ERSO - Acquisition, Reconstruction and Simulation of Real Objects[1]

Holger Diener, Ullrich Köthe
Fraunhofer-Institut für Graphische Datenverarbeitung
Joachim-Jungius-Straße 11
D-18059 Rostock, Germany
holger@egd.igd.fhg.de
ulli@egd.igd.fhg.de

Marcus Schreyer
Institut für Photogrammetrie und Kartographie
TU Darmstadt
Petersenstraße 13
D-64287 Darmstadt, Germany
marcus@gauss.phgr.verm.tu-darmstadt.de

Bernhard Ristow
Fraunhofer-Institut für Graphische Datenverarbeitung
Rundeturmstraße 6
D-64283 Darmstadt, Germany
ristow@igd.fhg.de

Ulf Stelbe
Zentrum für Graphische Datenverarbeitung e.V. Rostock
Joachim-Jungius-Straße 11
D-18059 Rostock, Germany
ust@rostock.zgdv.de

***Abstract* – A basic system for acquisition, reconstruction and simulation of real objects (ERSO) is presented. Our approach is to combine the knowledge of different research areas as photogrammetric, computer graphics, and computer vision to develop new techniques for generating three-dimensional models from images. The first two paragraphs give an introduction and overview of the system architecture. The following paragraphs describe in more detail the different parts of the reconstruction process.**

I. INTRODUCTION

In many application areas, such as architecture, virtual scenes, medicine, and photogrammetry, the acquisition of an abstract description of three-dimensional objects is an important topic. A common approach to this problem is based on photogrammetric methods, which require a large amount of manual work. Since efficient visualization and animation systems are increasingly available, the effort necessary for data acquisition turns more and more into a limiting factor for the use of these systems.

In the ERSO project our goal is the development of a basic system that integrates data acquisition with the reconstruction of scenes and objects. The system uses graphical interactive data handling methods and simulation tools to help architects, developers of virtual worlds or similar user groups to generate a 3D model from images of a real object and to use this model in their own applications.

Within the scope of the ERSO project new methods for the automation of data handling and acquisition processes will be developed. For this, automatic and interactive algorithms will be combined in order to maximize the efficiency of future intelligent vision systems.

II. SYSTEM ARCHITECTURE

To generate three-dimensional models from images an adequate pool of data is necessary, and it is important to get geometric data of the object as well as specific surface attributes (e.g. texture). Therefore, we developed a knowledge base to enable non-experts in photogrammetry to take pictures sufficient for the reconstruction process.

There are no specific hardware requirements for the camera. Amateur cameras can be used as well as professional measurement cameras. With the camera calibration module it is possible to calculate camera parameters (inner image orientation) and eliminate distortion in the images caused by non-professional cameras.

Reconstructing the 3D coordinates of a single object point needs at least two observed images of the object point and the orientation between these images. In our case the orientations between two arbitrary images are unknown. So the first step in the reconstruction process is to estimate the (exterior) image orientations. This is possible if there are more than eight correspondences of previously detected image points. Before starting the estimating process correspondences are found by a semi-automatic matching procedure.

