Features stereo matching based on fuzzy logic

Grzegorz Nieradka, Bohdan Butkiewicz
Warsaw University of Technology
00-665 Warsaw, Nowowiejska 15/19 str, Poland
Email: G.Nieradka@ise.pw.edu.pl, B.Butkiewicz@ise.pw.edu.pl

Abstract— *This paper presents an entire scheme for the estimation of a sparse disparity map from stereo pair images. In contrary to a dense disparity map, for which disparity values are calculated for each pixel of image, the sparse disparity map is determined only for some distinguished set of pixels from an image. The pixels belong to this set are called features. Therefore, in the first stage the algorithm for determining sparse disparity map has to be able to detect and specify some pixels as the feature pixels. In this article a method for specifying features is introduced. The core of the presented method relies on a fuzzy edges detector. The algorithm of fuzzy edges detection, as well as a manner enabling determination of the features' set, are introduced. The disparity is calculated at the fuzzy domain based on a similarity measure which is the correlation of fuzzy sets. The results of the obtained disparity maps for some benchmark stereo pairs, and the comparison with the well known Marr-Poggio-Grimson algorithm designed for sparse disparity map estimation are presented.*

Keywords— feature points, stereo matching, disparity map, edges detection, fuzzy logic

1 Introduction

Stereo vision is a vivid researched technique of machine vision area which allows to obtain a three dimensional image of a scene observed. Its importance arises from its main characteristics, hence, it is a passive method and the equipment which allows to use it is rather cheap. The estimation of disparity map is one of the central problems of stereo vision application.

In spite of thirty years of research there is no good enough method which always gives a satisfactory solution. The stereo vision and especially the stereo matching research could be divided into two main streams [1, 2]. One of them are algorithms which try to match all pixels from images. As a result of this kind of algorithms the so called dense disparity map is obtained. [3]. The "dense" means that the disparity values are estimated for each pixel of the images. These algorithms, however suffer from the calculation being time consuming and the matching in uniform regions of images being almost impossible. But doubtless the advantage of this group of methods is better model reconstruction possibility. This paper does not consider the methods which produce the dense disparity map.

An alternative approach towards finding a solution to the stereo vision problem is a feature matching strategy. Although, the obtained disparity map has established values in much less amount of pixels, in contrary to the former group of methods, the result can be used in many real tasks as navigation of autonomous robots, manipulation of industrial robots or view morphing. Moreover, the disparity values could be interpolated in the areas where the matching is not made. The disadvantage of this approach is an insufficient amount of disparity values for visually good reconstruction of observed scene. On the other hand, the methods based on features reduce the searching space from full image pixels set to the pixels which were recognized and assigned as the features. It attracts reduction of calculation time and in a natural way, it causes eliminating the ambiguity that may occur inside the uniform regions of images, because in these regions there are usually no feature points.

Each clearly distinct pixel from image can be selected as the features considered for matching task. However, the most popular features taken under consideration are corners [4, 5] and edges [6]. The edges matching techniques are more popular for the reason that the disparity map estimated with dependence only on corners is too sparse. Matching the edges of the images gives a lot of more dense disparity map therefore matching edges is considered in this paper.

In another group of feature based approaches image segmentation is done as the first step. Later, they try to match the segmented regions [7]. This group of methods is not considered here, and the comparison of results would not give meaningful results because the ability of estimation disparity map depends mainly on the segmentation step, which is a totally different issue in machine vision research.

In order to improve results of features stereo matching there is a big group of methods which aim at matching piecewise-linear edge segments [8, 9]. The matching of edge segments has advantages that the error of isolated pixels in edges has very little influence on the position and orientation of the edge segments. There are different modifications in the descriptions of edge segments [10] and the edges could be matched as edge chains [11]. This group of methods is out of our interest because they do not depend on an edge detector and have a different philosophy behind them than matching edges directly.

The well known algorithm for matching features is the Marr-Poggio-Grimson algorithm, MPG in short [12, 13]. This algorithm is an implementation of the human stereo vision theory that was developed by Marr and Poggio [14]. Because the MPG algorithm works mainly with reliance on the edge detection it has been used here as the reference algorithm. All the details of this algorithm are given in the Sec. 5.1.

The paper proceeds in three stages: the first gives a formal statement of the stereo matching problem; the second part contains an algorithm based on fuzzy logic for edges detection and a manner for establishing feature points as well as a method for the calculation of a disparity map; the third part of the paper presents evaluation results of the introduced method and the comparison with the MPG algorithm.

