RECOVERING 3-D OBJECT GEOMETRY USING A GENERIC CONSTRAINT DESCRIPTION

S.Weik, O.Grau
Institut für theoretische Nachrichtentechnik und Informationsverarbeitung
Universität Hannover - Germany
E.-Mail: weik@tnt.uni-hannover.de grau@tnt.uni-hannover.de

Commision V, Working Group 1

KEY WORDS: Close Range, Knowledge Base, Architecture, Reconstruction, Calibration, Graphics

ABSTRACT

A knowledge based approach for the surface reconstruction of buildings to be used in computer graphic applications is presented. Using a calibrated stereo camera pair, scene depth is estimated by correspondence analysis. To compensate for noisy and not dense depth maps we use a-priori knowledge about the scene to further increase the quality of the reconstruction results. Symbols are assigned to the image content which are used to establish an interpretation of the scene in form of a semantic net. Together with the scene description additional geometric constraints can be selected from a generic knowledge base. Each of these constraints describes a relationship either between parts of the model (e.g. the perpendicularity of two walls) or between the 3-D scene and extracted 2-D image features (e.g. edges or depth information). In the latter case 3-D edges and orientations of model parts are linked by constraints to the respective 2-D feature. Due to noisy data the resulting set of constraints normally lacks consistency. Numerical optimization is applied to solve this inconsistency, thus determining the model which best meets all imposed constraints. They are used both for restoring the object's geometry from the data and incrementally adding new camera viewpoints as parts to the scene description. In this mode the constraints are minimized by optimizing the camera pose and orientation.

1 Introduction

The visualisation of virtual environments, e.g. for driving simulators or architectural planning systems, ask for highly photorealistic 3-D models. The automatic or semi-automatic generation of these models is of current interest [1][5]. The data-driven approach [4] uses only the assumption of piecewise continuous surfaces and creates a triangular mesh from depth maps. Often models of well known regular objects like buildings appear unnatural due to errors in the restored object geometry. It is important for these models that certain expectations of human observers are met. Geometric distortions and symmetries are particularly critical. Inaccuracies are recognized immediately and appear to be disturbing.

On the other hand there are approaches to integrate explicit knowledge into the reconstruction process. The work presented here uses a model driven approach where the object properties are stored in an explicit knowledge base [6]. It is part of the scene interpretation system AIDA[1][2][3]. An interpretation identifies the objects and selects appopiate geometric constraints for surface reconstruction. In addition to these internal constraints the interpretation assigns image features like regions and contours to object parts. This measured data and the internal geometric constraints are used in a uniform scheme: The 3-D model under reconstruction is compared with the constraints and the back projected contours and regions using specific cost functions. The restoration of object geometry is usually not possible without conflicts. Therefore a numerical error compensation method is applied.

The constraint description in conjunction with the labeled data is a generic description of the objects in contrast to parameterized models. The latter models have a fixed topology and complexity. In an optimization process only size and orientation of the models are adapted to the data. The complexity of generic models is determined adaptively during the reconstruction.

The goal of the scene interpretation system AIDA is to perform most of the data labeling and the constraint assignment automatically. The surface reconstruction described in this contribution could also be part of a fully interactive system.

In this contribution the facilities of the system are shown for the application of modelling buildings. The model surfaces are approximated with polygonal meshes.

2 System Overview

The presented system AIDA (fig. 1) processes a sequence of stereo image pairs, taken with two cameras mounted on a rigid bar. One stereo pair is the input of the system for one sequential operation. In subsequent operational steps the following stereo pairs are integrated.

