Automatic 3D Modeling Based on Soft Computing Techniques

András Rövid
Dept. of Automobiles
Budapest University of Technology and Economics
Integrated Intelligent Systems- Japanese-Hungarian Laboratory
Budapest
Hungary
andras.rovid@auto.bme.hu

Annamária R. Várkonyi-Kóczy
Dept. of Measurement and Information Systems
Budapest University of Technology and Economics
Integrated Intelligent Systems- Japanese-Hungarian Laboratory
Budapest
Hungary
koczy@mit.bme.hu

Maria da Graça Ruano
University of Algarve
Campus de Gambelas
Portugal
mruano@ualg.pt

Abstract— **3D reconstruction plays a very important role in computer vision. The determination of the 3D model from multiple images is of key importance. In this paper a 3D reconstruction algorithm is introduced, which is capable to determine the 3D model without any external intervention.**

I. INTRODUCTION

3D reconstruction from images is a common issue of several research domains. In recent time the interest in 3D models has dramatically increased [1][2]. More and more applications are using computer generated models. The main difficulty lies with the model acquisition. Although, more tools are at hand to ease the generation of models, it is still a time consuming and expensive process. In many cases models of existing scenes or objects are desired. Traditional solutions include the use of stereo rigs, laser range scanners and other 3D digitizing devices. These devices are often very expensive, require careful handling and complex calibration procedures.

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 [3].

In this work an alternative approach is proposed which avoids most of the problems mentioned above. The object which has to be modeled is recorded from different viewpoints by a camera. The relative position and orientation of the camera and its calibration parameters will automatically be retrieved from the image data. In general, 3D reconstruction requires camera calibration which is performed by checking the correspondence between 3D geometry in world coordinates and 2D geometry in image coordinates. For the reconstruction we will use only the edge points of the reconstruction object. Using this alternative, the processing time can be significantly decreased.

The paper is organized as follows: Section I shows how to detect the edges of objects using a fuzzy based filter, Section II summarizes the basics of epipolar geometry, Section III is devoted to the estimation of the fundamental matrix, Section IV describes how to find the corresponding image points, Section V shows how to estimate the perspective projection matrix from image data, and finally, Section VI and Section VII report experimental results and conclusions.

II. NOISE ELIMINATION AND ESTIMATION OF THE OBJECT EDGES

A major task in the field of digital processing of measurement signals is to extract information from sensor data corrupted by noise [4][5]. For this purpose we will use a special fuzzy system characterized by an IF-THEN-ELSE structure and a specific inference mechanism. Different noise statistics can be addressed by adopting different combinations of fuzzy sets and rules [4][5].

Let $x(\mathbf{r})$ be the pixel luminance at location $\mathbf{r}=[r_1,r_2]$ in the noisy image where r_1 is the horizontal and r_2 the vertical coordinate of the pixel. Let **N** be the set of eight neighboring pixels (see Fig. 1a). The input variables of the fuzzy filter are the amplitude differences defined by:

$$\Delta x_j = x_j - x_0, j = 1,\dots,8 \tag{1}$$

where the x_j, j=1,…,8 values are the neighboring pixels of the actually processed pixel x_0 (see Fig. 1a).
Let y_0 be the luminance of the pixel having the same position as x_0 in the output signal.

x_1	x_2	x_3
x_4	x_0	x_5
x_6	x_7	x_8

Fig. 1a: The neighboring pixels of the actually processed pixel x_0

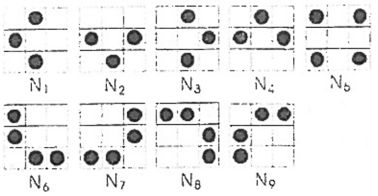

Fig. 1b: Pixel Patterns

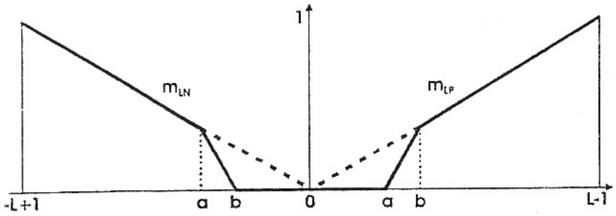

Fig. 2: Membership function m_{LP}. Parameters a and b are appropriate constant values

This value is determined by the following relationship:

$$y_0 = x_0 + \Delta y \tag{2}$$

where Δy is determined thereinafter (see eq. (5)). Let the rulebase deal with the pixel patterns $\mathbf{N}_1, \ldots, \mathbf{N}_9$ (see Fig. 1b.)

