată pentru scanarea suprafețelor plane. De aceea se folosesc ambele metode de scanare prin selectarea metodei de scanare optimă în funcție de amprenta codului Data Matrix.

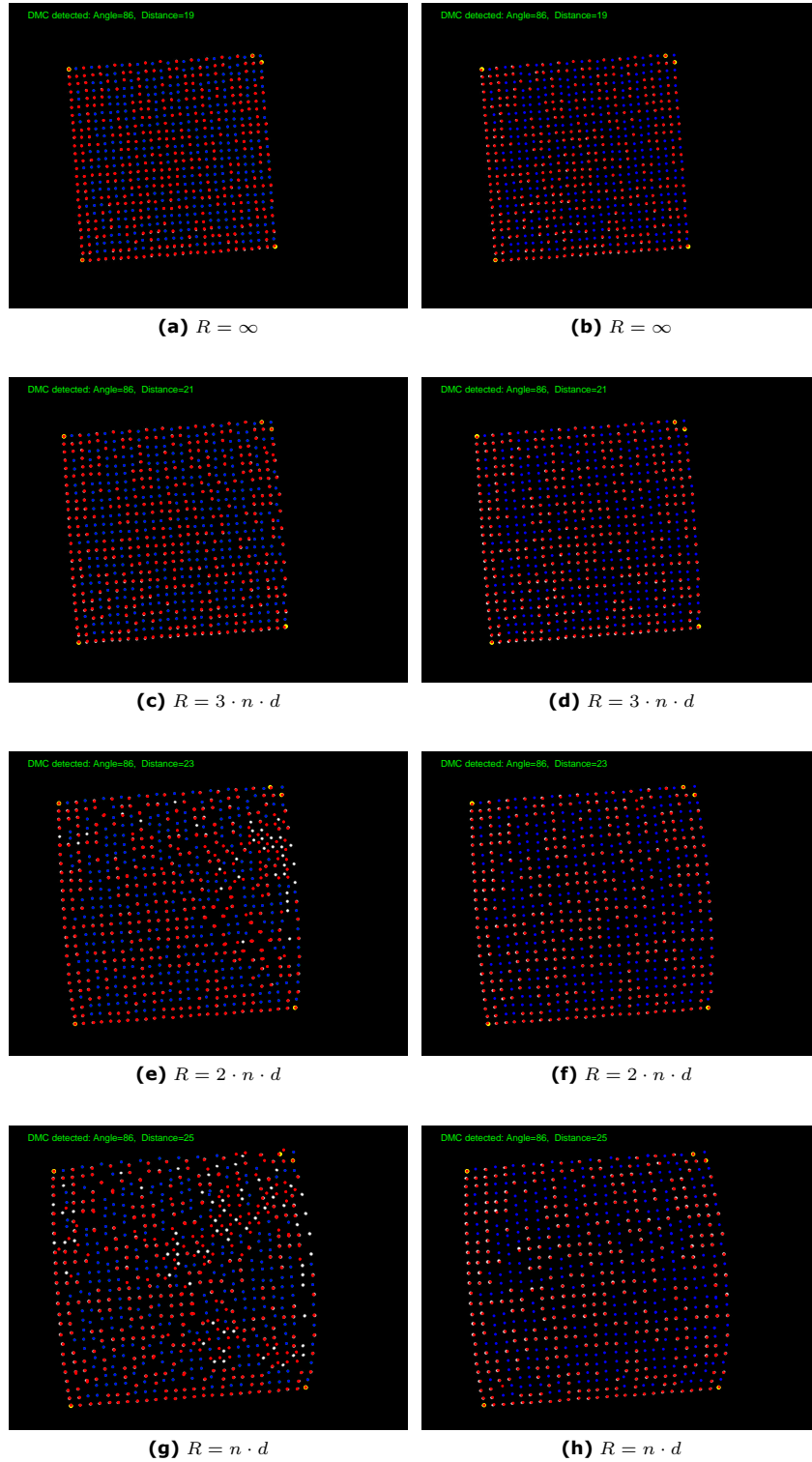


Figura 7.8: Amprentă DMC cilindrică, mărime = 26×26 , $d = 19px$, $\alpha = 86^\circ$. Partea stângă scanare vectorială, partea dreaptă scanare matricială

