3–D Surface Reconstruction from Stereoscopic Image Sequences

Reinhard Koch

Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung
Universität Hannover, Appelstr. 9A, 30167 Hannover, Germany

Abstract

A stereoscopic scene analysis system for 3–D modeling of objects from stereoscopic image sequences is described. A dense map of 3–D surface points is obtained by image correspondence, object segmentation, interpolation, and triangulation. Emphasis is put on the accurate measurement of image correspondences from grey level images. The surface geometry of each scene object is approximated by a triangular wire–frame which stores the surface texture in texture maps. Sequence processing serves to track camera motion and to fuse surfaces from different view points into a consistent 3–D surface model. From the textured 3–D models, highly realistic image sequences from arbitrary view points can be synthesized using computer graphics techniques.

1 Introduction

In this contribution a system for automatic 3–D surface reconstruction from stereoscopic image sequences is discussed. The system is aimed on automatically computing CAD–models of any object that is composed of smooth surface patches as is the case with buildings and objects containing coherent surfaces. This object type is needed for a wide range of applications where computer generated 3–D environments are desirable, like in architecture visualization [1], virtual television studios [2], virtual presence for video communications [3] and general "virtual reality" applications. It is not intended to model arbitrary surface geometries like trees or semi–transparent surfaces. Geometric and photometric models of the scene objects as well as object and camera motion are extracted from the image sequence. The geometry is measured from a stereoscopic image pair and the camera motion is tracked from the image sequence directly. All measurements obtained throughout the image sequence are integrated into a consistent 3–D scene model that contains not only the scene geometry but also texture maps of the object surface.

In previous research 3–D object shape, 3–D object motion, and object surface texture were treated separately. Great effort went into developing algorithms that estimate 3–D object shape from various sources, termed shape from motion, stereo, and others [4]–[6]. On the other hand research was conducted to find solutions to the problem of rigid object motion [7],[8]. Only recently the problem of dynamic nonrigid bodies and nonrigid motion was addressed [9],[10]. Some approaches are reported from monocular vision systems to compute dense depth maps and surface reconstruction for orthographic [11] and perspective projection [12]. The approach that is most related to this research can be found in [13].

Chapter 2 sketches the general outline of the complete system. In Chapter 3 the surface reconstruction from a single image pair is discussed. Chapter 4 deals with the sequence processing which consists of view point tracking and the fusion of surfaces from different view points.

2 System Overview

The goal of the surface reconstruction is to extract a complete 3–D surface model of the scene that contains the surface geometry and the surface colors. Input to the modeling is a stereoscopic image sequence recorded by a binocular stereoscopic camera. To model the complete scene the camera sensor is moved around the object to capture views from all sides. The structure of the analysis system is shown in Fig. 1. Three main modules can be identified.

Sensor Processing: In order to compute depth from the camera pair a sensor calibration is needed. In an off-line calibration process before the actual scene recording, the internal camera parameters focal length and radial distortion are computed together with the relative position and orientation of both cameras in a sensor calibration coordinate system. The real cameras may have arbitrary relative external orientation like a convergence angle, and may have differing internal parameters. During scene recording those differences are compensated in a rectification process. The images are rectified with projective mapping onto a virtual camera target in such a way that the virtual camera system has coplanar image planes. This rectification greatly simplifies the image pair processing in the next stage.

Image Pair Processing: For each camera view point a stereoscopic image pair is recorded. This image pair is used to extract the surface depth information from this particular view point. The analysis consists of depth estimation and