The value y_0 can be calculated, as follows [4]:

$$\lambda = MAX\{MIN\{m_{LP}(\Delta x_j) : x_j \in N_i\}; i = 1,\ldots,9\} \tag{3}$$

$$\lambda^* = MAX\{MIN\{m_{LN}(\Delta x_j) : x_j \in N_i\}; i = 1,\ldots,9\} \tag{4}$$

$$\begin{aligned} \Delta y &= (L-1)\Delta\lambda \\ y_0 &= x_0 + \Delta y \end{aligned} \tag{5}$$

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

Edge Detecting

Edge detection in an image is a very important step for a complete image understanding system. In fact, edges correspond to object boundaries and are therefore useful inputs for 3D reconstruction algorithms. The proposed fuzzy based edge detection [5] can very advantageously be used for this purpose.

Let $x_{i,j}$ be the pixel luminance at location $[i,j]$ in the input image. Let us consider the group of neighboring pixels which belong to a 3x3 window centered on $x_{i,j}$.

The output of the edge detector is yielded by the following equation [5]:

$$\begin{aligned} z_{i,j} &= (L-1)MAX\{m_{LA}(\Delta y_1), m_{LA}(\Delta y_2)\} \\ \Delta y_1 &= |x_{i-1,j} - x_{i,j}| \\ \Delta y_2 &= |x_{i,j-1} - x_{i,j}| \end{aligned} \tag{6}$$

where $z_{i,j}$ is the pixel luminance in the output image and m_{LA} is the used membership function (see Fig. 2). Pixels $x_{i-1,j}$ and $x_{i,j-1}$ are the luminance values of the left and the upper neighbor of the pixel at location $[i,j]$.

The fuzzy based technique compared to the classical methods provided better results with less (very small) processing time. Figs. 4-6 show an example for the filtering and edge detection results. In Fig. 4 the original photo corrupted by noise can be seen, Fig. 5 presents the filtered image of Fig. 4, while in Fig. 5 the results of the edge detection can be followed.

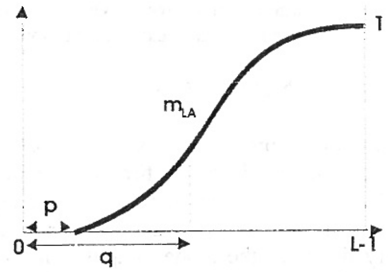

Fig. 3: Membership function m_{LA}. Parameters a and b are appropriate constant values

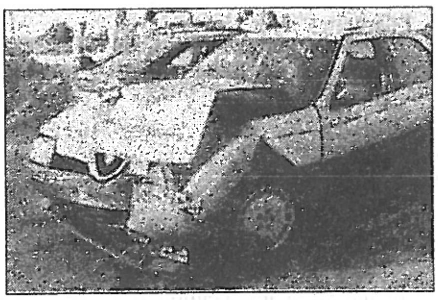

Fig. 4: Original photo of a crashed car corrupted by noise

Fig. 5: Fuzzy-filtered image of the photo in Fig. 4

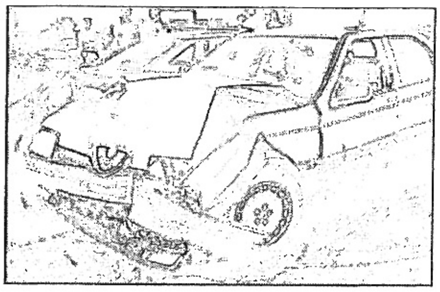

Fig. 6: Image of the photo in Fig. 5 after fuzzy based edge detection

III. EPIPOLAR GEOMETRY

Consider the case of two perspective images of a scene illustrated by Fig. 7. The 3D point **M** is projected to point $\mathbf{m}_1$ in the left image and $\mathbf{m}_2$ in the right one. Let $\mathbf{C}_1$ and $\mathbf{C}_2$ be the centers of projcction of the left and right cameras, respectively.