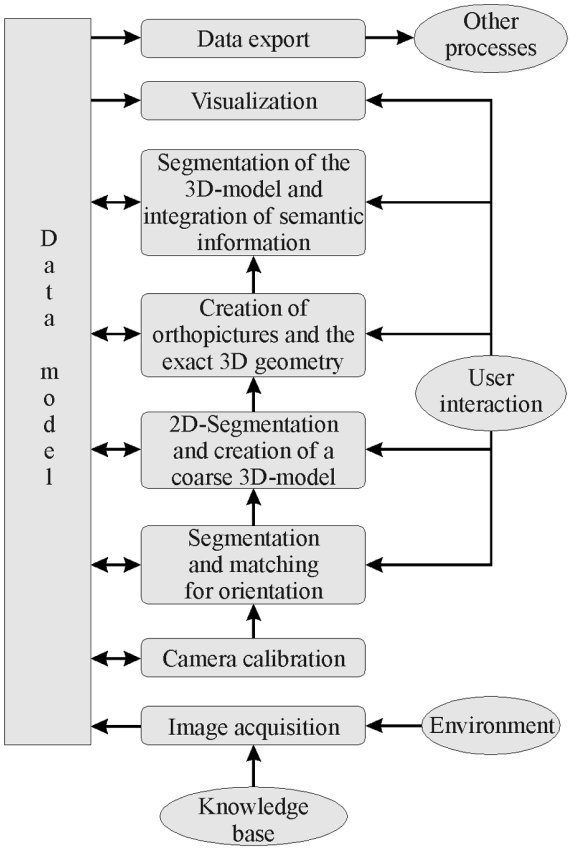

Fig. 1. System architecture

[1] The project is sponsored by the Volkswagen-Stiftung

To get more primitives for feature matching each image is segmented into points, edges and regions and their topological connections are stored in feature adjacency graphs. The construction of a coarse 3D-model is possible by matching these graphs.

The next step in the reconstruction process creates orthophotos (images that would be obtained by an orthogonal projection) and depth maps from different oriented views which include the exact 3D geometry of the object. Now it is possible to generate a polygonal model and texture mappings to display and control the reconstruction result.

Many applications need semantic information, therefore we are developing tools to support user defined selection and grouping within the polygonal model.

III. CAMERA CALIBRATION USING THE PHOTOGRAMMETRIC BUNDLE METHOD

The objective of camera calibration is to determine the interior geometric quantities of a camera used for image acquisition. Among others this is a main premise in order to achieve high accuracy and precision for reconstructing objects. No matter of using digital or analog devices there can usually be two approaches distinguished to calibrate a camera:

Test range calibration: In this case one or multiple pictures are taken of a specialized target area consisting of well known signal object points. Since this method requires an expert based target area and leads to an additional image scene beside the main application data this method has become quiet unpopular.

Self-calibration: The principle of this method is to include the task of calibration within the job of evaluating all the object points which are necessary to reconstruct the relevant object of the application. Usually this method leads to a higher accuracy in reference to the object points at the cost of a small set of additional parameters which have to be estimated. This method is widely used in many applications in industrial close range photogrammetry and shall be considered as the favorite approach for camera calibration in our project.

The backbone of the calibration procedure is the photogrammetric bundle method which has been fully developed for the case of metric cameras. It plays a critical role in the exploitation of the best potential accuracy and precision in photogrammetric applications such as object reconstruction. The basic idea behind it is to determine a set of interior as well as exterior camera parameters and a set of object points based on the relationship of multiple overlapping photographs. It is based on a mathematical camera model that describes the relationship between the 3D-objectspace and the 2D-imagespace.

A. Mathematical Camera Model

The functional model consists of the so called collinearity equations which are derived from the perspective transformation. They rely on the basic assumption that the object point, the perspective center of the camera and the corresponding image point form a straight line. In reference to each observed image point (x'_{ij}, y'_{ij}) of an image I'_j corresponding to an object point (X_i, Y_i, Z_i) this model is written in the following form:

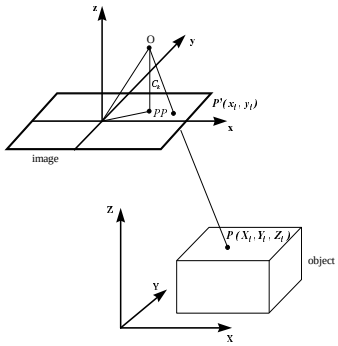

Fig. 2.: Collinearity principle

$$x_{ij}' = x_{ij} - x_0 - \Delta x'_{ij} = -c_k \frac{r_{11}(X_i - X_{0j}) + r_{21}(Y_i - Y_{0j}) + r_{31}(Z_i - Z_{0j})}{r_{13}(X_i - X_{0j}) + r_{23}(Y_i - Y_{0j}) + r_{33}(Z_i - Z_{0j})} \tag{3.1}$$