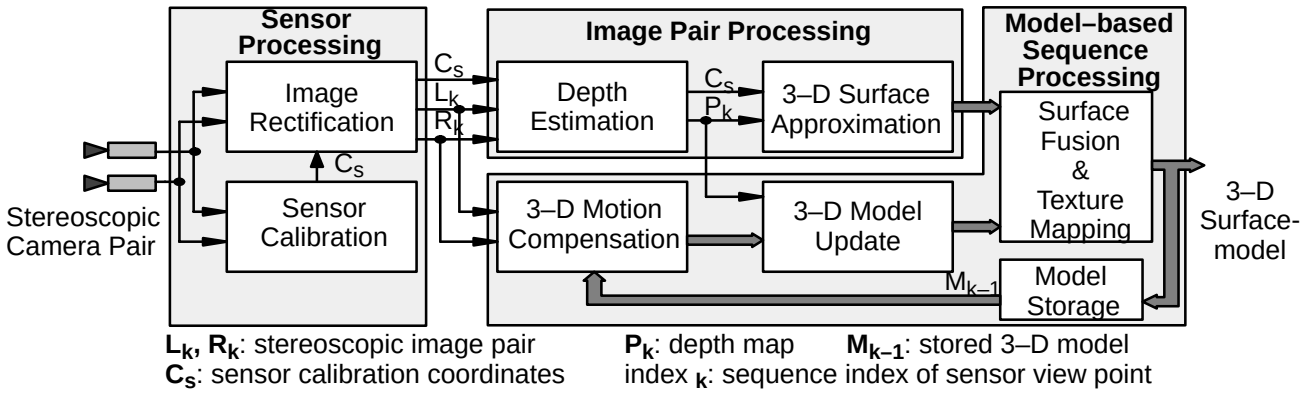

Fig.1: Structure of 3–D Scene Analysis from Sequences of Stereo Images

surface approximation. During depth estimation the correspondence between the stereoscopic image pair is evaluated to compute a dense depth map. The discrete depth map is then converted into a structured, parametric surface representation based on some surface constraints. As geometrical surface representation a triangular wire–frame is fitted to the surface. The surface color of the real object for this view point is stored as texture map on the wire–frame surface.

Model–based Sequence Processing: Image pair processing computes a 3–D surface reconstruction of a particular view point. The complete model is obtained by fusing the different view points into a consistent surface model. This fusion is performed in the sequence processing which consists of three main modules. The view–point dependency of the surfaces are removed by tracking the camera and object motion between the view points. The model is transformed into the current view point which allows to fuse the new surface with the existing model. If a particular model surface can be seen from several view points, then all the depth measurements of this surface are integrated in the model update phase using a kalman filter for each vertex of the surface wire–frame. When new object areas become visible the model wireframe is extended to include the newly visible parts as well. Finally all surface areas are textured to allow realistic visualization of the model.

3 The Image Pair Analysis

Image pair analysis computes a view point dependent 3–D surface model from the rectified stereoscopic image pair. It is assumed that the objects to be modeled consist of piecewise continuous, smooth surfaces. The steps to be performed are image correspondence analysis, surface segmentation, depth interpolation, and polygonal surface approximation. The modules of the analysis pipeline were described in detail in [25] and will be reviewed only briefly.

3.1 Camera Model

The imaging camera system is described as a pinhole camera with perspective projection as in Fig. 2. The camera is oriented arbitrarily in space with respect to a calibration coordinate system $\mathbf{C_s}$. The focal point of the camera is located at $\mathbf{C}$, while the camera orientation is described by a coordinate system $(\mathbf{A}, \mathbf{H_0}, \mathbf{V_0})$. $\mathbf{A}$ defines the optical axis and the imaging sensor is spanned by $\mathbf{H_0}$ for the image scan line and $\mathbf{V_0}$ for the image columns. This camera system corresponds to the camera model as introduced by Yakimowski and Cunningham [14]. The optical axis penetrates the image plane at center point $\mathbf{h} = (h_x, h_y)^T$ and the effective focal length c is scaled in pixel coordinates with the sensor pixel size $\mathbf{s} = (s_x, s_y)^T$. The spatial point $\mathbf{P}$ projects onto the camera target at $\mathbf{p}$ in pixel coordinates as intersection of the line of sight $\mathbf{S} = \mathbf{P} - \mathbf{C}$ with the image plane.

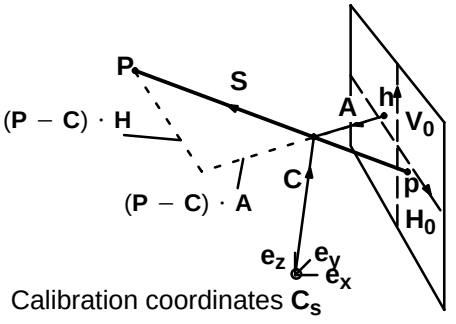

Fig. 2: Camera projection model

Perspective projection of $\mathbf{P}$ onto the image plane at $\mathbf{p}$ is computed with Eq. (1).

