

# A Fuzzy Logic Approach for Stereo Matching Suited for Real-Time Processing

M. Perez-Patricio

Tuxtla Gutierrez Institute of Technology  
Carr. panam. km. 1080  
Tuxtla Gtz, Mexico

A. Aguilar-Gonzalez

Tuxtla Gutierrez Institute of Technology  
Carr. panam. km. 1080  
Tuxtla Gtz, Mexico

J.L. Camas-Anzueto

Tuxtla Gutierrez Institute of Technology  
Carr. panam. km. 1080  
Tuxtla Gtz, Mexico

M. Arias-Estrada

National Institute of Astrophysics, Optics and Electronics  
Luis Enrique Erro 1, Santa Mara Tonatzintla  
Puebla, Mexico

## ABSTRACT

In this paper, a novel method that uses both area and feature based information as similarity measures for stereo matching is proposed. Area-based information is suited for non-homogeneous regions while feature information helps in homogeneous areas. In order to define a conjugate pair, a fuzzy logic approach that combines the similarity information is used. The proposed method preserves discontinuities while reducing matching errors in homogeneous regions. This proposal is suited for real-time processing using dedicated hardware. We demonstrate and discuss performance using synthetic stereo pairs.

## Keywords:

fuzzy logic; dense stereo vision; real-time processing; dedicated hardware; feature-based

## 1. INTRODUCTION

Stereo algorithms recover depth information from a scene by combining two images of that scene. Images are taken at the same time from slightly different viewpoints. In order to retrieve the depth information, primitives that correspond to the same point in the scene should be identified in both images. The correspondence problem consists in defining conjugate pairs of pixels, one from each image, that correspond to the same point of the scene. Because of the different viewpoints, coordinates of each pixel in a conjugated pair may be different. This difference is called the disparity. In order to reduce the complexity of the correspondence problem, images are rectified, i.e. the epipolar lines lie on horizontal lines at the same vertical position in both images. The depth of each primitive can be computed using the disparity value and a standard triangulation technique.

Correspondence algorithms can be grouped as feature-based or area-based. In feature-based algorithms, a pre-processing step yields the edges or corners in both images and features in both images are matched. A sparse depth map is obtained while disparity of pixels between edges are computed by interpolation.

In area-based algorithms, a rectangular neighborhood centered on a reference pixel in one of the images is compared to a similar neighborhood of every pixel in the same raster line of the other image. The neighborhoods can be compared using a correlation-based measure such as the Sum of Absolute Differences (SAD). A correlation coefficient is determined for each candidate pixel and the shift that minimizes the correlation coefficient is retained as the disparity. These methods yield a dense depth map, but they are very time consuming. Using these methods, selecting the size of the correlation window is a difficult task. With a large window, used to reveal a maxima of correlation in non-textured areas, edges are blurred and small details or small objects are removed from the disparity map. On the other hand, with a small window, the correlation coefficient is very sensitive to noise.

Several adaptive methods have been proposed to improve results in both depth discontinuities and homogeneous areas. Kanade and Okutomi [1] changed neighborhood size and shape iteratively according to the local variation of the intensity and current depth estimates. However, the algorithm was computationally expensive and sensible to the initial depth estimates [2]. Veksler [7] changed neighborhood size and shape by optimization over a large class of compact windows via minimum ratio cycle. Lotti and Giraudon [3] used edges in the reference image to determine the size of a rectangular window. Yang [8] proposed a method where a tree structure is used and the similarity between two pixels is computed using the distance on the tree. Min et al. [9] proposed the cost aggregation process from a perspective of a histogram, reducing the complexity of this process. In an article proposed by Chatterji [10], a measure called the fuzzy compactness is used to determine whether or not the intensity variation in the window is enough to split the window. In a paper by Zitnick [11], an iterative algorithm that uses the uniqueness and smooth assumptions and where the occlusions are explicitly detected is proposed. Due to its high algorithmic complexity, the processing time of these algorithms is high.

To simplify the adaptive methods and reduce time processing, efficient multiple windows methods have been proposed. Roberto et al. and Intille and Bobick [12, 4], computed correlation coefficients on nine windows, and the one yielding the lowest value was retained.

Hirschmüller [5] used a central window surrounded by several support windows. The correlation coefficients of the best support windows, i.e. the lowest values, were added to the coefficient computed using the central window. The reduced number of windows used in these methods cannot cover the whole range of different sizes and shapes required in all situations.

