PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 43, NO. 3, PP. 153–162 (1999)

INTELLIGENT STEREO VISION SYSTEM FOR ROBOTIC APPLICATIONS

Ferenc TÉL and Béla LANTOS

Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
H–1117 Budapest, Pázmány Péter sétány 1/D, Hungary
e-mail: tel@opsys.hu, lantos@iit.bme.hu

Received: November, 1999

Abstract

This paper describes an intelligent vision method, which is capable of reconstructing the robot environment. Algorithms and their implementation are presented for localising known objects in the scene and determine 3D Euclidean transformation (the relative position and orientation) between recognised objects. This can be done by reconstructing the projective 3D world of the scene and involve the metrical constraints with an object recognition method. The resulting displacement information can be used as the input of the intelligent robot control system and the calibrated virtual reality.

Keywords: stereo image processing, uncalibrated cameras, object recognition.

1. Introduction

More and more robotic applications require an intelligent extension. A possible solution for such a system is depicted in *Fig. 1*. This system is able to interact with the environment, including both humans (by VR) and objects (by vision). This paper describes the stereo vision part of the intelligent control system of the Puma 560 robot and the dextrous hand developed at the Technical University of Budapest [1].

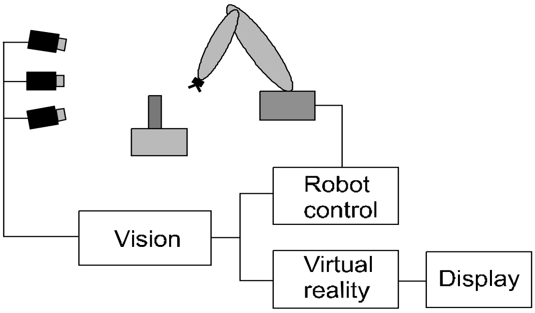

Fig. 1. Main blocks of the intelligent robot control system

The chapters are organised as follows. Chapter 2 gives an overview of the previous works, Chapter 3 describes our stereo system in detail. Some result is presented in Chapter 4. We draw the conclusion in Chapter 5.

2. Previous Work

There are many tasks in high level robot control that require information about the environment, such as path planning, collision avoidance. One source of these data can be a vision system.

There are two types of vision, an active and a passive one. The active systems usually use laser systems and triangulation in order to extract 3D metrical information about the environment. In case of passive vision system the traditional solution is the calibrated stereo rig. The drawback of this type of system is that it requires an accurate calibration method. Hence these types of solutions do not give the desired flexibility or sometimes cannot be applicable (e.g. in case of autozoom or moving cameras).

Newer stereo systems use uncalibrated cameras to avoid the drawbacks of the calibration process. As it was shown by FAUGERAS [2], in this case only the projective structure of the surrounding world can be reconstructed, but this does not contain any metrical information. Many researches work with these types of uncalibrated systems and show different properties of the resulting projective geometrical world (epipolar geometry, fundamental matrix, trifocal tensor) and entities in this world (points, lines, curves) [3], [4], [5], [6], [7], [8], [9].

It is also proven that constraints must be introduced in order to build the Euclidean world [10], [11], [12].

Another type of vision research aims to recognise complex objects in the scene. These methods usually use single image about the scene and model information stored in the database. The matching between the scene and database information is generally solved via probabilistic methods [13], [14], neural networks or in other ways. But these 2D object recognition methods are not used to calculate the geometrical relationships (position, orientation) between 3D objects.

Intelligent control systems usually require both types of information. High level control tasks such as grasp planning demand not only the positions of some entities in the space but also the types of them (grasp planning). The main contribution of our stereo method is that it tries to integrate the results of the projective reconstruction of the scene and the (2D) object recognition such that the object recognition is used to supply the additional constraints required by the projective. This yields that the output of the stereo system is the type of the recognised objects and the position and orientation with respect to each other.

3. Stereo System

The stereo system consists of the low-level image processing in the views, the reconstruction of the projective structure, the model based object recognition and the calculation of the Euclidean transformation between recognised objects.

3.1. Low-Level Image Processing

The first step during the processing is the low-level image processing. The goal of this step is to achieve the feature detection on camera images.

