A New Corner Detector Supporting Feature Extraction for Automatic 3D Reconstruction

Annamária R. Várkonyi-Kóczy[1,2], András Rövid[1,2], Endre Selényi[1]

[1] Dept. of Measurement and Information Systems, Budapest University of Technology and Economics

[2] Integrated Intelligent Systems- Japanese-Hungarian Laboratory
Budapest, Hungary
koczy@mit.bme.hu, rovid@mit.bme.hu, selenyi@mit.bme.hu

Abstract: Corner detection plays an important role in computer vision as well as in 3D reconstruction of a scene. With the help of the corners we can determine the most characteristic points of an object and so reconstruct them. Corners are useful in case of pattern recognition, as well. In this paper a new corner detection scheme is introduced which is based on fuzzy reasoning and applies a special local structure matrix.

Index Terms: corner detection, fuzzy image processing, noise elimination, noise smoothing, 3D modeling

1 Introduction

Corner detection plays an important role in computer vision as well as in pattern recognition [1], in shape and motion analysis [2] and in 3D reconstruction [3] of a scene. Motion is ambiguous along an edge, unambiguous at a corner. In most cases shapes can approximately be reconstructed from their corners [4]. 3D reconstruction from images is a common issue of several research domains. In recent time, the interest in 3D models has dramatically increased [5], [6].

More and more applications are using computer generated models. In many cases models of existing scenes or objects are desired. Creating photorealistic 3D models of a scene from multiple photographs is a fundamental problem in computer vision and in image based modeling. The emphasis for most computer vision algorithms is on automatic reconstruction of the scene with little or no user interaction. The basic idea behind 3D model reconstruction from a sequence of un-calibrated images can be defined in several steps [7]: first we need to relate the images in the whole sequence, then extract information based on pixel correspondences to be able to apply methods of epipolar geometry. In real life image sequences contain many of the points that are better suited for automated matching than others. The environment of these points contains significant intensity variations and is therefore easy to differentiate from others. The correspondence between such points of interest has to be done by some kind of matching procedure. A possible approach to select points of interest is the corner detection. Corners in a scene are the end points of the edges. As we know, edges represent object boundaries and are very useful in 3D reconstruction of a scene. There are two important requirements for the feature of the points. First, points corresponding to the same scene point should be extracted consistently over the different views. Secondly, there should be enough information in the environment of the points so that the corresponding points can be automatically matched. With the help of detected corners it is possible to determine the corresponding points. For example, if we are able to select points on the corners of objects we have a greater chance for matching the same corners in another image. There are several known corner detection algorithms for the estimation of the corner points. These detectors are based on different principles, characteristic for the algorithm. It is known, that there are corner detectors, whose functionality is based on a local structure matrix, consisting the first partial derivatives of the intensity function. An example of it is the Harris feature point detector [8]. Harris' method is based on a comparison: the measure of the corner strength – which is defined by the method and is based on a local structure matrix – is compared to an appropriately chosen concrete threshold. Another well known corner detector is the SUSAN (Smallest Univalue Segment Assimilating Nucleus) detector based on brightness comparison [9]. It does not depend on image derivatives. The SUSAN area reaches a minimum, when the nucleus lies on a corner point. This method is more resistant to image noise. The effectiveness of these above mentioned algorithms is acceptable. The main point in these algorithms is the following: When a calculated value of the mentioned detectors (which is characteristic for a corner) exceeds a given threshold the processed image point is usually detected as a corner.

The novelty of this paper is that the fuzzy based corner detection algorithm proposed by the authors in [4] is improved by an image smoothing procedure applied in the preprocessing phase together with the introduction of a fuzzy based technique assigning new attributes to the detected corner candidates (showing the membership value of being a „real" corner). This property of the detector can very advantageously be used at the matching of the corresponding image points in stereo image pairs and results in a better output.

The paper is organized as follows: Section II is devoted to the noise elimination of images and describes why it is important to smooth the images before corner detection. Section III details how to determine the corner points of an image. Section IV deals with the point correspondence matching in stereo image pairs. Section V shows experimental results and finally in Section VI conclusions are summarized.