Using graphics processing units (GPUs), several algorithms can process stereo images at a real-time rate. The GPU is a massively parallel computing device. For example, Wang implemented [13] an algorithm using adaptive cost aggregation and dynamic programming. The matching cost is aggregated based on both color and geometric proximity. Their implementation improves the speed by one or two orders of magnitude compared to the CPU implementation. Humenberger et al. [14] implemented a simplified Census Transform on embedded systems. In [15] the multi-window algorithm proposed in [5] has been implemented using a massively parallel SIMD architecture. Yang and Pollefeys [16] implemented a multi resolution scheme and a six shapes support windows method on a graphics card that achieved up to 289 million disparity evaluations per second. Chang et al. [6] implemented a 4x5 jigsaw matching template on a DSP architecture for efficient use of the memory. Ambrosch and Kubinger [17] implemented a modified version of the Census Transform that expanded the Census Transform to be processed over the intensity image and the absolute value of the gradient in  $x$  and  $y$  direction. In [18], a two step method was proposed where an initial disparity was computed and only reliable disparity values were aggregated to the cost function for the second step. Jin et al. [19] proposed a cost aggregation method that computed ground control points (GCP) in order to refine disparity values using the GCPs, while [20] adapted the support area using assumptions of similarity and proximity values. Using the graphics hardware, real-time processing can be achieved, but the implementation must be efficiently adapted and modifications to that structure will be difficult to carry out.

Another choice for hardware implementation of stereovision algorithms are the FPGA-based architectures. These devices are popular because of their inherent parallelism and high computational power. Ttofis et al. [21] first applied an edge detector in order to reduce the space search, and next used a correlation measure only on the edges detected. Area-based algorithms are the most adapted for implementation on FPGA devices because they have a regular structure with fixed execution time. Several systems that use these methods have been described in the literature [22, 23]. These algorithms use a fixed size window with Census Transform or SAD as similarity measures where the problem of determining the adequate window size is still a challenge.

We are interested in stereo matching algorithms that can be implemented on dedicated hardware for real-time processing. In this paper, a novel method is described that uses area and feature information in order to compute similarity between candidates for matching. Reliability of each candidate pixel is evaluated using a fuzzy logic system and the winner-take-all strategy is used to determine disparity. At this time, only three measures are used as similarity criteria, but the algorithm can be efficiently modified to increment the number of measures used. Results on synthetic images show that this algorithm preserves edges while reducing errors in homogeneous areas. Our proposal can be efficiently implemented on dedicated hardware for real-time processing.

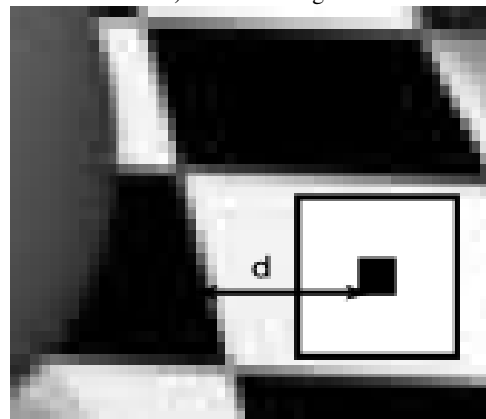
The rest of this paper is organized as follows : section 2 presents our algorithm. Experimental results with five synthetic stereo pairs and a comparison with similar algorithms is reported in section 3. Finally, section 4 concludes this paper.

## 2. THE PROPOSED METHOD

In order to explain the main idea behind this proposal, the image showed in the figure 1a is used. This image shows several objects at different depths, slanted areas and large homogeneous regions. In figure 1b, a zoom of a small region of the figure 1a is presented. This figure shows a pixel for matching and their support area, which is placed in an homogeneous area. The correlation function presents several minimals and the selection of a candidate is difficult. It is well known that the use of only correlation value in the matching procedure is not adequate in homogeneous areas. Nevertheless, the distance  $d$  of this pixel versus the left nearest edge must remain equal in the other image as observed by Yang [?]. This observation is valid except in presence of occlusion areas and for slanted regions. So, we propose the use of several similarity measures simultaneously in the matching procedure, ones that can help for homogeneous areas and other ones suited for non-homogeneous areas.



a) Corridor image



b) a small region

Fig. 1. Corridor image

Our proposal is depicted in figure 2 where several similarity measures are computed for the pixel for matching ( $I_l(x, y)$ ) and a candidate ( $I_r(x, y, s)$ ). All the similarity values are introduced in a fuzzy logic system that computes the reliability of the matching

$(R(x, y, s))$  for each disparity value ( $s$ ). The  $argmax$  of each reliability value  $R(x, y, s)$  is retained as the disparity. We can also observe that each data computed on each phase of the proposed algorithm is not dependant of another one, and they can be computed in parallel using dedicated hardware.

