Model-based 3-D Shape and Motion Estimation Using Sliding Textures

Eckehard G. Steinbach, Peter Eisert, and Bernd Girod

Information Systems Laboratory
Stanford University
{steinb,eisert,bgirod}@stanford.edu

Abstract

Given an accurate 3-D shape model of a scene, the motion parameters of a moving camera can be recovered with high accuracy using model-based motion estimation techniques. Shape errors, however, reduce the accuracy of this kind of motion estimation considerably. In this paper, model-based motion estimation is combined with simultaneous object shape refinement. A deformable 3-D shape model of low dimensionality is employed to approximate the object shape. Camera position and orientation for all views as well as object shape refinements are estimated simultaneously from the image data using an optical-flow-based approach. In comparison to traditional flexible body motion estimation, our formulation of the shape deformation allows the object texture to slide on the object surface. Experimental results illustrate that combined shape and motion estimation using sliding textures improves the calibration data of the individual views in comparison to fixed-shape model-based camera motion estimation.

1 Introduction

Model-based 3-D motion estimation algorithms use information about the 3-D shape of an object for motion parameter recovery. For an accurate 3-D shape description of an object, e.g., obtained from a 3-D scanner, the motion parameters of a moving camera can be recovered with high accuracy. Shape errors, however, reduce the accuracy of model-based motion estimation techniques considerably. Therefore, rigid body model-based motion estimation has been extended to combined shape and motion estimation [1]-[5]. Simultaneous shape and motion estimation has the advantage of a tight coupling of all available views since the estimated shape updates have to be consistent within all views.

In this paper we present a formulation of combined shape and motion estimation that differs from traditional flexible body motion estimation. Traditionally, the object texture is extracted from one frame and is mapped onto the 3-D object surface leading to a perfect reproduction of this frame after rendering. It is typically assumed that the original shape is very accurate and that the deformations we want to estimate occur after initialization. In case the object shape is only an estimate of the true shape, we face the problem that after object surface deformations the projection of the model leads to a distorted version of this initial frame. This requires to re-extract the texture from the frame and to re-map it on the altered object surface. In this work, the texture is not fixed to the object surface but can slide on it in combination with surface deformations. This *sliding texture* concept ensures that the projection of the object into the initial view always remains undistorted independent of the estimated shape refinement. This allows us to perform 3-D model-based motion estimation with a coarse approximative shape that is refined during 3-D motion parameter recovery.

We consider the case where many camera views of an object are available but only very limited or erroneous 3-D geometry information is available. Camera position and orientation for the camera views are unknown and are to be determined. We employ a generic subdivision surface model to approximate the object shape. This generic model is initially spherical and is adapted to the object using the object silhouette. The resulting approximative object shape is then used to estimate the camera position and orientation for all views together with object shape refinements. Our algorithm requires in its current formulation knowledge about the internal camera parameters which are estimated from a camera calibration step [6] using a reference object.

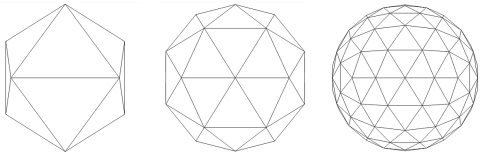

Figure 1: Shape model with increasing resolution.

2 3-D Shape Model

A generic 3-D shape model that is based on an icosahedron is used to describe the shape of the object. The icosahedron is defined by 12 control points which form a triangle mesh as illustrated on the left hand side of Fig. 1. Since 12 control points are not sufficient to describe arbitrary object shapes, the icosahedron is recursively subdivided until the desired resolution or the desired number of control points is reached. This subdivision is illustrated in Fig. 1. The number of control points as a function of the subdivision level l can be computed as

$$N_{CP} = 12 + 10(4^l - 1) \; . \tag{1}$$

The control points can be moved individually for shape approximation of an 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 deformations from global rotation and translation of the object. Due to the limited number of control points and the radial movement constraint, the shape modeling is of approximative nature.

3 3-D Shape Model Initialization

The generic 3-D shape model is by definition spherical and typically deviates considerably from the actual object shape. In order to facilitate the shape estimation we exploit object silhouette information, if available, to adapt the generic model to the individual object shape. In case we have a rough estimate of the camera position and orientation for all views, we exploit object silhouettes in all views. If this information is not available we use only the silhouette extracted from the first frame.