$$\boldsymbol{p} = \frac{1}{(\boldsymbol{P}-\boldsymbol{C})\cdot\boldsymbol{A}} \cdot \begin{pmatrix} (\boldsymbol{P}-\boldsymbol{C})\cdot\boldsymbol{H} \\ (\boldsymbol{P}-\boldsymbol{C})\cdot\boldsymbol{V} \end{pmatrix} \qquad (1)$$

$$\textit{with}: \quad \boldsymbol{H} = \frac{c}{s_x}\cdot\boldsymbol{H}_0 + h_x\cdot\boldsymbol{A} \quad , \quad \boldsymbol{V} = \frac{c}{s_y}\cdot\boldsymbol{V}_0 + h_y\cdot\boldsymbol{A}$$

The definition of four vectors for the camera is redundant because $(\mathbf{A},\mathbf{H_0}, \mathbf{V_0})$ form a right–handed coordinate system. The orientation of the camera is uniquely described by a rotation matrix $\underline{R}$ with three independent rotation angles $\mathbf{R} = (R_x, R_y, R_z)^T$ which transforms $\mathbf{C_s}$ into $(\mathbf{A},\mathbf{H_0}, \mathbf{V_0})$:

$$\boldsymbol{H}_0 = \underline{R}\cdot\boldsymbol{e}_x, \boldsymbol{V}_0 = \underline{R}\cdot\boldsymbol{e}_y, \text{and}\, \boldsymbol{A} = \underline{R}\cdot\boldsymbol{e}_z \qquad (2)$$

The external orientation and position $\mathbf{X} = (\mathbf{R},\mathbf{C})^T$ of each camera together with the internal parameters of focal length c and the squared radial distortion are calibrated from a

planar calibration pattern of known 3–D points [15]. The calibration is an extension of the calibration as described by Tsai [16] that is performed for both cameras of the stereoscopic sensor. By computing the orientation difference between the cameras the relative orientation in sensor coordinates is obtained that is invariant to sensor motion.

Calibration errors are handled by treating $\mathbf{X}$ as gaussian random vector with mean expectation value $\hat{\boldsymbol{X}}$.For each camera the external orientation vector is calibrated together with its associated Covariance $\underline{C_X}$ that is composed of 3x3 sub–matrices for rotation error $\underline{C_R}$,translation error $\underline{C_C}$, and cross–variance $\underline{C_K}$.

$$\underline{C_X} = \begin{pmatrix} \underline{C_R} & \underline{C_K} \\ \underline{C_K}^T & \underline{C_C} \end{pmatrix} \quad , \quad \hat{\boldsymbol{X}} = (\boldsymbol{R}, \boldsymbol{C})^T \tag{3}$$

3.2 Correspondence analysis

Correspondence analysis exploits the fact that a surface point $\mathbf{P}$ of the real object projects onto both camera targets of the sensor at $\mathbf{p_l}$ and $\mathbf{p_r}$. Because of perspective projection all points on the line $(\mathbf{P} - \mathbf{C_l})$ project onto the single point $\mathbf{p_l}$ in the left camera while in the right camera each point projects onto a different point $\mathbf{p_r}$ on the epipolar search line. A point $\mathbf{P}$ in infinity projects onto the starting point $\mathbf{p_e}$ of the epipolar line. The length of the epipolar line from the starting point to the corresponding point $\mathbf{p_r}$ is defined as disparity. Because of rectification the epipolar search line simply corresponds to the image scan line. From the rectified images a disparity map is obtained by searching along the epipolar lines using correlation matching and dynamic programming [17]. The cross correlation of a small image patch (typically 5*5 to 7*7 pixel) around each point on the epipolar line is computed as similarity measure.

The search for the best match between the points on the epipolar line is controlled by uniqueness and ordering constraints. These constraints are based on the fact that there can be no more than one match between left and right image points and that matches are in order for physical surfaces. All possible correspondences are evaluated in an optimum search procedure using dynamic programming that matches all correspondences between left and right image that lie on the same epipolar line. The dynamic programming algorithm was adapted from the work of Cox et al. [18]. The disparity value obtained for each candidate is recorded in a disparity map.

The disparity estimates computed with this approach give dense and reliable depth maps with a disparity resolution of 1 pixel due to quantization of the search space. This quantization leads to a discontinuous and rough surface that can not be tolerated in surface reconstruction. Therefore in a second stage the quantized disparity estimates are taken as a starting value for a sub–pixel accuracy gradient matching with affine transformation. Assuming that the object mostly consists of smooth surfaces with arbitrary orientation, the effect of perspective distortions between both views is corrected by an affine transformation of a small image patch of about 9x9 to 19x19 pixel. The difference in grey level distribution between the corresponding patches is minimized by fitting the patch with a robust least squares estimation of the affine transformation[19],[20]. With this approach the disparity is estimated to an average accuracy of 0.05 to 0.1 pixel, which is enough for most reconstruction tasks.