For demonstration purposes, only three similarity measures are used. Nevertheless, the number of similarity measures used can be efficiently modified to consider other ones. Each similarity measure used in this proposal is described as follows:

## 2.1 The area-based information

One of the similarity measures used is area-based information. It can be computed using SAD or Census Transform. For demonstration purposes, area-based information is computed using SAD which is computed using equation 1.

$$C_l(x, y, s) = \sum_{i=-w_x}^{i=w_x} \sum_{j=-w_y}^{j=w_y} |I_l(x+i, y+j) - I_r(x+s+i, y+j)| \quad (1)$$

where  $I_l(x+i, y+j)$  and  $I_r(x+i+s, y+j)$  are the gray levels of the pixels within the neighborhood in both images, called the left and right images respectively.  $(2 \times w + 1)^2$  is the window size,  $s$  is the shift of the neighborhood in the right image and the maximal shift of the correlation window in the right image is  $s_m$ . Correlation values are computed using small window sizes that preserve discontinuities.

## 2.2 The feature-based information

The feature-based information used is the distance to the left and right nearest edges. They are detected in both images using a Sobel filter (figure 3) using only the vertical component of the Sobel filter ( $G_y$ ).



Fig. 3. Sobel filter  $G_y$

The euclidean distance between each pixel with  $I(x, y)$  coordinates and the nearest left  $d_1$  and right  $d_2$  edges for  $s$  values of disparity of images of resolution  $X \times Y$  are computed as follows:

$$g_1 = \begin{cases} 1, & y > 1 \\ 0, & otherwise \end{cases} \quad (2)$$

$$k_1(x, y, s) = |I_\epsilon(x, y, s) - I_\epsilon(x, y - g_1 + s)| \quad (3)$$

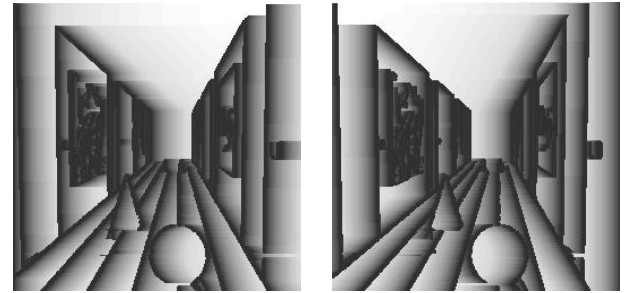
$$d_1(x, y, s) = \begin{cases} l = 0, & k_1(x, y, s) < \beta \\ l = l + 1, & k_1(x, y, s) > \beta \end{cases} \quad (4)$$

$$g_2 = \begin{cases} 1, & y < Y \\ 0, & otherwise \end{cases} \quad (5)$$

$$k_2(x, y, s) = |I_\epsilon(x, y, s) - I_\epsilon(x, y - g_2 + s)| \quad (6)$$

$$d_2(x, y, s) = \begin{cases} l = 0, & k_2(x, y, s) < \beta \\ l = l + 1, & k_2(x, y, s) > \beta \end{cases} \quad (7)$$

where  $\epsilon$  takes values of  $l$  or  $r$  for the left or right images, and  $\beta$  is a threshold value used to determine a pixel as a border. Fig. 4 shows the distance of each pixel to the nearest edge. Black gray levels represent shortest distances while white gray levels represent longest distances.



a) Distance to left edge b) Distance to right edge

Fig. 4. Images of distance to nearest edges

In order to determine similarity between distances of a pixel in the left image  $I_l$  and a pixel in the right image  $I_r$ , equation 8 can be used:

$$D_k(x, y, s) = |I_l(d_k(x, y, 1)) - I_r(d_k(x, y, s))| \quad (8)$$

where  $k$  is  $[1, 2]$ ,  $d_k$  is determined by eq.4 and eq.7

## 2.3 The fuzzy logic system

Correlation values  $C_l(x, y, s)$  and distance to the nearest left and right edges  $D_1, D_2$  are the inputs for a fuzzy inference system that determine similarity between candidates. For each one of these inputs, the three membership degrees in the corresponding fuzzy sets *good*, *medium* and *bad* are computed. This is realized by the membership functions  $\mu_{good}$ ,  $\mu_{medium}$  and  $\mu_{bad}$ :

$$\mu_{good}(x) = \begin{cases} 0, & \gamma \leq x \\ \frac{x-\gamma}{\alpha-\gamma}, & \alpha \leq x \leq \gamma \\ 1, & x \leq \alpha \end{cases} \quad (9)$$