ISBN: 978-989-95079-6-8

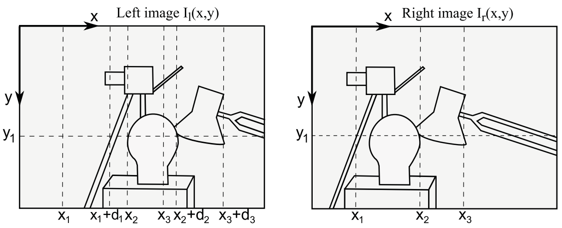


Figure 1: Depiction of the stereo matching problem

2 Problem statement

The stereo vision problem relies on establishing the correspondence between physically the same points in two different images representing the same scene. However, these two images have to be taken under some geometric restrictions in order to ensure they constitute a proper stereo pair [15, 16]. If two images constitute a proper stereo pair then there is a horizontal shift between these two images called *disparity*. Finding this shift is the main aim of almost every stereo matching algorithm. This task is illustrated in Fig. 1 which shows, that the same physical points (x_1, x_2, x_3 in the Fig. 1) in the left and the right images have different abscissa in the image coordinate system. The matching task is usually posed as the problem of finding points in the right image I_r corresponding to the selected points in the left image I_l. The value of abscissa shift of the determined point, called *disparity* is here refereed to as d. The task is to find the proper values of disparity d which correspond to the same physical points.

Usually, the epipolar constraint [15] restricts the search to a horizontal line of images. A further simplification is attained by limiting the maximum possible value of disparity d, for which a good estimate can exist. In this paper we restrict our considerations to a parallel rectified pair of cameras. This means that the searching is done for each line of images and the lines of the left and right images correspond to each other.

3 Features detector

Edge detection is a well known problem for the machine vision community. There are many more or less popular algorithms in the literature considering on image processing tasks [17, 18]. In this paper we used some new kind of an edge detector. This edge detector is based on fuzzy relation and with its characteristic feature being rather fat edges obtained as the detection results [19]. The thickness of the detected edges could be a disadvantage in some applications but in the matching problem has become an advantage, because the edges have more information about themselves and about their neighborhood.

3.1 Edges detection

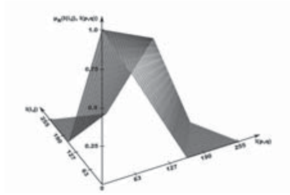
The principle behind the presented algorithm is the fact that for the pixel which belongs to an edge, the neighboring pixels have different values. Therefore if the edge is strong then in the determined small neighborhood of the pixel under consideration there are numerous pixels with the big difference of value.

The distinction between these pixels is made by using the fuzzy relation [20]. As it is pointed in [19] a lot different shapes of fuzzy relation can be used. In this paper we used the fuzzy relation in the triangular form (1). The shape of the triangular similarity relation is shown in Fig. 2(a) and its strength depends mainly on α value.

Formally the triangular fuzzy relation between the pixels $I(i,j)$ and $I(k,l))$ is expressed:

$$\mu_{RT}\left(I(i,j), I(k,l), \alpha\right) = \begin{cases} 1 - \frac{|I(i,j)-I(k,l)|}{\alpha} \\ \quad if \quad |I(i,j) - I(k,l)| < \alpha \\ 0 \\ \quad if \quad |I(i,j) - I(k,l)| \geq \alpha \end{cases} \tag{1}$$

In the edge detection stage the distinctions between one distinguished pixel $I(i,j)$ and its neighbors are calculated. If the differences are big, there are likely big changes in image, and there could be an edge present. The algorithm for edge detection is described as follows:



I(i-1,j-1)	I(i-1,j)	I(i-1,j+1)
I(i,j-1)	I(i,j)	I(i,j+1)
I(i+1,j-1)	I(i+1,j)	I(i+1,j+1)

(a) (b)

Figure 2: The shape of triangular fuzzy relation used for the edge detection (a) and the neighborhood (b) of distinguished pixel.

