Automatic Reconstruction of Stationary 3-D Objects from Multiple Uncalibrated Camera Views

Peter Eisert, Eckehard Steinbach, and Bernd Girod, *Fellow, IEEE*

***Abstract*—A system for the automatic reconstruction of real-world objects from multiple uncalibrated camera views is presented. The camera position and orientation for all views, the 3-D shape of the rigid object, as well as the associated color information, are recovered from the image sequence. The system proceeds in four steps. First, the internal camera parameters describing the imaging geometry are calibrated using a reference object. Second, an initial 3-D description of the object is computed from two views. This model information is then used in a third step to estimate the camera positions for all available views using a novel linear 3-D motion and shape estimation algorithm. The main feature of this third step is the simultaneous estimation of 3-D camera-motion parameters and object shape refinement with respect to the initial 3-D model. The initial 3-D shape model exhibits only a few degrees of freedom and the object shape refinement is defined as flexible deformation of the initial shape model. Our formulation of the shape deformation allows the object texture to slide on the surface, which differs from traditional flexible body modeling. This novel combined shape and motion estimation using *sliding texture* considerably improves the calibration data of the individual views in comparison to fixed-shape model-based camera-motion estimation. Since the shape model used for model-based camera-motion estimation is only approximate, a volumetric 3-D reconstruction process is initiated in the fourth step that combines the information from all views simultaneously. The recovered object consists of a set of voxels with associated color information that describes even fine structures and details of the object. New views of the object can be rendered from the recovered 3-D model, which has potential applications in virtual reality or multimedia systems and the emerging field of video coding using 3-D scene models.**

***Index Terms*—Shape and motion estimation, view calibration, 3-D reconstruction.**

I. Introduction

OBTAINING computer models of real world objects is a very active research area with applications in Virtual Reality (VR) and multimedia systems. A common approach to obtain photorealistic object descriptions uses multiple camera views from different positions around the object and attempts to fuse this information into a complete 3-D description of the object.

For 3-D reconstruction from multiple views, two basic classes of algorithms can be distinguished. The first class of algorithms computes depth maps from two or more views of the object and then registers the depth maps into a single 3-D surface model. The depth-map recovery often relies on sparse or dense matching of image points with subsequent 3-D structure estimation [1], [16], [23] or is supported by additional depth information from range sensors [33], [3], [19]. The second class of algorithms is based on volume intersection, and is often referred to as *shape-from-silhouette* algorithms [2], [25], [21], [30], [9]. The object shape is typically computed as the intersection of the outline cones which are back-projected from all available views of the object. This requires the reliable extraction of the object contour in all views which restricts the applicability to scenes where the object can be easily segmented from the background.

In this paper, we combine the shape-from-silhouette and stereo approaches by using a volumetric representation of the 3-D object and multi-hypothesis testing of the projected object surface voxels with the camera views [5]. The color of the surface is incorporated in a uniform framework, so that the object shape can be estimated accurately in regions, where the silhouette information is not sufficient.

All these approaches for 3-D reconstruction have in common that camera pose and orientation for all views must be known. If they are not available, view calibration has to be performed before the actual 3-D reconstruction can be conducted. Typically, point correspondences are tracked over all views as, for instance, proposed in [9] and the internal and external camera parameters are computed from these point correspondences [7], [12]. In contrast, the approach presented in this paper uses a model-based camera-motion estimator that exploits information from the entire image. Given an accurate 3-D model of a scene, e.g., a 3-D laser scan as used in [4], the motion parameters of a moving camera can be recovered with high accuracy using intensity gradient-based algorithms. However, model errors considerably reduce the accuracy of this kind of motion-estimation methods. We therefore extend traditional model-based motion estimation [18], [15] to combined shape and motion estimation. The advantage of simultaneous shape and motion estimation is a tight coupling of all available views since shape updates have to be consistent within all views.

Our formulation of the object shape refinement differs from traditional flexible body modeling. Traditionally, the texture is extracted from one frame and is mapped onto the 3-D surface leading to a perfect rendered reproduction of this frame independent of the object shape. After object surface deformation, however, the projection of the model leads to a distorted version of this initial frame. In our approach, the texture is not fixed to the object surface but can slide on it in combination with surface deformation. This *sliding texture* concept ensures that the projection of the object into the initial view always remains undistorted independent of the estimated shape refinement.

Manuscript received March 15, 1999; revised September 30, 1999. This paper was recommended by Guest Editor Y. Wang.

The authors are with the Telecommunications Laboratory, University of Erlangen-Nuremberg, D-91058 Erlangen, Germany (e-mail: eisert@lnt.de; steinb@lnt.de; girod@lnt.de).

Publisher Item Identifier S 1051-8215(00)02018-8.

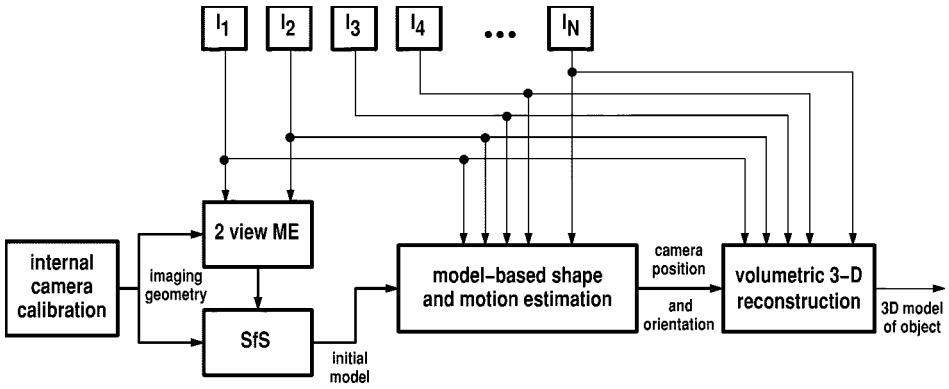


Fig. 1. Architecture of the automatic 3-D reconstruction system from N input views $I_1, \cdots, I_N$. *2-view ME:* 3-D camera-motion estimation from two views without knowledge about the object shape. *SfS:* dense map of depth value computation using the structure-from-stereo paradigm. *Model-based shape and camera-motion estimation:* simultaneous 3-D camera-motion estimation and shape refinement. *Volumetric 3-D reconstruction:* final 3-D structure and color recovery from N calibrated views.

A generic reduced resolution subdivision surface model is employed to approximate the object geometry. Initially, this generic model is spherical and is adapted to the object under investigation exploiting the silhouette of the object together with depth information from the first two views. This depth information is recovered using structure-from-stereo together with the relative camera-motion estimate obtained from the 3-D motion-estimation algorithm introduced in [28]. The resulting approximative object shape is then used to estimate the position and orientation of all cameras.

This paper is organized as follows. In Section II, the system architecture for automatic 3-D reconstruction from multiple uncalibrated camera views is presented. We discuss the different processing steps that involve: 1) camera calibration; 2) initial model construction; 3) view calibration using the *sliding texture* concept; and 4) volumetric 3-D reconstruction from all views. View calibration is treated in Section V, where we describe the combined 3-D rigid body shape refinement and camera-motion estimation algorithm. Simulation results illustrate the improved motion-estimation accuracy which can be achieved with the proposed combined motion and shape estimation framework. After these steps, the internal and external camera parameters are available for all views. These calibration data are exploited by the volumetric object reconstruction algorithm described in Section VI.

II. Automatic 3-D Reconstruction from Multiple Camera Views

Fig. 1 shows the proposed system architecture for automatic 3-D object reconstruction from multiple uncalibrated views. The block diagram in Fig. 1 includes the following processing units.

1) *Camera Calibration:* A reference 3-D object is used to calibrate the imaging geometry of the camera. The 3-D reference body is captured and the internal camera parameters are determined from the 3-D to 2-D point correspondences using the standard technique proposed by Tsai in [32]. After initial camera calibration, we assume the internal camera parameters to remain constant for all views.
2) *Camera-Motion Estimation from Two Views (2-view ME):* Using the first two views of the scene, I_1 and I_2, the relative motion between the camera and the object can be estimated under the assumption of rigid body motion. This assumption holds for a static rigid object in front of a moving camera.
3) *Structure-from-Stereo (SfS):* Given the relative camera-motion parameters from view I_1 to I_2, a dense map of depth values is computed using the SfS paradigm. Neighborhood, ordering, and smoothness constraints, as well as explicit occlusion detection, are used during depth computation in order to obtain a reliable depth estimate for each pixel in frame I_1.
4) *Model-Based Shape and Camera-Motion Estimation:* The depth map resulting from the SfS step is used to adapt a generic 3-D shape model to the object. The number of degrees of freedom of this initial shape model is limited resulting in an approximative geometry of the object. A linear, intensity gradient-based algorithm estimates the relative camera position and pose for all views, in combination with a shape refinement of the approximate object model.
5) *Volumetric Reconstruction:* Since all camera positions and the imaging geometry are now available, we discard the approximative geometry of the previous step and perform the actual 3-D reconstruction of the object under investigation. A volumetric reconstruction of the 3-D object is performed that leads to a set of object voxels with associated color information. Arbitrary new views can now be created via rendering of the reconstructed voxel volume.

III. Camera Calibration

All steps described so far require knowledge about the imaging geometry, i.e., the focal length, viewing angle, or pixel geometry of our pinhole camera model. These parameters are often referred to as the internal parameters of the camera. We determine these parameters for our experimental setup in advance, using a manufactured calibration object with known 3-D point coordinates.

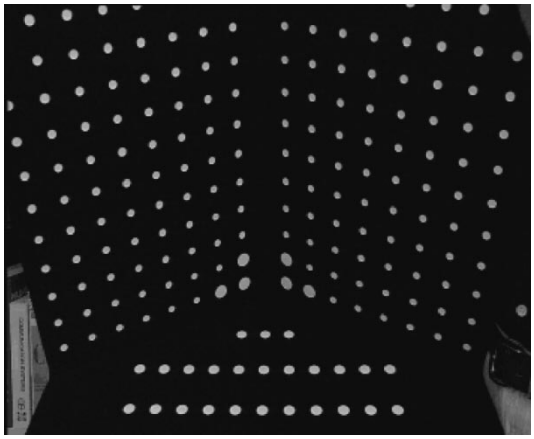


Fig. 2. View of the calibration object that is used to determine the internal pinhole camera parameters: focal length, viewing angle, and pixel geometry.

The calibration object shown in Fig. 2 is captured with the camera. The calibration points are automatically detected in the image with subpixel accuracy [14], [27] and the 3-D to 2-D point correspondences are used for camera calibration using the method described in [32]. The camera calibration delivers the focal length, the viewing angle and the pixel geometry which can be summarized by means of the horizontally and vertically scaled focal length values f_x and f_y. These two values specify the imaging geometry of a pinhole camera model and are assumed to be constant after calibration. The imaging geometry describes the projection of 3-D world object points (x, y, z) into pixel coordinates (X, Y) in the image plane as follows:

$$X = -f_x \frac{x}{z}, \quad Y = -f_y \frac{y}{z}. \tag{1}$$

IV. Shape Model Initialization from Two Views

We assume that no information about the 3-D shape of the object and the camera positions is available at the beginning. Therefore, we first extract an initial 3-D description of the object from the first two neighboring frames I_1 and I_2 using the 3-D motion-estimation algorithm proposed in [11] and [28], in combination with the computation of a dense map of depth values. This initial shape model is later refined during the view calibration of the remaining frames (Section V-C).