$$y_{ij}' = y_{ij} - y_0 - \Delta y'_{ij} = -c_k \frac{r_{12}(X_i - X_{0j}) + r_{22}(Y_i - Y_{0j}) + r_{32}(Z_i - Z_{0j})}{r_{13}(X_i - X_{0j}) + r_{23}(Y_i - Y_{0j}) + r_{33}(Z_i - Z_{0j})},$$

with the parameters of the interior and exterior orientation of the image I'_j:

x_0, y_0 and c_k :	Interior orientation (Principal point and principal focus)
X_{0j}, Y_{0j}, Z_{0j} :	Center of perspective of the exterior orientation
$\omega_j, \phi_j, \kappa_j$:	Angles of the orthogonal rotation matrix $R = (r_{kl})$ of the exterior orientation

$\Delta x'$ and $\Delta y'$ represent geometric correction terms that are applied to the image coordinate measurements to model and compensate the systematic errors due to the lens, solid state sensors, film deformation etc. These terms are essential in the case of camera calibration and usually different types of independent additional functions are applied to model the systematic errors. The set of parameters used for these functions are called the additional parameters (APs) of the bundle method. In order to make the bundle method applicable to a wide range of amateur cameras we decided to provide the following three independent types of functions to model systematic distortions.

Radial Lens Distortion: To describe radial lens distortion which has usually the greatest impact we preferred to use a simplified odd power polynomial proposed by Fryer & Brown [4] which leads to the following components in the image coordinate system:

$$\Delta x'_{rad} = (x - x_0) \cdot \left(K_3 r^2 + K_5 r^4 + K_7 r^6\right)$$
$$\Delta y'_{rad} = (y - y_0) \cdot \left(K_3 r^2 + K_5 r^4 + K_7 r^6\right) \tag{3.2}$$

In this approach *r* refers to the radial distance between the principal point and the image point. K_3, K_5 and K_7 represent the additional parameters of the radial distortion function.

Decentering Lens Distortion: In this case we also followed a simplified form derived by Fryer & Brown [4]

with the additional parameters P_1 and P_2:

$$\Delta x'_{asymm} = P_1 \cdot \left(r^2 + 2x'^2\right) + 2P_2 \cdot x'y'$$
$$\Delta y'_{asymm} = 2P_1 \cdot x'y' + P_2 \cdot \left(r^2 + 2y'^2\right), \quad (3.3)$$

where $\Delta x'_{asymm}$ and $\Delta y'_{asymm}$ are the distortion values in the image coordinates respectively and *r* with its components *x'* and *y'* is the radial distance to the principal point.

CCD Array Distortion: The application of a digital sensor requires an additional term to compensate a shear and tilt in the CCD array. In this context we followed an AP set suggested by Beyer [1] that has proven to be effective:

$$\Delta x'_{CCD} = -(x - x_0) \cdot s_x + (y - y_0) \cdot a$$
$$\Delta y'_{CCD} = (x - x_0) \cdot a, \quad (3.4)$$

where *a* represents a scale in *x* and s_x a shear of the CCD-array.

In summary we have expanded the conventional bundle method by three additive sets of distortion terms with the additional parameters K_3, K_5, K_7 of the radial distortion, P_1, P_2 of the decentering distortion and s_x and *a* of the CCD array distortion.