The disparity map is now converted into a depth map where each pixel $\mathbf{p_l}$ of the map contains the length $\tau_l = |\mathbf{S_l}|$ of the line of sight vector to the real surface point $\mathbf{P}$. The 3D–Position $\mathbf{P}$ is computed from the corresponding points in sensor coordinates by space intersection, which is formulated in Eq. (4) that computes the point in space where both line–of–sight vectors of the corresponding points meet closest

$$\begin{gathered} \boldsymbol{P} = \boldsymbol{C}_l + \tau_l \cdot \boldsymbol{S}_l = \boldsymbol{C}_r + \tau_r \cdot \boldsymbol{S}_r + \boldsymbol{M} \\ \textit{with minimum condition:} \quad \boldsymbol{S}_l^T \cdot \boldsymbol{M} = \boldsymbol{S}_r^T \cdot \boldsymbol{M} = 0 \end{gathered} \tag{4}$$

Solving Eq. (4) computes the length τ_l for the left line–of–sight

$$\tau_l = \frac{(\boldsymbol{C}_r - \boldsymbol{C}_l) \cdot (\boldsymbol{S}_l - s_{lr} \cdot \boldsymbol{S}_r)}{1 - s_{lr}^2} \quad \textit{with:} \quad s_{lr} = \boldsymbol{S}_l^T \cdot \boldsymbol{S}_r \tag{5}$$

$\mathbf{P}$ is treated as random vector with covariance $\underline{C_P}$. The uncertainty of the point in space is a function of calibration uncertainty and correspondence uncertainty. The position error is

$$\boldsymbol{dP} = \boldsymbol{dC}_l + d\tau_l \cdot \boldsymbol{S}_l + \tau_l \cdot \underline{S_l} \cdot \boldsymbol{dR}_l + d\tau \cdot \underline{S_l} \cdot \boldsymbol{dR}_l \tag{6}$$

and the resulting point covariance can therefore be computed as a function of the camera calibration covariance from Eq. (3) and the correspondence uncertainty $\sigma\tau^2$.

$$\underline{C_P} = \underline{C_{Cl}} + \tau_l^2 \cdot \underline{S_l} \cdot \underline{C_{Rl}} \cdot \underline{S_l}^T + 2\tau_l \cdot \underline{C_{Kl}} \cdot \underline{S_l}^T + (\boldsymbol{S}_l \cdot \boldsymbol{S}_l^T) \cdot \sigma\tau^2 \tag{7}$$

3.3 Scene segmentation and depth interpolation

Surface segmentation exploits the knowledge that the scene to be modelled mostly consists of smooth or even plane surfaces that may have some creases and breaks. The approach for the segmentation is therefore to extract regions of similar surface orientation and then to group these regions to form surfaces. For each pixel of the disparity map the local surface normal is computed and a two–dimensional histogram of the surface orientation is accumulated. The local peaks in the histogram correspond to the most likely surface orientations while the valleys correspond to the boundary between surfaces. The surface orientations are now clustered by searching for the valleys and assigning each peak a surface label.

As last step of surface reconstruction the object surface is interpolated by a parametric surface. A multi–grid surface reconstruction algorithm as described by Terzopoulos[21] was chosen to calculate the interpolation with a finite element approximation. It is assumed that each segmented area