Since the object is static and only the camera is moving, the result of the motion-estimation step is the relative camera motion between these two frames. The recovered motion parameters consist of the rotation around the coordinate axes given by the rotation matrix $\mathbf{R}$ and the direction of the translation vector $\mathbf{t}$. It is well known [31] that from two views of an image sequence, one can expect to recover 3-D rotation, but 3-D translation only up to a scale factor. Given this relative motion between views I_1 and I_2, it is now possible to recover scene structure information using the structure-from-stereo paradigm [31]. Since the absolute length of the translation vector cannot be determined, the recovered depth will contain the same scale factor. This scale factor remains constant during all following steps and can be neglected without loss of generality.

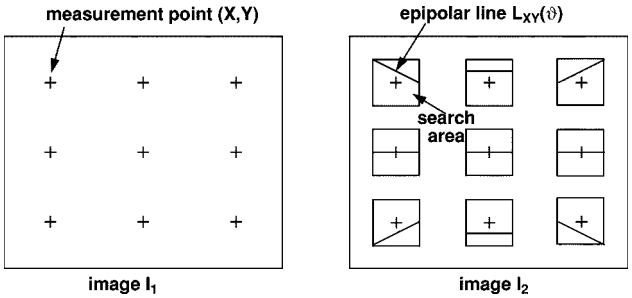


Fig. 3. Illustration of the rigid body constraint in the image plane. The epipolar line constrains the search space for point correspondences to a 1-D search space.

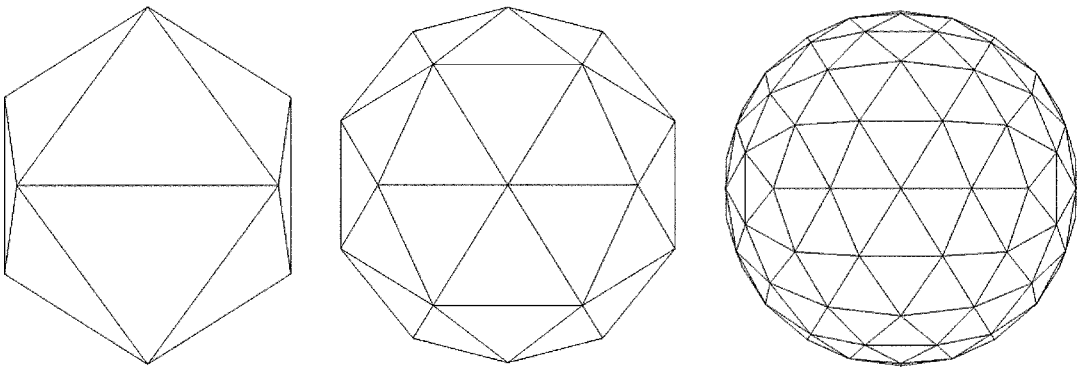
In the next section, we describe the 3-D motion-estimation approach. In Section IV-B, we explain how the dense map of depth values is recovered given the estimated relative camera motion between the two frames.

A. 3-D Camera-Motion Estimation

The motion-estimation approach in [11] and [28] is based on the observation that 3-D rigid body motion constrains the point correspondences in two views to lie on a straight line in the image plane, the epipolar line. The mathematical derivation of the straight line equation as a function of the 3-D motion parameters $\vartheta = \{\mathbf{R}, \mathbf{t}\}$, the imaging geometry, and the image-plane location (X, Y) is given, e.g., in [31]. The epipolar line for the motion parameter set ϑ computed for the pixel (X, Y) in frame I_1 will be denoted by $L_{XY}(\vartheta)$ in the following. Assuming a particular motion ϑ, we can calculate the epipolar line $L_{XY}(\vartheta)$ in I_2. A maximum horizontal and vertical displacement in the image plane (e.g., ± 30 pixels) defines a 2-D search area around the point (X, Y) inside which the point correspondences are assumed to fall. The rigid body constraint now reduces the 2-D search area in the vicinity of a measurement point to the intersection of the epipolar line with the 2-D search area. The point correspondence problem therefore becomes a 1-D search problem under a candidate motion parameter set ϑ. Fig. 3 illustrates the situation for $F = 9$ measurement points (+) in image I_1 and the corresponding 1-D search spaces for the point correspondences along the epipolar lines. Since the epipolar line is a function of the motion parameters and the image-plane location, the line typically changes its slope and intercept from measurement point to measurement point. All point correspondences have to lie on the corresponding epipolar lines $L_{XY}(\vartheta)$ in view I_2. In order to determine the position of the point correspondence along the epipolar line, the mean-squared error (MSE) evaluated over a measurement window of size (N, M) which is centered around the measurement point (X, Y)

$$\mathrm{MSE}(X, Y, d_x, d_y) = \frac{1}{MN} \sum_{r=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{s=-\frac{M-1}{2}}^{\frac{M-1}{2}} (I_1(X+r, Y+s) - I_2(X + d_x + r, Y + d_y + s))^2 \tag{2}$$

is evaluated, with I_1 and I_2 denoting the intensity values in views 1 and 2, respectively, and d_x, d_y being the displacements in horizontal and vertical direction. Since the actual motion parameter set ϑ is unknown *a priori*, candidate motion parameter



(a) (b) (c)

Fig. 4. Shape model with increasing resolution. (a) Icosahedron (level 0) with 12 control points and 20 triangles. (b) First level with 42 control points and 80 triangles.(c) Second level with 162 control points and 320 triangles.

sets have to be selected from the motion parameter space. In order to decide which candidate motion fits best to the actual image data, the following cost function is employed:

$$c(\vartheta) = \sum_{m=1}^{F} \min_{\{(X_m+d_x, Y_m+d_y) \in L_{X_m Y_m}(\vartheta)\}} \cdot \mathrm{MSE}(X_m, Y_m, d_x, d_y) \quad (3)$$

with F being the total number of measurement locations selected in view I_1. The cost function in (3) is the minimum MSE detected along the epipolar lines accumulated over all F measurement locations. The motion parameter set ϑ leading to the smallest value for $c(\vartheta)$ in (3) is the estimated motion. The actual minimization of the cost function in (3) is performed using initial estimates produced with the linear 8-point algorithm in [31], with subsequent optimization using Powell's conjugate direction search method [22].

B. Computation of a Dense Map of Depth Values

Given the motion parameters ϑ from view I_1 to I_2, we recover a dense map of depth values using the structure-from-stereo paradigm which says that the depth at pixel position (X, Y) is a function of the estimated motion parameters ϑ, the measured image-plane displacements for this point (d_x, d_y) and the imaging geometry of the camera described by the constants f_x and f_y

$$z = f(\vartheta, X, Y, d_x, d_y, f_x, f_y) \quad (4)$$

with

$$(X + d_x, Y + d_y) \in L_{XY}(\vartheta). \quad (5)$$

For explicit occlusion detection and incorporation of neighborhood constraints, we use a modified version of the stereo-depth estimation algorithm by Falkenhagen [6] which is based on dynamic programming and recovers depth estimates for all pixels in the image. Our modifications include the extension of stereo constraints (neighboring and ordering, smoothness, occlusion, etc.) to arbitrary epipolar geometry [7], and adaptive cost functions for the explicit determination of occlusions.

C. Generic 3-D Shape Model

Given the dense map of depth, we construct a 3-D shape model that is used in the following to determine the camera motion for all views with a model-based estimator. Since the shape information recovered from only two views is neither complete nor very accurate, the initial shape is refined when incorporating new views. In order to restrict the number of degrees of freedom for the shape and reduce the complexity of the shape refinement, we use an approximative 3-D shape model that is based on an icosahedron. The icosahedron is defined by 12 control points (vertices) which form a triangular mesh as illustrated on the left-hand side of Fig. 4. Only 12 control points are not sufficient to describe arbitrary object shapes. Therefore, the icosahedron is recursively subdivided until the desired resolution or the desired number of control points is reached. This subdivision is illustrated in Fig. 4. The number of control points as a function of the subdivision level l can be computed as

$$N_{\mathrm{CP}} = 12 + 10(4^l - 1). \quad (6)$$

The control points can be displaced for shape approximation of an individual object. For increased estimation robustness, we restrict the movement of control points to be radial only. The advantage of this restriction is a decoupling of local shape deformation from global rotation and translation of the entire object.

D. 3-D Shape Model Initialization

The generic 3-D shape model introduced in the previous section is initially spherical. In the general case, this spherical shape deviates considerably from the actual object shape. In order to facilitate the shape estimation, we exploit silhouette and depth information from the first view to adapt the generic model to the individual object shape.

In the first step, the icosahedron is placed in the 3-D space such that the projection into the first frame encloses the entire object. In the next step, the control points of the icosahedron that are projected outside of the object silhouette are scaled toward the object. This initialization process is illustrated in Fig. 5 for a 2-D cross section.

In addition to the silhouette initialization, the dense map of depth values computed for the first frame as described in Section IV-B is used for further 3-D shape model adaption. All con-

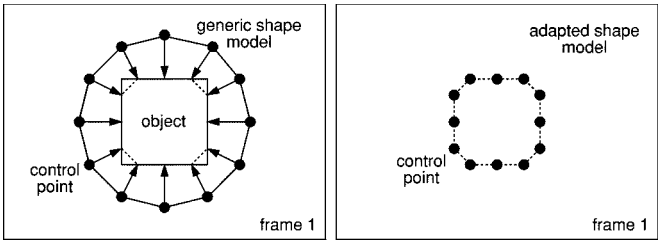


Fig. 5. Shape initialization of the generic model using silhouette information. The control points are radially scaled toward the silhouette.

trol points facing the camera are scaled according to the depth map. Those control points representing nonvisible parts of the object are not scaled. The integration of the depth information leads to a considerable improvement of the initial shape model and facilitates further shape estimation.

V. Model-Based 3-D Motion and Shape Estimation

With the adapted generic shape model, an initial 3-D description of the scene is available. As stated in the introduction, model-based motion estimation leads to good camera position and pose estimates if the available model is very accurate. This is true for instance, if the model stems from a laser scanner as described in [4]. If the model only approximates the 3-D shape of the object, as it is the case for the adapted generic shape model in Section IV-D, the motion estimates reflect these model errors. Conversely, if the motion estimate between two views contains an error, the structure-from-stereo algorithm will produce an erroneous depth-map. This mutual dependency of shape and motion motivated the investigations of this section.

In the following, we derive an algorithm that allows the simultaneous estimation of 3-D motion parameters and 3-D shape refinement from two or more views of an object. The approach is based on the evaluation of spatial and temporal intensity gradients and leads to a set of linear equations for the unknown motion and shape parameters that can be solved with low computational complexity.