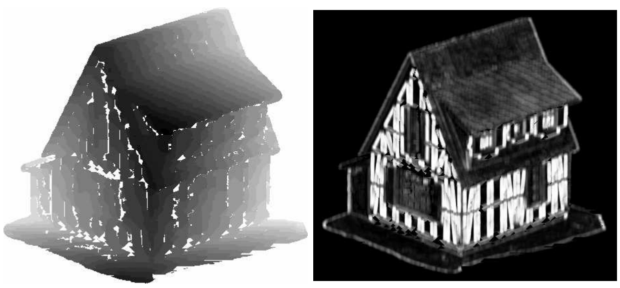

Figure 2: Gray coded depth- and certainty map of a building

In an initial processing step the system is calibrated using a regular calibration pattern of control points. The calibration estimates the internal camera parameters (internal camera orientation) and the relative external camera orientation using the method of Tsai [10]. With this information the input image pair is rectified to achieve epipolar geometry. In a next step a disparity map is calculated using a stereoscopic correspondence analysis based on the cross correlation of blockwise pixel luminance values. With the known relative camera orientation of the camera pair the disparity or parallax values are used for the calculation of (camera centered) depth values (fig. 2 left). During depth calculation an

[1]Automatic Image Data Analyzer

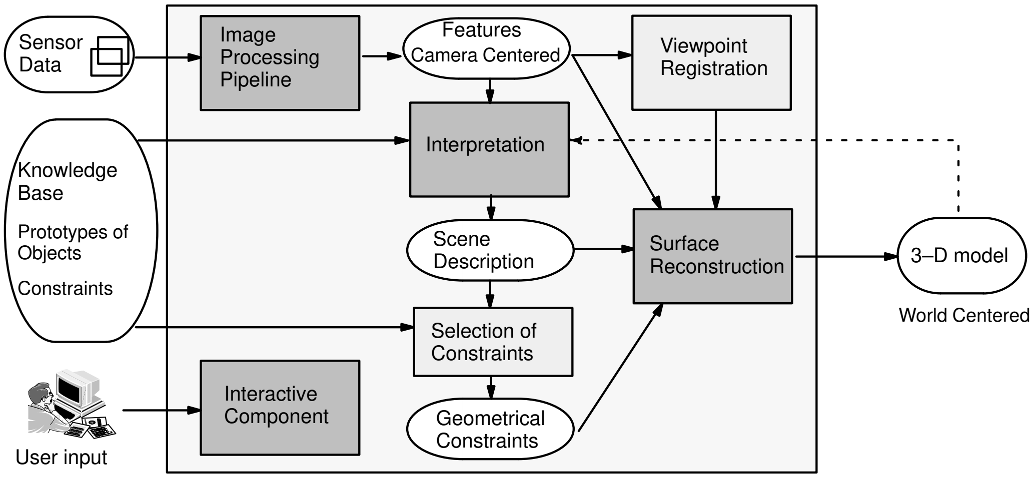

Figure 1: System overview

additional certainty map is created containing a reliability measure for the depth value. The certainty measurement is a combination of NCC[2] between left and right image and the image gradient (fig. 2 right).

In addition to the estimation of the depth maps, regions and contours are extracted. The segmentation into regions uses the criteria of the same orientation of surface points found in the depth map. The details of the stereo pair processing and the segmentation can be found in [8][4].

The central module of the system is the interpretation. It assigns a semantic meaning described in the knowledge base to the features extracted in the image processing pipeline before. The knowledge is formulated in a semantic net [6] and structured into three layers of abstraction (fig. 3): The top layer or scene layer describes the world in terms that are highly symbolic. The middle layer called world centered layer describes the appearance of the objects found in the model world in 3-D space and in absolute world coordinates. The bottom layer describes the objects appearance in camera centered coordinates, i.e. in 2-D space.

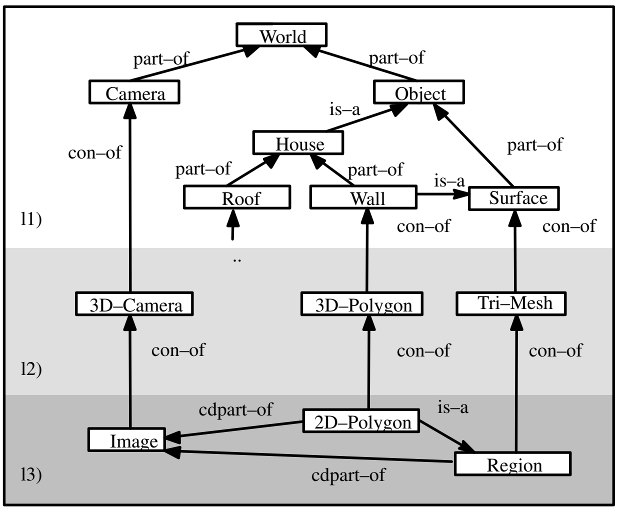

Figure 3: Example semantic net (part)