In a 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 the object silhouette are scaled towards the object. This initialization process is illustrated in Fig. 2.

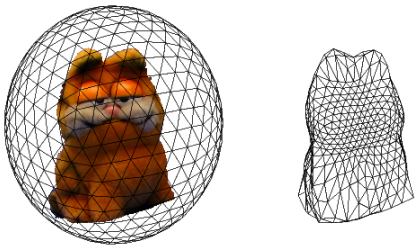

Figure 2: Shape initialization using silhouette information. **Left:** object silhouette and the generic shape model before initialization. **Right:** adapted object shape after initialization.

4 Model-based 3-D Motion and Shape Estimation

The adapted generic shape model provides us with an initial 3-D description of the object for model-based motion estimation. Model-based motion estimation permits accurate view calibration if the available model itself is very accurate. This is true, for instance, if the model stems from a 3-D laser scanner. If the model deviates from the actual shape of the object, as it is the case for the adapted generic shape model in Section 3, the motion estimates reflect these model errors. In this case, simultaneous estimation of motion and shape is required. 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 [7] and leads to a set of linear equations for the unknown camera motion and object shape parameters [9].

The 3-D model of the object as computed in Section 3 delivers shape but no texture information. Therefore, the texture is extracted from the first view I_1, where the object pose is initialized. 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. 3, a 3-D object point with respect to the first camera view I_1 is then described

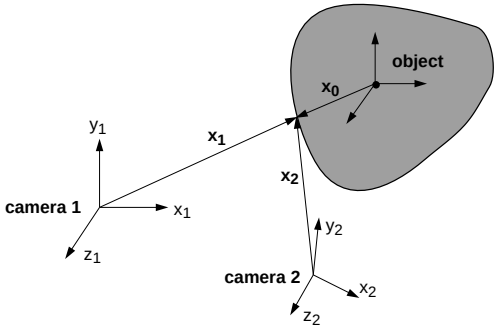

Figure 3: Object and camera coordinate systems for two camera views.

as

$$\mathbf{x_1} = \mathbf{R_1}\mathbf{x_0} + \mathbf{t_1} \ . \tag{2}$$

For a second view I_2 this transform becomes

$$\mathbf{x_2} = \mathbf{R_2}\mathbf{x_0} + \mathbf{t_2} \ . \tag{3}$$

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

$$X_1 = -f_x \frac{x_1}{z_1}, \quad Y_1 = -f_y \frac{y_1}{z_1} \tag{4}$$

with f_x, f_y being the scaled horizontal and vertical focal lengths, respectively, that relate world coordinates to pixel coordinates. 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, assuming Lambertian reflectance and the absence of occlusion. The relative motion between the two frames is described as

$$\begin{aligned} \mathbf{x_2} &= \mathbf{R_2}\mathbf{R_1^{-1}}(\mathbf{x_1} - \mathbf{t_1}) + \mathbf{t_2} \\ &= \mathbf{R_{12}}(\mathbf{x_1} - \mathbf{t_1}) + \mathbf{t_1} + \mathbf{t_{12}} \\ \mathbf{x_1} &= \mathbf{R_1}\mathbf{R_2^{-1}}(\mathbf{x_2} - \mathbf{t_2}) + \mathbf{t_1} \\ &= \mathbf{R_{12}^{-1}}(\mathbf{x_2} - (\mathbf{t_1} + \mathbf{t_{12}})) + \mathbf{t_1} \ . \end{aligned} \tag{5}$$

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 $\mathbf{x_2}$ from (X_2, Y_2) by

$$\mathbf{x_2} = \left[-\frac{X_2 z_2}{f_x}, -\frac{Y_2 z_2}{f_y}, z_2 \right]^T \ . \tag{6}$$

The depth z_2 at position (X_2, Y_2) is obtained by rendering the 3-D model geometry into a z-buffer for view I_2. The corresponding 3-D point $\mathbf{x_1}$ for the first view is computed using the relation in (5) and finally the color is extracted from the projection in (4).