The 3-D model of the object as computed in Section IV-D delivers shape but no texture information. Therefore, the texture is extracted from the first view I_1. The surface points of the 3-D shape model, with respect to the object center, are denoted as $\mathbf{x_0}$ in the following. As shown in Fig. 6, a 3-D object point with respect to the first camera view I_1 is then described as

$$\mathbf{x_1} = \mathbf{R_1 x_0} + \mathbf{t_1}. \tag{7}$$

For a second view I_2, this transform becomes

$$\mathbf{x_2} = \mathbf{R_2 x_0} + \mathbf{t_2}. \tag{8}$$

The color of an object point associated with $\mathbf{x_1}$ is determined in view I_1 by the color value at pixel position (X_1, Y_1), with

$$X_1 = -f_x \frac{x_1}{z_1}, \quad Y_1 = -f_y \frac{y_1}{z_1}. \tag{9}$$

The relative 3-D motion from view I_1 to view I_2, together with the shape information from the 3-D model, allows to generate a motion compensated approximation of frame I_2. The relative motion between the two frames is described by

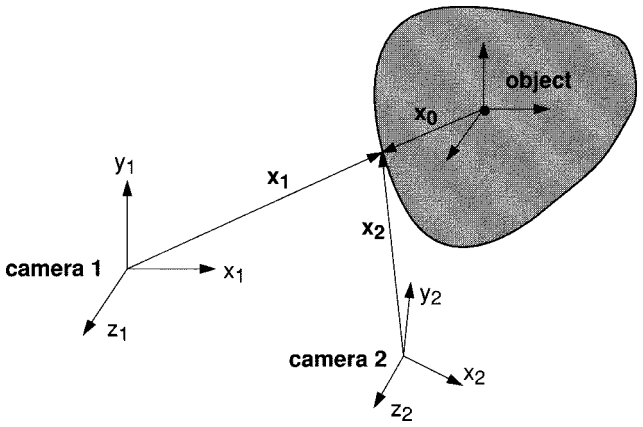


Fig. 6. Object and camera coordinate systems for two camera views.

$$\begin{aligned}\mathbf{x_2} &= \mathbf{R_2 R_1^{-1}}(\mathbf{x_1} - \mathbf{t_1}) + \mathbf{t_2} \\ &= \mathbf{R_{12}}(\mathbf{x_1} - \mathbf{t_1}) + \mathbf{t_1} + \mathbf{t_{12}} \\ \mathbf{x_1} &= \mathbf{R_1 R_2^{-1}}(\mathbf{x_2} - \mathbf{t_2}) + \mathbf{t_1} \\ &= \mathbf{R_{12}^{-1}}(\mathbf{x_2} - (\mathbf{t_1} + \mathbf{t_{12}})) + \mathbf{t_1}.\end{aligned} \tag{10}$$

The motion compensation for each pixel (X_2, Y_2) in frame I_2 requires the determination of the corresponding pixel coordinates (X_1, Y_1) in frame I_1. We first determine the 3-D object point coordinates $\mathbf{x_2}$ from (X_2, Y_2) by

$$\mathbf{x_2} = \left[-\frac{X_2 z_2}{f_x}, -\frac{Y_2 z_2}{f_y}, z_2 \right]^T. \tag{11}$$

The depth z_2 at position (X_2, Y_2) is obtained by projecting the 3-D shape into view I_2. The corresponding 3-D point coordinates $\mathbf{x_1}$ for the first view is computed using the relation in (10) and finally the color is extracted from the projection in (9).

To summarize, the color value at pixel position (X_2, Y_2) in frame I_2 is a function of the motion parameters $\mathbf{R_{12}}$ and $\mathbf{t_{12}}$, the object depth z_2, the initial object position $\mathbf{t_1}$, and the color value in frame I_1

$$I_2(X_2, Y_2) = f(I_1(X_1, Y_1), z_2, \mathbf{R_{12}}, \mathbf{t_{12}}, \mathbf{t_1}). \tag{12}$$

Erroneous motion parameters $\hat{\mathbf{R}}_{\mathbf{12}}$ and $\hat{\mathbf{t}}_{\mathbf{12}}$ or inaccurate depth $\hat{z}_2$ due to 3-D shape errors lead to an imperfect motion-compensated frame $\hat{I}_2$. In other words, the color differences between $\hat{I}_2$ and I_2 depend on the accuracy of the motion parameters $\hat{\mathbf{R}}_{\mathbf{12}}$ and $\hat{\mathbf{t}}_{\mathbf{12}}$ and the accuracy of the 3-D shape model employed. The frame difference between $\hat{I}_2$ and I_2 can be used to refine either the motion parameters or the shape, or both.

The following sections formalize this insight. Section V-A first derives the estimation equations for the case of correct shape but the wrong motion parameters. Section V-B then assumes correct motion and shows how shape errors can be estimated using a novel *sliding texture* formulation. Section V-C finally combines both effects into a common estimation framework.

A. Model-Based 3-D Rigid Body Motion Estimation

In this section, a correct 3-D shape model is assumed and the image synthesis error after motion compensation from I_1 to I_2 is used to refine the 3-D rigid body motion parameters. Explicit modeling of the motion parameter error leads to the following expression for the object point location $\mathbf{x_2}$ for view I_2

$$\begin{aligned}\mathbf{x_2} &= \Delta\mathbf{R}(\hat{\mathbf{x}}_\mathbf{2} - (\mathbf{t_1} + \hat{\mathbf{t}}_\mathbf{12})) + \mathbf{t_1} + \hat{\mathbf{t}}_\mathbf{12} + \Delta\mathbf{t} \\ &= \Delta\mathbf{R}(\hat{\mathbf{x}}_\mathbf{2} - \mathbf{x_c}) + \mathbf{x_c} + \Delta\mathbf{t} \end{aligned} \tag{13}$$

with the unknown motion errors $\Delta\mathbf{R}$ and $\Delta\mathbf{t}$ and the object center $\mathbf{x_c} = \mathbf{t_1} + \hat{\mathbf{t}}_\mathbf{12}$, with respect to $\hat{I}_2$. Under the assumption that the rotation angles in $\Delta\mathbf{R}$ are small, we can linearize the rotation matrix

$$\Delta\mathbf{R} \approx \begin{bmatrix} 1 & -\Delta R_z & \Delta R_y \\ \Delta R_z & 1 & -\Delta R_x \\ -\Delta R_y & \Delta R_x & 1 \end{bmatrix} \tag{14}$$

where ΔR_x, ΔR_y, and ΔR_z are the rotational angles around the x-, y-, and z-axis.

The resulting displacement error (u_m, v_m) between $\hat{I}_2(\hat{X}_2, \hat{Y}_2)$ and $I_2(X_2, Y_2)$ can then be described after first order Taylor expansion as [4]

$$\begin{aligned} u_m &= X_2 - \hat{X}_2 \\ &\approx f_x \left[-\Delta R_y \left(1 - \frac{z_c}{\hat{z}_2}\right) - \Delta R_z \left(\frac{\hat{Y}_2}{f_y} + \frac{y_c}{\hat{z}_2}\right) \right. \\ &\quad - \frac{\Delta t_x}{\hat{z}_2} + \frac{\hat{X}_2}{f_x}\left(\Delta R_x \left(\frac{\hat{Y}_2}{f_y} + \frac{y_c}{\hat{z}_2}\right)\right. \\ &\quad \left. \left. - \Delta R_y \left(\frac{\hat{X}_2}{f_x} + \frac{x_c}{\hat{z}_2}\right) - \frac{\Delta t_z}{\hat{z}_2}\right)\right] \\ v_m &= Y_2 - \hat{Y}_2 \\ &\approx f_y \left[\Delta R_x \left(1 - \frac{z_c}{\hat{z}_2}\right) + \Delta R_z \left(\frac{\hat{X}_2}{f_x} + \frac{x_c}{\hat{z}_2}\right) \right. \\ &\quad - \frac{\Delta t_y}{\hat{z}_2} + \frac{\hat{Y}_2}{f_y}\left(\Delta R_x \left(\frac{\hat{Y}_2}{f_y} + \frac{y_c}{\hat{z}_2}\right)\right. \\ &\quad \left. \left. - \Delta R_y \left(\frac{\hat{X}_2}{f_x} + \frac{x_c}{\hat{z}_2}\right) - \frac{\Delta t_z}{\hat{z}_2}\right)\right] \end{aligned} \tag{15}$$

with $\hat{z}_2$ being the depth obtained from the model after rendering with the erroneous motion parameters $\hat{\mathbf{R}}_\mathbf{12}$ and $\hat{\mathbf{t}}_\mathbf{12}$. Combining this description of rigid body motion with the optical flow constraint equation [13]

$$\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \cdot u_m + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \cdot v_m = \hat{I}_2 - I_2 \tag{16}$$

results in a linear equation for the six unknown motion parameters

$$\begin{aligned} a_0\Delta R_x + a_1\Delta R_y + a_2\Delta R_z + a_3\Delta t_x + a_4\Delta t_y + a_5\Delta t_z \\ = \hat{I}_2 - I_2 \end{aligned} \tag{17}$$

with a_0 to a_5 given as

$$\begin{aligned} a_0 &= \frac{\partial \hat{I}_2}{\partial \hat{X}_2}\hat{X}_2\left(\frac{\hat{Y}_2}{f_y} + \frac{y_c}{\hat{z}_2}\right) \\ &\quad + f_y \frac{\partial \hat{I}_2}{\partial \hat{Y}_2}\left(1 - \frac{z_c}{\hat{z}_2} + \frac{\hat{Y}_2}{f_y}\left(\frac{\hat{Y}_2}{f_y} + \frac{y_c}{\hat{z}_2}\right)\right) \\ a_1 &= -f_x \frac{\partial \hat{I}_2}{\partial \hat{X}_2}\left(1 - \frac{y_c}{\hat{z}_2} + \frac{\hat{X}_2}{f_x}\left(\frac{\hat{X}_2}{f_x} + \frac{x_c}{\hat{z}_2}\right)\right) \\ &\quad - \frac{\partial \hat{I}_2}{\partial \hat{Y}_2}\hat{Y}_2\left(\frac{\hat{X}_2}{f_x} + \frac{x_c}{\hat{z}_2}\right) \\ a_2 &= -f_x \frac{\partial \hat{I}_2}{\partial \hat{X}_2}\left(\frac{\hat{Y}_2}{f_y} + \frac{y_c}{\hat{z}_2}\right) + f_y \frac{\partial \hat{I}_2}{\partial \hat{Y}_2}\left(\frac{\hat{X}_2}{f_x} + \frac{x_c}{\hat{z}_2}\right) \\ a_3 &= -f_x \frac{\partial \hat{I}_2}{\partial \hat{X}_2}\frac{1}{\hat{z}_2} \\ a_4 &= -f_y \frac{\partial \hat{I}_2}{\partial \hat{Y}_2}\frac{1}{\hat{z}_2} \\ a_5 &= -\frac{\partial \hat{I}_2}{\partial \hat{X}_2}\frac{\hat{X}_2}{\hat{z}_2} - \frac{\partial \hat{I}_2}{\partial \hat{Y}_2}\frac{\hat{Y}_2}{\hat{z}_2}. \end{aligned} \tag{18}$$

At least six equations are necessary for the algorithm to determine the motion parameters, but due to the large number of object pixels, an over-determined linear system of equations is obtained, and is solved in a least-squares sense.

The inherent linearization of the intensity in the optical flow constraint and the approximations used for obtaining a linear solution do not allow dealing with large displacements between two views. To overcome this limitation, a hierarchical scheme is used for the motion estimation. First, an approximation for the parameters is computed from low-pass filtered and sub-sampled images where the linear intensity assumption is valid over a wider range. With the estimated parameter set, a motion-compensated image is generated by simply moving the 3-D model and rendering it at the new position. Due to motion compensation, the differences between the new synthetic image and the camera frame decrease. Then, the procedure is repeated at higher resolutions, each time yielding a more accurate motion parameter set. In our current implementation, we use three levels of resolution, starting from 88 × 72 pixels. For each new level, the resolution is doubled in both directions, leading to a final resolution of 352 × 288 pixels (CIF). Experiments with this hierarchical scheme show that displacements of up to 30 pixels between two frames can be estimated.