The edges and corner points are detected using the Canny and Harris detector, respectively. Then the detected corner points are attached to the edges if it can be done or dropped otherwise. The subsequent processing steps require the accurate determination of the corner points, but corner-edge relationships can be poor in case of junctions because of Gaussian filtering. Hence we develop a new method that is capable of localizing the correct position of the corner within a window around the initial guess.

After the refinement and edge segmentation process the edges are classified by a parameter fitting algorithm into the following classes: junction, point set, line, 2nd order curve.

The system produces the graph of the features. A node of the graph contains a feature, the branch of the graph holds the relation (2D position, orientation) between them. Hence in the subsequent steps instead of using predefined high level feature types we use (and store) only a relationship between the features within a certain distance.

3.2. Reconstruction of the Projective Structure

There are several methods to solve the projective reconstruction problem from point correspondences. The projective relationship between the images are expressed by the epipolar geometry [3], [4], [5] in two view case or by the trifocal tensor [6], [7], [8], [9] in three view case. Assuming the pinhole camera model, the imaging equations can be written in linear form [15],

$$\lambda_{ij}\mathbf{m}_{ij} = \mathbf{P}_j\mathbf{M}_i, \tag{1}$$

where $\mathbf{m}_{ij} = [u, v, w]^T$ is the projective coordinates of the image point, $\mathbf{P}_j = [p_{ij}]$ is the unknown projection matrix with size 3×4 for the jth camera, $\mathbf{M}_i = [x, y, z, t]^T$ contains the unknown projective coordinates of the 3D point.

Our system uses the method developed by MOHR et al. [11], [12] to rebuild the 3D projective structure (but any other method could be used too). This method eliminates the scaling factor and directly minimises the resulting equation by Levenberg-Marquardt method [16] . This is similar to the nonlinear camera calibration process described in [15], but for this case the 3D projective coordinates ($\mathbf{M}_i$) are also unknown.

3.3. Object Recognition

A 2D object recognition system is used to localize the known objects in the scene. In order to represent the occlusions of the features and to handle the uncertainty during the recognition process, probabilistic description was chosen. A similar method can be found in [13], [14]. The method can recognize only those objects that are predefined in a model database. The building of the database is supposed to be off-line. An object model in the database contains:

- Limited number of 2D views about the object. A view of the objects contains features and relations between them resulting by feature extraction method.
- The 3D coordinates of the same features (mostly as point coordinates) in the local, object based coordinate system. These are used during the calculation of the Euclidean transformation.

For example in *Fig. 2* the 2D feature (O) is the image of an 3D corner point of the cube. In the 2D world it is a junction in which there are 3 intersecting lines. The 2D information in the model database about feature O consists of the type of the feature (junction), viewpoint independent attributes (feature has 3 intersecting lines), the coordinates of the feature in each image ($[u_i, v_i]$ image coordinate pairs) and the uncertainty of the detection (e.g. response of the detector or empirical deviation).

The 3D information is the Euclidean 3D coordinates of the feature in an object relative coordinate system ($[x, y, z]$ coordinate triplet). This object relative 3D coordinate system can be chosen arbitrarily but must be predefined in the database. A possible selection is denoted by white coordinate vectors, in that system O has the $[0, 0, 0]$ coordinate triplet.

The recognition process tries to find the most probable configuration representation of objects with pairing the features of the scene and the model database. Note that in this case not only the database but also the scene consists of more than one view about the object.

The recognition requires the description of the quality of the matching by defining a cost function. Using the Bayesian estimation theory the quality measure can be associated with the probability $P(A|Q, T)$, where A denotes the hypothesis that the object of the model database is present in the scene, Q contains the pairings between views and T denotes viewpoint transformations. These are multidimensional joint probabilities, hence some simplification condition is used and the detection and the presence of the features are handled as independent events (feature independence simplification) [8], [9]. This yields that the probabilities can be approximated as the product of lower dimensional distributions.