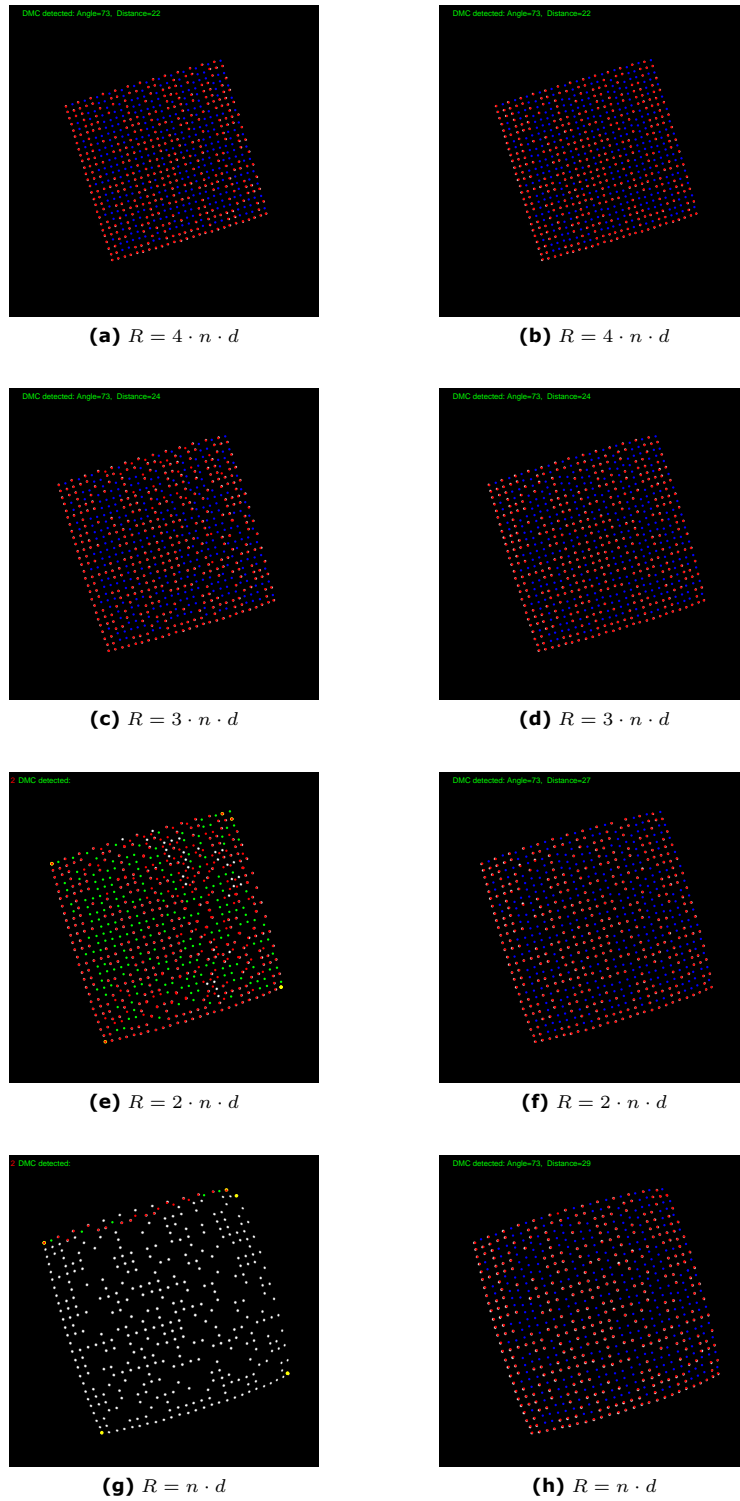


Figura 7.9: Amprentă DMC sferică, Mărime = 26×26 , $d = 22px$, $\alpha = 73^\circ$. Partea stângă scanare vectorială, partea dreaptă scanare matricială

Capitolul 8

Concluzii generale și contribuții

8.1 Concluzii

Teza de doctorat este dedicată unei probleme de mare interes în amprente digitale și anume achiziția codurilor Data Matrix în mediul industrial. Problema constă în localizarea corectă a amprentei și estimarea caracteristicilor ei, pentru a putea fi scanată. Metodele tradiționale de recunoaștere și scanare implementate pentru amprente de coduri Data Matrix standard sunt neutilizabile în cazul de față. Codurile Data Matrix standard sunt formate din celule pătratice având amprente de identificare formate din linii, pe când codurile Data Matrix industriale sunt formate în totalitate din puncte. Este mai ușoară recunoașterea unui set de linii, toate fiind orientate în aceeași direcție, decât o distribuție de puncte. Dacă se mai adaugă și factorii perturbatori produși de mediul industrial precum lumină slabă sau forma și structura materialului, atunci problema devine și mai complicată.