1. The determination of the differences between the distinguished pixel and its small established neighborhood. This situation of the distinguished pixel and its neighboring pixels is shown in Fig. 2(b). In this case we use the window size eight but many different sizes of this neighborhood can be used [19].
   The vector of homogenity description h_v is defined. The length of homogenity vector h_v is equal to the size of the neighborhood. For each pixel of the image the values of the vector h_v are calculated by (2):

$$h_v = \bigcup_{k,l=-\rho}^{\rho} \mu_{RT}\left[I(i,j), I(i+k, j+l)\right] \tag{2}$$

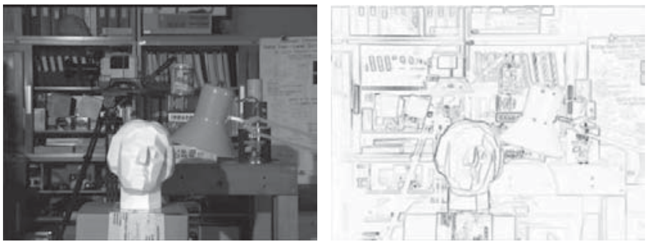
2. The final response of the edge detector is defined as the sum of the homogenity vector values:

$$h_e(i,j) = \sum h_v \tag{3}$$

If the region is uniform, the response of the edge detector would be big because all of the pixels have big similarity close to one. If the region is diverse, and if a strong edge is present in the region, the response of the detector is smaller. The results of the detector working for two images are shown in Fig. 3 and 4. We reverse the colors of the edge images in order to get the better presentation in black in white background. Originally the colors for the edges are brighter, the bigger response of the edge detector while the background is black.

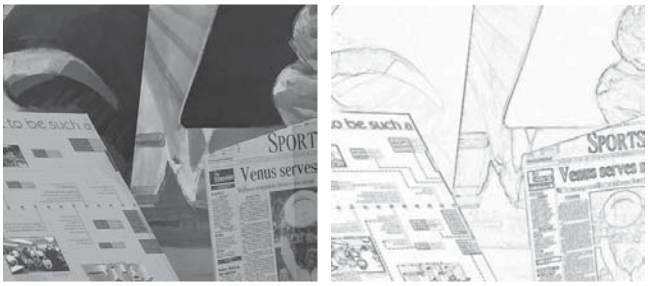
Moreover, the values of the edge detector response called "edginess" could be considered as values of some membership function. The bigger the response, the bigger the pixel's membership function value.

ISBN: 978-989-95079-6-8



(a) (b)

Figure 3: The left of stereo pair image called ”Tsukuba” (a) and the result of detected edges (b).



(a) (b)

Figure 4: The left image of an another stereo pair called ”Venus” (a) and the result of edges detection (b).

3.2 Edges thinning

The edges detected by the presented edges detection procedure are rather fat therefore we need to further reduce the number of points which are subject to match-making. In order to obtain the reduction of the edge points, a thinning algorithm is executed at the edge images. The thinning is done in three independent steps.

The full thinning algorithm proceeds in the following way:

1. *Thresholding:* The first step of the edge thinning is attenuation of small values by using the global threshold P_{th} with the value given by:

$$P_{th} = a_{th} \times E_{mean} \tag{4}$$

where a_{th} is an arbitrary chosen empirical threshold value. In this work it was set at the value equals 1.25. The term E_{mean} determines the average value of the edges image:

$$E_{mean} = \frac{1}{M \cdot N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} e(i,j) \tag{5}$$

where $e(i,j)$ is the pixel value of the edge image.

2. *Horizontal non-maximum suppression:* This step has similar idea to the non-maximum suppression algorithm. But in this step, it is made only in the horizontal direction of the image, as carried out:

$$p_e(i,j) = \begin{cases} p_e(i,j) & \textbf{if} \quad p_e(i,j) > p(i-1,j) \\ & \textbf{and if} \quad p_e(i,j) > p(i+1,j) \\ 0 & \textbf{if} \quad else \end{cases} \tag{6}$$

3. *Vertical non-maximum suppression:* The last step is practically the same as the previous step, but it is done in the vertical direction of the image:

$$p_e(i,j) = \begin{cases} p_e(i,j) & \textbf{if} \quad p_e(i,j) > p(i,j-1) \\ & \textbf{and if} \quad p_e(i,j) > p(i,j+1) \\ 0 & \textbf{if} \quad else \end{cases} \tag{7}$$

As the result the images with the edges of one pixel thickness are obtained. In these images the positions of the remaining pixels determine the locations of the features which are to be matched.

The examples of the results obtained by the edge thinning algorithm and the locations of the feature points for the images considered earlier (see Fig. 3 and Fig. 4) are shown in Fig. 5.