The objects are represented as nodes in the net. The nodes are connected via special edges or links: The *part-of* link enables an object decomposition. The *concrete-of (con-of)* link connects different layers of abstraction. The *is-a* link permits inheritance of attributes from general to specialised nodes. Another link is the *instance-of* link. It connects an instance node that was build up during the interpretation with its prototype in the knowledge base.

The creation of instances that describe the real scene is the goal of the interpretation. It assigns those node types to objects that are found in the scene, for example an instance of the node *House* in fig. 3. The process of the interpretation is described in [6] in detail. It is based on hypotheses and the their verification. In the beginning of the interpretation a hypothesis *House* is established. In the following the interpretation creates *sub-hypotheses* for obligate parts (like walls) of the house and tries to verify them in the image. After all obligate hypotheses have been verified the higher hypothesis *House* is validated.

From the resulting scene description geometrical constraints are selected. These are together with the measured features from the image processing pipeline the input of the surface reconstruction module, which is described in the following sections.

3 Surface and Camera Representation

To achieve our goal to optimize the resulting 3-D model according to some constraints, we need a scene representation that allows parts to be moved around under the influence of an global optimization algorithm.

As can be seen in figure 4, each part of the model (e.g. a wall or a roof) is represented as a plane in space. The 3-D model edges result from intersections between two neighbouring parts, thus leading to a polygon description of each model element. To control its position and orientation, each model part is assigned a local coordinate system. The origin t of the system is positioned above the center of the wall with the three axes $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$ spanning a righthanded coordinate system. The vector $(1, 1, 1)$ has the opposite direction of the surface normal. A local coordinate system for each part is necessary since the desired global optimization is sensitive to inhomogeneous coordinates. With the proposed local system all coordinates are treated with equal influence on optimization.

The scene description consists of elements of the actual model such as walls, as well as cameras. For camera representation the $cahv$-model is used [7]. There are six degrees of freedom for

[2] Normalized Cross Correlation

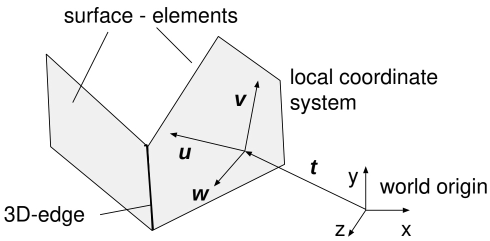

Figure 4: Used local coordinate system

the extrinsic parameters of a camera and three for each part of a model, which in the following is refered to as *model part* or *surface element*.

4 Constraint Description

We propose a number of different constraints that are intended to judge properties of man-made regular objects such that subjectively good-looking models result.

Every constraint is represented by a function c. The value of c corresponds to the deviation of the model from the respective restriction. An optimization module moves the parts to find the part positions which lead to the minimum sum of all constraint functions.

Two different kinds of constraints are used: *Internal* and *measurement constraints*. *Internal constraints* describe properties within objects, i.e. they relate model parts to each other (e.g. perpendicularities or parallelisms). They are described in section 4.1 through 4.4.

Measurement constraints, on the other hand, integrate sensor data like depth maps and a scene segmentation into the model (section 4.5 and 4.6). Hence internal properties and measured data are integrated in a uniform representation.