De aceea s-a propus să se facă cercetări în această direcție și să se dezvolte o aplicație de recunoaștere a codurilor destinată în special Codurilor Data Matrix industriale. Dacă până în prezent, metodele deja dezvoltate pentru codurile standard nu funcționează și pentru codurile industriale, asta nu înseamnă că nu sunt bune. Problema a fost privită altfel, dar cu o parte din dificultățile cu care ne confruntăm la achiziția codurilor industriale s-au confruntat și cei care au adus contribuții la achiziția codurilor standard. De aceea, mai întâi s-au implementat metode noi de recunoaștere și scanare pentru codurile industriale pe baza algoritmilor deja folosiți, cum ar fi: Binarizarea imaginii cu prag dinamic adaptiv, Morfologie matematică pentru detecția formei, Estimarea pe baza histogramei și a Transformării Hough pentru calcularea distanței dintre module, suprafața modulelor și orientarea amprentei.

Analizând aceste metode, am observat că zgomotul în imagine poate fi văzut ca și puncte din cod. Singura diferență între zgomot și cod este că modulele amprentei sunt așezate regulat, zgomotul fiind aleator. De aceea, am mers mai departe în căutarea de metode noi de estimare mult mai fiabile și mai precise. În acest caz, este nevoie de un estimator de densitate de puncte regulate. Analizând literatura de specialitate, am considerat că Algoritmii cu Translație la Medie ar putea satisface aceste cereri.

Încă de la prima implementare a acestuia, am observat diferența pozitivă a rezultatelor. Pentru a demonstra aceasta, am evaluat comparativ cu Metoda Transformării Hough performanțele estimatorului bazat pe Algoritmii cu Translație la Medie. Pe baza rezultatelor am demonstrat eficacitatea Algoritmilor cu Translație la Medie în estimarea caracteristicilor amprentei Codului Data Matrix industrial.

Pentru scanarea codului, inițial, am implementat metoda clasică vectorială pe baza orientării amprentei și a distanței dintre module. Această metodă este rapidă și funcționează bine pentru suprafețele curate, unde modulele sunt distribuite uniform. Însă, această metodă întâmpină dificultăți în cazul materialelor curbe. De aceea, pe baza Matricei de transformare, am implementat o nouă metodă de scanare care funcționează bine pentru toate tipurile de suprafețe. Pentru a evidenția avantajele acestei metode am realizat un studiu comparativ, analizând succesiv materiale reale cu suprafețe diferite: plane, cilindrice sau sferice. Pentru a avea un rezultat mult mai bine fundamentat, am comparat cele două metode pe amprente generate cu

ajutorul calculatorului. Acestea prezintă grade de deformare mult mai ridicate față de amprentele reale.

8.2 Contribuții

- Am implementat un algoritm semi-automat de recunoaștere a formei codului Data Matrix bazat pe morfologie matematică. Metoda propusă a fost publicată în [89]. Această metodă este îmbunătățită prin automatizare completă, astfel fiind capabilă să recunoască direct amprenta de identificare a codului Data Matrix, publicată în [51].
- Am implementat metoda Transformării Hough pentru estimarea orientării amprentei Codurilor Data Matrix industriale, prezentată în [90].
- Principala contribuție a tezei de doctorat este elaborarea și implementarea unei metode de achiziție și recunoaștere a codurilor Data Matrix industriale, bazată pe Algoritmul cu Translație la Medie, publicată în [90], [51]. Contribuțiile metodei se referă la o soluție de maximă verosimilitate, prin maximizarea densității de probabilitate în spațiul parametrilor. Algoritmul cu Translație la Medie folosit în localizarea maximelor densității de probabilitate a caracteristicilor amprentei (distanța dintre module, orientare, aria modulelor) este fundamentat pe estimarea neparametrică.
- Am realizat un studiu comparativ între Algoritmul cu Translație la Medie și algoritmul Transformării Hough pentru estimarea orientării amprentei codului Data Matrix. Acest studiu comparativ realizat și publicat în [90], a stabilit performanțele și acuratețea Algoritmului cu Translație la Medie. Experimentele realizate confirmă avantajele metodei bazată pe estimarea densității de probabilitate folosind funcții nucleu.
- Am implementat o metodă de scanare a codului Data Matrix marcat pe suprafețe plane, pe baza orientării amprentei și distanței dintre module, metodă publicată în [82]. Metoda fiind îmbunătățită pe baza Matricei de transformare geometrice, permite scanarea codurilor Data Matrix industriale ale căror imagini au defect de perspectivă sau sunt marcate pe suprafețe curbe. Rezultatele acestor îmbunătățiri sunt publicate în articolele [53], [54], [52].