B. 3-D Shape Estimation Using Sliding Textures

For 3-D shape estimation, the camera-motion parameters $\mathbf{R_{12}}$ and $\mathbf{t_{12}}$ are assumed to be correct and the object shape needs to be refined. The color value at pixel position (X_2, Y_2) in frame I_2 is a function of the motion parameters, object depth, and the initial object position, as shown in (12). Image synthesis after motion compensation from I_1 toward I_2 produces frame $\hat{I}_2$, which is a distorted version of I_2 due to the object shape errors. In the following, we describe how the intensity differences between $\hat{I}_2$ and I_2 can be exploited for object shape refinement.

As mentioned above, the control points of the shape model are constrained to move radially, with respect to the object center. Conventionally, the texture is extracted from frame I_1 and mapped onto the 3-D surface leading to a perfect reproduction of I_1 after rendering with an arbitrary shaped model. After object-surface deformation, however, the projection of the model leads to a distorted version of I_1. In our *sliding texture* approach, the texture is not fixed to the object surface, but can slide on it in combination with surface deformation. While the control points defining the object shape move radially, the texture slides along the line of sight for each pixel in I_1. This ensures that the projection of the model into I_1 always remains undistorted.

Fig. 7 illustrates the influence of the radial control point movement on the object surface and the *sliding texture* concept for a particular pixel location in view I_1. We assume that the model exhibits shape errors and denote the erroneous 3-D position of a visible object surface point caused by these errors as $\hat{\mathbf{x}}_\mathbf{1}$. Imagine that the 3-D point $\hat{\mathbf{x}}_\mathbf{1}$ is radially displaced to the new position $\hat{\mathbf{x}}_\mathbf{1} + (\hat{\mathbf{x}}_\mathbf{1} - \mathbf{t}_\mathbf{1})\Delta s$. Assuming a locally planar object surface described by the tangential plane in Fig. 7 and small shape refinements, the point $\mathbf{x}_\mathbf{1}$ is approximately lying on the object surface. This point represents the deformed object surface and is obtained via intersection of the shifted tangential plane through $\hat{\mathbf{x}}_\mathbf{1} + (\hat{\mathbf{x}}_\mathbf{1} - \mathbf{t}_\mathbf{1})\Delta s$ with the line of sight. Both 3-D points $\hat{\mathbf{x}}_\mathbf{1}$ and $\mathbf{x}_\mathbf{1}$ are projected to the same pixel position in the image plane. Please note that this deformation description differs from traditional flexible body modeling, where the color at $\hat{\mathbf{x}}_\mathbf{1}$ and $\hat{\mathbf{x}}_\mathbf{1} + (\hat{\mathbf{x}}_\mathbf{1} - \mathbf{t}_\mathbf{1})\Delta s$ would be identical. In our case, the color is not fixed for a 3-D point but slides along the line of sight. Therefore, the points $\hat{\mathbf{x}}_\mathbf{1}$ and $\mathbf{x}_\mathbf{1}$ have the same color which means that the texture moves from $\hat{\mathbf{x}}_\mathbf{1}$ to $\mathbf{x}_\mathbf{1}$ due to the object shape refinement.

For a given 3-D motion from view I_1 to I_2, surface deformations produce image-plane displacements which can be exploited for shape refinement. In order to arrive at a description of the image-plane displacements similar to the previous section, we first determine the point $\mathbf{x}_\mathbf{1}$ in Fig. 7. The tangential plane $\mathbf{x}_\mathbf{t}$ through a visible point $\hat{\mathbf{x}}_\mathbf{1}$ is given by

$$\mathbf{x_t} = \hat{\mathbf{x}}_\mathbf{1} + k\left[1, 0, \frac{\partial z_1}{\partial x_1}\bigg|_{\hat{\mathbf{x}}_\mathbf{1}}\right]^T + l\left[0, 1, \frac{\partial z_1}{\partial y_1}\bigg|_{\hat{\mathbf{x}}_\mathbf{1}}\right]^T \tag{19}$$

with

$$\frac{\partial z_1}{\partial x_1} = -\frac{\partial z_1}{\partial X_1}\frac{f_x}{z_1}, \quad \frac{\partial z_1}{\partial y_1} = -\frac{\partial z_1}{\partial Y_1}\frac{f_y}{z_1}. \tag{20}$$

Assuming that the surface normal at point $\hat{\mathbf{x}}_\mathbf{1}$ is facing the camera, it can be written as

$$\mathbf{n}_{\hat{\mathbf{x}}_\mathbf{1}} = \left[-\frac{\partial z_1}{\partial x_1}, -\frac{\partial z_1}{\partial y_1}, 1\right]^T\Bigg|_{\hat{\mathbf{x}}_\mathbf{1}}. \tag{21}$$

Since the control points can be moved in radial direction only, the deformation of the object surface can be locally modeled as a shift of the tangential plane as shown in Fig. 7. The shifted plane becomes

$$\mathbf{x_{ts}} = \mathbf{x_t} + (\hat{\mathbf{x}}_\mathbf{1} - \mathbf{t}_\mathbf{1})\Delta s. \tag{22}$$

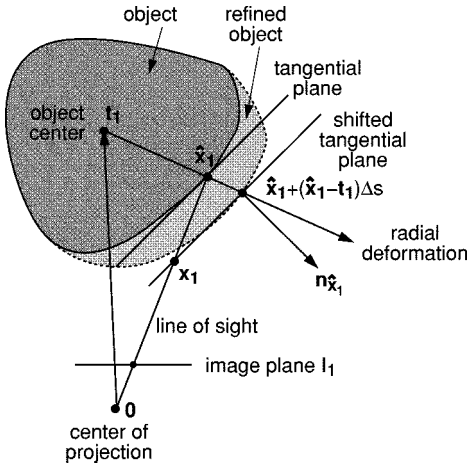


Fig. 7. Illustration of the radial shape deformation and the *sliding texture* concept.

This plane is then intersected with the line of sight $\mathbf{x_{ls}}$

$$\mathbf{x_{ls}} = \lambda\hat{\mathbf{x}}_\mathbf{1}, \tag{23}$$

leading to the new object point $\mathbf{x_1}$

$$\mathbf{x_1} = \hat{\mathbf{x}}_\mathbf{1}\left(1 + \Delta s\left(1 - \frac{\mathbf{t}_\mathbf{1}^T\mathbf{n}_{\hat{\mathbf{x}}_\mathbf{1}}}{\hat{\mathbf{x}}_\mathbf{1}^T\mathbf{n}_{\hat{\mathbf{x}}_\mathbf{1}}}\right)\right). \tag{24}$$

For a given 3-D motion $\mathbf{R_{12}}$ and $\mathbf{t_{12}}$ from frame I_1 to frame I_2, the points $\hat{\mathbf{x}}_\mathbf{1}$ and $\mathbf{x_1}$ project to the same image point in frame I_1 but to different image-plane positions in frame I_2. Assuming that $\hat{\mathbf{x}}_\mathbf{1}$ represents the erroneous object surface point position and $\mathbf{x_1}$ the correct position, the motion-compensated version of the erroneous object point $\hat{\mathbf{x}}_\mathbf{1}$ becomes

$$\hat{\mathbf{x}}_\mathbf{2} = \mathbf{R_{12}}(\hat{\mathbf{x}}_\mathbf{1} - \mathbf{t_1}) + \mathbf{t_1} + \mathbf{t_{12}}. \tag{25}$$

For the corresponding object point $\mathbf{x_1}$, after deformation we obtain

$$\mathbf{x_2} = \mathbf{R_{12}}(\mathbf{x_1} - \mathbf{t_1}) + \mathbf{t_1} + \mathbf{t_{12}}. \tag{26}$$

Projection into the image plane and Taylor series expansion truncated after the linear terms leads to the image displacements u_s and v_s in horizontal and vertical direction due to shape deformation

$$\begin{aligned} u_s &= X_2 - \hat{X}_2 \\ &\approx -\frac{\Delta s}{\hat{z}_2}\left(1 - \frac{\mathbf{t}_\mathbf{1}^T\mathbf{n}_{\hat{\mathbf{x}}_\mathbf{1}}}{\hat{\mathbf{x}}_\mathbf{1}^T\mathbf{n}_{\hat{\mathbf{x}}_\mathbf{1}}}\right)(f_x(\mathbf{r_1}\mathbf{t_1} - t_{1x} - t_{12x}) \\ &\quad + \hat{X}_2(\mathbf{r_3}\mathbf{t_1} - t_{1z} - t_{12z})) \\ v_s &= Y_2 - \hat{Y}_2 \\ &\approx -\frac{\Delta s}{\hat{z}_2}\left(1 - \frac{\mathbf{t}_\mathbf{1}^T\mathbf{n}_{\hat{\mathbf{x}}_\mathbf{1}}}{\hat{\mathbf{x}}_\mathbf{1}^T\mathbf{n}_{\hat{\mathbf{x}}_\mathbf{1}}}\right)(f_y(\mathbf{r_2}\mathbf{t_1} - t_{1y} - t_{12y}) \\ &\quad + \hat{Y}_2(\mathbf{r_3}\mathbf{t_1} - t_{1z} - t_{12z})) \end{aligned} \tag{27}$$

with

$$\mathbf{R_{12}} = \begin{bmatrix}\mathbf{r_1}^T \\ \mathbf{r_2}^T \\ \mathbf{r_3}^T\end{bmatrix}, \quad \mathbf{t_{12}} = \begin{bmatrix}t_{12x} \\ t_{12y} \\ t_{12z}\end{bmatrix}, \quad \mathbf{t_1} = \begin{bmatrix}t_{1x} \\ t_{1y} \\ t_{1z}\end{bmatrix}. \tag{28}$$

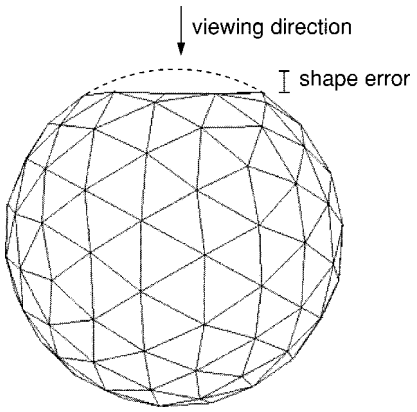


Fig. 8. Top view of the deformed sphere. In the simulations, the part of the sphere that is deformed is facing the camera for the two recorded views.