2 Noise Elimination and Gaussian Smoothing

Before starting to search for the corner points of an image it is necessary to eliminate the noise. For this purpose we use a special fuzzy system characterized by an IF-THEN-ELSE structure and a specific inference mechanism proposed by Russo [10]. Different noise statistics can be addressed by adopting different combinations of fuzzy sets and rules.

Let $I(\mathbf{r})$ be the pixel luminance at location $\mathbf{r}=[x,y]$ in the noisy image, where x is the horizontal and y the vertical coordinate of the pixel. Let $I_0 = I(\mathbf{r}_0)$ denote the luminance of the input sample having position $\mathbf{r}_0$ and being smoothed by a fuzzy filter. The input variables of the fuzzy filter are the amplitude differences defined by:

$$\Delta I_j = I_j - I_0,\ j = 1,\ldots,8 \tag{1}$$

where $I_j=I(\mathbf{r}_j)$, $j=1,\ldots,8$ values are the luminance values of the neighboring pixels of the actually processed pixel $\mathbf{r}_0$ (see Fig. 1a). Let K_0 be the luminance of the pixel having the same position as $\mathbf{r}_0$ in the output signal. This value is determined by the following relationship:

$$K_0 = I_0 + \Delta I \tag{2}$$

where ΔI is determined later (see eq. (5)).

Let $W = \bigcup_{i=1}^{9} W_i$ be defined by a subset of the eight neighboring pixels around $\mathbf{r}_0$ belonging to a 3x3 moving window (see Fig. 1a). Let the rule base deal with the pixel patterns $W_1,\ldots,W_9$ (see Fig. 1b.) The value K_0 can be calculated, as follows [10], [11]:

$$\lambda = MAX\left\{MIN\left\{m_{LP}(\Delta I_j) : r_j \in W_i\right\}; i = 1,\ldots,9\right\} \tag{3}$$

$$\lambda^* = MAX\left\{MIN\left\{m_{LN}(\Delta I_j) : r_j \in W_i\right\}; i = 1,\ldots,9\right\} \tag{4}$$

$$\begin{aligned} \Delta I &= (L-1)\Delta\lambda \\ K_0 &= I_0 + \Delta I \end{aligned} \tag{5}$$

where $\Delta\lambda=\lambda-\lambda^*$, L is the maximum of the gray level intensity, m_{LP} and m_{LN} correspond to the membership functions and $m_{LP}(I)=m_{LN}(-I)$ (see Fig. 2). The filter is recursively applied to the input data.

In Figs. 3-4 the effectiveness of the noise smoothing can be followed. Fig. 3 shows the photo of a crashed car corrupted by noise, while in Fig. 4 the filtered image can be seen. Starting from the above detailed algorithm new corner detection procedures can be built.

r_1	r_2	r_3
r_4	r_0	r_5
r_6	r_7	r_8

Figure 1a

The neighboring pixels of the actually processed pixel $\mathbf{r_0}$

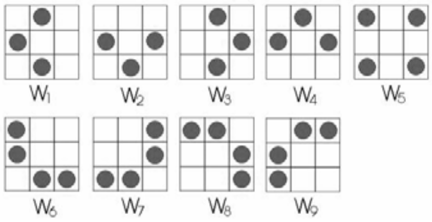

Figure 1b

Pixel Patterns

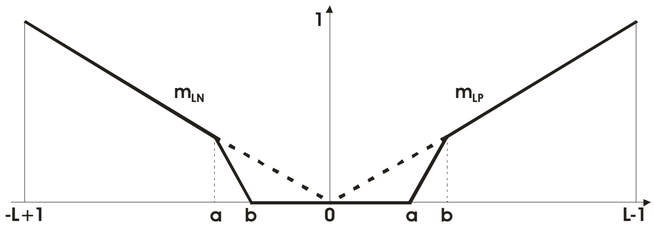

Figure 2

Membership functions m_{LN} (large negative) and $m_{LP,}$ (large positive), *a* and *b* are parameters for the tuning of the sensitivity to noise of the filtering

