

## A Real Time Stereo Disparity Architecture for FPGA/ASIC Implementation

Victor Moisa, Georgica Iacobescu, Ioan Jivet<sup>1</sup>

**Abstract** – The paper presents an original architecture for real time stereo disparity computation. State of the art algorithms in recent reported implementations are analyzed in their adequacy to real time. For the selected SAD (sum of absolute differences) algorithm several improvements are proposed to enhance the disparity calculation. A fully pipelined design is disclosed and details of its implementation in a FPGA are presented. For the disparity computation module a weighting extension is proposed aiming to increase robustness to noise in the images. A post processing filter implementation is also described. Details of VHDL coding and synthesis are not among the objectives of the paper and only an outline is given on its feasibility.

**Keywords**— real time stereo disparity calculation, SAD algorithm (sum of absolute differences) FPGA/ASIC implementation .

### I. INTRODUCTION

The main objective of the work reported in the present paper was the development of an architecture for stereo disparity computation in hardware, capable to operate in real time.

The survey of recent literature on stereo disparity indicated that the algorithms used in the recent projects in stereo disparity calculation span a large variety [1], [2]. The simplest and most often used was found to be the sum of absolute differences (SAD) using a window of size 4 to 9, in spite of its major disadvantage - it fails in cases of different illumination as viewed from the two cameras [1]. The maximum disparity determines the number of parallel blocks to be used. The front end and back end pre-processing of rectification and respectively left/right check are used to increase the versatility of the method.

The performance of the representative solutions from literature has been compared with the target application specifications for best choice of algorithm to be used.

Typical recent implementation in ASIC, as reported in the literature, have been analysed against the resources possible in present day FPGA's [3].

The technological constraints of mainstream FPGA limit the frame rate to about 1000 fr/s. For an image of 512 x 512 at 1000 fr/s the serial pixel input rate of 2Gb/s is compatible with the high speed input available in FPGA's (Xilinx Rocket I/O MGT). The internal block RAM memory capacity is sufficient to accommodate two frames of image (4 Mb BRAM total is common to medium size FPGA).

The review of the architectures used for stereo disparity calculations in recent implementations revealed as typical ASIC recent solution the following:

- stereo processor, fabricated in a 0.18-um standard CMOS technology, 120 MHz clock frequency, achieving over 100 frames/s depth maps, 320 x 240 maximum image size and 16 - 64 disparity levels [5].

Recent research directions and results were also reviewed to determine calibrating assumptions for the objectives of the present work.

The target application considered was the automotive stereo based collision estimator [7].

The following general specifications have been used:

- Stereo camera systems (parallel optical axes)
- Integrated in the car electronics
- Detection and measurement of movement data
  - lateral coming objects
- Stereo Camera distance 500 mm
- Min resolution 800 x 600 pixels
- 8 bits / pixel gray-scale
- Frame-rate 100 Hz
- Detection range 3 .. 100m
- View angle 60 .. 100°

Given the target application specification an original architecture for FPGA/ASIC implementation was derived.

### II. STEREO ALGORITHMS AND DISPARITY MAPS COMPUTATION

Stereo images contain the information necessary to calculate a dense disparity map and redundancy of data is a well known issue [6].

There are also a number of known problems inherent to disparity calculation with no universal solution derived.

<sup>1</sup> Facultatea de Electronică și Telecomunicații, Departamentul Comunicații Bd. V. Pârvan Nr. 2, 300223 Timișoara, e-mail ioan.jivet@etc.upt.ro

Ideally the disparity map should be smooth over the interior of the exposed surface of the objects and with boundaries precisely delineated. Exposed surface elements of different objects should be detected as separately distinguishable regions.

Though obviously desirable, it is known that it is not possible to satisfy these requirements at the same time by a particular stereo algorithm.

The two key elements to devise improvements are the window size and averaging formula used in the disparity computation.

Several choices from literature reporting recent work indicate that window sizes producing a smooth disparity map tend to miss the details and those that can produce a detailed map tend to be noisy.

In area-based stereo methods, matching neighbouring pixel values by calculating correspondence parameters over a window - the selection of an appropriate window size is critical to achieving a smooth and detailed disparity map [4].