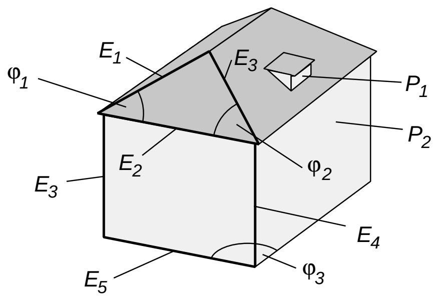

Figure 5: Internal constraints used for the system

4.1 Angle constraint

Model parts can be arranged to form a certain angle between them. In figure 5 the angle φ_3 between the front and side wall of the model is intended to be 90 degrees. The use of the angle constraint is not restricted to 90 degrees, though. The constraint function is

$$c_\varphi = (\varphi_s - \varphi(\boldsymbol{p}))^2 \cdot w_\varphi, \tag{1}$$

where φ_s denotes the angle as it should be. $\varphi(\boldsymbol{p})$ describes the actual angle as function of the position vector $\boldsymbol{p}$ of the involved parts. w_φ is a weight that controls the relative influence of the constraints among each other. Every constraint function is weighted with some weight w that empirically has to be determined once. The function equals zero if the actual angle equals the intended angle.

4.2 Parallel constraint for parts

Two model parts can be constrained to be parallel. Extensions of buildings, for example, can be aligned parallel to larger walls, whose spatial orientation can be estimated more precisely from image data. In figure 5 the walls P_1 and P_2 are an example where an *parallel constraint for parts* could be used. This constraint is a special case of the angle constraint. For $\varphi_s = 0$ the constraint function is:

$$c_{par} = (\varphi(\boldsymbol{p}))^2 \cdot w_{par}. \tag{2}$$

4.3 Parallel constraint for edges

In addition to model parts two model edges, which are the intersections of two parts, can be constrained to be parallel, e.g. two edges of a house should be parallel like edge E_3 to E_4 or E_2 to E_5 in figure 5. The constraint function is the same as above except that the angle between the edges is used instead of the angle between the plane normals.

4.4 Symmetry constraint

Symmetries are important for subjectively good looking objects. Human observers are very sensitive to violations of expected geometric relations. For this purpose a *symmetry constraint* is introduced that judges the difference between two supposedly equal angles. In figure 5 this concerns the angles φ_1 and φ_2, which means the slope of the roof should be identical for the front and back part.

The cost function is:

$$c_{sym} = w_{sa} \cdot (\varphi_1(\boldsymbol{p}) - \varphi_2(\boldsymbol{p}))^2. \tag{3}$$

where φ_1 and φ_2 must equal each other to minimize the function's value to zero.

We propose the following *measurement constraints* to incorporate 2-D image features:

4.5 Position constraint

We introduce a position constraint to ensure that depth information (fig. 2) matches with the actual part position and orientation. A segmented image and a depth map are jointly used to estimate an initial orientation $\boldsymbol{n}_{init}$ and an initial center of gravity $\boldsymbol{c}_{init}$ for each part by plane regression. If the part is being moved by the optimization algorithm the actual position $\boldsymbol{c}(\boldsymbol{p})$ and $\boldsymbol{n}(\boldsymbol{p})$ may differ from their initial values (cf. figure 6).

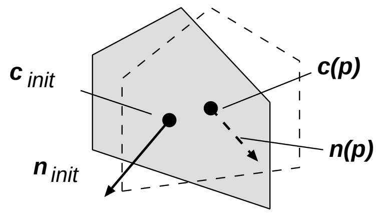

Figure 6: Moving a surface element

The cost function

$$c_{dir} = w_c \left[\angle^2(\boldsymbol{n}_{init}, \boldsymbol{n}(\boldsymbol{p})) + w_n \cdot ((\boldsymbol{c}(\boldsymbol{p}) - \boldsymbol{c}_{init}) \cdot \boldsymbol{n}_{init})^2\right] \tag{4}$$

values the match between initial and current position and orientation respectively. For orientation the angle between initial normal direction $\boldsymbol{n}_{init}$ and moved normal $\boldsymbol{n}(\boldsymbol{p})$ contributes to the cost function whereas for position the projection of $(\boldsymbol{c}(\boldsymbol{p}) - \boldsymbol{c}_{init})$ onto the plane normal is used. Thus a position shift perpendicular to the plane normal does not have any influence on the cost function.