In corner detection, besides the noise another problem can be that the digital image is stored as a collection of discrete pixels, thus an edge is represented as a serie of discrete pixels causing that small brakes may occure in the edge which in some cases are detected as (false) corners. In Fig. 5 the left image illustrates how a line appears when the resolution of the image is finite. For increasing the effectiveness of the corner detection algorithm, the false corners should be eliminated before applying the corner detector. For this purpose we implement a Gaussian smoothing algorithm, which is usually used to 'blur' images and to remove unimportant details and noise. In Fig. 5 the right image shows how a line after smoothing appears in the image.

Figure 3
Original photo of a crashed car corrupted by noise

Figure 4
Fuzzy-filtered image of the photo in Fig. 3

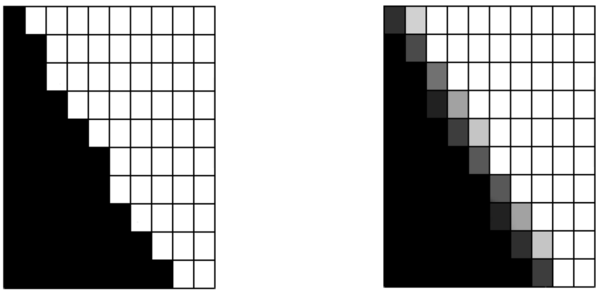

Figure 5
Edge representation without smoothing (left), edge representation after applying the smoothing algorithm

The algorithm uses a kernel of size *N*x*N* that represents the shape of a Gaussian hump. This kernel has special properties detailed below. A circularly symmetric Gaussian has the form of [12]:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{6}$$

where x and y stand for the 2D coordinates of a point and σ is a parameter with the help of which we can shape the distribution. An example of this distribution can be seen in Fig. 9. The main idea of Gaussian smoothing is to use this 2-D distribution as a „point-spread" function which can be achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution [12].

The above detailed smoothing algorithm is used by the next section for obtaining the entries, which corresponds to the first partial derivates of the intensity function $I(x,y)$, as well (see section III). Fig. 6 illustrates the smoothed image got by smoothing the intensity function of Fig. 4.

3 Detection of Corner Points

Corners are local image features characterized by locations where the variations of the intensity function $I(x,y)$ are high in both x and y directions, i.e. both partial derivatives I_x and I_y are large.

Most of the corner detection algorithms are derived from a so called local structure matrix, which has the form of

$$\mathbf{L}(x,y) = G(x,y) * \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}, \tag{7}$$

where $G(x,y)$ represents the above mentioned Gaussian hump and * stands for the convolution. This matrix can be derived from locally approximating the autocovariance function of a real valued stochastic signal $I(x,y)$ (generated by a stochastic process) in the origo [13].

Figure 6

Gaussian Smoothed image

One of the corner detection algorithms, which uses the above local structure matrix is the Förstner's one. Förstner determines corners as local maxima of function $H(x,y)$ [14].

$$H(x,y) = \frac{\left(\frac{\partial I}{\partial x}\right)^2 \left(\frac{\partial I}{\partial y}\right)^2 - \left(\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}\right)^2}{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{8}$$

Starting from the algorithm of Förstner a new improved corner detection algorithm can be developed by combining it with fuzzy reasoning. This is used for the characterization of the continuous transient between the localized and not localized corner points, as well. The algorithm consists of the following steps. First, the picture, in which we have to find the corners, is preprocessed. As a result of the preprocessing procedure the noise is eliminated. For this purpose we apply the intelligent fuzzy filters described in [10] and [11]. The filtering is followed by the smoothing applying the convolution of the filtered intensity function with the 2D Gaussian hump. After preprocessing, the first derivatives of the intensity function $I(x, y)$ are calculated in each image point. For this purpose we apply the following convolution masks:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ for determining } \frac{\partial I}{\partial x} \text{ and } \frac{\partial I}{\partial y}\text{, respectively.}$$