The best choice of window size depends on the local image properties like the amount of variation in texture. In general, a smaller window is desirable to avoid unwanted smoothing of the resulting disparity map. In areas of small amplitude of texture, however, a larger window is performing better due to the fact that the window contains intensity variation sufficient to achieve reliable correspondence.

In areas with low reflexive surface objects a large window is more effective due to the fact that the correspondence parameter values are significant to raise above the noise level.

In the present work more subtle problems like projective distortion have not been addressed – all objects were considered presenting an exposed surface of type fronto-parallel.

Methods of disparity computation that are extremely computationally intensive involving multiplication have not been considered. Algorithms requiring only addition and subtraction have been preferred for their efficiency in FPGA/ASIC type computational architectures implementations.

In order to determine object positions in 3D – the disparity map from stereo - computation for correspondence of pixels for every pair in the two image is verified. The correspondence can be expressed as a disparity vector i.e. if the corresponding pixels are at positions  $x_l$  or  $x_r$  in the left and right image respectively, then the disparity map  $D(x_l, x_r)$  is the difference of their original image co-ordinate. The output of a stereo algorithm is therefore the disparity map that matches every pixel from one image to only one in the other. The standard approach is to simplify the disparity vector to one dimension - horizontal. The two images are considered perfectly aligned in vertical dimension.

In order to provide results for longer term of the work reported in the paper the disparity correspondence data was retained for all candidate pixels.

The result is a 4D graph presented in figure 1 for only a column of points in the reference image. Each point

in centre of the graph is associated with a function displayed in frontal section in figure 1. MathLab was used as computational environment. The reference in the example located (at a position in the middle of the graph) given in the figure is at about the middle of the right side in the graph (indicated by an arrow).

The disparity is obtained at ‘best correspondence’ in this case the minimum of the value of the correspondence parameter – SAD value.

As can be seen in figure 1 a) - there are more than one minimum on a range of candidate pixels considered for correspondence. The ‘absolute minimum’ – ideally zero is the real coincidence calculated from the two matrices around the reference and candidate points.

In order to reduce the risk of selecting the wrong minimum the classic SAD algorithm was modified to favour one zone or another in the matrix in the computation of the correspondence parameter.

The pixels close to the centre were given lower weights than the ones at the sides by a Gaussian like distribution. This results in values of SAD - the correspondence parameter with less noise, is presented in figure 1 b).

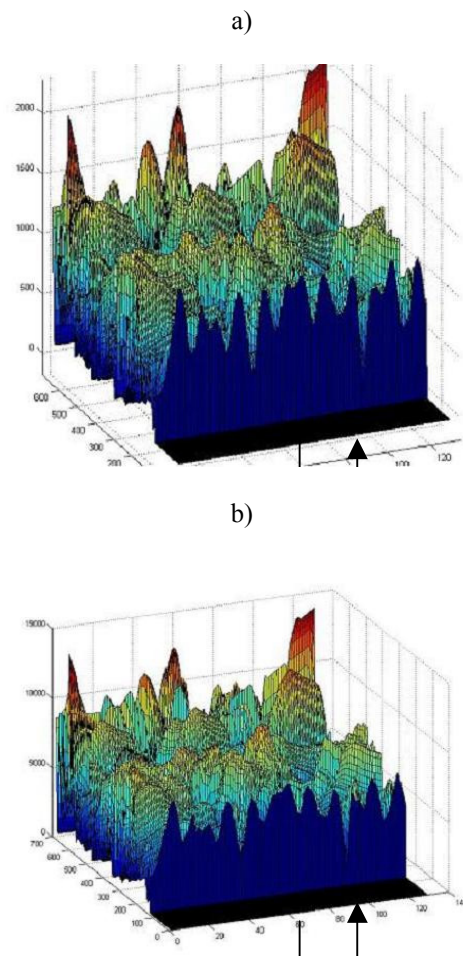


Fig 1. Gaussian kernel like weighting matrix for disparity calculation - results for a original ‘uniform’ SAD window a) ‘cup’ Gaussian like weighting matrix results b).