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

$$I_2(X_2, Y_2) = f\left(I_1(X_1, Y_1), z_2, \mathbf{R_{12}}, \mathbf{t_{12}}, \mathbf{t_1}\right) \ . \tag{7}$$

Inaccurate 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. From an estimation point of view, 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 describe the formulation of these estimation problems. Section 4.1 first derives the estimation equations for the case of correct shape but inaccurate motion parameters. Section 4.2 then assumes correct motion and shows how shape errors can be estimated using a novel *sliding texture* formulation. Section 4.3 finally combines both effects into a common estimation framework.

4.1 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 $\hat{I}_2$ to I_2 is used to refine the 3-D rigid body motion parameters. Explicit modeling of the motion parameter error leads to the object point location $\mathbf{x_2}$ for view I_2 using the following expression

$$\begin{aligned} \mathbf{x_2} &= \Delta\mathbf{R}(\hat{\mathbf{x}}_{\mathbf{2}} - (\mathbf{t_1} + \hat{\mathbf{t}}_{\mathbf{12}})) + \mathbf{t_1} + \hat{\mathbf{t}}_{\mathbf{12}} + \Delta\mathbf{t} \\ &= \Delta\mathbf{R}(\hat{\mathbf{x}}_{\mathbf{2}} - \mathbf{x_c}) + \mathbf{x_c} + \Delta\mathbf{t} \end{aligned} \tag{8}$$

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 of $\Delta\mathbf{R}$ are small, we can linearize the rotation matrix

$$\Delta\mathbf{R} \approx \begin{bmatrix} 1 & -\Delta R_z & \Delta R_y \\ \Delta R_z & 1 & -\Delta R_x \\ -\Delta R_y & \Delta R_x & 1 \end{bmatrix} \ , \tag{9}$$

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

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

$$\begin{aligned} u_m \approx\ & f_x \left[-\Delta R_y (1 - \frac{z_c}{\hat{z}_2}) - \frac{\Delta R_z}{f_y} Y_n - \frac{\Delta t_x}{\hat{z}_2} + \right. \\ & \left. \frac{\hat{X}_2}{f_x} \left(\frac{\Delta R_x}{f_y} Y_n - \frac{\Delta R_y}{f_x} X_n - \frac{\Delta t_z}{\hat{z}_2} \right) \right], \\ v_m \approx\ & f_y \left[\Delta R_x (1 - \frac{z_c}{\hat{z}_2}) + \frac{\Delta R_z}{f_x} X_n - \frac{\Delta t_y}{\hat{z}_2} + \right. \\ & \left. \frac{\hat{Y}_2}{f_y} \left(\frac{\Delta R_x}{f_y} Y_n - \frac{\Delta R_y}{f_x} X_n - \frac{\Delta t_z}{\hat{z}_2} \right) \right] \end{aligned} \tag{10}$$

with $Y_n = \hat{Y}_2 + f_y \frac{y_c}{\hat{z}_2}$, $X_n = \hat{X}_2 + f_x \frac{x_c}{\hat{z}_2}$, and $\hat{z}_2$ being the depth obtained from the model after rendering it into a z-buffer with the known 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 [7]

$$\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \cdot u_m + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \cdot v_m \approx \hat{I}_2 - I_2 \tag{11}$$

results in a linear equation for the six unknown motion parameters

$$\begin{aligned} & a_0 \Delta R_x + a_1 \Delta R_y + a_2 \Delta R_z + a_3 \Delta t_x + \\ & + a_4 \Delta t_y + a_5 \Delta t_z = \hat{I}_2 - I_2, \end{aligned} \tag{12}$$

with a_0 to a_5 given as

$$\begin{aligned} a_0 =& \frac{\partial \hat{I}_2}{\partial \hat{X}_2} \frac{\hat{X}_2}{f_y} Y_n + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \left(f_y - f_y \frac{z_c}{\hat{z}_2} + \frac{\hat{Y}_2}{f_y} Y_n \right) \\ a_1 =& -\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \left(f_y - f_y \frac{y_c}{\hat{z}_2} + \frac{\hat{X}_2}{f_x} X_n \right) - \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \frac{\hat{Y}_2}{f_x} X_n \\ a_2 =& -\frac{f_x}{f_y} \frac{\partial \hat{I}_2}{\partial \hat{X}_2} Y_n + \frac{f_y}{f_x} \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} X_n \\ a_3 =& -f_x \frac{\partial \hat{I}_2}{\partial \hat{X}_2} \frac{1}{\hat{z}_2}; \quad a_4 = -f_y \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \frac{1}{\hat{z}_2} \\ a_5 =& -\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \frac{\hat{X}_2}{\hat{z}_2} - \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \frac{\hat{Y}_2}{\hat{z}_2}. \end{aligned} \tag{13}$$