For increasing the effectiveness of the corner detection it is proposed to smooth each of the entries I_x^2, I_y^2, I_xI_y, in eq. (8), which correspond to the first partial derivates of the intensity function $I(x,y)$ (here x,y denote the 2D coordinates of the pixels). This can be done by applying a Gaussian 6x6 convolution kernel with σ=1, illustrated in Fig 9. [12]. As the following step, the values $H(x,y)$ are calculated for each image point with the help of the previously determined I_x^2, I_y^2, and I_xI_y smoothed values. If the detected corners are neighbors, then we should keep only the corner having the largest calculated value $H(x,y)$. The others are to be ignored. In most cases we can not unambiguously determine that the analyzed image point is a corner or not with only the help of a certain concrete threshold value, therefore in the proposed algorithm, fuzzy techniques are applied for the calculation of the values (corners) which increases the rate of correct corner detection. As higher the calculated H value as higher the membership value, that the analyzed pixel is a corner. Fuzzifying the H values into fuzzy sets, and applying a fuzzy rulebase we can evaluate the "cornerness" of an analyzed pixel. This property of the pixel can advantageously be used by the searching for the corresponding corner points in stereo image pairs, which is an indefinite step of the automatic 3D reconstruction. The antecedent and consequent fuzzy sets of the detector are illustrated in Figs. 7 and 8, respectively. In Fig. 7 (antecedent fuzzy sets) the horizontal axis represents the calculated H values. These values are classified into fuzzy sets $\mathbf{C}_{\mathrm{WEAK}}$, $\mathbf{C}_{\mathrm{MEDIUM}}$ and $\mathbf{C}_{\mathrm{STRONG}}$, which represent the fuzzy set of values corresponding to WEAK, MEDIUM and STRONG corners, respectively, while in Fig. 8 (consequent fuzzy sets) on the horizontal axis the intensity values of the pixels are shown. With the help of the parameters H_k (k=1,2,3,4,5) (see Fig. 7) the shape of the membership functions $\mu_{C_{WEAK}}$, $\mu_{C_{MEDIUM}}$, $\mu_{C_{STRONG}}$ can be modified and so the sensitivity of the described detector can be changed. If the pixel is not a corner (no fuzzy rules are fired) then its intensity will be zero, in other cases the "cornerness" is evaluated by the fuzzy rulebase below. The fuzzy rulebase of the detector has the following rules:

IF ($H(x,y)$, $\mathbf{C}_{\mathrm{WEAK}}$) THEN ($I(x,y)$, $\mathbf{I}_{\mathrm{LOW}}$),

IF ($H(x,y)$, $\mathbf{C}_{\mathrm{MEDIUM}}$) THEN ($I(x,y)$, $\mathbf{I}_{\mathrm{MEDIUM}}$),

IF ($H(x,y)$, $\mathbf{C}_{\mathrm{STRONG}}$) THEN ($I(x,y)$, $\mathbf{I}_{\mathrm{HEIGH}}$),

which means that if the $H(x,y)$ value is member of the fuzzy set $\mathbf{C}_{\mathrm{WEAK}}$ then let be the output intensity of the pixel low, if the $H(x,y)$ value the member of the fuzzy set $\mathbf{C}_{\mathrm{MEDIUM}}$ then let the output intensity of the pixel be medium and so on. Let μ (.) be the membership function of the consequent fuzzy set generated as the superposition of the rule consequents. As defuzzification algorithm we use the center of gravity method, and so we can get the intensity value of a pixel in the output image by the following way:

$$I_o(x,y) = \frac{\sum_{i=1}^{I_{\max}} \mu(I_i) I_i}{\sum_{i=1}^{I_{\max}} \mu(I_i)}, \tag{9}$$