B. Parameter Estimation

The first step in estimating the unknown orientation parameters as well as the object point coordinates is to establish the pair of equations formulated in (3.1) for each pair of image coordinates (x'_{ij}, y'_{ij}) observed on each image I'_j. Due to the non-linearity of the equation (3.1) a linearization is necessary and each observation which is accompanied by a measurement error gets an additional correction term v_i. By assuming initial values for the unknown parameters and uncorrelated image coordinate observations we obtain to the following Gauss-Markov Model:

$$\vec{l}_k + \vec{v}_k = A_k \cdot d\vec{x}_k \quad \text{and} \quad \Sigma_{kll} = \sigma_{k0}^2 \cdot P^{-1} \quad (3.5)$$

The design matrix *A* is of dimension $n \times m$ (*n* is the number of observations, *m* the number of unknown parameters) with $n \geq m$ and $\text{rank}(A) = m$. *P* describes the weight matrix, Σ_{kll} the covariance matrix and σ_{k0} the standard deviation of unit weight for the observations vector $\vec{l}_k$. The iterative process is described by the index *k*. By applying the least square minimization the unknown correction parameters are given by the normal equations:

$$N_k \cdot d\vec{x}_k = \vec{n}_k \text{ with } N_k = A_k^T \cdot P \cdot A_k \text{ and } \vec{n}_k = A_k^T \cdot P \cdot \vec{l}_k \quad (3.6)$$

The precision of the estimated parameters is controlled by the covariance matrix:

$$\Sigma_{kll} = \sigma_{k0}^2 \cdot (A_k^T \cdot P \cdot A_k)^{-1} \text{ with } \sigma_{k0}^2 = \frac{v_k^T \cdot P \cdot v_k}{n - m} \quad (3.7)$$

To obtain the solution for the desired parameters this leads to the Gauss-Newton iteration where in each iteration equation (3.6) has to be solved:

$$\vec{x}_{k+1} = \vec{x}_k + d\vec{x}_k \quad , k = 0,1,\ldots, \quad (3.8)$$

with $d\vec{x}_k$ derived from (3.6). The iteration is stopped if the norm of the correction vector falls below a certain constant value close to zero.

IV. SEGMENTATION AND MATCHING FOR ORIENTATION OF THE IMAGES

In the process of reconstruction each image have orientation parameters in objectspace (exterior orientation) and cameraspace (interior orientation) to perform the perspective transformation. We assume that these parameters are unknown. Therefore, they have to be estimated in a separate procedure. In this procedure we detect corners in the images and find their correspondences in other images. Finally, with all correspondences the orientation of the images can be estimated.

A. Corner Detection

There are several image filters to detect corners in images. We decided to use two types of corner detectors which are basically different. The principle idea of the SUSAN operator [13] is to compare the intensities of all pixel in a circular mask to the intensity of the center pixel. In the case of a corner most of the pixel in the mask differ from the center pixel.

One major property of the SUSAN corner detector is the good performance in the presence of noise. Other corner detectors make use of the image derivatives which are strongly influenced by noise.

The second corner detector is a statistically motivated approach [3]. Therefore, the average of the gradient image are used to maximize the following function:

$$\frac{\det C(\vec{x})}{\text{trace } C(\vec{x})} \rightarrow \max \text{ with } C(\vec{x}) = \begin{pmatrix} \overline{g}_x^2 & \overline{g}_x \overline{g}_y \\ \overline{g}_x \overline{g}_y & \overline{g}_y^2 \end{pmatrix} \quad (4.1)$$

Once we have detected some corners in the images with one of the two methods described above we try to match them semi-automatically.

B. Point Matching Algorithm

When working on dense image sequences the cross correlation is successfully used to match features between images. The cross correlation considers the similarity of the gray values in a window around the corners. In the case of wide baselines between the images, as in our case, the

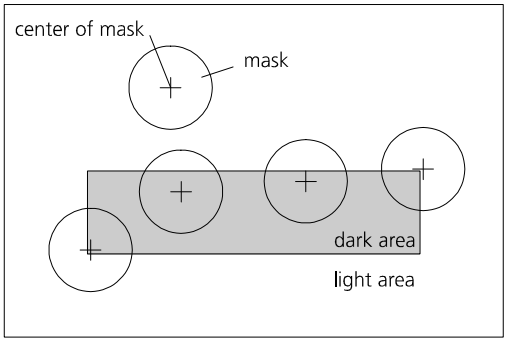

Fig. 3. The SUSAN Principle

cross correlation between the corners in different images can not be used. The perspective transformation changes the gray values in a window so that only a translation of a window does not determine the correspondence in an other image (see Fig. 4).