As can be seen in the figure 1. b) the value close to zero of the correct correspondence point is preserved and all other values are less dispersed in amplitude. The result is that none of them is low enough to compete with the correct minimum and thus the error possibility is reduced.

The results are similar to the case of using a larger matrix in the correspondence computation procedure. The modification as proposed, when used in the SAD valuable determination in a hardware implementation, is important since it reduces the circuit complexity. Weighting has been approximated with binary weights easy to implement in hardware by data shifts.

### III. PARALLEL PIPELINED HARDWARE IMPLEMENTATION OF A SAD ALGORITHM

A digital implementation of the modified SAD algorithm was implemented in hardware using a VHDL description of the circuit suitable for FPGA implementation.

In figure 2. is presented the general view of the architecture devised for the implementation of the modified SAD algorithm.

The left image is explored sequentially, one line at a time, pixel by pixel while the right image is explored in parallel on all the possible candidates points for the accepted range of disparity on the current line.

At each pixel step all windows over right image candidates compute a SAD value. A cascade of compare and select circuits determines the disparity value (relative index of minimum) and passes it to the output disparity map result.

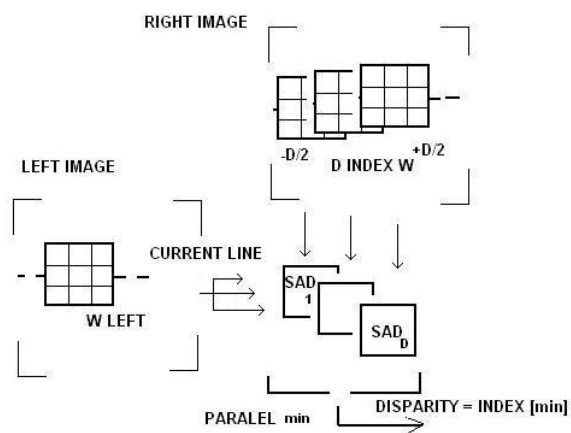


Fig 2. Architecture for the SAD (sum of absolute differences) multiple parallel windows implementation in a FPGA.

The architecture as described is fully pipelined and offers the highest speed in excess of 400 fps for image resolution of 512x512 when implemented in a common FPGA.

In order to have all the data necessary for all the matrices used in the computation the last five lines of both images are stored in line long (512 pixels) FIFO shift registers.

If some of the operations like the minimum calculation is done sequentially the amount of resources necessary is reduced at the cost of lower processing speed in terms of frames per second achievable.

One of the known problems of 3D from stereo is the sensitivity to noise in the areas with small amplitude texture. In order to improve the performance in this respect an improvement in the disparity determination was devised and experimented.

It is well known that the correspondence of areas using a matrix of pixels around the candidate one improves noise immunity when compared to the single pixel vs pixel correspondence.

The matrix SAD algorithm is still very 'local' in the sense that one pixel away from the correct correspondence position the SAD value changes abruptly even though the two matrixes contain almost all the right corresponding data.

A variation of the SAD computation algorithm was implemented as presented in figure 4. The correspondence of two areas of the two images is verified for partial superposition as well.

A circular permutation of one of the matrix by columns are compared with the reference matrix in parallel. Partial superposition will thus be detected and the SAD value at correspondence will drop gradually. This effect will be found to improve the error margin in the disparity minimum SAD determination.

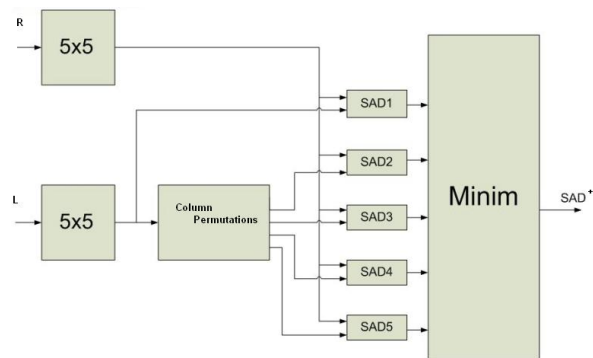


Fig 4. Improved SAD window implementation – partial overlap 'roll in' principle.