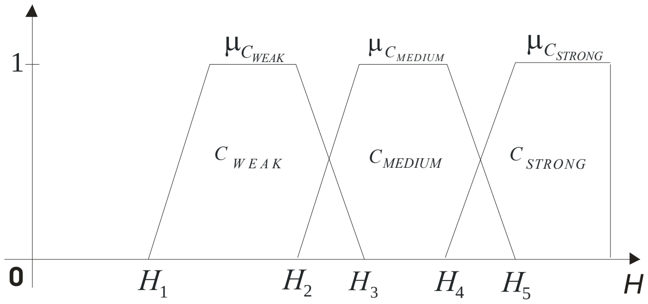

Figure 7

Illustration of antecedent fuzzy sets $\mathbf{C}_{WEAK}$, $\mathbf{C}_{MEDIUM}$ and $\mathbf{C}_{STRONG}$. The values H_k (k=1,2,3,4,5) solve for shaping the membership functions (changing the sensitivity of the detector) $\mu_{C_{WEAK}}$, $\mu_{C_{MEDIUM}}$, $\mu_{C_{STRONG}}$. The axis H is the axis of the calculated $H(x,y)$ values.

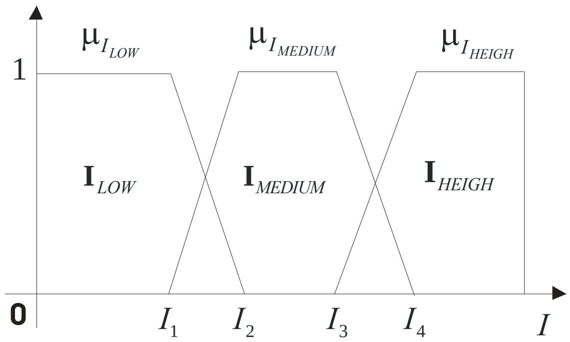

Figure 8

Illustration of consequent fuzzy sets $\mathbf{I}_{LOW}$, $\mathbf{I}_{MEDIUM}$ and $\mathbf{I}_{HEIGHT}$. The values I_k (k=1,2,3,4) solve for shaping the membership functions $\mu_{C_{LOW}}$, $\mu_{C_{MEDIUM}}$, $\mu_{C_{HEIGHT}}$. The axis I is the axis of the pixel's intensity.

0.0003	0.0023	0.0062	0.0062	0.0023	0.0003
0.0023	0.0168	0.0458	0.0458	0.0168	0.0023
0.0062	0.0458	0.1244	0.1244	0.0458	0.0062
0.0062	0.0458	0.1244	0.1244	0.0458	0.0062
0.0023	0.0168	0.0458	0.0458	0.0168	0.0023
0.0003	0.0023	0.0062	0.0062	0.0023	0.0003

Figure 9

Gaussian 6x6 convolution kernel with σ=1

where $I_{\max}$ stands for the maximal intensity. The value $I_o(x,y)$ represents the intensity value of the pixel in the output image at the position $[x,y]$.

4 Point Correspondence Matching in Stereo Image Pairs

Feature matching is commonly referred to as the correspondence problem. The problem is how to automatically match corresponding features from two images, while at the same time not assigning matches incorrectly. The common approach for corners is to take a small region of pixels around the detected corner (referred to as a correlation window) and compare this with a similar region around each of the candidate corners in the other image. Each comparison yields a score, a measure of similarity. The match is assigned to the corner with the highest matching score. The most popular measure of similarity is the cross-correlation. Most matching algorithms include constraints to complement the similarity measure. These may take the form of constraints on which corners are selected as candidate matches: a maximum disparity, or corners which agree with some known relationship between the images (such as the epipolar geometry). Constraints such as uniqueness or continuity may also be applied after candidate matches have been found. With the help of the epipolar constraint we can reduce the number of candidate image points. We have to search only along the epipolar line corresponding to the actually chosen image point in the source image. This epipolar line can be determined using the so called fundamental matrix [7]. This is a 3x3 matrix, which defines the relation between the corresponding image points. As we know, the images in which we have to find the corresponding feature points are taken from different camera positions. If the angle of the camera positions is relatively small, we have greater chance to match the mentioned feature points, because of the small deformation of image pixels between two views. In this case the corresponding points can be found with high reliability in each image. Feature point mentioned in this section can be either corners or edge points. Using the above detailed corner detector for detecting the feature points, the number of candidate feature points can be effectively reduced. For this purpose we use the knowledge, that if the angle of the camera positions is relatively small the corresponding corners have similar properties and therefore it is possible to reduce the number of candidate corner points. For details of the algorithm, see [15].