Points $\mathbf{m}_1$ in the first image and $\mathbf{m}_2$ in the second image are the imaged points of the point **M** of the 3D space. Points $\mathbf{e}_1$ and $\mathbf{e}_2$ are the so-called *epipoles*, and they are the intersections of the line joining the two cameras $\mathbf{C}_1$ and $\mathbf{C}_2$ and the two image planes. The plane formed by the three points $[\mathbf{C}_1\mathbf{M}\mathbf{C}_2]$ is called *epipolar plane*. The lines $\mathbf{l}_{m1}$ and $\mathbf{l}_{m2}$ are called *epipolar lines* and are formed when the epipoles and image points are joined. Point $\mathbf{m}_2$ is constrained to lie on the epipolar line $\mathbf{l}_{m1}$ of point $\mathbf{m}_1$. This is called *epipolar constraint*. Epipolar line $\mathbf{l}_{m1}$ is the intersection of the epipolar plane mentioned above with the second image plane Image2. This means that image point $\mathbf{m}_1$ can correspond to any 3D point on the line $[\mathbf{C}_1\mathbf{M}]$ and that the projection of $[\mathbf{C}_1\mathbf{M}]$ in the second image Image2 is the line $\mathbf{l}_{m1}$. All epipolar lines of the points in the first image pass through the epipole $\mathbf{e}_2$ and form thus a pencil of planes containing the baseline $[\mathbf{C}_1\mathbf{C}_2]$. The above definitions are symmetric, in a way such that the point of $\mathbf{m}_1$ must lie on the epipolar line $\mathbf{l}_{m2}$ of point $\mathbf{m}_2$.

An important practical application of epipolar geometry is to aid the search for corresponding points, reducing it from the entire second image to a single epipolar line. The epipolar geometry can easily be found from a few point correspondence [7][8].

Let $\mathbf{m}_1 = (x, y, z)^t$ be the homogeneous coordinates of a point in the first image and $\mathbf{e}_1 = (u, v, w)$ be the coordinates of the epipole of the second camera in the first image. The epipolar line through $\mathbf{m}_1$ and $\mathbf{e}_1$ is represented by the vector

$$\mathbf{l}_{m2} = (a,b,c)^t = \mathbf{m}_1 \times \mathbf{e}_1 \qquad (7)$$

The mapping $\mathbf{m}_1 \rightarrow \mathbf{l}_{m2}$ is linear and can be represented by a 3x3 rank 2 matrix **C**:

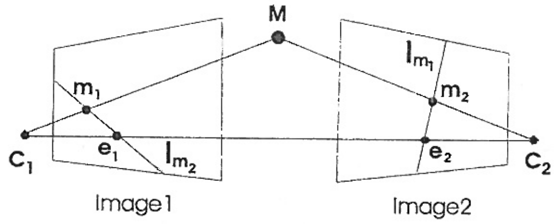

Fig. 7: Illustration of the epipolar geometry

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} yw - zv \\ zu - xw \\ xv - yu \end{pmatrix} = \begin{pmatrix} 0 & w & -z \\ -w & 0 & u \\ z & -u & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \qquad (8)$$

The mapping of epipolar line $\mathbf{l}_{m2}$ from Image1 to the corresponding epipolar line $\mathbf{l}_{m1}$ in Image2 is a collineation defined on the 1D pencil of lines through $\mathbf{e}_1$ in Image1. Let **A** be one such collineation, i.e. $\mathbf{l}_{m1} = \mathbf{A}\ \mathbf{l}_{m2}$. Since **A** has eight degrees of freedom and we only have five constraints, it is not fully determined. Nevertheless, the matrix $\mathbf{F} = \mathbf{AC}$ is fully determined [7][8]. **F** is called *fundamental matrix*. We get

$$\mathbf{l}_{m1} = \mathbf{ACm}_1 = \mathbf{Fm}_1 \qquad (9)$$

It is a fact that all epipolar lines in the second image pass through $\mathbf{e}_2$ for all transferred $\mathbf{l}_{m1}$.

$$\mathbf{e}_2^T . \mathbf{l}_{m1} = 0 . \qquad (10)$$

F defines a bilinear constraint between the coordinates of the corresponding image points. If $\mathbf{m}_2$ is the point in the second image corresponding to $\mathbf{m}_1$, it must lie on the epipolar line $\mathbf{l}_{m1}$.

$$\mathbf{l}_{m1} = \mathbf{Fm}_1 \qquad (11)$$

and hence

$$\mathbf{m}_2^T . \mathbf{l}_{m1} = 0 . \qquad (12)$$

The epipolar constraint can therefore be written as

$$\mathbf{m}_2^T \mathbf{Fm}_1 = 0 \qquad (13)$$

Linear Solution for the Fundamental Matrix

Each point match gives rise to one linear equation in the unknown entries of matrix **F**. The coefficients of this equation can easily be written in terms of the known coordinates of $\mathbf{m}_1$ and $\mathbf{m}_2$. Specifically, the equation corresponding to a pair of points $\mathbf{m}_1$ and $\mathbf{m}_2$ will be