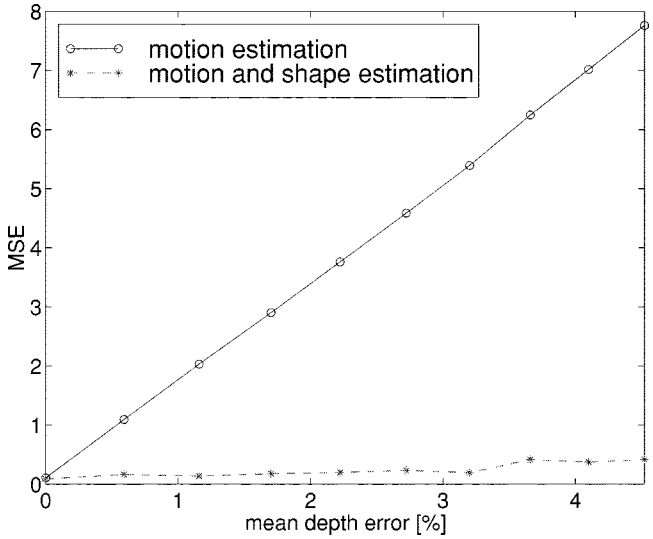


Fig. 9. MSE of the motion compensated second view for the two cases of 3-D motion estimation with and without simultaneous shape estimation.

The value $\hat{z}_2$ represents the depth of the object point $\hat{x}_2$ in view $\hat{I}_2$ and is computed by perspective projection of the object model into a z buffer.

Equation (27) is valid for every object surface point $\hat{\mathbf{x}}_\mathbf{1}$. The surface, however, is modeled using a finite set of control points. Each object surface point is described by a linear combination of three control points. We, therefore, replace Δs in (27) by

$$\Delta s = b_i \Delta s_i + b_j \Delta s_j + b_k \Delta s_k \tag{29}$$

with b_i, b_j, b_k being the barycentric coordinates for the object point $\hat{\mathbf{x}}_\mathbf{1}$ in the triangle formed by control points $\mathbf{c_i}$, $\mathbf{c_j}$, and $\mathbf{c_k}$. The quantities Δs_i represent the radial scaling factor of control point $\mathbf{c_i}$. Combination of (27) and (29) with the optical flow constraint

$$\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \cdot u_s + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \cdot v_s = \hat{I}_2 - I_2 \tag{30}$$

leads again to a linear equation for the unknown parameters $\Delta s_0 \cdots \Delta s_{N_{\mathrm{CP}}-1}$. Due to the local influence of the control points, each equation depends only on three unknowns

$$a_i \Delta s_i + a_j \Delta s_j + a_k \Delta s_k = \hat{I}_2 - I_2. \tag{31}$$

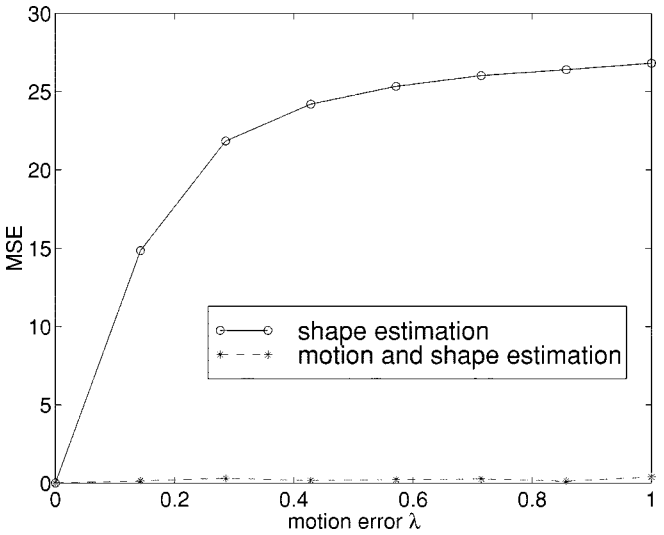


Fig. 10. MSE of the motion compensated second view for the two cases of 3-D shape estimation with and without simultaneous motion estimation.

The three indices i, j, and k represent the three control points of the triangle enclosing the surface point $\hat{\mathbf{x}}_\mathbf{1}$. The coefficients a_i, a_j, and a_k are given as

$$\begin{aligned} a_i &= b_i \left(\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \frac{u_s}{\Delta s} + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \frac{v_s}{\Delta s} \right) \\ a_j &= b_j \left(\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \frac{u_s}{\Delta s} + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \frac{v_s}{\Delta s} \right) \\ a_k &= b_k \left(\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \frac{u_s}{\Delta s} + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \frac{v_s}{\Delta s} \right). \end{aligned} \tag{32}$$

Similar to Section V-A, the resulting over-determined linear system of equations can be solved in a least-squares sense.

C. Combined 3-D Shape and 3-D Motion Estimation

Consideration of both motion and shape errors is achieved by superimposing the pixel displacements (u_m, v_m) in (15) and (u_s, v_s) in (27). Together with the optical flow constraint, we now obtain the linear equation

$$\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \cdot (u_m + u_s) + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \cdot (v_m + v_s) = \hat{I}_2 - I_2 \tag{33}$$

with the $N_{\mathrm{CP}} + 6$ unknown parameters $\Delta s_0 \cdots \Delta s_{N_{\mathrm{CP}}-1}$ and ΔR_x, ΔR_y, ΔR_z, Δt_x, Δt_y, and Δt_z. This equation can be set up for each pixel position that is covered by the object. Since the number of object pixels typically exceeds the number of unknowns, the resulting over-determined linear system of equations can be solved in a least-squares sense. Please note that the inherent linearization requires an iterative solution using the hierarchical estimation scheme described at the end of Section V-A.

So far we have considered only two frames: I_1 and I_2. In the case of N available views $I_1 \cdots I_N$ the combination of motion and shape estimation has the additional advantage that the simultaneous shape update generates a 3-D model that is consistent with all frames. This leads to a tight

Fig. 11. Frames 1 and 10 of the *cassette* sequence.

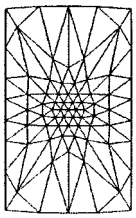


Fig. 12. Initial object geometry used for simultaneous shape refinement and motion estimation.

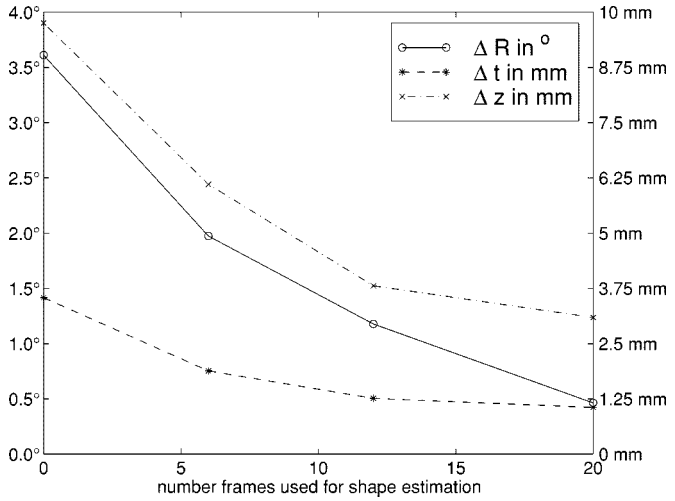


Fig. 13. Average rotational and translational motion error as a function of the number of frames used for simultaneous shape and motion estimation. Δz represents the average deviation of the estimated from the original object depth.

coupling of the multiple motion-estimation problem across all views in comparison to the traditional model-based motion-estimation approach, where the motion for each frame is estimated independently. The number of unknowns in the resulting linear system of equations increases correspondingly to $N_{\mathrm{CP}} + 6(N-1)$.

D. Simulation Results for Combined 3-D Shape and Motion Estimation

This section provides simulation results that illustrate the improvement in 3-D motion estimation when shape refinement and motion estimation are combined. In the first experiment, we use a spherical test object and a random texture to produce two synthetic object views at CIF resolution (352 $\times$ 288 pixels). We fix the camera motion $\mathbf{R_{12}}$, and $\mathbf{t_{12}}$ between the two views and deform the object surface by scaling seven control points toward the object center. The resulting erroneous object shape is then used for model-based motion estimation. Fig 8 shows the deformation of the original sphere from the top to illustrate the introduced shape errors. Given the new object shape, model-based motion estimation with and without simultaneous shape refinement is performed between the two views. The estimation error is illustrated in terms of the mean-squared intensity difference between the original second view and the synthesized second view, which is obtained by motion compensation of the first view given the estimated motion and shape refinement parameters. This is shown in Fig. 9, where the mean-squared intensity error is plotted as a function of the relative shape error for the two cases:

1) model-based motion estimation without shape estimation (Section V-A);
2) combined model-based motion and shape estimation (Section V-C).

The mean shape error in Fig. 9 is the magnitude of the relative depth error between the original and the erroneous object shape

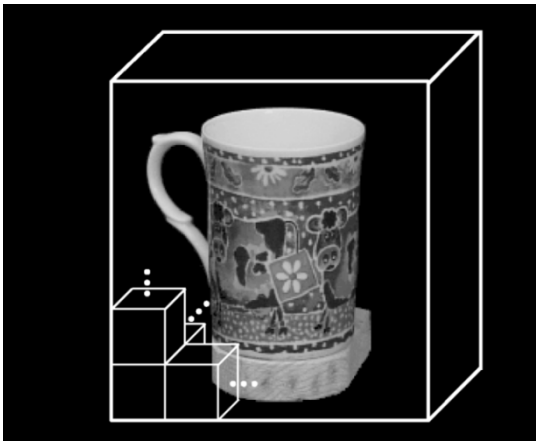


Fig. 14. Bounding box of the volume.

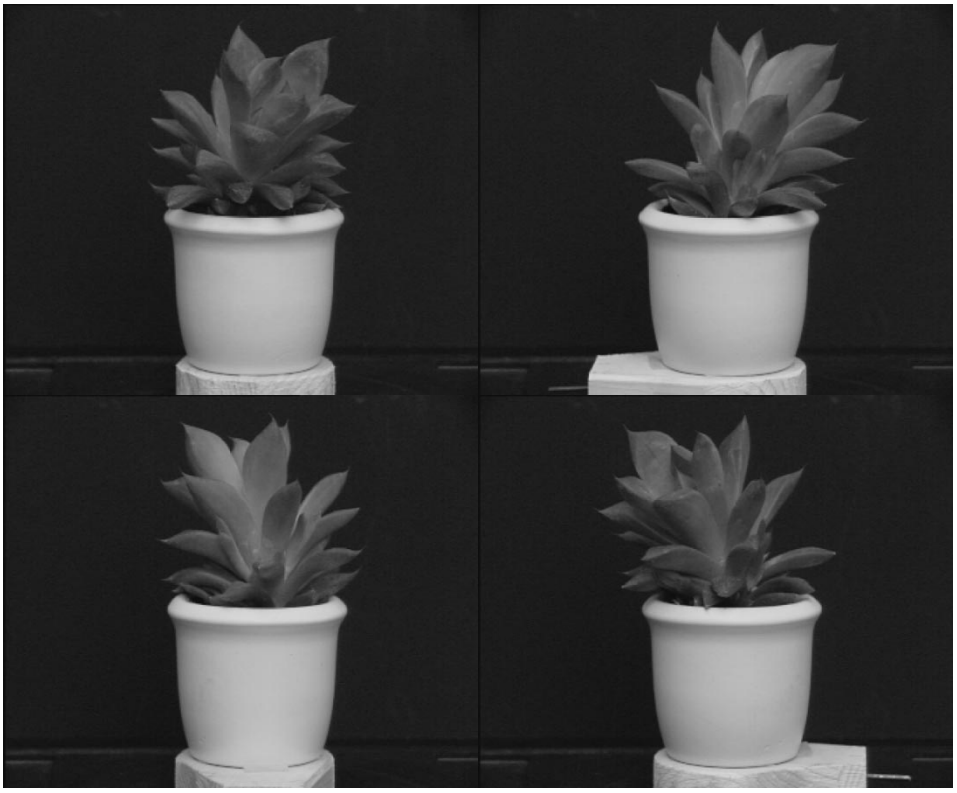


Fig. 15. Frames 1, 4, 7, and 10 of the *Plant* sequence.