At least six equations are necessary for the algorithm to determine the motion parameters. For robustness, we set up (12) for each pixel corresponding to the object and solve the resulting overdetermined system of linear equations in the 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 displacement vectors 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 the 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 x 72 pixels. For each new level the resolution is doubled in both directions leading to a final resolution of 352 x 288 pixels (CIF). Experiments with this hierarchical scheme show that displacements of up to 30 pixels between two frames can be estimated.

4.2 3-D Shape Estimation Using Sliding Textures

In the case of 3-D shape estimation, the camera motion parameters $\mathbf{R}_{\mathbf{12}}$ and $\mathbf{t}_{\mathbf{12}}$ are assumed to be correct and the object shape to be erroneous. The color value I_2 at pixel position (X_2, Y_2) in frame I_2 is a function of the motion parameters, the object depth, and the initial object position as shown in (7). Image synthesis after motion compensation from I_1 towards 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 before, the control points of the shape model are constrained to move radially with respect to the object center. Traditionally, the texture is extracted from frame I_1 and is mapped onto the 3-D surface leading to a perfect reproduction of I_1 after rendering of the model with arbitrary shape. After object surface deformations, 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 deformations. While the control points defining the

object shape move radially, the texture movement is restricted 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. 4 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 inaccurate 3-D position of an object surface point caused by these errors as $\hat{\mathbf{x}}_1$. Imagine that the 3-D point $\hat{\mathbf{x}}_1$ is radi-

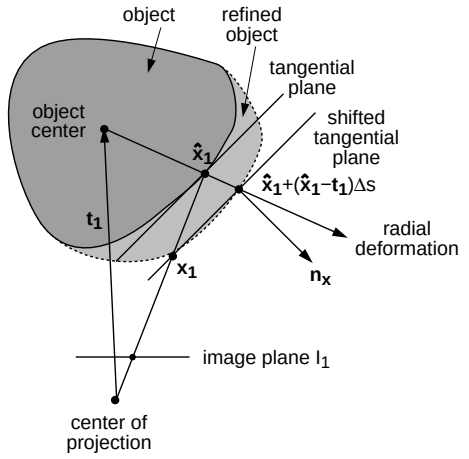

Figure 4: Illustration of the radial shape deformation and the *sliding texture* concept.

ally moved to the new position $\hat{\mathbf{x}}_1 + (\hat{\mathbf{x}}_1 - \mathbf{t}_1)\Delta s$. Assuming a locally planar object surface described by the tangential plane in Fig. 4, both 3-D points $\hat{\mathbf{x}}_1$ and $\mathbf{x}_1$ are projected to the same pixel position in the image plane. The 3-D point $\mathbf{x}_1$ represents the deformed object surface and is obtained via intersection of the shifted tangential plane through $\hat{\mathbf{x}}_1 + (\hat{\mathbf{x}}_1 - \mathbf{t}_1)\Delta s$ with the line of sight. Please note, that this deformation description differs from traditional flexible body modeling where the color at $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_1 + (\hat{\mathbf{x}}_1 - \mathbf{t}_1)\Delta s$ would be identical. In our case, the color is not fixed for a 3-D point but along the line of sight. Therefore, the points $\hat{\mathbf{x}}_1$ and $\mathbf{x}_1$ have the same color which means that the texture slides from $\hat{\mathbf{x}}_1$ to $\mathbf{x}_1$ 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 $\mathbf{x}_1$ in Fig. 4. The tangential plane $\mathbf{x}_t$ through point $\hat{\mathbf{x}}_1$ is given by

$$\mathbf{x}_t = \hat{\mathbf{x}}_1 + k \left[1, 0, \left.\frac{\partial z_1}{\partial x_1}\right|_{\hat{\mathbf{x}}_1}\right]^T + l \left[0, 1, \left.\frac{\partial z_1}{\partial y_1}\right|_{\hat{\mathbf{x}}_1}\right]^T \tag{14}$$

with

$$\frac{\partial z_1}{\partial x_1} = -\frac{\partial z_1}{\partial X_1}\frac{f_x}{z_1}, \qquad \frac{\partial z_1}{\partial y_1} = -\frac{\partial z_1}{\partial Y_1}\frac{f_y}{z_1} \; . \tag{15}$$