Therefore, we use a semi-automatic matching procedure. In a first step we select manually the necessary number of points to estimate the epipolar geometry. Based on this estimation we use the A* algorithm to find additional correspondences. The cost function is a combination of the distance to the epipolar line and the direction of the flow vector in comparison to the flow vector of the nearest found correspondence.

C. Estimation of the orientation

First the relative orientation between all image pairs is estimated using the eight point algorithm [2]. The object point and the projected object points in image one and two define a plane. This plane intersects the image plane. The intersection line is called the epipolar line. This fact can be expressed by the following formula:

$$\vec{u}_2^T \cdot A_2^T \cdot TR \cdot A_1 \cdot \vec{u}_1 = \vec{u}_2^T \cdot F \cdot \vec{u}_1 = 0 \qquad (4.2)$$

The matrix A contains the parameters of the perspective transformation, T and R the parameters of the orientation in space. With at least eight correspondences we can solve a least square minimization problem to estimate the parameters of the exterior orientation. With all relative orientations we build a reconstruction tree and try to find the path with the minimum error of back projection. Using this path all relative orientations are transformed into one coordinate system. Finally, all interior and exterior parameters are optimized in a bundle block procedure.

V. 2D-SEGMENTATION AND CREATION OF A COARSE 3D-MODEL

A. Segmentation

So far, all processing steps were solely based on points in 2D and 3D. However, this is not sufficient to obtain the higher level descriptions we want to generate. We must detect a richer set of features within each image and store a description of their different properties.

A feature adjacency graph contains points, edges, and regions and represents their topological and geometrical relationships.

There is a wealth of segmentation algorithms in the literature. Therefore, we first had do study several algorithms to assess their suitability for the ERSO system. Somewhat to our surprise, relatively simple algorithm outperformed much more complicated ones. Nevertheless, no single algorithm is sufficiently accurate to be used in isolation. To combine several algorithms, we use an improved version of the method described in Köthe [7]. The basic algorithms are the following:

Difference of Exponential Edge Detector: We convolve an image *I* with an exponential filter *E* and take the difference to the original:

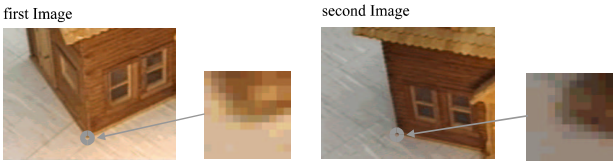

Fig. 4. Windows around a corner in two different images

$$DoE(x,y) = I(x,y) - (E * I)(x,y)$$
$$\text{with } E(x,y) = N \exp\left((|x|+|y|)/\sigma\right).$$

where σ denotes the scale of the operator. In this image, zero crossings are detected to yield an edge image. Since the edges in this image still contain gaps, they will only be used to initialize a region growing algorithm.

Scale selection: Using a scale selection procedure as described in Köthe [8], we locally compute plausible scales for edges and regions. By comparing the edge detection results at different scales with the local appropriate scales, we can merge multiple edge images into one.

Seeded Region Growing: Starting from a set of seed regions, in this case obtained by inverting and eroding the edge image, each pixel not already labeled is assigned to the region into which it fits best. By using a global priority queue, it is ensured that good fits are processed first, so that two regions will always meet at pixels that don't fit well into either one.

The obtained segmentation is subjected to certain topoplogical transformations to avoid well known connectivity problems. From this we finally extract the feature adjacency graph which represents the position and neighborhood information of the features as well as any additional information (such as gray levels) that will be useful in the subsequent feature matching step.

B. Feature Matching

To combine the feature adjacency graphs of all images we decided to use the graduated assignment algorithm for graph matching by Gold and Rangarajan [5]. This algorithm compares properties of nodes and edges in a graph to calculate an approximate solution of the matching problem in an iterative process and is described briefly.