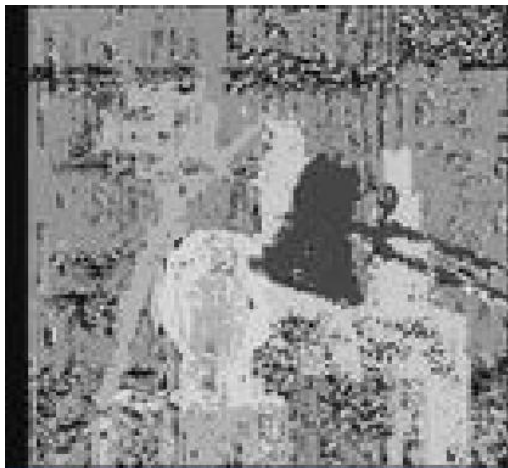
In order to improve further on noise in the disparity map a disparity filtering was experimented. A number of filtering alternatives have been studied using MathLab computational environment. In figure 5 a median filter results is presented.

The disparity map filtering is dependent on the subsequent processing of the disparity map as required by the practical application – object localization and tracking. Of equal importance is the implementation in the same FPGA/ASIC with minimal resources.

The output of the disparity map computation module was concatenated in the same FPGA code with additional three storage FIFO shift registers.

It was found that the FPGA extra resource necessary for filter implementation adds only a very small percentage to the overall design.

During the implementation phase of the project VHDL representation of the circuit was used for simulation and then synthesis.



a)



b)

Fig 4. Sample disparity map result using a) simple SAD algorithm , b) smoothed map when using a median filter b).

The Xilinx ISE 10.1 development tool was used to test the resource requirements for the design.

The high demand for logic resources becomes very important when a fully pipeline parallel implementation is synthesised as shown in an example in figure 5. Optimisation of code is absolute necessary in practice.

Aspects of the coding and implementation details in

sum_code Project Status		
Project File:	sum_code.isc	Current State:
Module Name:	sum_code	• Errors:
Target Device:	xc4vfx20-12ff672	• Warnin
Device Utilization Summary (estimated values)		
Logic Utilization	Used	Available
Number of Slices	10001	8544
Number of Slice Flip Flops	768	17088
Number of 4 input LUTs	18170	17088

Fig 5. Example of resource ‘explosion’ for a fully pipelined multiple parallel SAD implementation in a FPGA.

FPGA are not among the objectives of the present paper.

#### IV. CONCLUSIONS

The paper presents an architecture for real time stereo disparity computation targeting as application collision estimation for automotive. The SAD (sum of absolute differences) algorithm was selected for implementation and improvements are discussed for the disparity calculation immunity to false minima. The design of a fully pipelined architecture is presented and its implementation in a FPGA outlined. For the disparity computation module a proposed window weight extension is shown to increase robustness. A post processing filter implementation is also described. The VHDL code was synthesised and used for determination of the demand on resources. It was estimated that the resource count is very high in the fully pipelined architecture. Implementation aspects are outlined only, resource optimisation and implementation performance are the objectives of future work.

#### V. REFERENCES

- [1] Ambrosch, K., Humenberger, M., Kubinger, W., Steininger, A. Hardware implementation of an SAD based stereo vision algorithm, IEEE Conference on Computer Vision and Pattern Recognition, 2007 pp: 1 -6.
- [2] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV, 2002.
- [3] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. CVPR 2005, PAMI 30(2):328-341, 2008
- [4] Z. Wang and Z. Zheng. A region based stereo matching algorithm using cooperative optimization. CVPR 2008.
- [5] Q. Yang, C. Engels, and A. Akbarzadeh. Near real-time stereo for weakly-textured scenes. BMVC.
- [6] D. Scharstein, Middlebury Stereo Vision Page, [www.middlebury.edu/stereo](http://www.middlebury.edu/stereo)
- [7] S. Alvarez et al, Vehicle and Pedestrian Detection in eSafety Applications, Proc Word Congres on Eng. and Comp. Science 2009