(a) (b)

Figure 5: The feature points after edge thinning applied for the images in Fig. 3 (a) and Fig. 4 (b) respectively.

4 Feature points matching

A typical algorithm for finding a stereo matching solution works for an arbitrary range of disparity values from d_{min} to the d_{max}. There are two windows established, one of which, called a reference window is set at a stable position in one (usually left) of the images constituting a stereo pair and the other called a search window moves along the horizontal line in the other image (usually right) within the established disparity range. For each position of the search window some kind of matching measure is calculated. The value of d for which position of the search window the value of measure attains its supreme is taken as the proper value.

The presented algorithm works in the mentioned scheme, that is reference and search windows, but during the matching stage the feature image and the edges images are used. The feature image determines the position of the feature points received after using the edge thinning. It is used for establishing the position of the reference window in the reference edge image exactly at location of the feature point. After establishing the position of the reference window the matching is carried out by moving the search window in the other edge image.

The measure of matching is calculated based on the edge images. For each point which have determined as the feature point in the feature images, the calculation of matching score in the edge images is done. As it was mentioned earlier the values of the edge detector can be interpreted as the values of the membership function which gives the degree of pixel belonging to the ”edginess” variable. For this reason the calculation of matching score is done in a fuzzy fashion.

ISBN: 978-989-95079-6-8

The matching measure used in this algorithm is the correlation between fuzzy sets [21]. Let us assume that there is a sequence of paired data $((\mu_A(x_1), \mu_B(x_1)), \ldots, (\mu_A(x_n), \mu_B(x_n)))$ which corresponds to the grades of the membership functions of fuzzy sets A and B defined on X. The correlation coefficient, $f_{A,B}$, between the fuzzy sets A and B is calculated as:

$$r_{A,B} = \frac{\sum_{i=1}^{n} (\mu_A(x_i) - \bar{\mu}_A)(\mu_B(x_i) - \bar{\mu}_B) / (n-1)}{S_A \cdot S_B} \quad (8)$$

where $\bar{\mu}_A$ and $\bar{\mu}_B$ denote the average membership grades of the fuzzy sets A and B according to the dependence:

$$\bar{\mu}_A = \frac{\sum_{i=1}^{n} \mu_A(x_i)}{n} \quad (9)$$

and S_A and S_B are the standard deviations of fuzzy sets A and B calculated as: $S_A = \sqrt{S_A^2}$, where S_A^2 represents the degree of variation of the membership function determined by:

$$S_A^2 = \frac{\sum_{i=1}^{n} (\mu_A(x_i) - \bar{\mu}_A)^2}{n-1} \quad (10)$$

The measure of correlation (8) is simply adapted to the measure of correlation between "edginess" in the reference window and "edginess" in the search window. As in a typical matching algorithm the score is calculated for each shift from the chosen range and the value d_k for which the correlation attains its maximal value is taken as the proper disparity value.

5 Results evaluation and comparison

For the testing purposes we used some benchmark stereo pairs made public accessible [3] from the Middlebury Stereo Vision Page [22]. The big advantage of these stereo pairs is that they have the true disparity map available. This allows to make a reasonable comparison of different algorithm in reference to the true result which should be obtained.

5.1 Reference feature matching algorithm

As the reference algorithm the Marr-Poggio-Grimson (MPG) has been selected. This algorithm is very popular, easy to implementation and gives good results. The implementation which is used here was based on the scheme described in [12, 13].

The MPG algorithm works in the coarse to fine scheme. In the description of this algorithm the word "channel" is used and it refers to image filtering with a specific filter window with established size w. The coarse channel means the filtering with the bigger size of filter window and the finest channel means that the image was filtered by using filter with smaller size.

The typical scheme of the MPG algorithm consists of six steps. The steps are outlined below:

1. *LOG filtering:* Started from the coarsest channel the left and right image of stereo pair is convolved with LOG filter. The LOG is a smoothed second derivative of the image signal and assumes the following form:

$$\nabla^2 (G(x,y)) = \left(\frac{x^2+y^2}{\sigma^2}\right) \exp\left(\frac{-\left(x^2+y^2\right)}{2\sigma^2}\right) \quad (11)$$

where ∇^2 is the Laplacian $\nabla^2 = \left(\frac{\partial^2}{\partial x^2}\right) + \left(\frac{\partial^2}{\partial y^2}\right)$ and $G(x,y))$ is the Gaussian function, which acts as a low-pass-filter to the image:

$$G(x,y) = \sigma^2 \exp\left(\frac{-\left(x^2+y^2\right)}{2\sigma^2}\right) \quad (12)$$

where the width of the channel "*w*" is related to σ as follows: $w = \sqrt{2}\sigma$

2. *Zero crossing extraction:* In the filtered image the detection of zero crossing is done by scanning the image horizontally for adjacent elements of opposite sign or for three horizontally adjacent elements, where the middle one is zero. Also the sign of zero crossing is searched. The located position of zero crossing with their signs are remembered.