The results of comparing nodes *a* and *i* from different graphs and all their adjacent edges are stored in the element w_{ai} of the weight matrix *W*. The approach of the algorithm is to minimize the cost function

$$E(M) = -\tfrac{1}{2} \sum_{ai} w_{ai}(M) \cdot m_{ai} \qquad (5.1)$$

with a doubly stochastic matrix $M = (m_{ai})$ that means the elements of every row and every column of *M* are non-negative and sum up to 1. An element m_{ai} gives the probability of a correct correspondence between the node *a* and the node *i*. As written in (5.1) the comparing results are combined with *M* and are calculated as follows:

$$w_{ai}(M) = C_n(a,i) + \sum_{b \in nb(a)} \sum_{j \in nb(i)} m_{bj} \cdot C_e\big((a,b),(i,j)\big)$$

with *(a,b)* the edge from *a* to *b*, *nb(a)* all neighbor nodes

of *a*, C_n and C_e the compare functions of nodes and edges respectively.

The development of a Taylor series of (5.1) shows that minimizing the cost function *E* means to maximize the function

$$E_{ass}(M) = \sum_{ai} w_{ai}(M^{(0)}) \cdot m_{ai} \tag{5.2}$$

with a fixed matrix $M^{(0)}$. This is an assignment problem and it is solved for each iteration by the softassign algorithm described in [5]. The inner loop for solving (5.2) is either stopped on convergence of M or after a fixed number of iterations.

In the outer loop a control parameter for the softassign algorithm is increased and the process is stopped after a given number of iterations.

VI. CREATION OF ORTHOPICTURES AND THE EXACT 3D-GEOMETRY

The objective of Facets Stereo Vision (FAST-Vision), a method developed by Wrobel 1987 [14], is to reconstruct simultaneously the object surface *Z(X,Y)* and its optical density *G(X,Y)* with *X, Y* as independent surface coordinates of a regular quadratic *XY*-raster. Usually, *G(X,Y)* can be regarded as an orthophoto of the object. The relationship between a point on a surface (surfel) and its pixel representations in the images P', P'';... can be described with regard to geometric and radiometric characteristics. The geometric relation between a surfel and its imagepoints is given by the well-known perspective equations already described in (3.1). To describe the radiometric relationship between the object and its images we chose to use linear transfer functions T', T'',... that have proven to be effective:

$$G(X,Y,Z) = T'(G'(x',y')) = T''(G''(x'',y'')) = \ldots \tag{6.1}$$

with *G', G'',...* as the density values of the images *P', P'',...* Sensors that represent different camera models can be chosen instead of (3.1) and the transfer function does not necessarily has to be linear.

Since (6.1) includes the object models as non-linear parameters the development of a Taylor series to the first order is applied which leads to:

$$\begin{aligned} T'(G'(x',y')) = {} & G(X^0,Y^0) + dG(X^0,Y^0) + \\ & \frac{\partial G(X^0,Y^0)}{\partial X} X_Z' \cdot dZ + \frac{\partial G(X^0,Y^0)}{\partial Y} Y_Z' \cdot dZ \end{aligned} \tag{6.2}$$

with $X_Z' = \dfrac{X^0 - X_0}{Z^0 - Z_0}$ and $Y_Z' = \dfrac{Y^0 - Y_0}{Z^0 - Z_0}$.