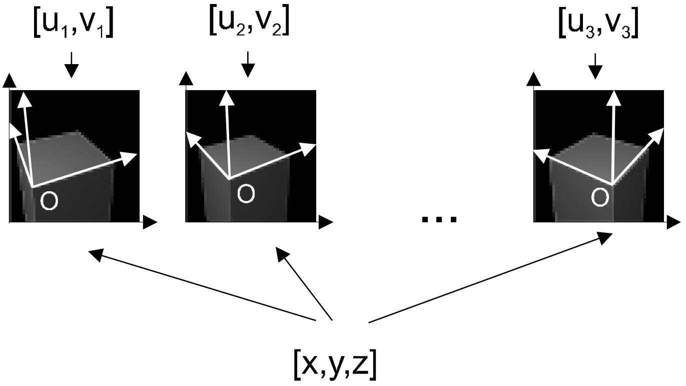

Fig. 2. Two types of information (coordinate representation) of the features in the model database

The features are represented by type t_{ij}, attribute vector (representation of independent properties of the given feature) α_{ij}, mean of the position μ_{ij}, covariance of the position Σ_{ij}. Using these, the described probability can be approximated.

As can be seen the described method requires the calculation of the transformation between views in the scene and/or the model. This is supposed to be affine. The transformation can be described by 6 parameters $\mathbf{t} = [a_t, b_t, c_t, d_t, x_t, y_t]^T$. The pose of the features can be characterized by $\mu_k = [t_x, t_y, \beta, \gamma]^T = [t_x, t_y, \theta, s]^T$, where t_x, t_y is the position, $\theta = \mathrm{atan}(\gamma/\beta)$ is the orientation, $s = \sqrt{\beta^2 + \gamma^2}$ is the scaling. Introducing β and γ yields that the effect of the transformation can be written in linear form:

a)
$$\mu_k' = \begin{bmatrix} t_x & t_y & 0 & 0 & 1 & 0 \\ 0 & 0 & t_x & t_y & 0 & 1 \\ \beta & \gamma & 0 & 0 & 0 & 0 \\ 0 & 0 & \beta & \gamma & 0 & 0 \end{bmatrix} \begin{bmatrix} a_t \\ b_t \\ c_t \\ d_t \\ x_t \\ y_t \end{bmatrix},$$

b)
$$\mu_k' = \begin{bmatrix} a_t & b_t & 0 & 0 & x_t & 0 \\ c_t & d_t & 0 & 0 & 0 & y_t \\ 0 & 0 & a_t & b_t & 0 & 0 \\ 0 & 0 & c_t & d_t & 0 & 0 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ \beta \\ \gamma \\ 1 \\ 1 \end{bmatrix}. \tag{2}$$

To calculate the transformation and determine the possible pairings, prediction-verification with a tree-search method is applied. A new feature pair is added to the sufficient node of the tree, if the features have the same type, the distance between the feature's parameters is below a threshold, the transformation error is small and the insertion in the tree is consistent with the actual content of the tree.

The last step of the recognition is to search for the object that gives the best representation (have the greatest probability).

If the object is recognized, the local 3D Euclidean coordinates from the model database can be attached to the features (e.g. corner points). For the recognized objects, the same features in the database and in the image can be localized, hence the 3D coordinates of the image points in the scene view can be determined with respect to the object's own 3D Euclidean coordinate system.

3.4. Calculation of the Euclidean Transformation

Reaching this point of the processing the coordinates of the recognized objects are expressed in two coordinate systems. The first one is the projective system that is common for all the objects. The second one is the local Euclidean frame for each object. The relative Euclidean transformation between the objects in the scene can be computed using these two types of the coordinate representation as illustrated in *Fig. 3*. This calculation can be achieved in two phases.

The first phase of the displacement computation is the calculation of the transformations (collineations, ${}^{(X)}\mathbf{W}$) between the object frames and the common projective system. Using the two types of the coordinate representation of an object, the transformation can be written in the following matrix form,

$$\mathbf{W}\mathbf{M}_{P,i} - \lambda_i \mathbf{M}_{E,i} = \mathbf{0}, \tag{3}$$

where $\mathbf{M}_{P,i}$ and $\mathbf{M}_{E,i}$ represent the projective and Euclidean coordinates respectively, the λ_i is the scaling factor for the ith point. Collect all of these equations into one system of equations $\mathbf{Ax} = \mathbf{b}$ (n is the number of the corresponding points)