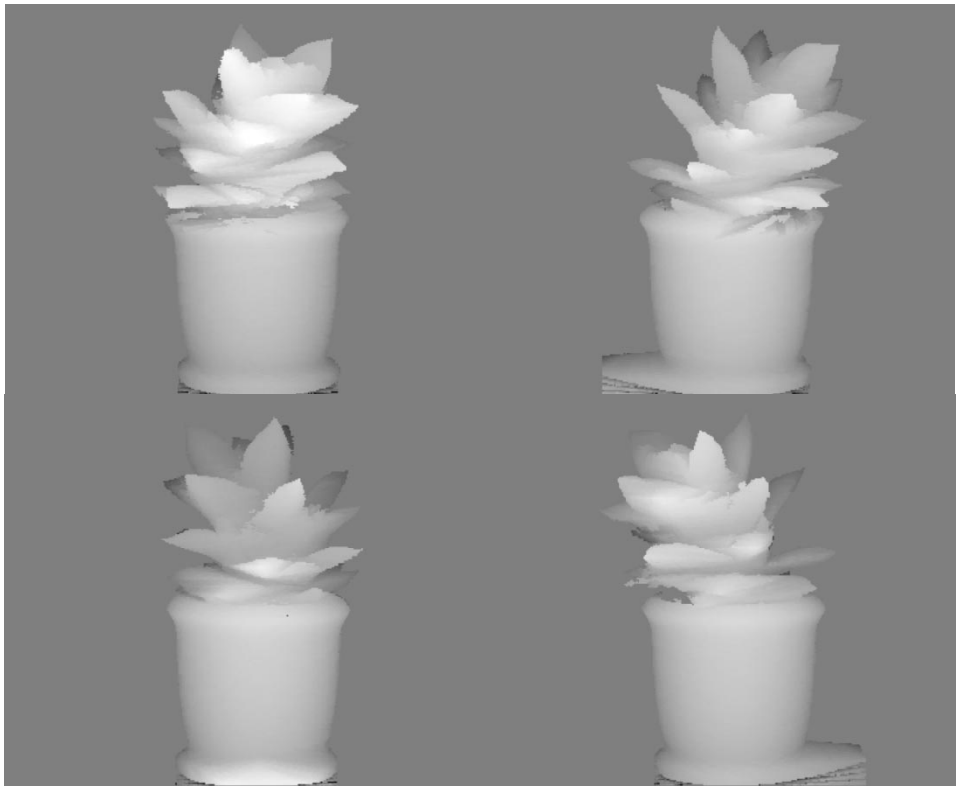


Fig. 16. Rendered depth maps for frames 1, 4, 7, and 10 of the *Plant* sequence.

averaged over all pixels. The larger the relative depth error, the more the sphere in Fig. 8 is deformed. It can be seen from Fig. 9 that for motion estimation without shape refinement, the approximation error of view I_2 increases considerably with shape

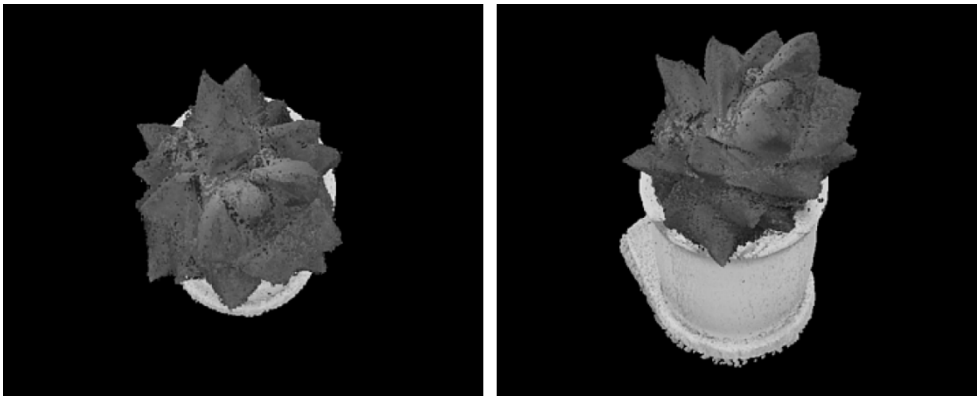


Fig. 17. New views of the plant that are not part of the original set of images.

Fig. 18. Original frames 1 and 5 of the *Cup* sequence.

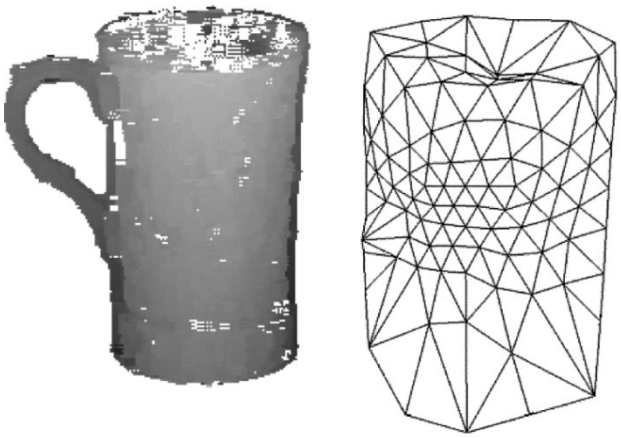
error. If the shape is estimated in combination with the 3-D motion parameters, the errors in the motion-compensated second view are significantly reduced.

In the second experiment, we fix the shape error of the deformed object to an average depth error of 2.5%. The assumed motion error between the two frames I_1 and I_2 is now varied and the improvement of the motion compensation error due to shape refinement is plotted as a function of the assumed motion error between the two views. The curves in Fig. 10 show the two cases:

1) model-based shape estimation without motion estimation (Section V-B);
2) combined model-based motion and shape estimation (Section V-C).

The rotational error is $\lambda \cdot 5^\circ$ and the translational error corresponds to a horizontal and vertical translation of $\lambda \cdot 15$ and $\lambda \cdot 25$ pixels in the image plane. For the measurement points in Fig. 10, we vary λ uniformly from 0 to 1. Hence, $\lambda = 0$ corresponds to no motion error and $\lambda = 1$ corresponds to maximum motion error. It can be seen from Fig. 10 that an increased motion error leads to a considerable increase in image synthesis error for the second view if only shape is estimated. The correct object deformation is only recovered in the case of no motion error ($\lambda = 0$ in Fig. 10). These experiments underline the strong interdependency of motion and shape errors. For erroneous motion, the constrained shape deformation cannot produce a perfect synthesized second view. For combined motion and shape estimation, the MSE is much smaller and remains almost constant for increasing motion error, which shows that the shape and motion errors are successfully estimated at the same time.

In a third experiment, we use 21 frames of a synthetic sequence showing a video cassette of size 12 cm $\times$ 20 cm $\times$ 4 cm. Fig. 11 shows two frames of the sequence and Fig. 12 the initial object model. The camera remains fixed for all frames while the object motion varies between $\pm 45^\circ$ for the rotation and ± 5 cm for the translation. For the first frame, the initial model is manually placed at the correct position, adapted to the silhouette, and the extension in z-direction of the cassette is erroneously selected to be 6 cm. This introduces a considerable shape error which prevents the 3-D model-based motion estimator from providing accurate motion parameters for the 21 frames. The combined shape and motion estimator as described in the previous section, however, can correct the shape errors and improves the motion parameter estimates. Fig. 13 shows the rotational $(\Delta R_i = (1/3)(|\Delta R_{xi}| + |\Delta R_{yi}| + |\Delta R_{zi}|))$ and translational $(\Delta t_i = ||\Delta \mathbf{t}_i||)$ motion errors averaged over all frames. The error measure is determined by comparing the correct motion parameters with the estimates from the model-based motion estimator in Section V-A for different object shapes. These different object shapes are obtained by simultaneous shape and motion estimation as described in Section V-C for a varying number of frames of the sequence. In addition to the motion errors, Fig. 13 also shows the average deviation of the refined object shape from the correct shape of the cassette in millimeters. It can be seen from Fig. 13 that for the original erroneous



(a) (b)

Fig. 19. (a) Depth map of frame 5 of the *Cup* sequence obtained from the structure-from-motion step. (b) Generic shape model after initialization.

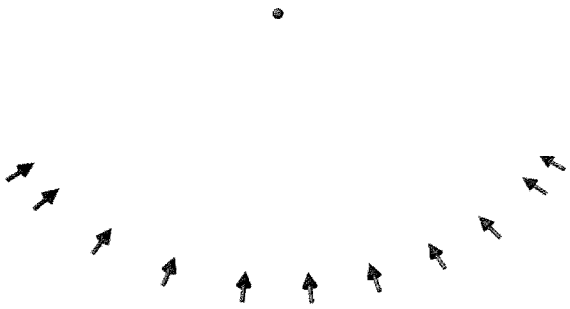


Fig. 20. Estimated camera position and orientation for the 11 views of the *Cup* sequence.

object shape (number frames $= 0$), we obtain a considerable motion error. Increasing the number of frames used for simultaneous shape and motion estimation improves the object shape (Δz in Fig. 13) and leads to more accurate motion estimates.

VI. Volumetric Reconstruction from Multiple Calibrated Camera Views

The processing steps so far provide us with the camera position and orientation for all views. As described in Section V-B the object shape used to get these parameters is modeled as a subdivided icosahedron whose control points can be moved in radial direction only. This restricts the quality of the object modeling, but offers the advantage of a small number of degrees of freedom for shape updates which can be estimated robustly from the image data. Remember that the shape deformation modeling has been developed to support the 3-D motion-estimation algorithm so that it is able to deal with 3-D model errors. Since our aim is to reconstruct a photorealistic 3-D model of the object, the calibrated views are now used as input to a separate 3-D reconstruction algorithm [5] that is based on a tesselation of the 3-D space by voxels. Since voxels can be set or removed independently from neighboring shape information, we are able to reconstruct fine structures or nonconvex parts of the 3-D object. We now describe this voxel-based 3-D reconstruction algorithm that leads to a high-quality 3-D description of the object under investigation.

TABLE I
Mean Absolute Estimation Error Averaged over 11 Frames of the *Cup* Sequence

ΔR_x	ΔR_y	ΔR_z	ΔT_x	ΔT_y	ΔT_z
0.4^o	1.36^o	0.60^o	5.0 mm	0.6 mm	4.0 mm

Fig. 21. Depth maps rendered from the reconstructed 3-D volume for the same viewing positions as in frames 1 and 5 of the *Cup* sequence.

In our final 3-D reconstruction approach, all operations are performed on voxels, that are unique for the 3-D object, and not on pixels where many representatives in different views correspond to the same 3-D point. Therefore, we avoid the search for corresponding points and the fusion of several incomplete depth estimates. The proposed algorithm proceeds in four steps:

1) volume initialization;
2) color hypothesis generation for all voxels from all available camera views;
3) consistency check and hypothesis elimination considering all views;
4) determination of the best color hypothesis for the remaining surface voxels.