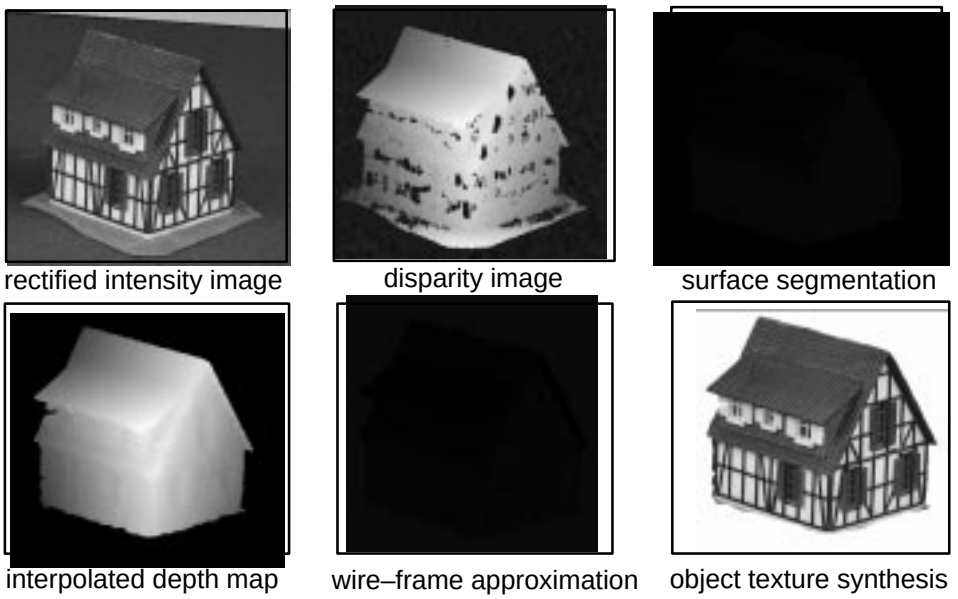

Fig. 3: 3–D surface reconstruction from one view point of scene HAUS.

contains a smooth coherent surface that can be modeled as a thin plate with a certain stiffness and that inside such a region the disparity measurement is either corrupted by noise or no estimate is available. The interpolation solves the problem of minimizing the potential energy function of the thin plate that is deformed by the disparity measurements.

3.4 3–D surface approximation

The amount of data for storing the finite element surface is very high, so an approximation of the surface is needed. The interpolated disparity map is converted into a parametric 3–D surface description by spanning a triangular wire–frame in space for each segmented object surface while preserving the discontinuities at the surface boundaries.

The triangular mesh was chosen because it is capable to approximate arbitrary surface geometries without singularities. On the surface of each triangle the object surface texture is stored in a texture map from which naturally looking views can be synthesized with texture mapping [23]. Fig. 3 shows the modeling process for a surface model from the scene HAUS. All steps of the modeling process are shown. The results of rectification, disparity estimation, segmentation, surface interpolation, wire–frame generation, and texture mapping are displayed.

4 Model–based Sequence Processing

The modeling of complex scenes requires that more than one view is evaluated to account for occluded objects and to improve the object geometry. Surface models from different view points have to be registered into a consistent object coordinate system. This registration is achieved by tracking the camera and object motion directly from the image sequence. Surface shape is improved by updating the visible surface area with a kalman filter. In areas where new surfaces become visible new surfaces are fused with the existing surfaces to complete the scene model.

4.1 Camera motion tracking

Direct tracking of moving objects observed by a static camera was discussed in detail in [22],[24]. This approach is extended to a moving camera system in the following section. Each object in the scene is treated as an arbitrarily shaped, rigid surface. The shape of the surface is modeled by the control points $\mathbf{P_i}$ of the triangular surface mesh. The object is observed from different view points. Tracking of the camera motion can be computed directly from the spatio–temporal intensity gradients. The motion from t_0 to t_1 is described by a rotation $\mathbf{R_v}$ of the camera axes **A**, **H**, **V** and the focal point **C** around the rotational center **G** and by a translation of **G** with **Tv**. The rotation is defined by a rotation matrix $\underline{R} = \underline{I} + \underline{Rv}$.

The imaging situation for the moving camera is depicted in Fig. 4. For small rotations the matrix $\underline{R_v}$ is linearized and the camera motion is expressed as

$$\begin{aligned} \boldsymbol{C}_t &= \boldsymbol{C} + \underline{Rv} \cdot (\boldsymbol{C} - \boldsymbol{G}) + \boldsymbol{Tv} \\ \boldsymbol{A}_t &= \boldsymbol{A} + \underline{Rv} \cdot \boldsymbol{A} = \boldsymbol{A} + \underline{A} \cdot \boldsymbol{Rv} \\ \boldsymbol{H}_t &= \boldsymbol{H} + \underline{Rv} \cdot \boldsymbol{H} = \boldsymbol{H} + \underline{H} \cdot \boldsymbol{Rv} \\ \boldsymbol{V}_t &= \boldsymbol{V} + \underline{Rv} \cdot \boldsymbol{V} = \boldsymbol{V} + \underline{V} \cdot \boldsymbol{Rv} \end{aligned} \tag{8}$$

with unknown motion parameters $\underline{R_v}$ and $\mathbf{T_v}$.