$$\begin{gathered}
A = \begin{bmatrix}
\mathbf{M}^T_{P,1} & 0^T & & -M_{E,1,1} & 0 & & & \\
0^T & \mathbf{M}^T_{P,1} & & -M_{E,1,2} & 0 & & & \\
0^T & 0^T & \dots & -M_{E,1,3} & 0 & & \ddots & \\
0^T & 0^T & & -M_{E,1,4} & 0 & & & \\
\mathbf{M}^T_{P,2} & 0^T & & 0 & M_{E,2,1} & 0 & & \\
& \mathbf{M}^T_{P,2} & & & -M_{E,2,2} & 0 & & \\
& & \ddots & & & & \ddots & -M_{E,n,4} \\
0 & & & & & & 0 & 1
\end{bmatrix}, \\
\mathbf{x} = [w_{11}\ w_{12}\ w_{13}\ w_{14}\ \cdots\ w_{44}\ \lambda_1 \cdots \lambda_n]^T, \\
\mathbf{b} = [0 \cdots 0\ 1]^T,
\end{gathered} \tag{4}$$

where the size of **A** is $(4n+1)\times(n+16)$ and the last equation states that $\lambda_n = 1$ (this is not a restriction, while all equations can be divided by $\lambda_n \neq 0$). The size of the unknown vector **x** is $(n+16)$ and the size of vector **b** is $(4n+1)$. The linear estimation of the solution can be determined in least squares (LS) sense as

$$\mathbf{x}_{n+16} = (\mathbf{A}^T\ \mathbf{A})^{-1}_{n+16,n+16}\mathbf{A}^T_{n+16,4n+1}\mathbf{b}_{4n+1}. \tag{5}$$

Similar methods are given in [18].

The second phase is the calculation of the displacement. Let's suppose we want to determine the transformation between objects **A** and **B**. Using the graph in the *Fig.* 3 the Euclidean coordinates of the point M of object **B** in the frame of object **A** can be expressed in two ways. The first one is the direct application of the collineation ${}^{(A)}\mathbf{W}$, the second one is to transform the coordinates into the frame of object **B** using ${}^{(B)}\mathbf{W}$ then apply ${}^{(AB)}\mathbf{D}$. In equation form

$${}^{(A)}\mathbf{W}\ {}^{(B)}\mathbf{M}_{P,i} = {}^{(AB)}\mathbf{D}\ {}^{(B)}\mathbf{W}\ {}^{(B)}\mathbf{M}_{P,i}, \tag{6}$$

where ${}^{(B)}\mathbf{M}_{P,i}$ are the projective coordinates of the ith point of object B. This equation can be rewritten into the form ${}^{(A)}\mathbf{V}_i = {}^{(AB)}\mathbf{D}\ {}^{(B)}\mathbf{V}_i$. Rescaling all of the $\mathbf{V}_i$ in order to represent Euclidean coordinates the displacement can be calculated in closed form using quaternions [19].

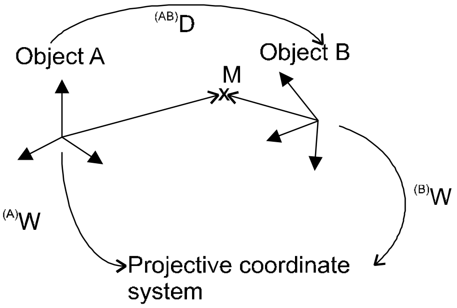

Fig. 3. Calculation of the Euclidean transformation

In order to put all the results together, a refinement step is also developed. The relation can be written into a similar form as in the *Eq.* (6). Expanding the resulting equation yields

$$\lambda\sum_{k=1}^{4}{}^{(A)}\mathbf{W}_{j,k}\mathbf{M}_{P,i,k} - \sum_{k=1}^{4}\sum_{l=1}^{4}{}^{(AB)}\mathbf{D}\ {}^{(B)}\mathbf{W}_{k,l}\mathbf{M}_{P,i,l}, \quad j = 1\ldots 4, \quad i = 1\ldots n. \tag{7}$$