Am implementat o aplicație în mediul de programare Matlab destinată achiziției și scanării codurilor Data Matrix industriale. Am avut în vedere următoarele aspecte:

- Selecția modului de lucru pentru recunoașterea codurilor, semi-automat sau automat. În modul semi-automat se poate configura aplicația cu date fizice despre amprenta codului și caracteristicile sistemului video folosit.
- Estimarea caracteristicilor amprentei codului pe baza Algoritmului cu Translație la Medie. Se estimează distanța dintre module, suprafața modulelor și orientarea amprentei Codului Data Matrix industrial.
- Clasificarea suprafeței pe care este marcat codul (plană, cilindrică sau sferică) și scanarea amprentei codului Data Matrix generând o matrice cu coordonatele modulelor ei.

Aplicația soft implementată, permite conectarea unei camere foto la calculator și afișarea informațiilor rezultate în urma recunoașterii și scanării codului Data Matrix industrial. Această aplicație este testată pe materiale reale de diferite dimensiuni și suprafețe, astfel fiind demonstrată aplicativitatea metodei propuse.

8.3 Direcții viitoare de cercetare

Rezultatele teoretice și experimentale ale cercetărilor dezvoltate în prezenta teză de doctorat deschid noi orizonturi de cercetare în recunoașterea codurilor Data Matrix industriale.

- Una din direcțiile viitoare de cercetare este optimizarea metodelor de recunoaștere și scanare, implementate în teză, cât și dezvoltarea de noi tehnologii cu posibilitatea de funcționare în timp real. Se dorește dezvoltarea unei aplicații hardware ce poate funcționa independent de calculator.
- O altă direcție de cercetare este studierea tehnicilor de marcare pentru codurile Data Matrix marcate pe suprafețe sferice. În acest caz, marcarea modulelor codului se face neregulat datorită suprafeței sferice a materialului. Cunoscând principiul de funcționare a acestor tehnici de marcare, se poate corecta această neregularitate a modulelor codului pe baza prelucrării imaginii achiziționate a codului Data Matrix industrial.
- Teza de doctorat a fost realizată în vederea colaborării cu un doctorand german al Universității de Științe Aplicate din Karlsruhe. Acest doctorand se ocupă de partea de decodare a codului Data Matrix industrial. Se dorește conectarea celor două aplicații: cea de Recunoaștere a codului și cea de Decodare, și implementarea unui sistem complex ce poate funcționa în buclă, prin reglarea automată a parametrilor de recunoaștere în funcție de deciziile de corecție luate de sistemul de decodare.