Because of the special structure of $\underline{R_v}$ the Matrix–Vector product $\underline{R_v} \cdot \mathbf{X}$ can be written as $\underline{X} \cdot \mathbf{R_v}$ which reduces $\underline{R_v}$ to its three independent components. This notation will be used for the Vectors **A,H,V** and their corresponding matrices $\underline{A}$, $\underline{H}$, $\underline{V}$.

$$\underline{X} = \begin{pmatrix} 0 & X_z & -X_y \\ -X_z & 0 & X_x \\ X_y & -X_x & 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{R_v} = \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix}.$$

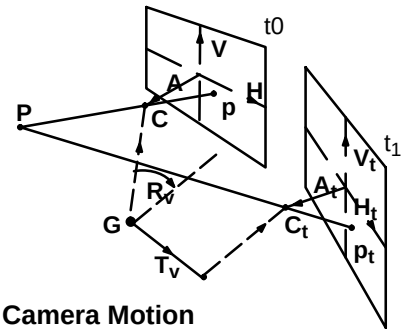

Fig. 4: Camera Motion

The projection of **P** onto the moving image plane for the times (0, t) generates an optical flow $\mathbf{d} = (d_x, d_y)^T = \mathbf{p}_t - \mathbf{p}$. Insertion of the camera motion from Eq. (8) in Eq. (1) and solving with some small linearization delivers an optical flow as function of the camera motion parameters

$$\boldsymbol{d} = \begin{pmatrix} \dfrac{(\boldsymbol{P}-\boldsymbol{G})^T \cdot (\underline{H} - p_x \cdot \underline{A}) \cdot \boldsymbol{Rv} - (\boldsymbol{H} - p_x \cdot \boldsymbol{A})^T \cdot \boldsymbol{Tv}}{(\boldsymbol{P}-\boldsymbol{C})^T \cdot \boldsymbol{A} + (\boldsymbol{P}-\boldsymbol{G})^T \cdot \underline{A} \cdot \boldsymbol{Rv} - \boldsymbol{A}^T \cdot \boldsymbol{Tv}} \\ \dfrac{(\boldsymbol{P}-\boldsymbol{G})^T \cdot (\underline{V} - p_y \cdot \underline{A}) \cdot \boldsymbol{Rv} - (\boldsymbol{V} - p_y \cdot \boldsymbol{A})^T \cdot \boldsymbol{Tv}}{(\boldsymbol{P}-\boldsymbol{C})^T \cdot \boldsymbol{A} + (\boldsymbol{P}-\boldsymbol{G})^T \cdot \underline{A} \cdot \boldsymbol{Rv} - \boldsymbol{A}^T \cdot \boldsymbol{Tv}} \end{pmatrix} \tag{9}$$

The optical flow **d** is derived directly from the spatio–temporal image gradients when the image model of locally linearized image intensities is assumed. This brightness–continuity constrained is widely used in flow computation.

$$\Delta I = I(x,y)_t - I(x,y) = \boldsymbol{g}(x,y)^T \cdot \boldsymbol{d}(x,y) \tag{10}$$

The relationship between the image gradients **g** and the camera motion parameters is found by substituting the flow equation (9) into the measurement Eq. (10):

$$\begin{aligned} &\Delta I \cdot (\boldsymbol{P}-\boldsymbol{C})^T \cdot \boldsymbol{A} = (\boldsymbol{P}-\boldsymbol{G})^T \cdot \underline{F} \cdot \boldsymbol{Rv} - \boldsymbol{F}^T \cdot \boldsymbol{Tv} \\ &\text{with } \boldsymbol{F} = g_x \cdot \boldsymbol{H} + g_y \cdot \boldsymbol{V} - (g_x p_x + g_y p_y + \Delta I) \cdot \boldsymbol{A} \\ &\qquad \underline{F} = g_x \cdot \underline{H} + g_y \cdot \underline{V} - (g_x p_x + g_y p_y + \Delta I) \cdot \underline{A} \end{aligned} \tag{11}$$

This direct motion equation can be computed for each point in the scene that holds image gradients above a noise threshold. From all points the motion vector $\mathbf{X_v} = (\mathbf{R_v}, \mathbf{T_v})^T$ and its associated covariance $\underline{C_v}$ is computed using a Gauss–Marcov minimum variance estimator over all measurements.

4.2 3–D Model Update

The depth measurements from the different view points are integrated during the model update using a kalman filter [26] for each surface control point **P**. The kalman filter consists of prediction and update phase.