$$xx'f_{11} + xy'f_{12} + xf_{13} + yx'f_{21} + yy'f_{22} + yf_{23} + x'f_{31} + y'f_{32} + f_{33} = 0 \quad (14)$$

where the coordinates of $\mathbf{m}_1$ and $\mathbf{m}_2$ are $(x, y, 1)'$ and $(x', y', 1)'$, respectively. Combining the equations obtained for each match gives a linear system that can be written as $\mathbf{Aw} = \mathbf{0}$, where $\mathbf{w}$ is a vector containing the 9 coefficients of $\mathbf{F}$ and each row of $\mathbf{A}$ is built of the coordinates $\mathbf{m}_1$ and $\mathbf{m}_2$ of a single match. Since $\mathbf{F}$ is defined only up to an overall scale factor, we can restrict the solution for $\mathbf{w}$ to have norm 1. We usually have more than the minimum number (8) of points, but these are perturbed by noise so we will look for a least squares solution [8]:

$$\min_{\|\mathbf{w}\|=1} \|\mathbf{Aw}\|^2 \quad (15)$$

As $\|\mathbf{Aw}\|^2 = \mathbf{w}^T\mathbf{A}^T\mathbf{Aw}$, we have to find the eigenvector associated with the smallest eigenvalue of the 9x9 symmetric, positive semidefinite normal matrix $\mathbf{A}^T\mathbf{A}$. However, this formulation does not enforce the rank constraint, so a second step must be added to the computation to project the solution $\mathbf{F}$ onto the rank 2 subspace. This can be done by taking the Singular Value Decomposition (SVD) of matrix $\mathbf{F}$ and setting the smallest singular value to zero. Basically, SVD decomposes any real valued matrix $\mathbf{F}$ in the form of

$$\mathbf{F} = \mathbf{Q\,D\,R} \quad (16)$$

where $\mathbf{D}$ is diagonal and $\mathbf{Q}$ and $\mathbf{R}$ are orthogonal matrices. Setting the smallest diagonal element of $\mathbf{D}$ to 0 and reconstituting gives the desired result.

IV. IMAGE POINT MATCHING

First the vertex correspondences are determined which is followed by the determination of the edge correspondences. This latter can be based on the comparison of a well-defined small region around the analyzed image point with the corresponding regions around each of the candidate image points in the other image. For this purpose we need several images taken from different camera positions. If the angle between the camera positions is relatively small then the corresponding points can be calculated automatically with high reliability in each image. As we had seen in the previous section the searching procedure can be reduced to 1D (along a line) with the help of the determined corresponding epipolar line. First we have to find the most characteristic image points. These points are the corners of the analyzed object. Corners can be effectively detected with the help of a fuzzy based corner detector [9]. For each detected corner we have to determine the corresponding epipolar line. Then we assign the corner points of this line. Thereinafter the fuzzy measure of the differences of the environment of the so gotten points are minimized by fuzzy based searching algorithm. The same procedure is applied to edge points.

V. ESTIMATION OF THE PERSPECTIVE PROJECTION MATRIX

There exists a collineation, which maps the projective space to the camera's retinal plane: 3D to 2D. The coordinates of a 3D point $\mathbf{M} = [M_X, M_Y, M_Z]^T$ (determined in an Euclidean world coordinate system) and the retinal image coordinates $\mathbf{m} = [m_x, m_y]^T$ (see Fig. 8.) are related by the following equation:

$$\begin{bmatrix} m_xW \\ m_yW \\ W \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & 1 \end{bmatrix} \begin{bmatrix} M_X \\ M_Y \\ M_Z \\ 1 \end{bmatrix} \quad (17)$$

where W is a scale factor, $\mathbf{m} = [m_x, m_y, 1]^T$ and $\mathbf{M} = [M_X, M_Y, M_Z, 1]^T$ are the homogeneous coordinates of points $\mathbf{m}$ and $\mathbf{M}$, and $\mathbf{P}$ is a 3×4 matrix representing the collineation 3D to 2D. One parameter of $\mathbf{P}$ can be fixed ($l = 1$). $\mathbf{P}$ is called the perspective projection matrix. Values $a, b, c, d, e, f, g, h, i, j, k$ are the elements of the projection matrix $\mathbf{P}$. The optical axis passes through the center of projection (camera) $\mathbf{C}$ and is orthogonal to the retinal plane. The focal length f_l of the camera is also shown, which is the distance between the center of projection and the retinal plane. Even if only the perspective projection matrix $\mathbf{P}$ is available, it is possible to recover the coordinates of the optical center or camera [7]. It is clear that