5 Experimental Results

In the followings the effectiveness of the new corner detection algorithm is illustrated and is compared to other well known algorithms through a simple example. Fig. 10 shows a part of a corridor with several lamps and doors. In Fig. 11 the corners detected by our algorithm can be seen. By analyzing the results we can see that all the corners are detected and no false corner was found. Figs. 12, 13, and 14 illustrate the results obtained by the Förstner's, Harris and SUSAN corner detection algorithms, respectively. All of these latter algorithms failed to find all the corners. The last example (Figs. 15 and 16) shows the difference of the results with (Fig. 16 (right image)) and without image smoothing (Fig. 16 (left image)).

Conclusions

In this paper an improved corner detection algorithm has been reported. This approach can very advantageously be used even if the images are corrupted by noise. The method is based on fuzzy techniques and also utilizes the results of the algorithm introduced by Förstner. With the help of this technique the most significant image points can be extracted which is very important e.g. at 3D modeling and pattern recognition.

Figure 10

The input image after noise filtering

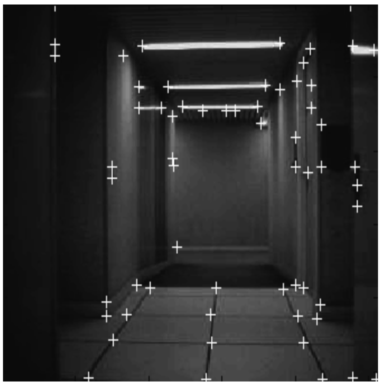

Figure 11

Corner detection applying the proposed new fuzzy based corner detection algorithm

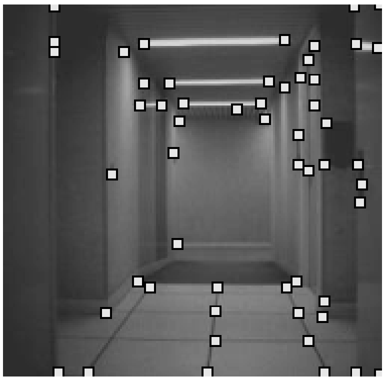

Figure 12
Corner detection applying the Förstner's corner detector

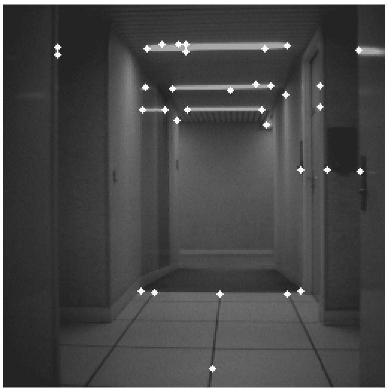

Figure 13
Corner detection applying the Harris corner detector (k=0.001)

Figure 14
Corner detection applying the SUSAN detector

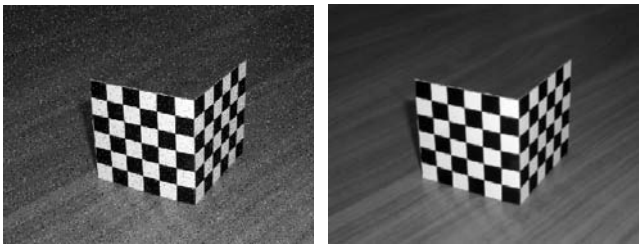

Figure 15

Image corrupted by impulse noise (left), the fuzzy filtered image (right)

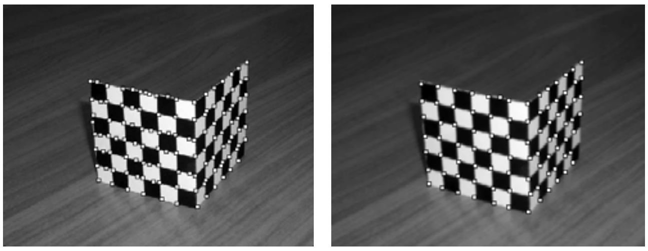

Figure 16

Detected corners in the fuzzy filtered image using fuzzy based detector without image smoothing (left), detected corners in the fuzzy filtered image using the same detector but with image smoothing

Acknowledgement

This work was sponsored by the Hungarian Fund for Scientific Research (OTKA T049519) and the Hungarian-Portuguese Intergovernmental S&T Cooperation Programme (P-24/03).