Bibliografie

- [1] R. Palmer, *The bar code book: A comprehensive guide to reading, printing, specifying, evaluating, and using bar code and other machine-readable symbols*. Trafford Publishing, 2007.
- [2] B. Raj, *Bar Codes*. Tata McGraw-Hill Publishing Company, 2001.
- [3] L. B.N, D. M.C., and I. Mang, "From Codabar to ISBT 128: Implementing Barcode Technology in Blood BankAutomation System," *27th Annual International Conference of the Engineering in Medicine and Biology Society*, pp. 542–545, 2006.
- [4] J. Wenbin, "A 2D Barcode-Based Mobile Payment System," *Third International Conference on Multimedia and Ubiquitous Engineering*, pp. 320–329, 2009.
- [5] Y.M. Li and L. Zeng, "Research and application of the EAN-13 barcode recognition on iphone," *2010 International Conference on Future Information Technology and Management Engineering (FITME)*, pp. 92–95, 2010.
- [6] Xianyong F., Fuli W., Bin L., Haifeng Z., and Peng W., "Automatic Recognition of Noisy Code-39 Barcode," *The 16th International Conference on Artificial Reality and Telexistence Workshops*, pp. 79–82, 2006.
- [7] The Institution of Engineering and Technology, "The 2D data matrix barcode," *Computing & Control Engineering Journal*, vol. 16, no. 6, p. 39, 2005.
- [8] H. Kato, K. Tan, and D. Chai, *Barcodes for Mobile Devices*. Cambridge University Press, 2010.
- [9] L. Xue, Y. Chao, L. Liu, and X.Zhang, "Information Hiding Algorithm for PDF417 Barcode," *ICNC '09. Fifth International Conference on Natural Computation*, pp. 578–581, 2009.
- [10] S. Cole, *The Aztec Code*. Bloomsbury Publishing PLC, 2007.
- [11] S. Lisa and G. Piersantelli, "Use of 2D Barcode to Access Multimedia Content and the Web from a Mobile Handset," *Global Telecommunications Conference, IEEE GLOBECOM*, pp. 1–3, 2008.
- [12] H. Kawakita, Y. Nishimoto, and T. Inoue, "Fast generation method of 2D code on DTV receivers," *IEEE International Conference on Consumer Electronics (ICCE)*, pp. 771–772, 2011.
- [13] T. Sarjakoski, J. Kovanen, R. M. M. nneberg, K. hko, J. nen, and L. T. Sarjakoski, "Data matrix technology for linking mobile maps in a web-based multi-channel service," *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*, pp. 1–5, 2010.
- [14] F. Xiaofei and J. Xingzhong, "A Blind Watermarking Method with Strong Robust Based on 2D-Barcode," *ITCS 2009. International Conference on Information Technology and Computer Science*, vol. 2, pp. 452–456, 2009.

- [15] Benhaim M, *An introduction and technical overview of the most advanced GS1 Application Identifiers compliant symbology*. GS1, 2010.
- [16] INTERNATIONAL STANDARD, "Information technology — International symbology specification — Data matrix," 2000.
- [17] Carl W. Gerst III, *New Standards Reliably Verify 2D Data Matrix Codes*. Cognex Corporation, 2009.
- [18] K. Changgeun, R. Sunwook, K. Juhee, and J. Yong, "Product Reed-Solomon Codes for Implementing NAND Flash Controller on FPGA Chip," *2010 Second International Conference on Computer Engineering and Applications (ICCEA)*, vol. 1, pp. 281–285, 2010.
- [19] National and International Standards, "ASSOCIATION FOR AUTOMATIC IDENTIFICATION AND MOBILITY AIM DPM," 2006.
- [20] I. C. Dita and M. Otesteanu, "Factors that Influence the Image Acquisition of Direct Marking Data Matrix Code," *17th Telecommunications forum TELFOR 2009*, pp. 929–926, 2009.
- [21] ISO/IEC 15415:2004, "Information technology – Automatic identification and data capture techniques – Bar code print quality test specification – Two-dimensional symbols," 2011.
- [22] T. M. Tsai, *Camera Calibration for a Manufacturing Inspection Workstation: NIST Interagency/Internal Report (NISTIR)*. National Institute of Standards, 2005.
- [23] I. C. Dita, "Image Acquisition of Direct Marking Data Matrix Code," *Doctor ETC 2009*, pp. 111–116, 2009.
- [24] H. Yang, A. C. Kot, and X. Jiang, "Knowledge guided adaptive binarization for 2D barcode images captured by mobile phones," *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 1046–1049, 2010.
- [25] H. Dong, T. Jianfu, Y. Zhaoxuan, P. Yanwei, and W. Meng, "2D barcode image binarization based on wavelet analysis and Otsu's method," *2010 International Conference on Computer Application and System Modeling (ICCASM)*, vol. 5, pp. 30–33, 2010.
- [26] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man and Cybernetics, DOI - 10.1109/TSMC.1979.4310076 (Systems, Man and Cybernetics, IEEE Transactions on)*, pp. 62–66, 1979.
- [27] Z. Yudong and W. Lenan, "Fast Document Image Binarization Based on an Improved Adaptive Otsu's Method and Destination Word Accumulation," *School of Information Science and Engineering, Southeast University, Nanjing China*, pp. 1886–1892, 2011.
- [28] D. Hu and H. X. Tan, "Radon transformation applied in two dimensional barcode image recognition," *Journal of Wuhan University*, vol. 5, pp. 584–588, 2005.
- [29] C. Lian and Du Sidan, "Rotation, scale and translation invariant image watermarking using Radon transform and Fourier transform," *Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, 2004.*, vol. 1, pp. 281–284 Vol.1, 2004.