All values $w(i,j)$ of the part's area in the certainty map (fig. 2 right) are integrated into the constraint function weight w_c according to the following formula:

$$w_c = \sum_{(i,j)\in R} w(i,j).$$

Larger walls (regarding their area in the image plane) with more reliable certainty values lead to model parts that are less moveable by the optimization than smaller parts. Together with internal constraints (e.g. the parallel constraint) smaller extensions are aligned to larger parts whose spatial position can be estimated more accurately from image data.

4.6 Edge constraint

As shown in figure 7 the goal of an edge constraint is to make the projection of a model's 3-D edge congruent to the corresponding image contour. To have the 3-D edge's projection lie on the image edge is particularly important for texturing, which is the backprojection of image information onto the 3-D model.

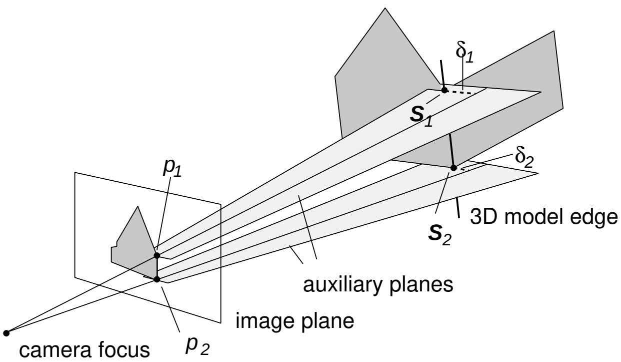

Figure 7: Using edge information for 3-D modelling

The image edge is represented by the points p_1 and p_2, which are used to span two auxiliary planes. These planes are perpendicular to the so called *plane of sight* which is determined by the focal point of the camera, p_1 and p_2. The 3-D model edge intersects with the auxiliary planes in S_1 and S_2. The orthogonal distances δ_1 and δ_2 between the intersections and the *plane of sight* have influence on the cost function:

$$c_{edge}(\boldsymbol{p}) = w_{edge} \cdot (\delta_1^2(\boldsymbol{p}) + \delta_2^2(\boldsymbol{p})), \qquad (5)$$

with the weight w_{edge} that controls the relative influence of the contraint during optimization.

All above constraints are designed to meet the needs of human spectators regarding geometric properties.

5 Incremental Surface Reconstruction

The goal of surface reconstruction is to find the object's shape and perform the accompanying viewpoint registration. The latter is used to integrate new information if further viewpoints are added to the scene description.

As described in section 3 the model is represented a planes in space which can be moved through the parameter vector $\boldsymbol{p}$. Together with the model parts a number of geometric constraints, as described in the previous section, are derived from the generic knowledge base during interpretation. Due to faulty image data the resulting set of constraints generally lacks consistency. Numerical optimization is applied and leads to a model which best meets all constraints. Therefore the optimizing algorithm finds the parameter vector $\boldsymbol{p}$ such that the global sum of all N cost functions $c^{(i)}$ is minimized:

$$c_{glob}(\boldsymbol{p}) = \sum_{i=1}^{N} w_i \cdot c^{(i)}(\boldsymbol{p}) \rightarrow \text{Min}, \qquad (6)$$

where w_i denotes the respective weight for the cost function. For details on how to choose the weights refer to[9].

In the approach presented here the method of conjugated gradients is employed for minimizing the global sum of cost functions. The use of a general minimization technique like conjugated gradients makes it easy to extend the existing set of constraints. Only a new cost function is needed to integrate the judgement of new features into the system.

To avoid local minima and to reduce the parameter space needed for optimization, the search for the best fit is done hierarchically, by use of the system's control structure. In a the first step the shape is only approximated by a few planes and then refined in following minimizations.

In addition the cost functions are used for verification of hypotheses generated during interpretation.If for instance an angle between two parts has been wrongly predicted to 90 degrees instead of, say, 135 degrees, the affiliated cost function will return an unusually high value. This indicates that the hypothesis has probably to be rejected by the interpretation.