The surface normal at point $\hat{\mathbf{x}}_1$ can then be written as

$$\mathbf{n}_{\hat{\mathbf{x}}_1} = \left[-\frac{\partial z_1}{\partial x_1}, -\frac{\partial z_1}{\partial y_1}, 1\right]^T \Bigg|_{\hat{\mathbf{x}}_1} \; . \tag{16}$$

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. 4. The shifted plane becomes

$$\mathbf{x}_{ts} = \mathbf{x}_t + (\hat{\mathbf{x}}_1 - \mathbf{t}_1)\Delta s \; . \tag{17}$$

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

$$\mathbf{x}_{ls} = \lambda \hat{\mathbf{x}}_1 \; , \tag{18}$$

leading to the new object point $\mathbf{x}_1$

$$\mathbf{x}_1 = \hat{\mathbf{x}}_1 \left(1 + \Delta s \left(1 - \frac{{\mathbf{t}_1}^T \mathbf{n}_{\hat{\mathbf{x}}_1}}{\hat{\mathbf{x}}_1^T \mathbf{n}_{\hat{\mathbf{x}}_1}}\right)\right) \; . \tag{19}$$

For a given 3-D motion $\mathbf{R}_{12}$, and $\mathbf{t}_{12}$ from frame I_1 to frame I_2 the points $\hat{\mathbf{x}}_1$ 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 $\hat{\mathbf{x}}_1$ represents the inaccurate object surface point position and $\mathbf{x}_1$ the correct position, the motion-compensated version of the inaccurate object point $\hat{\mathbf{x}}_1$ becomes

$$\hat{\mathbf{x}}_2 = \mathbf{R}_{12}(\hat{\mathbf{x}}_1 - \mathbf{t}_1) + \mathbf{t}_1 + \mathbf{t}_{12} \; . \tag{20}$$

For the corresponding object point $\mathbf{x}_1$ after deformation we obtain

$$\mathbf{x}_2 = \mathbf{R}_{12}(\mathbf{x}_1 - \mathbf{t}_1) + \mathbf{t}_1 + \mathbf{t}_{12} \; . \tag{21}$$

Projection into the image plane and Taylor series expansion of first order leads to the image displacements u_s and v_s in horizontal and vertical direction

due to shape deformation

$$
\begin{aligned}
u_s &= X_2 - \hat{X}_2 \approx -\frac{\Delta s}{\hat{z}_2}\left(1 - \frac{\mathbf{t_1}^T \mathbf{n}_{\hat{\mathbf{x}}_1}}{\hat{\mathbf{x}}_1^T \mathbf{n}_{\hat{\mathbf{x}}_1}}\right) \cdot \\
&\left(f_x\left(\mathbf{r_1 t_1} - t_{1x} - t_{12x}\right) + \hat{X}_2\left(\mathbf{r_3 t_1} - t_{1z} - t_{12z}\right)\right) \\
v_s &= Y_2 - \hat{Y}_2 \approx -\frac{\Delta s}{\hat{z}_2}\left(1 - \frac{\mathbf{t_1}^T \mathbf{n}_{\hat{\mathbf{x}}_1}}{\hat{\mathbf{x}}_1^T \mathbf{n}_{\hat{\mathbf{x}}_1}}\right) \cdot \\
&\left(f_y\left(\mathbf{r_2 t_1} - t_{1y} - t_{12y}\right) + \hat{Y}_2\left(\mathbf{r_3 t_1} - t_{1z} - t_{12z}\right)\right)
\end{aligned} \tag{22}
$$

with

$$
\mathbf{R_{12}} = \begin{bmatrix} \mathbf{r_1} \\ \mathbf{r_2} \\ \mathbf{r_3} \end{bmatrix}, \mathbf{t_{12}} = \begin{bmatrix} t_{12x} \\ t_{12y} \\ t_{12z} \end{bmatrix}, \mathbf{t_1} = \begin{bmatrix} t_{1x} \\ t_{1y} \\ t_{1z} \end{bmatrix}. \tag{23}
$$