- [30] C. Zhao and N. H. R. Yao, "The study of two dimensional barcode identification technology based on HOUGH transform," *Journal of Changchun Normal University*, vol. 4, pp. 94–98, 2007.
- [31] M. Nakanishi and T. Ogura, "Real-time line extraction using a highly parallel Hough transform board," *International Conference on Image Processing*, pp. 582–585, 1997.
- [32] W. Cai and Z. Liu, "Two dimensional barcode localization algorithm based on convex," *Journal of Zhejiang University*, vol. 4, pp. 669–672, 2008.
- [33] Q. Enzhuo and D. S. Lalush, "A Faster Ordered-Subset Convex Algorithm for Iterative Reconstruction," *IEEE Nuclear Science Symposium Conference Record*, vol. 6, pp. 3424–3427, 2006.
- [34] L. Ming and Z. Wei, "Visiting power laws in cyber-physical networking systems," *Mathematical Problems in Engineering*, 2012.
- [35] L. Ming, C. C., and C. SY, "Viewing sea level by a one-dimensional random function with long memory," *Mathematical Problems in Engineering*, 2011.
- [36] H. Qiang, C. Wen-Sheng, H. Xiao-Yan, and Z. Ying-Ying, "Data Matrix Code Location Based On Finder Pattern Detection And Bar Code Border Fitting," *Mathematical Problems in Engineering*, pp. 1–13, 2012.
- [37] M. Myllyla, J. R. Cavallaro, and M. Juntti, "Architecture Design and Implementation of the Metric First List Sphere Detector Algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, DOI - 10.1109/TVLSI.2010.2041800 (*Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*), vol. 19, pp. 895–899, 2011.
- [38] C. Chung-Hua, Y. De-Nian, and C. Ming-Syan, "Extracting Barcodes from a Camera-Shaken Image on Camera Phones," *IEEE International Conference on Multimedia and Expo*, pp. 2062–2065, 2007.
- [39] F. Miller, A. Vandome, and J. McBrewster, *Edge Detection*. Alphascript Publishing, 2010.
- [40] H. Huaqiao, X. Wenhuan, and H. Qiang, "A 2D Barcode Extraction Method Based on Texture Direction Analysis," *Fifth International Conference on Image and Graphics, 2009. ICIG '09*, pp. 759–762, 2009.
- [41] X. Wei and S. McCloskey, "2D Barcode localization and motion deblurring using a flutter shutter camera," *IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 159–165, 2011.
- [42] C. Yizong, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 790–799, 1995.
- [43] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, DOI - 10.1109/34.1000236 (*Pattern Analysis and Machine Intelligence, IEEE Transactions on*), vol. 24, pp. 603–619, 2002.