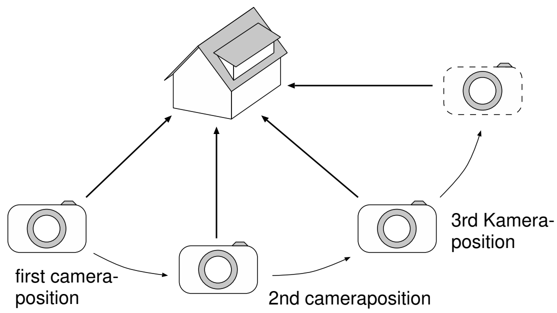

Figure 8: Registration of new camera viewpoints

Internal constraints apply only to objects. The *measurement constraints* like *edge constraint* or *position constraint*, on the other hand, are used for integrating camera related image features like edges or depth information into the model. Thus two ways of minimizing the latter constraints are possible: Either the model parts or the camera model can be moved in space by numerical optimization. This allows not only to further optimize the model but also to incrementally add new camera viewpoints to the scene description. In this mode the constraints are minimized by moving the new camera to the position best matching the constraints.

As depicted in figure 8 the first camera viewpoint is used as a reference position. Together with a first shape approximation the second viewpoint is determined. Using this shape approximation the following camera positions are estimated. In final steps the model is completed with smaller extensions.

Constraints are hence exploited both for improved surface reconstruction and camera pose estimation, thus leading to a consistent and closed surface description from multiple viewpoints.

6 Experimental Results

This section deals with some experimental results that have been achieved with the system. In the first part we show the incremental reconstruction of an existing appartment building. The second part deals with the view point registration that can be achieved with the system.

6.1 Hierarchical Surface Reconstruction

Figure 9 shows the input from the first of two camera viewpoints for the modelling process: The original camera image (a), a depth mep of the scene (b), which has a large number of dropouts, and a manually created segmentation of the scene (c).

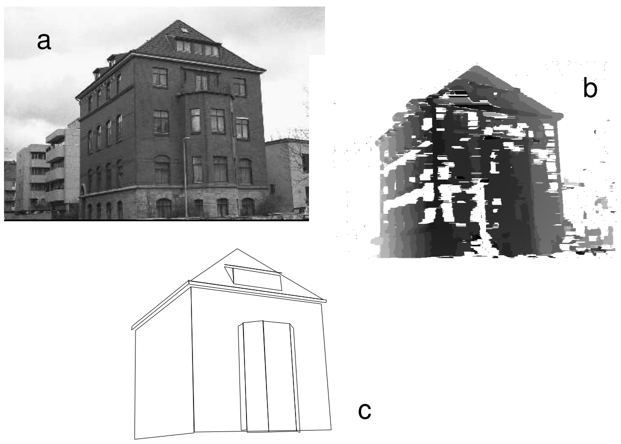

Figure 9: Input data for the modelling process: image (a), depth map (b), segmentation (c)

In a first step the approximate shape is reconstructed consisting only of the main walls and the front roof (fig. 10 left). The knowledge base inserts the back walls although they are occluded in the first viewpoint. A number of edge, position and angle constraints are then imposed on the model, which serve to improve the initial model's shape.

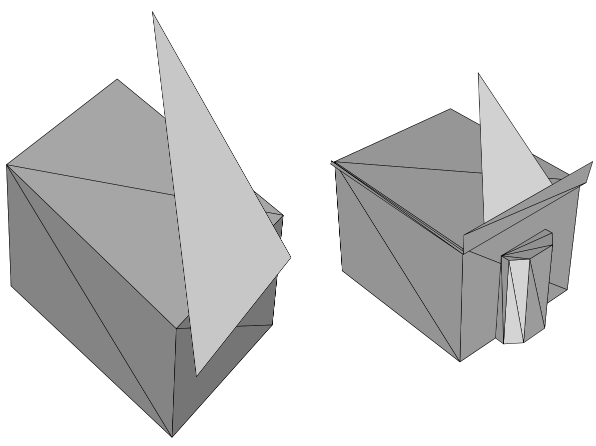

Figure 10: Incremental surface reconstruction

In the same optimization process the second viewpoint position is estimated through edge and position constraints that link the model to the other camera position. After computing the basic form, the main parts are fixed and used as references for the incremental refinement of the model. As shown in fig. 10 (right) smaller parts like oriels are aligned through parallel constraints to the already determined main walls.