A. Volume Initialization

The first step is to define a volume in the reference coordinate system that encloses the 3-D object to be reconstructed. The volume extensions are determined by placing a conservative bounding box around the approximative object geometry obtained in Section V-C. The volume is discretized in all three dimensions leading to an array of voxels with associated color, where the position of each voxel in the 3-D space is determined by its indices (l, m, n). Initially, all voxels are transparent. Fig. 14 shows an example of the initial volume with large voxels for illustration purposes. Typical numbers are $200 \times 200 \times 200$ voxels.

B. Hypothesis Generation

During the hypothesis generation step, a set of color hypotheses is assigned to each voxel of the predefined volume. The k-th hypothesis H^k_{lmn} for a voxel V_{lmn} with voxel index (l, m, n) is

$$H^k_{lmn} = [R(X_i, Y_i), G(X_i, Y_i), B(X_i, Y_i)] \qquad (34)$$

where (X_i, Y_i) is the pixel position of the perspective projection of the voxel center (x_l, y_m, z_n) into the i-th camera view. R, G,

Fig. 22. Rendered object views from the reconstructed 3-D voxel model for the same viewing positions as in Fig. 18.

Fig. 23. Original frames 4 and 9 of the *Shoe* sequence.

and B are the three color components. The projection of the voxel center for view i is obtained as

$$X_i = -f_x \frac{x_{li}}{z_{ni}}$$
$$Y_i = -f_y \frac{y_{mi}}{z_{ni}} \tag{35}$$

with

$$[x_{li}, y_{mi}, z_{ni}]^T = \mathbf{R_i}[x_{l0}, y_{m0}, z_{n0}]^T + \mathbf{t_i}. \tag{36}$$

$\mathbf{R_i}$ and $\mathbf{t_i}$ are the object rotation and translation in view i with respect to the reference coordinate system.

Hypothesis H^k_{lmn} is associated with voxel V_{lmn} if the projection of V_{lmn} into at least one other camera view $j \neq i$ leads to an absolute color difference that is less than a predefined threshold Θ

$$|R(X_i, Y_i) - R(X_j, Y_j)| + |G(X_i, Y_i) - G(X_j, Y_j)| + |B(X_i, Y_i) - B(X_j, Y_j)| < \Theta. \tag{37}$$

This equation has to be evaluated for each possible pair (i, j) from N available views. For all combinations of i and j that fulfill (37), a hypothesis H^k_{lmn} is stored with the color taken from view i according to (34). Please note that the voxel need not be visible in all views due to occlusions and that it might not be visible in any view at all if it is inside the object. At this stage of the algorithm, we make no assumptions about the object geometry and cannot decide whether a voxel is visible or not. We therefore have to remove those hypotheses from the overcomplete set that do not correspond to the correct color of the object surface.

C. Consistency Check and Hypothesis Elimination

In the previous step, we stored multiple hypotheses for each voxel of the working volume. Those hypotheses were extracted from two or more consistent views, but might lead to contradictions with other views where the voxel is visible as well. Hence, we refine our voxel set by iterating over all views. We start from the voxel surface of the predefined volume and remove voxels until the *maximal photo-consistent* [17] 3-D shape of the object is recovered.

For each view, we determine the currently visible voxels and compare all associated hypotheses with the corresponding pixel color at the pixel position given by (35). The similarity measure is again the absolute difference of the color components in (37). If this error exceeds threshold Θ, we eliminate the corresponding hypothesis for this voxel. If all hypotheses for one voxel are removed, the voxel is transparent and the visible surface for the next view moves one voxel toward the interior of the volume. This implies that during the first iteration only voxels on the surface of our volume can be removed. We, therefore, iterate several times over all available views until no more hypotheses are removed and the number of transparent voxel converges. The remaining nontransparent voxels constitute the volumetric description of our 3-D object. The color values associated with the resulting nontransparent voxels which are on the object surface can now be used for rendering.

D. Visible Surface Determination

The hypothesis testing and subsequent hypothesis elimination for each view i require the determination of the visible surface voxels from the current view of the volume. This visibility test can be carried out by indexing the voxels with increasing depth from the camera origin. Processing the voxels in their order of visibility for a particular view can be achieved by volume index permutation in combination with a decision of whether to index the voxels in increasing or decreasing order for each dimension. This leads to a total of 48 different cases of volume traversal.

The algorithm used for identifying the volume traversal direction for a particular view works as follows. We first determine the plane of the volume bounding box that is visible and most parallel to the image plane of the current view. This is achieved by rotating the optical axis $\mathcal{O} = (0, 0, -1)^T$ of the camera of view i according to the object pose

$$\mathcal{O}_i = \mathbf{R_i^{-1}}\mathcal{O}. \tag{38}$$

The largest scalar product of the transformed optical axis $\mathcal{O}_i$ and the six surface normals of the bounding box $(0, 0, 1)$, $(0, 0, -1)$, $(0, 1, 0)$, $(0, -1, 0)$, $(1, 0, 0)$, and $(-1, 0, 0)$ identifies the desired plane. Each of the six surfaces corresponds to one permutation of the three volume indices (l, m, n).

In order to determine if the loop indices have to be evaluated in decreasing or increasing order, we transform the eight corners of the volume into the camera coordinate system of view i and compute the distance of these eight points to the camera projection center. The corner corresponding to the smallest distance determines for each loop index l, m, and n if we have to increment or decrement it.

From an implementation point of view, the visibility order is obtained by simply exchanging the loop indices l, m, n when stepping through the volume. Using the resulting voxel ordering we store for each pixel in the camera image the index of the first voxel that is projected into that pixel. All the following voxels that are projected to the same pixel are considered to be invisible. A considerable speed-up of the reconstruction process can be achieved if all views that lead to the same voxel ordering are combined into a group of pictures. The hypothesis and voxel removal step, as described in Section VI-C, can then be performed in all these views simultaneously.

E. Determination of the Voxel Color

Having obtained a final set of voxels representing the geometry of the 3-D object, many voxels may still contain more than one color hypothesis consistent with the pixel colors at the corresponding image-plane location. In order to determine the best color value for this voxel, we first determine all views I_v where this voxel is visible using the approach described in Section VI-D. We then select the hypothesis H_{lmn}^{opt}, which leads to the smallest value of

$$\min_{\forall H_{lmn}^k} \left\{ \underset{\forall I_v}{\text{median}} \| H_{lmn}^k - (R(X_v, Y_v), G(X_v, Y_v), B(X_v, Y_v) \|_1 \right\} \tag{39}$$

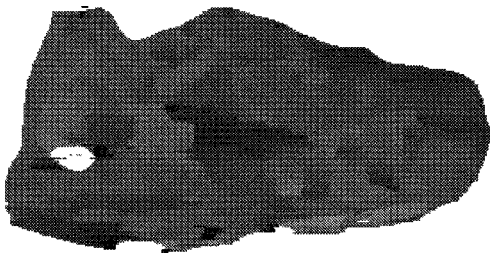


Fig. 24. Depth map of frame 1 of the *Shoe* sequence obtained from the structure-from-stereo step.

Fig. 25. Estimated camera position and orientation for the 12 views of the *Shoe* sequence.

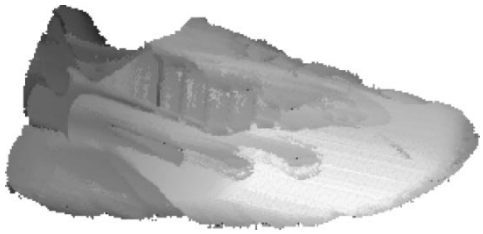


Fig. 26. Depth map rendered from the reconstructed 3-D volume for the same viewing position as in frame 12 of the *Shoe* sequence.

yielding the projection into the original views with the smallest median color error according to the l_1-norm $\| \cdot \|_1$.

F. Generation of Arbitrary New Views

Once the volumetric description of the object is determined, we can render views from new viewing positions which are not part of the set of available views. For that purpose, we transform the volume to the desired viewing position according to (36). The pixels in the virtual views are generated by perspective projection (35). A simple z buffer ensures that only visible voxels are rendered when stepping through the volume. The depth map for the view can be taken directly from the z buffer.

In the following two sections, experimental results for calibrated and uncalibrated views are presented. For the calibrated scene in Section VII, the position and viewing direction of the camera are available because the object is moved on a turn table whose motion is known. For the uncalibrated sequences in Section VIII, the camera is moved to arbitrarily selected positions without knowledge about the camera motion. For those experiments, the views are calibrated from the image data themselves using the model-based view calibration technique described in Section V-C.

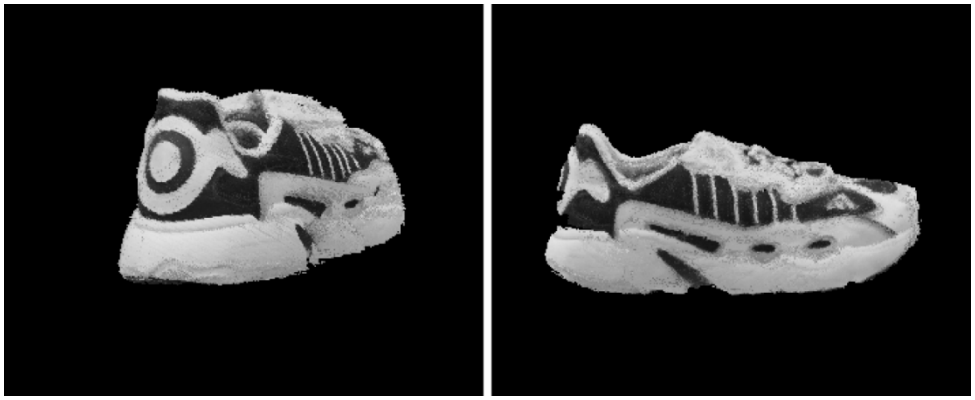


Fig. 27. Rendered views of the reconstructed object for the same viewing positions as in Fig. 23.

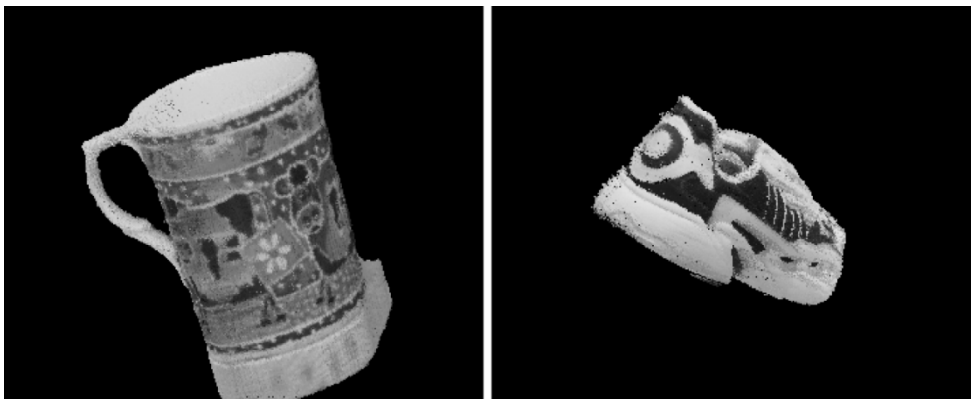


Fig. 28. New views of the two test objects which are not part of the set of initially available shots.