The unknowns are the elements of the ${}^{(AB)}\mathbf{D}$, ${}^{(A)}\mathbf{W}$, ${}^{(B)}\mathbf{W}$ and λ. Constraints must be introduced for ${}^{(AB)}\mathbf{D}$ to hold the desired form. Using the properties of the rotation matrix in the displacement, the constraints are the following:

$$\sum_{k=1}^{3} {}^{(AB)}\mathbf{D}_{j,k}\ {}^{(AB)}\mathbf{D}_{l,k} = 0,\ \ j,l = 1\ldots 3 \text{ (orthogonality of the rows)} \tag{8}$$

$$\sum_{k=1}^{3} {}^{(AB)}\mathbf{D}_{k,j}\ {}^{(AB)}\mathbf{D}_{k,l} = 0,\ \ j,l = 1\ldots 3 \text{ (orthogonality of the columns)} \tag{9}$$

$$\sum_{k=1}^{3} {}^{(AB)}\mathbf{D}^2_{4,k} = 0 \ \text{ (first three elements of the last row are zero)} \tag{10}$$

$${}^{(AB)}\mathbf{D}_{4,4} - 1 = 0 \ \text{ (scaling is one).} \tag{11}$$

These systems of equations can be minimised with Levenberg-Marquardt method. The initial values of the unknowns are the results of the LS estimation and the calculation based on quaternions.

4. Results

The implemented software is based on the developed algorithms and robust numerical methods. We tested the accuracy of the algorithms with simulated data in order to evaluate robustness under different noise conditions. The simulated scene was 2 cubes with the size 400 mm and three cameras view the scene from approximately 2.5 m. Both of the cubes consist of 16 points. Noise was added to the camera matrices in order to simulate the distortions and other effects. Gaussian noise with mean value 0 and deviation 1 is added to the 3D model database to model the inaccuracy in the recognition process. (This means approximately 2–3 mm errors in the 3D model coordinates). During the simulation the mean value of the pixel noise was 0, the deviation is changed between 0 and 1, so the maximum error was approximately 3–4 pixel.

The errors shown in *Fig. 4* are average values of the results of some simulations. The first part of the figure shows the errors in rotation around x, y and z axis with solid, dotted and dashed lines, respectively. The angles are given in grades. The second part contains the translation error in the x, y, and z directions again with solid, dotted and dashed lines, respectively. The third part is the scaling, which is 1 for the Euclidean case. As can be seen, the errors are below 1 degree in angles, the position error usually remains under 5 mm–1.5 cm. This means that the relative errors are approximately 1%, where the relative error means the ratio of the error (of position) and the overall size of the scene (approximately 2 m). The scaling is very close to 1. The peaks at the pixel noise level 2 are the result of the effect of those cases, when the Levenberg-Marquardt method has not been converged properly.

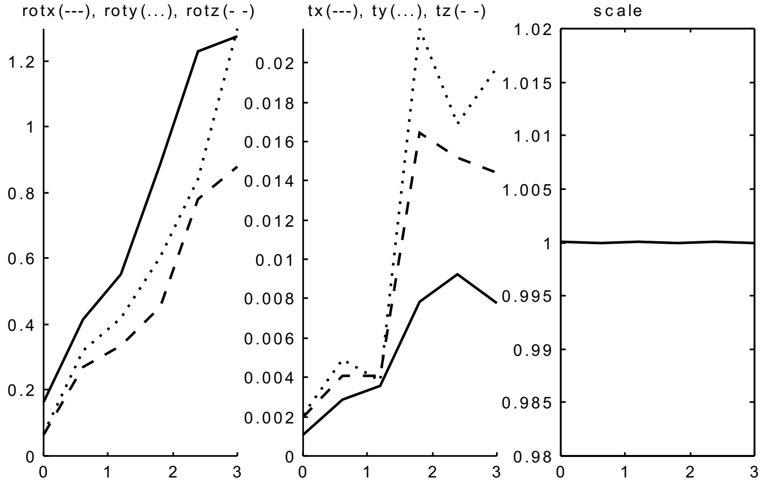

Fig. 4. Average displacement error vs. pixel noise

5. Conclusion

This paper has shown the 'visual' part of an intelligent robot control system. Algorithms are presented for localising known objects in the scene and determine 3D Euclidean transformation (relative position and orientation) between recognised objects.

The software system consists of the low-level image processing in the views, the reconstruction of the projective structure, the model based object recognition and the calculation of the Euclidean transformation between recognised objects. The implementation is based on the developed algorithms and robust numerical methods. The presented results demonstrate the robustness and the accuracy of the algorithms under different noise conditions.

The uncalibrated stereo method can be applied to collect information about the surrounding world. The resulting displacement information serves as the input of the intelligent robot control system and the calibrated virtual reality.