$$W = iM_X + jM_Y + kM_Z + 1 \quad (18)$$

From equation (17) we can compute the coordinates of point $\mathbf{m}$ (m_x, m_y), as follows:

$$m_x = \frac{aM_X + bM_Y + cM_Z + d}{W} \quad (19)$$

$$m_y = \frac{eM_X + fM_Y + gM_Z + h}{W} \quad (20)$$

$$M_Xa + M_Yb + M_Zc + d + 0e + 0f + 0g + 0h - M_Xm_xi - M_Ym_xj - M_Zm_xk = m_x \quad (21)$$

$$0a + 0b + 0c + 0d + M_Xe + M_Yf + M_zg + h - M_xm_yi - M_ym_yj - M_zm_yk = m_y \quad (22)$$

All together we have eleven unknowns – these are the elements of the projection matrix that means that we need six points to determine the projection matrix.

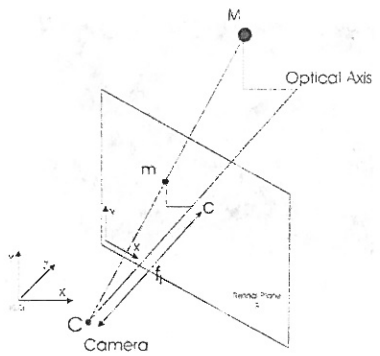

Fig. 8: Perspective projection – illustration of points $\mathbf{M}=[X,Y,Z]$ and its projection $\mathbf{m}=[x,y]$ in the retinal plane R.

After substitutions and equivalent transformations we get the following equations and matrices:

$$\mathbf{AQ} = \mathbf{B} \tag{23}$$

where matrix **A** and vectors **Q** and **B** are described thereinafter. See eqs. (25), (26), (27), (28), (29).

$$\mathbf{A}^T\mathbf{AQ} = \mathbf{A}^T\mathbf{B} \tag{24}$$

From eq. (23) the projection matrix can be obtained:

$$\mathbf{Q} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{B} \tag{25}$$

$$\mathbf{A} = \begin{bmatrix} M_{1X} & M_{1Y} & M_{1Z} & 1 & 0 & 0 & 0 & 0 & -m_{1x}M_{1X} & -m_{1x}M_{1Y} & -m_{1x}M_{1Z} \\ 0 & 0 & 0 & 0 & M_{1X} & M_{1Y} & M_{1Z} & 1 & -m_{1y}M_{1X} & -m_{1y}M_{1Y} & -m_{1y}M_{1Z} \\ & & & & & & \vdots & & & & \\ M_{nX} & M_{nY} & M_{nZ} & 1 & 0 & 0 & 0 & 0 & -m_{nx}M_{nX} & -m_{nx}M_{nY} & -m_{nx}M_{nZ} \\ 0 & 0 & 0 & 0 & M_{nX} & M_{nY} & M_{nZ} & 1 & -m_{ny}M_{nX} & -m_{ny}M_{nY} & -m_{ny}M_{nZ} \end{bmatrix} \tag{26}$$

The first two lines of matrix **A** correspond to points $\mathbf{M}_1$ and $\mathbf{m}_1$, the second two lines correspond to points $\mathbf{M}_2$ and $\mathbf{m}_2$, etc. With the help of these points we can compute the elements of the projection matrix **P**:

$$\mathbf{P} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & 1 \end{bmatrix} \tag{27}$$

$$\mathbf{Q} = \begin{bmatrix} a & b & c & d & e & f & g & h & i & j & k \end{bmatrix}^T \tag{28}$$

$$\mathbf{B} = \begin{bmatrix} m_{1x} & m_{1y} & m_{2x} & m_{2y} & . & . & . & m_{nx} & m_{ny} \end{bmatrix}^T \tag{29}$$

The elements of vector **B** are the coordinates of points $\mathbf{m}_i$, where $i=1...n$

VI. EXAMPLE

In this section a simple example is presented to illustrate the introduced modeling procedure. Fig. 9a and Fig. 9b show the input images which are corrupted by noise. Images represented by Figs. 10a and 10b are the filtered ones of the above mentioned input images. Fig. 11a and Fig. 11b represent some corresponding points as example of the matching algorithm's functionality, Fig 12a and 12b show the results after the edge detection and finally, Fig 13 illustrates the resulted 3D model.