Prediction: Each control point from the previous measurement is predicted to the current position and updated with a weighted depth estimate from the current position to improve surface shape. The prediction phase in the case of stationary objects is very simple:

$$\hat{\boldsymbol{P}}_{(k)} = \boldsymbol{P}_{(k-1)} \quad , \quad \hat{\underline{C}}_{P(k)} = \underline{C}_{P(k-1)} \tag{12}$$

Update: The new measurement $\mathbf{P_M}$ from the current frame is fused with the predicted position using the kalman filter algorithm

$$\underline{K}_{(k)} = \hat{\underline{C}}_{P(k)} \cdot \left(\hat{\underline{C}}_{P(k)} + \underline{C}_{PM(k)} \right)^{-1} \tag{13}$$

$$\boldsymbol{P}_{(k)} = \hat{\boldsymbol{P}}_{(k)} + \underline{K}_{(k)} \left(\boldsymbol{P}_{M(k)} - \hat{\boldsymbol{P}}_{(k)} \right) , \quad \underline{C}_{P(k)} = \left(\underline{I} - \underline{K}_{(k)} \right) \hat{\underline{C}}_{P(k)} \tag{14}$$

The filter is computed for each control point of the visible surface. Fig. 5 gives an impression about the average improvement achieved with the filtering. The model HAUS was tracked over 9 images and the mean standard deviation of the depth estimate was computed for all visible surface points. The binocular camera with a base line of 230 mm was positioned 1.3 m from the object and rotated around the object with a view point change of 4 degrees between two successive images. Graph (1) shows the error of depth integration computed from known view points. The mean depth error dropped from 3.5 mm for a single view point to about 0.6 mm after integration of 9 view points. Graph (2) shows the depth error during joint estimation of motion and depth. Even with the additional error introduced from motion tracking the depth error approaches 1 mm, which is below 1 $^0/_{00}$ of sensing range.

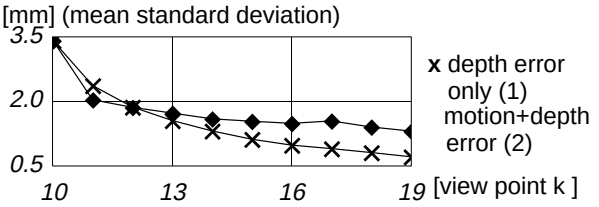

Fig. 5: Depth fusion with kalman filtering

4.3 Surface Fusion

Surface update with a kalman filter can be applied as long as the surface is visible in all views. Once a new and previous unseen surface needs to be modeled, this surface must be appended to the existing surface model. This surface fusion is carried out by searching along the existing open surface boundaries. The open boundaries of a surface are projected into the current image and compared with the surface segmentation from the current image pair. After finding a new object surface in the neighborhood of a boundary, the 3–D wire–frame generation is applied to the new surface and the new surface is included into the model. The resulting surface model is closed by using this process repeatedly for a number of view points around the object.

5 Conclusions

An automatic 3–D scene modeling system was discussed that aims to build complete surface models from stereoscop-

ic image sequences. It uses calibrated and rectified image sequences to obtain dense depth maps of the scene, compute 3–D surface models from different view points, and to fuse the surfaces from different view points into a consistent object model. The system successfully models a variety of scenes assuming piecewise smooth surfaces.

The objects modeled so far were taken in a laboratory environment to control the different scene parameters and to judge the performance. The next step will be to record outdoor scenes of real buildings and to look at the influence of such conditions onto the robustness of the modeling process. A portable stereoscopic image recording system is currently being built and tested.

Acknowledgement

This work has been supported by a grant of the German postal service TELEKOM and by the "Stiftung Volkswagenwerk". I thank Federico Pedersini for his version of camera calibration, Michael Werman to discuss kalman filtering, and many students for their contribution to implement the actual system.

References

[1] Durisch, P. "Photogrammetry and Computer Graphics for Visual Impact Analysis in Architecture", *Proceedings of ISPRS Conference 1992*, Vol. 29, B5, pp. 434–445.

[2] Blonde, L.,"The MONA LISA Project: General Presentation", *Proceedings on the European Workshop in Combined Real and Synthetic Image Processing for Broadcast and Video Production*, VAP Media Centre, Hamburg, Germany, Nov. 1994.

[3] H. Harashima, F. Kishino, "Intelligent Image Coding and Communications with Realistic Sensations – Recent Trends,", *IEICE Transactions*, Vol. E 74 (6), pp. 1582–1592, June 1991.

[4] R. A. Jarvis, "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5 (2), pp. 122–139, March 1983.