Figure 11 depicts the completed wireframe model of the building. In a last reconstruction step the original image information is backprojected onto the model, which can be seen in fig. 12. It has been generated from two stereo images, depth maps, a manual scene segmentation and constraints derived from the knowledge base. 28 surface elements are connected by 51 constraints. The model is completely closed and has been textured only from two original images. The camera position of the second camera has been determined together with the model's construction.

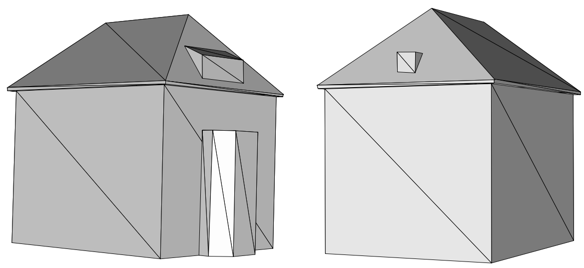

Figure 11: Reconstructed Wireframe

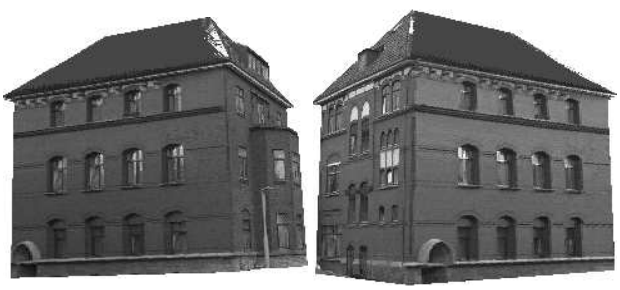

Figure 12: Textured Model

6.2 Viewpoint Registration

During view point registration new camera positions are added to the scene description. They can be used for incrementally refining the model. To show the performance of the system during viewpoint registration a reference object of well known geometry has been used. Figure 13 shows two original images (front and right side). The estimated shape in form of a simple box is overlayed.

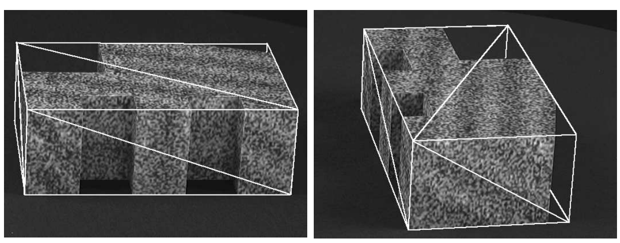

Figure 13: Two views of the reference object using already registered camera positions

	front	right	rear	left
reference angle (degrees)	0	85	195	265
calibration result (degrees)	0	85,07	195,02	265,02

Table 1: Viewpoint calibration results

The registration procedure is illustrated in figs. 14 and 15. Beginning with the front camera position a new viewpoint is registered. The left image of fig. 14 shows the camera image of a new viewpoint with the model's edges projected in the (yet) wrong camera position. The projected edges do not match with the image edges. Fig.14 (right) shows the model's geometry together with the initial front camera position in form of a pyramid.