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

Equation (22) is valid for every object surface point $\hat{\mathbf{x}}_1$. The surface, however, is modeled using a finite set of control points. Each object surface point is described by a linear combination of 3 control points. We therefore replace Δs in (22) by

$$
\Delta s = b_i \Delta s_i + b_j \Delta s_j + b_k \Delta s_k \tag{24}
$$

with b_i, b_j, b_k being the barycentric coordinates for the object point $\hat{\mathbf{x}}_1$ 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 (22) and (24) with the optical flow constraint

$$
\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \cdot u_s + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \cdot v_s \approx \hat{I}_2 - I_2 \tag{25}
$$

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

$$
a_i \Delta s_i + a_j \Delta s_j + a_k \Delta s_k = \hat{I}_2 - I_2 \ . \tag{26}
$$

The three indices i, j, and k represent the three control points of the triangle enclosing the surface point $\hat{\mathbf{x}}_1$. The coefficients a_i, a_j, and a_k are given as

$$
\begin{aligned}
a_i &= b_i \left(\frac{\partial \hat{I}_2}{\partial \hat{X}_2}\frac{u_s}{\Delta s} + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2}\frac{v_s}{\Delta s}\right) \\
a_j &= b_j \left(\frac{\partial \hat{I}_2}{\partial \hat{X}_2}\frac{u_s}{\Delta s} + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2}\frac{v_s}{\Delta s}\right) \\
a_k &= b_k \left(\frac{\partial \hat{I}_2}{\partial \hat{X}_2}\frac{u_s}{\Delta s} + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2}\frac{v_s}{\Delta s}\right) \ .
\end{aligned} \tag{27}
$$

Similar to Section 4.1 the resulting over-determined linear system of equations can be solved in a least-squares sense.

4.3 Combined 3-D Shape and 3-D Motion Estimation

Both, motion and shape errors, are considered by combining the displacements (u_m, v_m) in (10) and (u_s, v_s) in (22). Together with the optical flow constraint we now obtain the linear equation

$$
\frac{\partial \hat{I}_2}{\partial \hat{X}_2} \cdot (u_m + u_s) + \frac{\partial \hat{I}_2}{\partial \hat{Y}_2} \cdot (v_m + v_s) \approx \hat{I}_2 - I_2 \tag{28}
$$

with the $N_{cp} + 6$ unknown parameters $\Delta s_0 \ldots \Delta s_{N_{cp}-1}$ and ΔR_x, ΔR_y, ΔR_z, Δt_x, Δt_y, Δt_z. This equation can be setup for each pixel position that is covered by the object. Since the number of pixels corresponding to the object 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 again requires an iterative solution using the hierarchical estimation scheme described at the end of Section 4.1.

So far we have considered only two frames I_1 and I_2. In the case of N available views $I_1 \ldots I_N$ 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 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 $N_{cp} + 6(N-1)$.

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

In the first experiment, we use 20 frames of a synthetic test sequence showing a video cassette of size $12cm \times 20cm \times 4cm$. Fig. 5 depicts two frames of the sequence. The camera remains fixed for all frames while the object motion varies between $\pm 45^o$ for the rotation and $\pm 5cm$ for the translation. The initial model is adapted to the silhouette

Figure 5: Frame 1 and 10 of the cassette sequence as well as the initial object geometry.

of the first frame (Fig. 5) and the extension in z-direction of the cassette is erroneously selected to be $6cm$. This introduces a considerable shape error which prevents the 3-D model based motion estimator from providing accurate motion parameters for the 20 frames. The combined shape and motion estimator as described in the previous section, however, can correct the shape errors and improve the motion parameter estimates. Starting from the initial object shape, the relative motion for all 20 views is estimated using the rigid body model-based motion estimator in Section 4.1. This corresponds to the left-most points on the curves in Fig. 6. It can be seen that the erroneous shape causes a considerable motion error. We then use the simultaneous shape and motion estimation as described in Section 4.3 to obtain a shape refinement of our initial object model. The shape refinement is performed using a varying number of frames of the sequence (6, 12, and 20). Fig. 6 shows the average deviation of the refined object shape from the correct shape of the cassette in mm as a function of the number of frames employed. The resulting object shape is then again used to determine the relative motion for all views. It can be observed that increasing the number of frames used for simultaneous shape and motion estimation improves the object shape (Δz in Fig. 6) and leads to more accurate motion estimates (ΔR and Δt in Fig. 6). Best results are obtained when estimating shape and motion simultaneously from all available views which corresponds to the right-most points on the curves in Fig. 6.