- [44] F.-B. Daniela, "Tehnici de aliniere a imaginilor utilizand estimare neparametrica de densitate de probabilitate: Teze de doctorat," Ph.D. dissertation, UPT, Timisoara, 2009.
- [45] J.-S. Pan, A. Abraham, and C.-C. Chang, *A Scanning Method for Dotted Data Matrix: Eighth International Conference on Intelligent Systems Design and Applications*. IEEE Computer Society, 2008.
- [46] K. V. Asari, S. Kumar, and D. Radhakrishnan, "A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation," *IEEE Transactions on Medical Imaging*, vol. 18, pp. 345–354, 1999.
- [47] T. Liu and A. Dong, "Precisely Correct Radial and Oblique Distortion of Camera Image," *Journal of Image and Graphics*, pp. 1935–1938, 2007.
- [48] P. Junhee, B. Seong-Chan, and L. Byung-Uk, "Lens distortion correction using ideal image coordinates," *IEEE Transactions on Consumer Electronics*, vol. 55, pp. 987–991, 2009.
- [49] L. Lei, H. Weiping, and Z. Wei, "Distortion Correction of Data Matrix Symbol Directly Marked on Cylinder Surface," *2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI)*, vol. 2, pp. 225–229, 2010.
- [50] I. C. Dita, M. Ottesteanu, and F. Quint, "Scanning Industrial Data Matrix Codes," *19th Telecommunications Forum (TELFOR)*, pp. 1211–1214, 2011.
- [51] I. C. Dita, V. Gui, M. Ottesteanu, and F. Quint, "Using Mean Shift Algorithm in the Recognition of Industrial Data Matrix Codes," *accepted for publishing at 11th WSEAS International Conference on SIGNAL PROCESSING (SIP '12)*, 2012.
- [52] I. C. Dita, M. Ottesteanu, F. Quint, and N. Annamdevula, "A Scanning Method for Industrial Data Matrix Codes marked on Spherical Surfaces," *accepted for publishing at 11th WSEAS International Conference on TELECOMMUNICATIONS and INFORMATICS (TELE - INFO 2012)*, 2012.
- [53] I. C. Dita, F. Quint, and M. Ottesteanu, "A Scanning Method for Industrial DMC Marked on Cylindrical Surfaces," *accepted for publishing at 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems*, 2012.
- [54] I. C. Dita, M. Ottesteanu, and F. Quint, "Scanning of Industrial Data Matrix Codes in non Orthogonal View Angle Cases," *accepted for publishing at 6th Euro American Conference on Telematics and Information Systems EATIS 2012*, 2012.
- [55] P. S. Swaroop, D. Patra, and N. P. Kumar, "Adaptive Thresholding Based Image Segmentation with Uneven Lighting Condition," *IEEE Region 10 and the Third international Conference on Industrial and Information Systems*, pp. 1–6, 2008.
- [56] R. Driggers, Ed., *Encyclopedia of Optical Engineering*. Optical Engineering Press, 2003.
- [57] T. Acharya and A. K. Ray, "Image Processing: Principles and Applications," *Electrical Communication*, vol. 15, no. 3, 2005.
- [58] D. M. Williams and M. C. Ebach, *The data matrix*. Publications Scientifiques du Muséum national d'Histoire naturelle, 2006, vol. 28.

- [59] O. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, DOI - 10.1109/34.1000236 (*Pattern Analysis and Machine Intelligence, IEEE Transactions on*), vol. 17, no. 3, pp. 312–315, 1995.
- [60] A. Z. Arifin and A. Asano, "Image segmentation by histogram thresholding using hierarchical cluster analysis," *Pattern Recognition Letters*, vol. 27, pp. 1515–1521, 2006.
- [61] V. Gui, D. Lacrama, and D. Pescaru, *Tehnici de estimare robusta in prelucrarea imaginilor*. Editura Politehnica Timisoara, 1999.
- [62] Z. Ye, Q. Hongsong, and W. Yanjie, "Adaptive Image Segmentation Based on Fast Thresholding and Image Merging," *16th International Conference on Artificial Reality and Telexistence Workshops, 2006. ICAT '06*, pp. 308–311, 2006.
- [63] V. Gui, D. Lacrama, and D. Pescaru, *Prelucrarea imaginilor*. Editura Politehnica Timisoara, 1999.
- [64] A. E. Savakis, "Adaptive document image thresholding using foreground and background clustering," *International Conference on Image Processing, 1998. ICIP 98. Proceedings*, vol. 3, pp. 785–789, 1998.
- [65] Y. Zhao and R. M. Haralick, "Binary shape recognition based on an automatic morphological shape decomposition," *International Conference on Acoustics, Speech, and Signal Processing, 1989. ICASSP-89*, pp. 1691–1694, 1989.
- [66] A. Bovik, Ed., *Handbook of Image and Video Processing*. Elsevier Academic Press, 2005.
- [67] C. Beumier, "Straight-Line Detection Using Moment of Inertia," *IEEE International Conference on Industrial Technology, 2006. ICIT 2006.*, pp. 1753–1756, 2006.
- [68] Z. Xinglong, Y. Xiaojun, Z. Junjun, Z. Ying, and Z. Jjping, "An Analytical Method of Determining the Plane Based on Dual Ellipse Equations," *International Conference on Artificial Intelligence and Computational Intelligence (AICI)*, vol. 1, pp. 468–473, 2010.
- [69] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [70] P. Meer, G. Medioni, and S. B. Kang, "ROBUST TECHNIQUES FOR COMPUTER VISION," *Emerging Topics in Computer Vision*, pp. 107–190, 2004.
- [71] R. O. Duda and P. E. Hart, *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Comm. ACM, January, 1972.
- [72] H. A. Cohen, "Image restoration via N-nearest neighbour classification," *International Conference on Image Processing, 1996. Proceedings*, vol. 1, pp. 1005–1008, 1996.
- [73] Medina-Carnicer, R. Garrido-Castro J.L., Collantes-Estevez E., and Martinez-Galisteo A., "Fast detection of marker pixels in video-based motion capture systems," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 432–439, 2009.