This equation is fundamental in the process of FAST-Vision. It shows on the left hand side the density of an imagepoint corresponding to a certain surfel and provides a linear relationship to the improvement of the surface model *dZ* and the density model *dG*. To reduce the number of unknown parameters the quadratic *XY*-raster is independently resolved in a grid consisting of quadratic facets for the geometric and the density model. Each facet within the raster defines the region of validity for a local interpolation

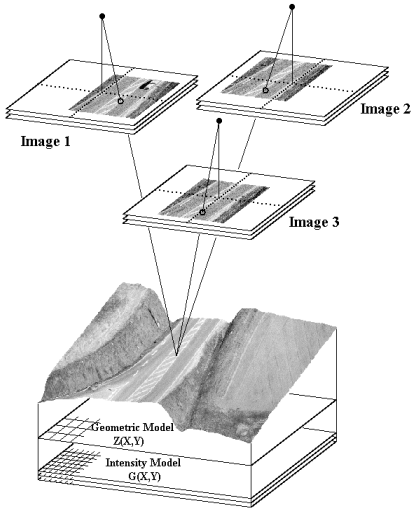

Fig. 5. FAST-Vision concept

function used for the description of the geometric model and the density model respectively. This leads to the following differentials for the models:

$$\begin{aligned} dZ(X,Y) &= \sum_k \sum_l a_{kl}(X,Y) \cdot dZ(X,Y) \\ dG(X,Y) &= \sum_k \sum_l \alpha_{kl}(X,Y) \cdot dG(X,Y) \end{aligned} \tag{6.3}$$

So far we prefer to use a bilinear interpolator but basically all interpolation functions that represent the geometric or density model in an appropriate way can be chosen for (6.3). The unknown parameters that basically have to be estimated are the nodes of the facets belonging to the interpolation functions *dZ* and *dG* respectively .

In order to reconstruct an object with the derived method at least a second image is required. The values associated with image P' in (6.2) have to be replaced with those associated with the images *P'',P'''* etc. The evaluation of (6.2) for all pixels from all pictures leads to a linear Gauss-Markov model analog to (3.5). This overdetermined problem is again solved by a least square minimization of the residuals which leads to the well-known normal equations already described in (3.6). The non-linear nature of FAST-Vision with its linearization in (6.2) requires a Gauss-Newton iteration to obtain the solution for the geometric model and the density model simultaneously.

VII. SEGMENTATION OF THE 3D-MODEL AND INTEGRATION OF SEMANTIC INFORMATION

First we have to generate a polygonal model of the object geometry which is defined by several oriented depth images.

A. Generating a Polygonal Model

Depth images are typically considered as a set of points, we consider the range images as partially defined surfaces. By interpolating between the grid points the surface can easily be completed continuously. Now we use the implicit function $h(\vec{x}) = 0$ for defining the surface [11] where $\vec{x}$ is a point in 3D space. For the definition of $h(\vec{x})$ we define

for each depth image f_i a function $g_i(\vec{x})$ which measures the signed distance between a point in space and the interpolated surface in $f_i(u,v)$. The corresponding point in the depth image is found by projecting the point $\vec{x}$ onto the parameter grid of the range image. A positive distance indicates that the point lies between the observer and the surface, a negative indicates that it lies below the surface. The synthesis of several depth images is achieved by combining the functions $g_i(\vec{x})$ as follows:

$$h(\vec{x}) = \max_i \left\{ g_i(\vec{x}) \right\} \tag{7.1}$$

The point $\vec{x}$ is called visible if $h(\vec{x}) \geq 0$. This criterion is used for each voxel in a volume which contains the complete object. After the generation of the volumetric model, the marching cube algorithm with a look-up table that resolves ambiguous cases (see Fig. 6) can be applied to generate a polygonal representation [9] [10]. The accuracy of this polygonal mesh is improved by moving the vertices of the mesh onto the surface implicitly defined by the registered range images. The point is moved between the voxels of the cube until the zero crossing of the signed distance function is found. The exact coordinates of the moved vertices are interpolated in the range images. This step is similar to a ray casting algorithm which is combined with the visibility criterion. This way, the full accuracy of the scanning device is exploited at all vertices of the mesh, and concave object shapes are also modeled correctly. The amount of polygons can be significantly reduced by applying a polygon reduction [12] [6].