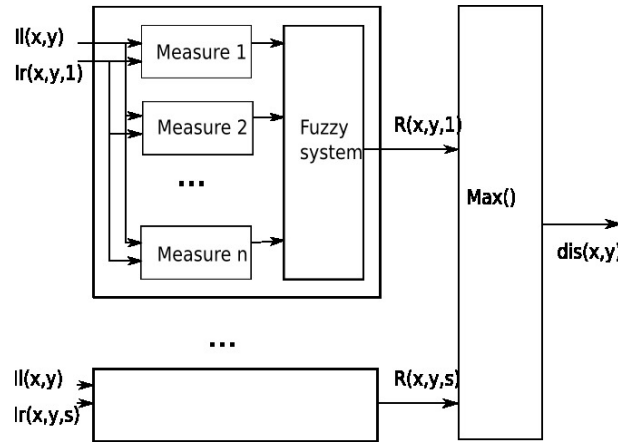


Fig. 2. The proposed algorithm

$$\mu_{medium}(x) = \begin{cases} \frac{x-\gamma}{\phi-\gamma}, & \gamma \leq x \leq \phi \\ \frac{x-\gamma}{\alpha-\gamma}, & \alpha \leq x \leq \gamma \\ 0, & otherwise \end{cases} \quad (10)$$

$$\mu_{bad}(x) = \begin{cases} 0, & x \leq \gamma \\ \frac{x-\gamma}{\phi-\gamma}, & \gamma \leq x \leq \phi \\ 1, & \phi \leq x \end{cases} \quad (11)$$

where  $\alpha, \phi$  and  $\gamma$  are empirically defined by 0.0, 0.5 and 1.0 respectively.

In this algorithm greatest fuzzy value determines the best candidate. For this purpose, output fuzzy sets *good*, *medium* and *bad* are proposed. Membership degrees are defined by the membership functions  $\psi_{good}$ ,  $\psi_{medium}$  and  $\psi_{bad}$ :

$$\psi_{good}(x) = \begin{cases} 0, & \gamma \leq x \\ \frac{x-\gamma}{\alpha-\gamma}, & \alpha \leq x \leq \gamma \\ 1, & x \leq \alpha \end{cases} \quad (12)$$

$$\psi_{medium}(x) = \begin{cases} \frac{x-\rho}{\sigma-\rho}, & \rho \leq x \leq \sigma \\ \frac{x-\rho}{\sigma-\rho}, & \sigma \leq x \leq \rho \\ 0, & otherwise \end{cases} \quad (13)$$

$$\psi_{bad}(x) = \begin{cases} 0, & x \leq \gamma \\ \frac{x-\gamma}{\phi-\gamma}, & \gamma \leq x \leq \phi \\ 1, & \phi \leq x \end{cases} \quad (14)$$

where  $\alpha, \phi, \gamma, \rho$  and  $\sigma$  are empirically defined by 0.0, 0.5, 1.0, 0.4 and 0.6 respectively.

Rules in the form of IF-THEN and connective AND defined by the Zadeh operator are used. Rules of the fuzzy system are determined as showed in 1

Fuzzy inference is determined by the Mamdani operator  $RM$  as:

$$RM = \min(\mu_n, \mu_w(x)), \forall x \quad (15)$$

where  $\mu_w$ , is the membership function of the output set  $w$  and  $\mu_n$  is the rule antecedent value for the n-sima rule. Singleton inputs are used and fuzzy inference is simplified as follows:

Table 1. Rules of the fuzzy system

Correlation	$D_1$	$D_2$	output
good	good	none	good
good	medium	none	good
good	bad	none	medium
medium	good	none	medium
medium	medium	none	bad
medium	bad	none	medium
bad	good	none	medium
bad	medium	none	medium
bad	bad	none	bad
good	none	good	good
good	none	medium	medium
good	none	bad	medium
medium	none	good	bad
medium	none	medium	medium
medium	none	bad	medium
bad	none	good	medium
bad	none	medium	bad
bad	none	bad	bad

$$RM = \mu_n \quad (16)$$

Rule aggregation is realized using the *sum* operator as follows:

$$\mu_\phi(x) = \mu_1 + \mu_2 + \dots + \mu_n, \forall x \quad (17)$$

where  $\mu_\phi(x)$  is the membership function of the output and  $\mu_1, \mu_2, \dots, \mu_n$  are  $RM$  values for n rules of the system.