3. *Matching:* For each scan line in the left image is centered reference window at the founded zero crossing point as the possible candidate to matching. In the right image the search window is moved in established disparity range. If in the right image is a zero crossing point on the search window and has the same sign as the left point, then this zero-crossing point produces a match. The disparity, difference between locations of zero crossing points in the left and right image, is stored in a dynamic buffer.

   Based on the matching process, the left zero crossing is marked as:

   - *unique match*, if only one right image zero crossing is matched with the left one
   - *multiple matches*, if more than one match is found
   - *no match*, if no match is found

4. *Disambiguation:* In this step the disparity map is checked for possible double matches. This kind of ambiguity is resolved by checking the disparities within the same region of the representation at the previous channel. If there is no coarser channel, or there is no a disparity value within this region in the coarser channel, or the disparity is not consistent with coarser level disparity, then the disparity is discarded. The disparity can be approximated by taking the average of the multiple match.

5. *Loop:* When the final map disparity for the current channel has been completed, the process return to the convolution to the next finer representation.

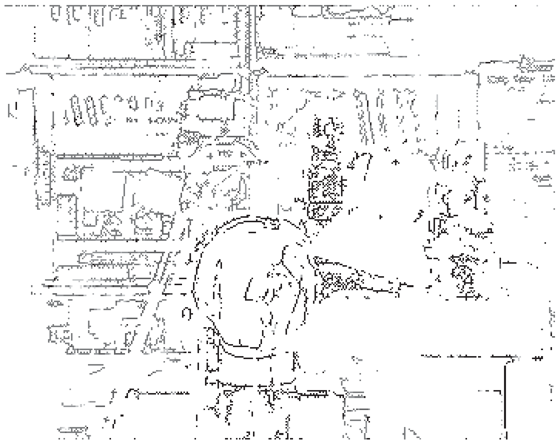
6. *Consistency:* After calculation of disparity in each channel has been completed, there is one final test done. Each disparity value at the finest channel is tested for consistency by checking its value with values at the coarser channel. If the values are inconsistent the matching is eliminated.

In this paper we used, as in the Grimson [12] implementation, four channels with size of 5, 9, 13 and 17 pixels respectively.

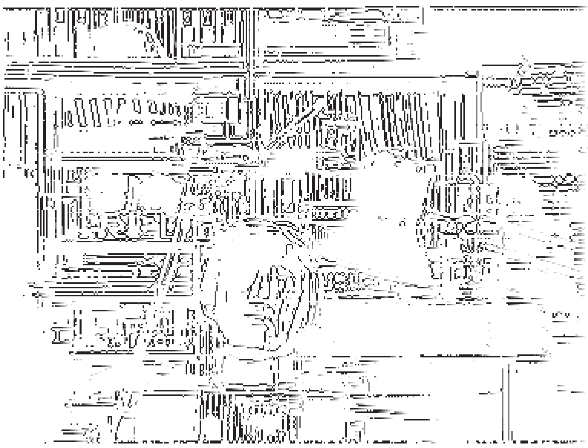
ISBN: 978-989-95079-6-8

5.2 Results evaluation

The algorithm proposed here and the MPG algorithm were applied for two different stereo pairs. The Fig. 6 presents the estimated disparity maps for "Tsukuba" stereo pair. Similar as in the edge images the colors were reversed for better visual presentation. If the color at the disparity map is darker the original disparity value has bigger value, and the element has bigger disparity value. The second image Fig. 7 presents the disparity maps obtained for the "Venus" stereo pair images.



(a) The result of the proposed algorithm.