Acknowledgement

Support for the research of stereo image processing and calibrated virtual reality for robots is provided by the Hungarian National Research programs under grant No. FKFP 0417/1997 and OTKA T 029072.

References

[1] LANTOS, B. – KLATSMÁNYI, P. – LUDVIG, L. – TÉL, F.: Intelligent Control System of a Robot with Dextrous Hand. *Proc. IEEE International Conference on Intelligent Engineering Systems INES'97*, Budapest, 1997. pp. 129–134.

[2] FAUGERAS, O. D.: What Can be Seen in Three Dimensions with an Uncalibrated Stereo Rig. *Proc. 2*$^{\text{nd}}$ *European Conf. Computer Vision*, 1992.

[3] LUONG, Q. T. – FAUGERAS, O. D.: The Fundamental Matrix: Theory, Algorithms and Stability Analysis. *Int. Journal of Computer Vision*, **17**, No. 1, (1995), pp. 43–76.

[4] HARTLEY, I. R.: Projective Reconstruction from Line Correspondences. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994, pp. 903–907.

[5] AVIDAN, S. – SHASHUA, A.: Tensor Embedding of the Fundamental Matrix, Institute of Computer Science, *Post-ECCV SMILE Workshop*, 1998, Frieburg, Germany. Springer LNCS series, Vol. 1506. Internet: http://www.cs.huji.ac.il/∼shashua/)

[6] SHASHUA, A. – WERMAN, M.: Fundamental Tensor: On the Geometry of Three Perspective Views. *Inst. of Computer Science, Hebrew University*, 1997, Internet: http://www.cs.huji.ac.il/labs/vision/biblo.html

[7] SHASHUA, A.: The Fundamental Construct of Multiple View Geometry and its Applications. *Inst. of Computer Science, Hebrew University*, 1997, Internet: http://www.cs.huji.ac.il/∼shashua/

[8] HARTLEY, I. R.: Lines and Points in Three Views and the Trifocal Tensor. *Int. Journal of Computer Vision*, **22** (2), (1997), pp. 125–140.

[9] TORR, P. H. S. – ZISSERMAN, A.: Robust Parametrization and Computation of the Trifocal Tensor. *Journal of Image and Vision Computing*, **15** (1997), pp. 591–605, Internet: http://imogen.robots.ox.ac.uk:20000/∼geoff/papers.cgi)

[10] DEVERNAY, F. – FAUGERAS, O. D.: From Projective to Euclidean Reconstruction, *INRIA*, France, 1995.

[11] BOUFAMA, B. – MOHR, R. – VEILLON, F.: Euclidean Constraints for Uncalibrated Reconstruction. *Proc. 4th International Conference on Computer Vision*, Berlin, Germany, pp. 466–470, 1993, Internet: http://www.inrialpes.fr/movi/pub/Publications/en/par_annee.html)

[12] MOHR, R. – QUAN, L. – VEILLON, F.: Relative 3D Reconstruction Using Multiple Uncalibrated Images. *Int. Journal of Robotic Research*. MIT Press **14**, No. 6, (1995), pp. 619–632.

[13] WELLS, W. M.: Statistical Object Recognition. Ph.D. Thesis, MIT, 1993.

[14] POPE, A. R.: Learning to Recognize Objects in Images: Acquiring and Using Probabilistic Models of Appearance, Ph.D. Thesis, University of British Columbia, Canada, 1995.

[15] FAUGERAS, O. D.: Three Dimensional Computer Vision (A Geometric Viewpoint). MIT Press, 1992.

[16] PRESS, W. H. – FLANNERY, B. P. – TEUKOLSKY, S. A. – VETTERLING, W. T.: Numerical Recipes in C. Cambridge University Press, 1988.

[17] WELCH, G. – BISHOP, G.: An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, via Internet, 1997.

[18] CSURKA, G. – DEMIRDJIAN, D. – HORAU, R.: Finding the Collineation between two Projective Reconstructions. *INRIA RR* No. 3468, 1998,
Internet: http://www.inrialpes.fr/movi/people/Horaud/Radu-publications.html

[19] LANTOS, B.: 3D Image Processing Methods. (Manuscript, in Hungarian) Technical University of Budapest, 1994.