VII. Experimental Results for Calibrated Views

In the first experiment, we apply the volumetric reconstruction algorithm as described in Section VI to a 24-view sequence of a plant recorded in front of a homogeneous background. The original frames 1, 4, 7, and 10 are shown in Fig. 15. The camera is calibrated using the calibration object shown in Fig. 2. The motion parameters from view to view are known in this experiment since the object is rotated on a turn table. Fig. 16 shows the depth values of the rendered surface voxels for the same viewing positions as in Fig. 15. As described in Section VI-F arbitrary new views can be rendered from the reconstructed object model. Fig. 17 shows two new views of the plant that are not part of the original sequence. Please note that the selected virtual views lie far off the set of input images.

VIII. Experimental Results for Uncalibrated Views

The first uncalibrated sequence is an 11-view sequence of a cup with homogeneous background recorded with a video camera. The original frames 1 and 5 are shown in Fig. 18. The 3-D rigid body motion-estimation algorithm described in Section IV first estimates the relative camera motion between frame 5 and 6. The dense map of depth value computation from two views (5 and 6) then leads to the depth map reproduced in Fig. 19. The generic 3-D model in Fig. 4 is now adapted to the initial depth map, as shown in Fig. 19. Note that the handle of the cup is excluded from the generic model by manual removal from the depth map. From an estimation point of view, this simply means that fewer parts of the object are used for relative camera pose and orientation estimation. Using the initial model, the combined motion- and shape-estimation algorithm as described in Section V-C is used to calibrate all available views of the cup. The recovered camera positions for these views are shown in Fig. 20. For objective assessment of the accuracy of the estimated camera positions, we compare them with calibration data that has been obtained by defined movement of the camera. Table I shows the average deviation of the estimated versus the calibrated camera movements. After calibration of all views, the refined generic object shape approximates the original shape for large parts of the object but fails to describe small details like, e.g., the handle of the cup. The volumetric reconstruction algorithm as described in Section VI is capable to reconstruct those fine details and leads to the final 3-D description of the object.

In order to judge the quality of this reconstruction, the 3-D voxel model can be rendered into a z buffer to show the depth maps that are obtained from the model. These depth maps are shown in Fig. 21 for the same camera positions as in Fig. 18. The corresponding rendered views from the 3-D model are shown in Fig. 22.

In a second experiment with uncalibrated image data, we use 12 views of the shoe reproduced in Fig. 23. The 3-D rigid body motion-estimation algorithm in Section IV first estimates the relative camera motion between frame 1 and 2. Using the estimated motion the depth map reproduced in Fig. 24 is computed. The generic 3-D model is then adapted to the initial depth map

and the combined motion and shape estimation algorithm as described in Section V-C is used to calibrate all available views of the shoe. The recovered camera positions for these views are shown in Fig. 25. Using the calibration data for all views, the volumetric reconstruction algorithm as described in Section VI recovers a 3-D description of the object.

In order to judge the quality of the reconstruction, the 3-D voxel model is rendered into a z buffer to show the depth maps that are obtained from the model. Fig. 26 shows a depth map for the camera position determined for frame 12. The rendered views from the 3-D model are shown in Fig. 27. As stated in Section VI.F new views of the object which are not in the set of initially available views can be rendered from the reconstructed 3-D voxel volume. New views for the two test objects *Cup* and *Shoe* are reproduced in Fig. 28.

IX. Conclusion

We have presented a system for the automatic reconstruction of real world objects from multiple uncalibrated camera views. The main system features are: 1) internal camera parameter calibration using a reference object; 2) initial 3-D model construction from the first two views; 3) view calibration using a novel intensity gradient-based approach that simultaneously estimates 3-D motion parameters and shape refinements; and 4) volumetric reconstruction of the object shape and color from all available views. The view calibration extends existing gradient-based 3-D motion-estimation techniques to the simultaneous estimation of shape refinements and uses the novel concept of sliding textures. This concept allows to keep the projection into the initial or reference view undistorted after shape modifications. The shape update is consistent with all available views leading to a tight coupling of the shape and motion-estimation problem for multiple views. The experiments show that a considerable accuracy improvement of the motion estimates is obtained when combining shape and motion estimation into a common framework. Once all camera positions are calibrated, a volumetric 3-D reconstruction process is performed that recovers surface voxels and associated color information of the object. This last step relaxes the severe shape restrictions used in the steps before and allows to recover fine structures of the object. The result is a set of voxels with associated color information describing shape and texture of the object. New views of the object can be rendered from the recovered 3-D description. Experimental results for calibrated and uncalibrated sequences illustrate the excellent visual quality of the reconstructed 3-D computer models.

References

[1] P. Beardsley, P. Torr, and A. Zisserman, "3D model acquisition from extended image sequences," in *Proc. ECCV '96*, Cambridge, U.K., 1996, pp. 683–695.

[2] E. Boyer, "Object models from contour sequences," in *Proc. ECCV'96* , Cabridge, U.K., 1996, pp. 109–118.

[3] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *ACM Siggraph '96*, June 1993, pp. 303–312.

[4] P. Eisert and B. Girod, "Model-based 3-D motion estimation with illumination compensation," in *Proc. IPA '97*, Dublin, Ireland, July 1997, pp. 194–198.

[5] P. Eisert, E. Steinbach, and B. Girod, "Multi-hypothesis, volumetric reconstruction of 3-D objects from multiple calibrated camera views," in *Proc. ICASSP '99*, Phoenix, AZ, Mar. 1999, pp. 3509–3512.

[6] L. Falkenhagen, "Depth estimation from stereoscopic image pairs assuming piecewise continuous surfaces," in *Image Processing for Broadcast and Video Production*, Y. Paker and S. Wilbur, Eds. Hamburg, Germany: Springer Great Britain, 1994, pp. 115–127.

[7] O. Faugeras, *Three-Dimensional Computer Vision*. Cambridge, MA: MIT Press, 1993.

[8] O. Faugeras, Q.-T. Luong, and S. Maybank, "Camera self-calibration—Theory and experiments," in *Proc. ECCV'92*, 1992, pp. 563–578.

[9] A. W. Fitzgibbon, G. Cross, and A. Zisserman, "Automatic 3D model construction for turn-table sequences," in *Proc. ECCV'98 Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, Freiburg, Germany, June 1998.

[10] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," in *Proc. ECCV'98*, Freiburg, June 1998, pp. 311–326.

[11] B. Girod and E. Steinbach, "A new method for simultaneous estimation of displacement, depth, and rigid body motion parameters," in *Proc. 9th IMDSP Workshop*, Belize, Mar. 1996, pp. 122–123.

[12] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *Proc. ECCV '92*, 1992, pp. 579–587.

[13] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[14] A. Huertas and G. Medioni, "Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 651–664, May 1986.

[15] R. Koch, "Dynamic 3-D scene analysis through synthesis feedback control," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 556–568, June 1993.

[16] R. Koch, M. Pollefeys, and L. Van Gool, "Multi viewpoint stereo from uncalibrated sequences," in *Proc. ECCV'98*, Freiburg, Germany, 1998, pp. 55–71.

[17] K. N. Kutulakos and S. M. Seitz, "What Do N Photographs Tell Us About 3D Shape?," Univ. Rochester, Rochester, NY, Tech. Rep. 680, Jan. 1998.

[18] H. Li, P. Roivainen, and R. Forchheimer, "3-D motion estimation in model-based facial image coding," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 545–555, June 1993.

[19] F. C. M. Martins and J. M. F. Moura, "Video representation with 3D entities," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 71–85, Jan. 1998.

[20] A. N. Netravali and J. Salz, "Algorithms for estimation of three-dimensional motion," *AT&T Tech. J.*, vol. 64, no. 2, pp. 335–346, 1985.

[21] W. Niem and J. Wingbermühle, "Automatic reconstruction of 3D objects using a mobile monoscopic camera," in *Proc. Int. Conf. Recent Advances in 3D Imaging and Modeling*, Ottawa, ON, Canada, May 1997.

[22] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C, The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[23] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool, "Flexible acquisition of 3D structure from motion," in *Proc. 10th IMDSP Workshop 1998*, Austria, July 1998, pp. 195–198.

[24] M. Pollefeys, R. Koch, and L. Van Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in *Proc. Int. Conf. Computer Vision (ICCV)*, Bombay, India, Jan 1998, pp. 90–95.

[25] S. Sullivan and J. Ponce, "Automatic model construction, pose estimation, and object recognition from photographs using triangular splines," in *Proc. Int. Conf. Computer Vision (ICCV)*, Bombay, India, Jan 1998, pp. 90–95.

[26] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," in *Proc. Computer Vision and Pattern Recognition (CVPR '97)*, Puerto Rico, 1997, pp. 1067–1073.

[27] M. R. Shortis, T. A. Clarke, and T. Short, "A comparison of some techniques for the subpixel location of discrete target images," *SPIE Videometrics III*, vol. 350, pp. 239–249, 1994.

[28] E. Steinbach, S. Chaudhuri, and B. Girod, "Robust estimation of multicomponent motion in image sequences using the epipolar constraint," in *ICASSP 97*, Munich, Germany, Apr. 1997, pp. 2689–2692.

[29] E. Steinbach, P. Eisert, and B. Girod, "Motion-based analysis and segmentation of image sequences using 3-D scene models," *Signal Processing*, vol. 66, no. 2, pp. 233–248, April 1998.

[30] R. Szeliski, "Rapid octree construction from image sequences," in *Proc. Computer Vision, Graphics and Image Processing CVGIP '93*, July 1993, pp. 23–32.
[31] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 13–27, Jan 1984.
[32] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 323–344, Aug. 1987.
[33] B. C. Vemuri and J. K. Aggarwal, "3-D model construction from multiple views using range and intensity data," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami Beach, FL, 1986, pp. 435–437.

Peter Eisert received the Dipl. Ing. degree in electrical engineering from the University of Karlsruhe, Germany, in 1995. He is currently working on his Ph.D. thesis.

He then joined the Image Communication Group at the Telecommunications Institute, University of Erlangen-Nuremberg, Germany, where he is a member of the Center of Excellence *3-D Image Analysis and Synthesis*. His research interests include model-based video coding, 3-D object reconstruction, image communication, and computer vision.

Eckehard Steinbach was born in Germany in 1969. He studied electrical engineering at the University of Karlsruhe, Germany, the University of Essex, U.K., and ESIEE, Paris. He is currently working on his Ph.D. thesis.

After receiving his diploma in 1994, he joined the Image Communication Group at the Telecommunications Institute, University of Erlangen-Nuremberg, Germany. His research interests include image processing, computer vision, and robust video transmission for mobile applications.

Bernd Girod (S'83–M'89–SM'97–F'98) received the engineering doctorate from University of Hannover, Hannover, Germany, and the M.S. degree from Georgia Institute of Technology, Atlanta, GA.

He is a Chaired Professor of Telecommunications in the Electrical Engineering Department, University of Erlangen-Nuremberg, Germany. Since 1996, he has served as Director of the Center of Excellence *3-D Image Analysis and Synthesis* in Erlangen. Prior visiting or regular faculty positions include MIT, Georgia Tech, and Stanford. He has been involved with several start-up ventures, among them PictureTel, Polycom, Vivo Software, 8×8, and RealNetworks. His research interests are in the areas of image communication, 3-D image analysis and synthesis, and multimedia systems.

Prof. Girod was recently elected a Fellow of the IEEE for his contributions to the theory and practice of video communications.