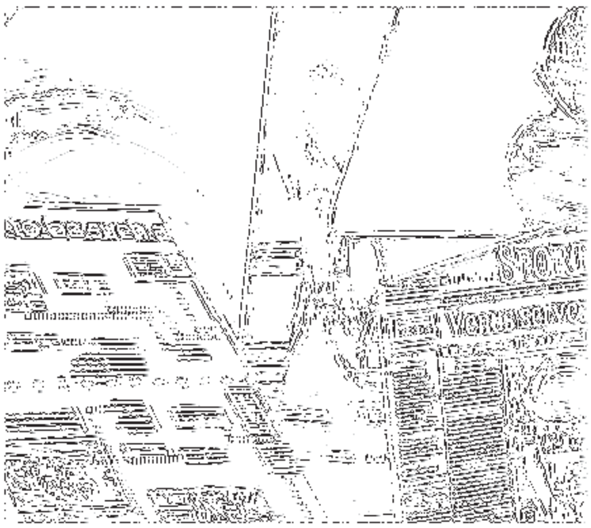
(b) The result of the MPG algorithm.

Figure 6: Disparity maps for the stereo pair "Tsukuba" (a) estimated by the proposed fuzzy algorithm and the MPG (b).

Visually these results are not so different. In the "Tsukuba" stereo pair can be seen that MPG algorithm gives some different matching but the comparison is very hard to done. Just comparison of numerical statistic given in the Tab.1 gives better view at obtained results. It can be seen that the amount of points classified as the features points is similar in both algorithms but the proposed algorithm based on fuzzy edge detection gives much more true matches. During generation of the results enclosed in Tab. 1 a point was classified as the proper matching point if the difference between its disparity and the true disparity taken from the true disparity map was equal zero or one. In this case the small mistake equal one was allowed. If the difference between estimated disparity value and the true value was bigger than one the matching point was classified as wrongly matched point.

In order to check the characteristics of these algorithms in some more deep manner the second experiment was done. In this case their behavior in the presence of noise in images was investigated. For this reason some amount of Gaussian noise was added to the both images consist of stereo pair. The amount of added noise is determined in the logarithmic scale and was established at $30dB$ value. The value denotes the amount of noise and is calculated by using expression:

(a) The result of the proposed algorithm



(b) The result of the MPG algorithm

Figure 7: Disparity maps for the stereo pair "Venus" (a) estimated by the proposed fuzzy algorithm and the MPG (b).

$$SNR = 10 \cdot \log_{10} \left(\frac{P_{image}}{P_{noise}} \right) \qquad (13)$$

For discrete image I with size $M \times N$ and intensities $I(i,j)$ of pixels the power P_{image} is expressed as following:

$$P_{image} = \frac{1}{M \cdot N} \sum_{i,j=1}^{M,N} I^2(i,j) \qquad (14)$$

For the purpose to produce the image with assigned SNR to the image the Gaussian noise was added iteratively and the power of noise was calculated for the noise image:

$$I_{noise} = I_{distorted} - I_{original} \qquad (15)$$

ISBN: 978-989-95079-6-8

Table 1: The quantitative results of comparison for matching algorithms.

Image	Points	Right	Bad	Right %	Bad %
Results obtained by the fuzzy matching					
Tsukuba	11579	10392	1187	89.8	10.2
Venus	18588	16479	2109	88.7	11.3
In the presence of noise in image; $SNR = 30dB$					
Tsukuba	11898	2629	9269	22.1	77.9
Venus	19269	3313	15956	17.2	82.8
Results obtained by the Marr-Poggio-Grimson algorithm					
Tsukuba	11579	9549	2030	82.4	17.6
Venus	19143	12449	6694	65.1	34.9
In the presence of noise in image; $SNR = 30dB$					
Tsukuba	12345	582	11763	4.7	95.3
Venus	22705	564	22141	2.5	97.5

where the power $I_{original}$ had been calculated (13) before adding noise and the P_{noise} was calculated for the noised image. The amount of noise had been increasing to the moment for which the SNR attained the value equal to $30dB$.

As it is shown in Tab.1 both algorithms have very small resistance at the noise. The number of features does not decrease but the the amount of the proper matching is decreasing very significantly. But the proposed fuzzy approach still give better results.

6 Conclusion

In this paper the full scheme for features stereo matching based on fuzzy edge detection has been presented. The application of fuzzy logic allowed to obtain results better than the good and stable MPG algorithm.

However, both of these algorithms are very sensitive to the noise present in the images under matching consideration. It seems that the most sensitive stage is the edge detection. This aspect of these algorithms should be researched more deeply because the noise is very important in the real applications of stereo vision. In the real world numerous kind of distortions should be considered and this topic should be examined more carefully.