- [74] V. C. Raykar and D. R., "Fast optimal bandwidth selection for kernel density estimation," *Proceedings of the sixth SIAM International Conference on Data Mining*, pp. 524–528, 2006.
- [75] P. van Kerm, "Adaptive kernel density estimation," *Stata Journal*, vol. 3, no. 2, pp. 148–156(9), 2003.
- [76] M.C. Jones and D.A. Henderson, "Maximum likelihood kernel density estimation on the potential of convolution sieves," *Computational Statistics & Data Analysis*, pp. 3726–3733, 2009.
- [77] H. Rob L., Z. Xibin, and L. K. Maxwell, *Bandwidth Selection for Multivariate Kernel Density Estimation Using MCMC*. Econometric Society 2004 Australasian Meetings, 2004.
- [78] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–577, 2003.
- [79] M. Ezio and M. Eric, "Experiments with robust estimation techniques in real-time robot vision," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 223–228, 2006.
- [80] H. Guo, "A Simple Algorithm for Fitting a Gaussian Function [DSP Tips and Tricks]," *IEEE Signal Processing Magazine*, vol. 28, pp. 134–137, 2011.
- [81] Z. Yefeng, B. Georgescu, L. Haibin, S. K. Zhou, M. Scheuering, and D. Comaniciu, "Constrained marginal space learning for efficient 3D anatomical structure detection in medical images," *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, pp. 194–201, 2009.
- [82] I. C. Dita, M. Otesteanu, and F. Quint, "Data Matrix Code - A Reliable Optical Identification of Microelectronic Components," *2011 IEEE 17th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pp. 39–44, 2011.
- [83] N. Dahotre and S. Harimkar, *Laser fabrication and machining of materials*. Springer, 2008.
- [84] G. V. Puskorius and L. A. Feldkamp, "Camera calibration methodology based on a linear perspective transformation error model," *Proceedings IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1858–1860, 1988.
- [85] Z. Jie-xian, F. Xiang, L. Si-Xin, T. Bing, and Y. De-sheng, "An Improved Method For Camera Calibration," *2nd International Congress on Image and Signal Processing CISP '09*, pp. 1–5, 2009.
- [86] B. Zhang, S. Sun, J. Sun, Z. Chi, and C. Xi, "3D Reconstruction Method from Biplanar Radiography Using DLT Algorithm: Application to the Femur," *First International Conference on Pervasive Computing Signal Processing and Applications (PCSPA)*, pp. 251–254, 2010.
- [87] R. Gonzalez and R. Woods, *Digital image processing*. Pearson / Prentice Hall, 2008.

-
- [88] B. Zitova, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, pp. 977–1000, 2003.
- [89] I. C. Dita, M. Ottesteanu, and F. Quint, "A Robust Localization Method for Industrial Data Matrix Code," *TRANSACTIONS on ELECTRONICS and COMMUNICATIONS*, vol. Tom 56(70), Fascicola 2, pp. 12–17, 2011.
- [90] I. C. Dita, V. Gui, F. Quint, and M. Ottesteanu, "Comparison of Hough Transform and Mean Shift Algorithm for Estimation of the Orientation Angle of Industrial Data Matrix Codes," *accepted for publishing at ICDIP 2012 - International Conference on Digital Image Processing*, 2012.