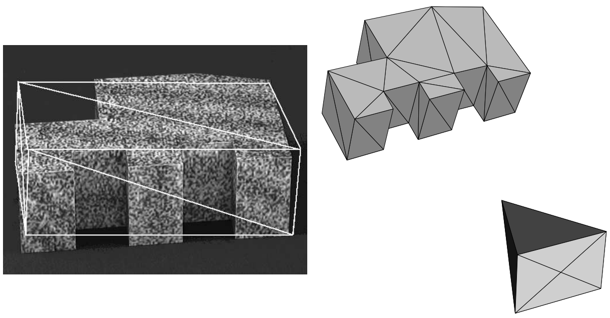

Figure 14: Before registration of a new viewpoint

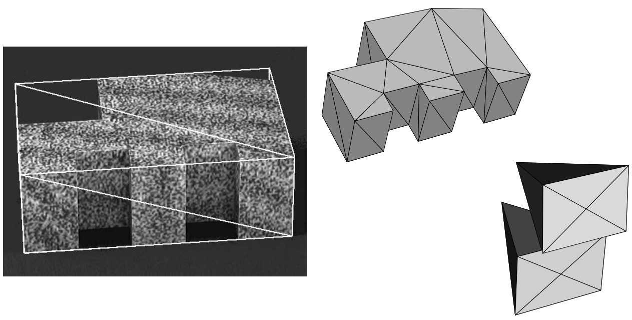

Figure 15: After registration of a new viewpoint

The situation after registration is depicted in fig. 15. The new position has been estimated leading to a projection of the geometry that matches the image edges (fig. 15 left). The second pyramid (fig. 15 right) depicts the estimated camera position.

An angle of about 45 degrees between two viewpoints could be estimated without encountering numerical instabilities. In case larger movements have to be estimated an initial guess had to be made. The images of the reference object were taken with one camera while the object was rotating in front of it on a computer controled turntable. Due to the very accurate positioning of the turntable the rotational angle between the images is exactly known. Table 1 shows the calibration results for some selected viewpoints. It should be stated that the presented estimations were achieved only using a manually generated segmentation. By use of a subpixel-accurate edge detection the precision could probably be further increased.

7 Conclusions

We have discussed the surface reconstruction of the system AIDA, which extracts 3-D geometry from a sequence of stereo images. It is particularly employable when reconstructing scenes consisting of man-made objects like buildings. Through the use of an explicit scene description in form of a generic semantic net geometric constraints are selected during interpretation. They restrict the model geometry to fulfill some expectations human observers have about such scenes like rectangular, plane walls, straight edges or symmetries.

This allows to further increase the model's quality compared to the results that can be achieved by only using image data. In particular properties that are important to human observers, like symmetries and parallelisms, are improved. Especially the edge constraint has proved to be very important since texturing depends directly on the exact congruence between projected model edges and image contours.

The approach presented in this contribution leads to realistic models that can be employed e.g. for driving simulators or in virtual scenes for film production.

References

[1] C. Braun, Th. Lang, F. Schickler, Steinhage, Cremers, W. Förstner, L. Plumer: *On the Models for Photogrammetric Building Reconstruction* Computer & Graphics, Vol. 19, No. 1, 1995

[2] O. Grau: *Ein Szeneninterpretationssystem zur Modellierung dreidimensionaler Körper*, 17. DAGM-Symposium Mustererkennung 1995, 13.-15. Sept. '95

[3] O. Grau: *A Scene Analysis System for the Generation of 3-D Models*. Submitted Paper

[4] R.Koch: *3-D Surface Reconstruction from Stereoscopic Image Sequences*, International Conference of Computer Vision ICCV '95 Cambridge, MA., USA, June 1995

[5] F. Leberl, M. Gruber, P. Uray, F. Madritsch: *Trade-Offs in the Reconstruction and Rendering of 3-D Objects*, Mustererkennung 1994, 16. DAGM Symposium und 18. Workshop der ÖAGM, Wien 1994

[6] C.-E. Liedtke, O. Grau, S. Growe: *Use of Explicit Knowledge for the Reconstruction of 3-D Object Geometry*, 6th International Conference CAIP '95 Computer Analysis of Images and Pattern. Sep. 6-8 1995 Prague

[7] Y. Yakimovski, R. Cunningham: *A System for Extracting 3-D Measurements from a Stereo Pair of TV Cameras*, Computer Graphics and Image Processing Vol 7, pp. 195-210, 1978

[8] Pakzad, Kian: *Segmentierung von Tiefenkarten aufgrund ähnlicher Flächenorientierung*, Studienarbeit an der Universität Hannover, 1994

[9] S. Weik: *Erzeugung dreidimensionaler Oberflächenmodelle unter Einhaltung gemetrischer Randbedingungen*, Diplomarbeit Universität Hannover, Aug. 1995

[10] R. Tsai: *A Versatile Camera Calibration Technique for High-Accuracy 3-D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*, IEEE Journal of Robotics and Automation, Vol. RA-3, No.4, pp.323-344, Aug.1987