Finally the bisector of area eq.18 is used for the defuzzification process.

$$R(x, y, z) = \int_\alpha^{z_0} \mu_\phi(x) dx = \int_{z_0}^\beta \mu_\phi(x) dx \quad (18)$$

where  $\mu_\phi(x)$  is the membership function of the output,  $\alpha = \min\{x|x \in X\}$ ,  $\beta = \max\{x|x \in X\}$ ,  $X$  is the output set and  $z_0$  divides the area into two regions with the same area.

The maximum of the values of the output determines disparity value according to equation 19

$$disparity(x, y) = argmax_z R(x, y, x) \quad (19)$$

The output of the fuzzy system is used as a confidence value in order to determine the reliability of the matching procedure and for subpixel matching. A parabolic equation is evaluated with the output of the fuzzy system, and the minimum value of the equation determines the disparity with subpixel accuracy.

### 3. EXPERIMENTAL EVALUATIONS

Several stereo pairs have been processed to demonstrate the effectiveness of our algorithm. A left-right validation has been applied to the disparity maps as a post-processing step in order to reduce errors near discontinuities. Disparity maps are presented as gray level images, using a linearly stretched gray level range to improve readability : black for the minimum disparity, white for the maximum one.

For the first experiment, we used the ground truth of the Corridor image showed in figure 5a. In this image there are several objects with different depths, homogeneous areas, and small objects. The difficulty is to detect small objects and preserve discontinuities, such as the ball in front, and to have robust results in homogeneous areas simultaneously. Figure 5b shows the disparity image obtained with the algorithm that uses a fixed size window and SAD as similarity measure, while figure 5c shows results of our proposal. Both use a small window of 3x3 pixels.

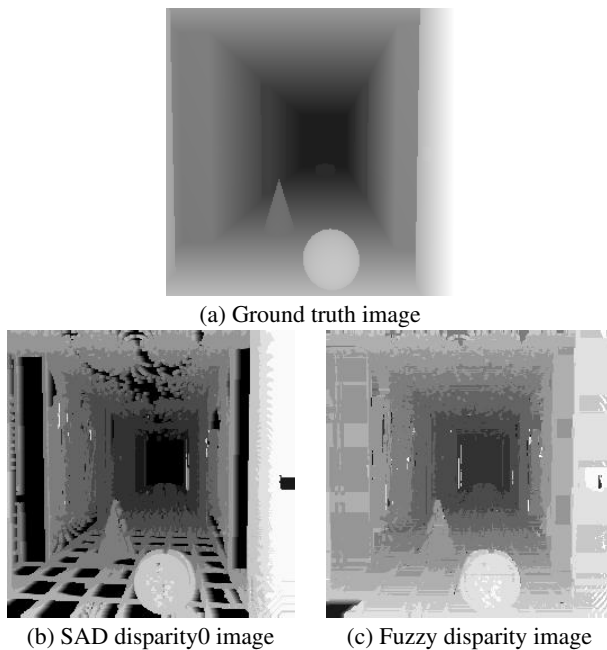


Fig. 5. Comparison of disparity maps

As a small window is used, discontinuities are well preserved in both algorithms. Nevertheless, the SAD algorithm yields incorrect disparities in homogeneous areas— areas that appear black in the images, but should appear white. On the other hand, disparity values obtained with our proposal are more accurate and contain fewer errors in these areas. We can conclude that the use of distance of the nearest edge helps in homogeneous areas while correlation values are used in non-homogeneous areas. Nevertheless, several mismatches have occurred. For example, in figure (c) the white line is an error; because the edge is misread it appears as white when it should appear dark. A detailed analysis of the computed data

showed that errors occur for photometric distortion and edges that are not well detected. Using one edge is not sufficient for consistent results, it is much better to use both the left and right nearest edges. Figure 6 shows other test images and their corresponding disparity maps computed using our proposed algorithm and a window size of 11x11 pixels. These images were taken from the Middlebury dataset. In the third column, the images show pixels with mismatches in black and pixels correctly matched in white.

The Tsukuba image has small objects, homogeneous areas, and several repetitive patterns, while the other ones present large homogeneous areas and slanted objects. Only the Venus, Cones, and Teddy images have subpixel disparities information. For our proposal, subpixel accuracy was achieved using the computed fuzzy value.

We can observe that errors in homogeneous areas are reduced, while a more accurate result is achieved near discontinuities. We can also observe that the fuzzy value can be efficiently used for obtaining subpixel accuracy results. Using images in the third column, it can be observed that a lot of mismatches are present on the left or right borders where there is not border information. In the Venus image, errors can be observed in the bottom-middle of the image caused by incorrect border information. Errors are also detected near discontinuities because this algorithm does not include an explicit discontinuity correction method.