VII. CONCLUSIONS

This paper introduces a method for 3D model reconstruction from images taken from different camera positions. The method applies fuzzy filtering, fuzzy edge detection and a new method based on the recent results of epipolar geometry. With the help of this technique 3D models can be produced without any external (human) intervention, thus it can be advantageous in many 3D applications and in computer vision.

Fig 9a. : The first input picture from camera position C1

Fig 9b. : The second input picture from camera position C2

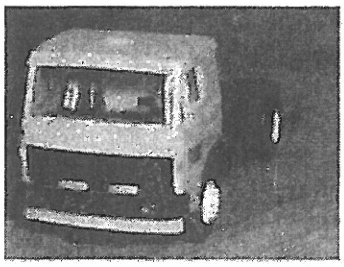

Fig 10a : Picture 9a after filtering

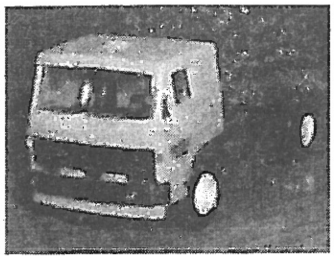

Fig 10b : Picture 9b after filtering

Fig 11a: An example of some calculated matching points (camera position C1)

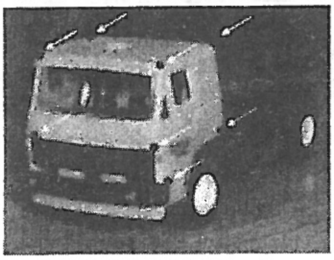

Fig 11b An example of some calculated matching points (camera position C2)

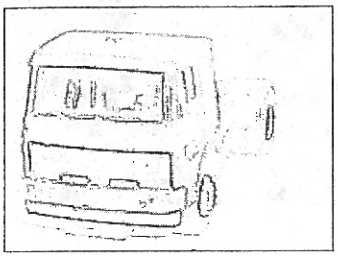

Fig 12a: The picture represented by Fig 10a after edge detection

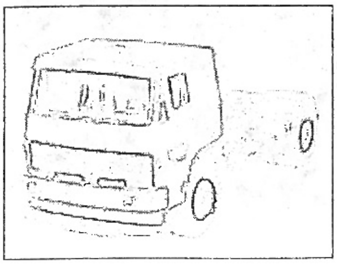

Fig 12b: The picture represented by Fig 10b after edge detection

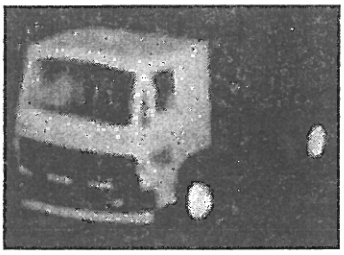

Fig. 13. : The resulted 3D object model from a different camera position

ACKNOWLEDGMENT

This work was sponsored by the Hungarian Fund for Scientific Research (OTKA T 035190) and the Hungarian-Portugese Intergovern. S&T Cooperation Programme (P-24/03).

REFERENCES

[1] C. Taylor, P. Debevec and J. Malik, "Reconstructing Polyhedral Models of Architectural Scenes from Photographs," *Computer Vision - ECCV'96*, Lecture Notes in Computer Science, Vol. 1065, Vol. II, pp 659-668, 1996.

[2] R. Hartley, "Euclidean reconstruction from uncalibrated views," in : J.L. Mundy, A. Zisserman, and D. Forsyth (eds.), *Applications of Invariance in Computer Vision*, Lecture Notes in Computer Science, Vol. 825, Springer-Verlag, pp. 237-256, 1994.

[3] R. Hartley and A. Zisserman, "*Multiple View Geometry in Computer Vision*," Cambridge University Press, 2000.

[4] 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.

[5] 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.

[6] Russo, F., "Edge Detection in Noisy Images Using Fuzzy Reasoning," *IEEE Transactions on Instrumentation and Measurement*, Vol. 47, No. 5, Oct. 1998, pp. 1102-1105.

[7] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig", *Computer Vision - ECCV'92*, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 563-578, 1992.

[8] R.Hartley. "In defence of the 8-point algorithm," *In Proc. of the 5th International Conference on Computer Vision*, Cambridge, Massachusetts, USA, pp. 1064-1070, June 1995.

[9] Rövid, A., A.R. Várkonyi-Kóczy, "Corner Detection in Digital Images Using Fuzzy Reasoning," *submitted to 2nd IEEE Int. Conf. on Computational Cybernetics*, August 30-Sep. 1, 2004 Vienna, Austria.