The obtained results are probably also image depended. In the case where the image is more complicated and there are many more edges in an image, as in "Tsukuba" stereo pair, the number of properly matched points is always bigger. In the "Venus" stereo pair, in which long straight edges are present, both algorithms found much more points classified as the feature points but the matching in both cases was worse. In that kind of images probably the algorithms based on segments edge matching would give better results. But when the image encloses a big number of short edges algorithms presented here should have better properties. This is probably a clue for further research but it is out of scope of this paper at this moment.

References

[1] M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *IEEE Trans. on Pattern Analys. and Machine Intell.*, 25(8):993–1008, 2003.

[2] U.R. Dhond. Structure from stereo - a review. *IEEE Transc. on Systems, Man and Cybernetics*, 19(6):1489–1510, 1989.

[3] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 47(1/2/3):7–42, 2002.

[4] F. Tiesheng, N. Bing, W. Qingsong, W. Tao, and Q. Dapeng. Novel stereo matching method on multi-scale harris corner points. *IEEE Proc. of the Int. Symposium on Comp. Science and Computat. Techn.*, pages 167–170, 2008.

[5] D. Zhao, D. Liu, and Y. Yang. A new stereo matching method based on sub-pixel corner detection. In *IEEE Proc. of the Int. Conf. on Comp. Science and Soft. Eng.*, pages 899–902, 2008.

[6] X. Su and T.M. Khoshgoftaar. A progressive edge-based stereo correspodence method. In *Proc. of the Int. Symposioum on Visual Comput. (ISVC)*, volume 1, pages 248–257, 2007.

[7] K. Kyung-Yoo, K. Sung-Yeol, and H. Yo-Sung. Stereo matching using global and local segmentation. In *Proc. of the Int. Technical Conf. on Circuts/Systems, Comp. and Communicat. (ITC-CSCC'07)*, pages 1225–1226, 2007.

[8] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *Proc. of the IEEE Comp. Society Conf. on Comp. Vision and Pattern Recogn. (CVPR'05)*, 2005.

[9] A. Luo, W. Tao, and H. Burkhardt. A new multilevel line-base stereo vision algorithm based on fuzzy techniques. In *Proc. of the Int. Conf. on. Pattern Recogn. (ICPR'96)*, volume 1, page 383, 1996.

[10] X. Shenghua, L. Jiping, P. Jianping, and W. Yong. Straight line extraction for stereoimage feature matching. In *IEEE Proc. of the Congress on Image and Signal Processing*, 2008.

[11] B. Tang, D. Ait-Boudaoud, J. Matuszewski, and L. Shark. An efficient feature based matching for stereo images. In *Proc. of the Geometric Modeling and Imaging - New Trends (GMAI'06)*. IEEE, 2006.

[12] W. Grimson. Computational experiments with a feature based stereo algorithm. *IEEE Transc. on Pattern Analysis and Machine Intell.*, 7(1):17–34, 1985.

[13] D.J. Braunegg. Stereo feature matching in disparity space. *IEEE*, pages 796–803, 1990.

[14] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proc. of the Royal Society of London*, B(204):301–328, 1979.

[15] O.D. Faugeras. *Three-dimensional computer vision: a geometric viewpoint.* MIT Press, Cambridge, 1993.

[16] E. Trucco E. and A. Verri. *Introductory techniques for 3-D computer vision.* Prentice Hall, New Jersey, 1998.

[17] W.K. Pratt. *Digital image processing.* John Wiley & Sons, Inc., third edition edition, 2001.

[18] R. Nevatia. *Machine perception.* Prentice-Hall, Inc., 1982.

[19] G. Nieradka. Method for edge detection in images using fuzzy relation. In *Proceedings of the SPIE*, volume 6937, pages 69373E–69373E–8, 2008.

[20] G.J. Klir and B. Yuan. *Fuzzy sets and fuzzy logic. Theory and applications.* Prentice Hall PTR, 1995. Upper Saddle River, New Jersey.

[21] D.-A. Chiang and N.P. Lin. Correlation of fuzzy sets. *Fuzzy sets and systems*, 102(2):221–226, 1999.

[22] Middlebury Stereo Vision Page. Available from: `http://vision.middlebury.edu/stereo`. [cited: 20.01.2009].

ISBN: 978-989-95079-6-8