For quantitative analysis, we present table 2 where each image of the dataset has been analyzed using several window sizes and for  $\beta = 20$ . Disparity maps have been compared using the method proposed in [?], in which the percentage of pixels with a disparity error greater than one is computed. Three percentages are computed, one for all non-occluded pixels, one for all pixels which includes the half-occluded regions, and one for non-occluded pixels near depth discontinuities.

Table 2. Performance under several windows sizes

image	ws	nonocc	all	disc
tsukuba	3x3	5.32	7.31	19.5
	5x5	5.17	7.15	19.0
	7x7	5.03	7.03	18.9
	9x9	4.96	6.99	19.5
	11x11	4.97	6.97	20.3
venus	3x3	7.89	8.87	32.6
	5x5	6.60	7.61	32.3
	7x7	5.99	6.99	32.7
	9x9	5.67	6.69	33.5
	11x11	5.43	6.40	33.6
teddy	3x3	22.3	29.3	35.5
	5x5	21.4	28.6	35.1
	7x7	20.8	28.0	34.3
	9x9	20.4	27.6	35.6
	11x11	20.3	27.5	38.3
cones	3x3	17.6	24.8	29.9
	5x5	16.3	23.5	28.6
	7x7	15.7	23.0	28.3
	9x9	15.6	23.0	28.8
	11x11	15.2	22.6	29.0

It can be observed that the percentage computed in the *all* column reduces with the greater window sizes. Detailed analysis of data shows that for several pixels there is not enough information using small windows and distance to nearest edges information and larger windows are required in this areas. On the other hand, in