References

[1] X. Xie, R. Sudhakar and H. Zhuang, ”Corner detection by a cost minimization approach, Pattern Recognition,” Vol. 26, No. 8, pp. 1235-1243, 1993

[2] Velastin S. A., Yin J. H, Davies A. C., Vicencio-Silva M. A., Allsop R. E. and Penn A., 1994: "Automated Measurement of Crowd Density and Motion using Image Processing", 7th IEE International Conference on Road Traffic Monitoring and Control, 26-28 April 1994, London, UK, pp. 127-132

[3] Rövid, A., Várkonyi-Kóczy A. R., Maria da Graça Ruano, "Automatic 3D Modeling Based on Soft Computing Techniques," *In Proc. of the 8th IEEE International Conference on Intelligent Engineering Systems*, INES 2004, Cluj-Napoca, Romania, September 19-21, 2004

[4] Rövid, A., Várkonyi-Kóczy, A. R., "Corner Detection in Digital Images Using Fuzzy Reasoning," *In Proc. of the 2nd IEEE Int. Conference on*

Computational Cybernetics, Vienna, Austria, August 30-September 1, 2004, pp. 95-99

[5] P. E. Debevec, C. J. Taylor, J. Malik, “Modeling and rendering architecture from photographs a hybrid geometry and image based approach,” *In Proc. SIGGRAPH,* August 1996

[6] E. Grossmann, D. Ortin, J. Santos-Victor, “Single and multi-view reconstruction of structured scenes,” *The 5th Asian Conference on Computer Vision*, 23-25 January 2002, Melbourne, Australia

[7] Rövid, A., A. R. Várkonyi-Kóczy, P. Várlaki, “3D Model Estimation from Multiple Images,” In Proc. of the *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2004*, July 25-29, 2004, Budapest, Hungary, pp. 1661-1666

[8] C. Harris and M. Stephens, ”A combined corner and edge detector”, *Proc. 4th Alvey Vision Conference*, pp. 189-192, 1988

[9] S. M. Smith and M. Brady, ”SUSAN - a new approach to low level image processing”, *International Journal of Computer Vision*, Vol. 23(1), 45-78, 1997

[10] Russo, F., “Fuzzy Filtering of Noisy Sensor Data,” *In Proc. of the IEEE Instrumentation and Measurement Technology Conference*, Brussels, Belgium, 4-6 June 1996, pp. 1281-1285

[11] Russo, F., “Recent Advances in Fuzzy Techniques for Image Enhancement,” *IEEE Transactions on Instrumentation and Measurement*, Vol. 47, No. 6, Dec. 1998, pp. 1428-1434

[12] F. Catté, P.-L. Lions, J.-M. Morel, T.Coll, “Image selective smoothing and edge detection by nonlinear diffusion,” *SIAM Journal on Numerical Analysis*, 32:1895-1909, 1992

[13] M. Felsberg, “Low-Level Image Processing with the Structure Multivector,” PhD thesis, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel, 2002, TR no. 0203

[14] W. Förstner, “A feature based correspondence algorithm for image matching,” *Int. Arch. Photogramm. Remote Sensing*, Vol. 26, pp. 150-166, 1986

[15] Rövid, A., Várkonyi-Kóczy, "Point Correspondence Matching for 3D Reconstruction Using Fuzzy Reasoning," *In Proc. of the IEEE 3rd International Conference on Computational Cybernetics*, Mauritius, April 13-16, 2005, pp. 87-92