[5] H.H. Baker, T.O. Binford: "Depth from edge and intensity based stereo," *Proc. seventh Int. joint Conf. Artif. Intell*. pp. 632–636, 1981.

[6] J. Aloimonos, D. Shulman, "Integration of Visual Modules," *Academic Press*, San Diego, USA, 1989.

[7] J.K. Aggarwal, N. Nandhakumar, "On the Computation of Motion from Sequences of Images –A Review," *Proc. of the IEEE*, Vol. 76 (8), pp. 917–935, Aug. 1988.

[8] A.N. Netravali, J. Salz, "Algorithms for Estimation of Three–Dimensional Motion," *AT&T Technical Journal*, Vol. 64 (2), 1985.

[9] D. Terzopoulos , D. Metaxas, "Dynamic 3D Models with Local and Global Deformations: Deformable superquadratics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13 (7), pp. 703–714, July 1991.

[10] A. Pentland, B. Horowitz, "Recovery of Nonrigid Motion and Structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13 (7), pp. 730–742, July 1991.

[11] Tomasi, C., Kanade, T, "Shape and motion from image streams under orthography: A factorization method", *IJCV 9*, 1992, pp. 137–154.

[12] Robert, Ph, Ogor, F.,"Joint Estimation of depth maps and camera motion in the Construction of 3D Models from a Mobile Camera", *Proceedings on the European Workshop in Combined Real and Synthetic Image Processing for Broadcast and Video Production*, VAP Media Centre, Hamburg, Germany, Nov. 1994.

[13] Tirumalai, A.P., Schunck, B.G., Jain, R.C.,"Dynamic Stereo with Self–Calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14 (12), pp. 1184–1189, Dec. 1992.

[14] Yakimowski,Y., Cunningham,R,"A System for Extracting 3D Measurement from a Stereo Pair of TV Cameras", *CVGIP, Vol 7*,pp 195–210, 1978.

[15] Pedersini, F., Tubaro, S., Rocca, F., 1993, "Camera Calibration and Error Analysis. An application to Binocular and Trinocular Stereoscopic System.", *4th International Workshop on time–varying Image Processing and moving–object recognition*, Florence, Italy.

[16] Tsai, R.Y.,"A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology using off–the–shelf TV Cameras and Lenses", *IBM Research Report RC 11413*, Sept. 1985, Yorktown Heights, NY, USA.

[17] Falkenhagen, L.," Depth Estimation from Stereoscopic Image Pairs Assuming Piecewise Continuos Surfaces", *European Workshop on Combined real and synthetic image processing for broadcast and video productions*, 23–24. 11. 1994, Hamburg, Germany.

[18] Cox, I., Hingorani, S., Maggs, B., Rao, S.," Stereo without Regularisation", *British Machine Vision Conference*, Leeds, UK, pp. 337–346, David Hogg & Roger Boyle (ed.), Springer Verlag, 1992.

[19] Gruen, A.W.,"Adaptive least squares correlation: A powerful image matching technique", *South African Journal of Photogrammetry, remote Sensing and Cartography*, Vol 14(3), 1985.

[20] Brink, G.,"Subpixelgenaue Disparitätsschätzung aus Stereobildpaaren", *Master Thesis*, Universität Hannover, 1994.

[21] Terzopoulos, D., "The computation of visible–surface representations", *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol 10, pp.417–438, 1988.

[22] Kappei, F., "Modellierung und Rekonstruktion bewegter dreidimensionaler Objekte aus einer Fernsehbildfolge", *Ph.D. Thesis*, University of Hannover,1988.

[23] Koch, R. "Automatic Modelling of Natural Scenes for Generating Synthetic Movies". *In: Vandoni, C.E. and Duce, D.A. (ed.) Eurographics Association 1990.* Elsevier Science Publishers B.V. (North–Holland).

[24] R. Koch,"Dynamic 3D Scene Analysis through Synthesis Feedback Control", *IEEE Trans. Patt. Anal. Mach. Intell., Special issue on analysis and synthesis*, Vol. 15:6, June 1993.

[25] Koch, R., "Automatic Reconstruction of Buildings from Stereoscopic Image Sequences", *Eurographics '93*, Barcelona,Spain, 1993.

[26] Brammer, K, Siffling, G. "Kalman–Bucy–Filter: Deterministische Beobachtung und stochastische Filterung". *R. Oldenbourg Verlag*, München, 1985.