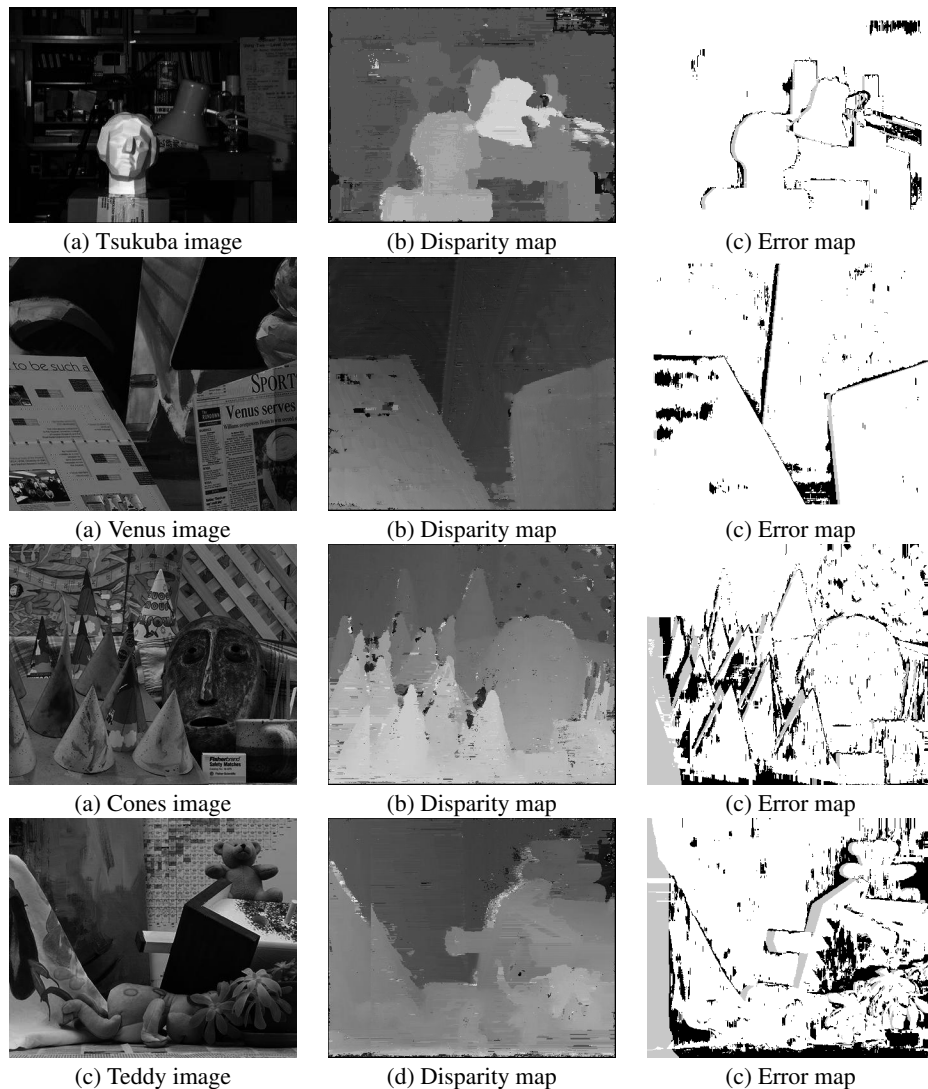


Fig. 6. Comparison of disparity maps using test images

the *disc* column, error grows while window size also grows. So, performance of this proposal is similar to SAD algorithm, but the error rate is smaller. Nevertheless, the difference between errors obtained with largest and smallest windows are very similar (less than 1 percent of difference) in the Tsukuba image, and this proposal can be efficiently used with small windows. The importance of this remark is that the use of smaller windows reduces time processing in a CPU implementation and reduces hardware resources and memory requirements in a FPGA implementation. An additional post-processing step to left-right validation can be used in order to reduce errors using the fuzzy value as a confidence measure.

As previously mentioned, results depend also on border detection. In order to determine an adequate  $\beta$  value, we evaluate performance of the algorithm for different values of  $\beta$  and for  $w = 2$ . Results are shown in figure 7.

We can observe that for values of  $\beta$  smaller than 20 and greater than 40, performance of the algorithm are worse, and better results can be obtained for  $\beta$  values between 20 and 40. We can conclude that

a small  $\beta$  value produces a lot of edges that can produce errors of multiple similar edges. On the other hand, larger values of  $\beta$  reduce the number of edges, which are necessary in homogeneous areas.

In order to compare our proposal with other similar real-time stereo algorithms, we present table 3, where the Tsukuba image has been analyzed. In this table, the column *Platform* describes the target hardware architecture, while the *Algorithm* column shows the type of algorithm used and the *Error<sub>rate</sub>* column describes the percentage of error obtained in the *all* column of the Middlebury dataset.

Results obtained show that our proposal is only outperformed by algorithms that use adaptive cost aggregation algorithms that require a more complex implementation. Adaptive cost-aggregation algorithms perform better near discontinuities because they only use pixels with the same disparity in the matching procedure, while our proposal includes pixels with different disparity values in the support windows that introduce errors. Nevertheless, the proposed algorithm outperforms algorithms that use a fixed-window size. This

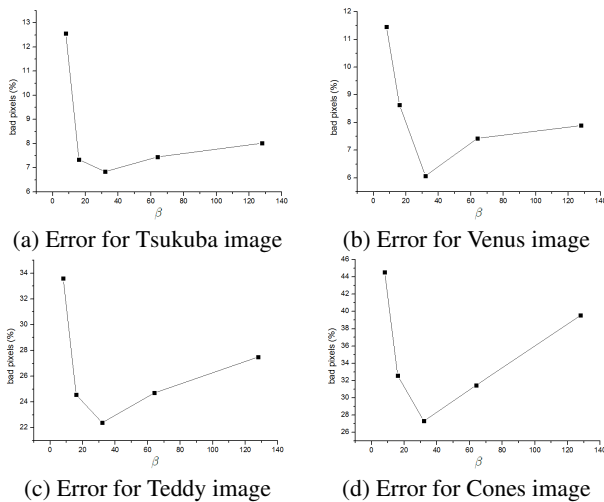
Fig. 7. Effect of  $\beta$  values

Table 3. Comparison with other real-time algorithms

Author	Platform	Algorithm	Error
Jin et al. [19]	FPGA	Cost-aggregation	2.17
Ding et al. [20]	FPGA	Cost-aggregation	5.65
our proposal	FPGA	Fixed-window	6.97
Yang et al. [16]	GPU	Cost-aggregation	7.07
Jin et al. [23]	FPGA	Fixed-window	11.5
Ttofis et al. [21]	FPGA	Fixed-window	10.4
Chang et al. [6]	DSP	Fixed-window	21.7
Hosseini et al. [15]	SIMD	Fixed-window	23.6
Ambrosch et al. [17]	FPGA	Fixed-window	23.7

performance is achieved by the reduction of the error rate in homogeneous areas where the distance to the nearest edges helps significantly.

#### 4. CONCLUSION AND OUTLOOKS

We have presented a new fuzzy logic algorithm for stereo matching suited for real-time processing using dedicated hardware. The algorithm uses local and global information in order to reduce errors in homogeneous areas and to preserve edges near discontinuities. At the moment, we only use three similarity measures, but the process can be adapted to increase the number of similarity measures used. We will analyze other similarity criteria in order to reduce error-rate when dealing with similarity of edges. An empirical value for determining a pixel as edge has been defined in order to improve results, but it's necessary to reduce the dependency of this value using other strategies of edge detection. We also hope to study the implementation of the algorithm using specialized hardware.