B. Integration of Semantic Information

The polygonal model does not contain any semantic information such as this polygon belongs to a window of a house. Therefore parts of the polygonal model must be clustered and represented by a semantic description. It is necessary to use a scene description which allows a semantic tree where the nodes describe semantic parts of the object. These nodes consist of the semantic information such as „door“ and the geometric representation (a subset of polygons) in the polygonal model. Because of the complexity of semantic information we offer some tools to group parts of the geometric model and give them a semantic label. There are manipulators which allow to group polygons manually or to select one and search automatically for other polygons which are similar to the selected. The similarity of the polygon may be the topological information, e.g. search for all quadrangles, the geometric information, e.g. search for all polygons with the same area, or the textural information, e.g. search for all red areas.

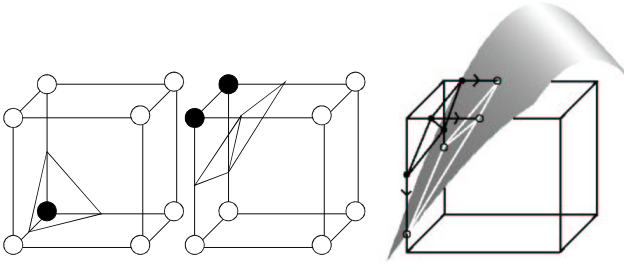

Fig. 6. Triangulation of a cube by applying Marching-Cubes and improving the mesh

VIII. CONCLUSION

The ERSO project will be sponsored until May 1998. Currently the modules are integrated in the user interface and a demonstration application is prepared.

IX. REFERENCES

[1] H.A. Beyer, "Geometric and radiometric Analysis of a CCD camera based on a Photogrammetric Close range system.", Thesis, Institute for Geodesy and Photogrammetry, Zurich, 1992, pp. 121-123.

[2] O. Faugeras, *Three Dimensional Computer Vision*. The MIT-Press Cambridge, Massachusetts, London, England, 1993.

[3] W. Foerstner, "A Feature Based Correspondence Algorithm for Image Matching", *Int. Arch. Photogramm. Remote Sensoring*, vol. 26, 1986, pp. 150-166.

[4] J. Fryer and D. Brown, "Lens Distortion for Close-Range Photogrammetry", *Photogrammetric Engineering and Remote Sensing*, 52(1), 1986, pp. 51-58.

[5] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.18, no.4, 1996, pp. 377-388.

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, „Mesh optimisation“, in *Proc. of SIGGRAPH '93*, Anaheim, California (1993), pp 19-26.

[7] U. Köthe, "Primary Image Segmentation", in *Sagerer et al. "Mustererkennung 1995", Proc. of 17. DAGM Symposium, Springer 1995*

[8] U. Köthe, "Local appropriate scale in morphological scale-space", in *Proceedings 4. European Conference on Computer Vision*, Springer 1996

[9] W.E. Lorensen and H.E. Cline, „Marching cubes: A high resolution 3d surface construction algorithm“, in *Proc. of SIGGRAPH '87*, Anaheim, California (1987), pp. 163-169.

[10] C. Montani, R. Scateni, and R. Scopigno, "A modified look-up table for implicit disambiguation of marching cubes", *Visual Computer* 10 (1994).

[11] P.J. Neugebauer, "Reconstruction of Real-World Objects via Simultaneous Registration and Robust Combination of Multiple Range Images", *International Journal of Shape Modeling*, Vol.3, No. 1&2 (1997), pp. 71-90.

[12] W. Schroeder, J. Zarge, and W. Lorensen, „Decimation of triangle meshes“, in *Proc. of SIGGRAPH '92*, Chicago, Illinois (July 1992), pp. 65-70.

[13] S.M. Smith and J.M. Brady, *A New Approach to Low Level Image Processing*. DRA Technical Report TR95SMS1c, 1995

[14] B. Wrobel, "Facets Stereo Vision (FAST Vision) - A New Approach to Computer Stereo Vision and to Digital Photogrammetry", FPPD 1987