#### 5. REFERENCES

- [1] T. Kanade and M. Okutomi (1991) A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment. Proceedings of the 1991 IEEE International Conference on Robotics and Automation. Sacramento, CA, USA
- [2] S. Scherer and W. Andexer and A. Pinz (1998) Robust adaptive window matching by homogeneity constraint and integration of descriptions. Proceedings of the 14th International Conference on Pattern Recognition. Brisbane, Australia
- [3] J. Lotti and G. Giraudon (1994) Correlation algorithm with adaptive window for aerial image in stereo vision. Proceedings of the Image and Signal Processing for Remote Sensing. Rome, Italy
- [4] S. Intille and A. Bobick (1994) Disparity-Space Images and Large Occlusion Stereo. Proceedings of the European Conference on Computer Vision
- [5] H. Hirschmuller (2001) Improvements in real-time correlation-based stereo vision. Proceedings of IEEE workshop on Stereo and Multi-Baseline Vision. Kauai, Hawaii
- [6] N. Chang and L. Ting and T. Tsung and T. Yu (2007) Real-Time DSP Implementation on Local Stereo Matching. Proceedings of the International Conference on Multimedia and Expo. Hefei, China
- [7] O. Veksler (2002) Stereo Matching by Compact Windows via Minimum Ratio Cycle. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol:24 1654–1660
- [8] Q. Yang (2014) Stereo Matching using tree filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence. DOI 10.1109/TPAMI.2014.2353642
- [9] D. Min and J. Lu and M. Do (2013) Joint Histogram-Based Cost Aggregation for Stereo Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol: 35 2539-2545
- [10] G. Chatterji (2004) Fuzzy Compactness Based Adaptive Window Approach for Image Matching in Stereo Vision. Lecture Notes in Computer Sciences. Vol: 3316 935-940
- [11] C. Zitnick and T. Kanade (2000) A cooperative algorithm for stereo matching and occlusion detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol: 22 1-10
- [12] A. Fusiello, V. Roberto and E. Trucco (2000) Symmetric stereo with multiple windowing. International Journal of Pattern Recognition and Artificial Intelligence. Vol: 8 1053-1066
- [13] L. Wang and R. Yang and M. Gong and M. Liao (2014) Real-time stereo using approximated joint bilateral filtering and dynamic programming. Journal of Real-Time Image Processing. Vol: 9 447-461
- [14] M. Humenberger and C. Zinner and M. Weber and W. Kubinger and M. Vincze (2010) A fast stereo matching algorithm suitable for embedded real-time systems. Computer vision and Image Understanding. Vol: 114 1180-1202
- [15] F. Hosseini and A. Fijany and S. Safari and J. Fontaine (2013) Fast implementation of dense stereo vision algorithms on a highly parallel SIMD architecture. Journal of Real-Time Image Processing. Vol: 8 421-435
- [16] R. Yanga and M. Pollefeys (2005) A versatile stereo implementation on commodity graphics hardware. Real-Time Image. Vol: 11 7-18
- [17] K. Ambrosch and W. Kubinger (2010) Accurate hardware-based stereo vision. Computer vision and Image Understanding. Vol: 114 1303-1316
- [18] X. Sun and X. Mei and S. Jiao and M. Zhou and Z. Liu and H. Wang (2014) Real-time local stereo via edge-aware disparity propagation. Pattern Recognition Letters. Vol: 49 201-206
- [19] M. Jin and T. Maruyama (2014) Fast and Accurate Stereo Vision System on FPGA. ACM Transactions on Reconfigurable Technology and Systems. Vol: 7 1-24
- [20] J. Ding and J. Liu and W. Zhou and H. Yu and Y. Wang and X. Gong (2011) Real-time stereo vision system using adaptive weight cost aggregation approach. EURASIP Journal on Image and Video Processing. Vol: 20 1-19

- [21] C. Ttofis and S. Hadjitheophanous and A. Georghiades and T. Theocharides (2013) Edge-Directed Hardware Architecture for Real-Time Disparity Map Computation. *IEEE Transactions on Computers*. Vol: 62 690-704
- [22] X. Zhang and Z. Chen (2013) SAD-Based Stereo Vision Machine on a System-on-Programmable-Chip (SoPC). *Sensors*. Vol: 13 3014-3027
- [23] S. Jin and J. Cho and X. D. Pham and K. M. Lee and S.-K. Park and M. Kim and J. W. Jeon (2010) FPGA Design and Implementation of a Real-Time Stereo Vision System. *IEEE Transactions on Systems and Circuits for Video Technology*. Vol: 20 15-26