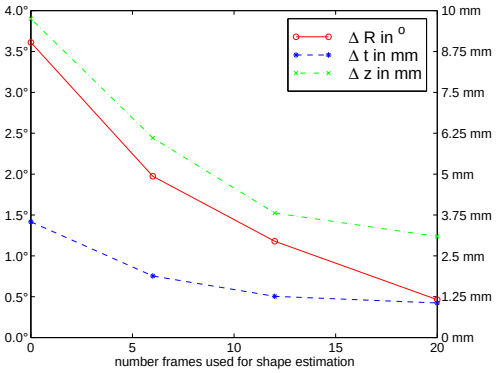

Figure 6: Average rotational and translational motion error as a function of the number of frames used for simultaneous shape and motion estimation. Δz represents the average deviation of the estimated from the original object depth.

In our second experiment we use 29 frames of the $Garfield$ sequence captured using a camera mounted on a robot arm [8]. Fig. 7 shows three views after object silhouette extraction and background removal. The views are calibrated using a reference calibration object [6]. Given these view calibration parameters we compare the following three cases for object shape recovery. In the first case we use the shape-from-silhouette step in Section 3 only. In the second case we refine this initial object shape using the shape estimation algorithm described in Section 4.2. For the third case we employ combined shape and motion estimation as described in Section 4.3 for object shape refinement.

Figure 7: Three views of the test sequence *Garfield* after silhouette extraction and background removal.

We measure the MSE between motion-compensated images and the original views for these three cases. For motion compensated prediction we map a reference image on the estimated shape model and render it with view parameters of neighboring frames. The larger the deviation of the model shape from the actual object shape, the larger the mean squared error after motion compensation. MSE values are converted into PSNR values using

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \qquad (29)$$

where larger PSNR values correspond to better motion-compensated prediction. In order to simulate the influence of inaccurate view calibration

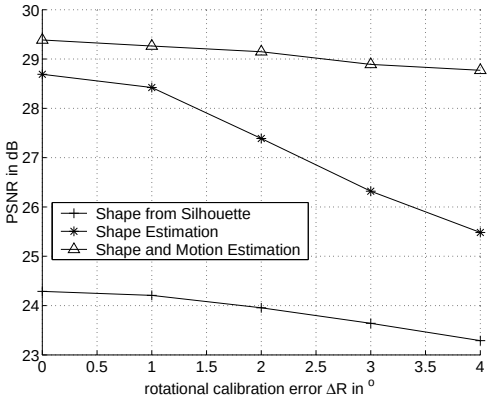

Figure 8: PSNR after motion-compensated prediction as a function of initial rotational calibration error.

data, we modify the rotational component of the calibration data in steps of 1 degree. A rotational error of zero degree means we use the original view calibration data that we obtained from camera calibration. The larger the initial calibration error the larger the deviation of the initial object shape from the actual shape. Fig. 8 shows the PSNR values after motion compensation as a function of the initial rotational calibration error. The motion-compensated images show poor quality (small PSNR values) if the object shape after the shape-from-silhouette step is used. After shape estimation as described in Section 4.2 we observe reduced prediction error (larger PSNR). The quality of the motion-compensated pictures decreases, however, rapidly as the calibration error increases. The third curve in Fig. 8 shows the results after combined motion and shape estimation as proposed in Section 4.3. Here, even for large initial calibration errors the shape and motion errors are corrected which leads to significantly higher PSNR values for the motion compensated images.

6 Conclusions

In this paper we present a formulation for simultaneous shape refinement and motion parameter estimation from multiple camera views. Image displacements due to erroneous shape and motion are linearly related to the observable image intensity gradients. Shape and motion refinements are estimated simultaneously in order to exploit their mutual dependency. Our formulation of object deformation allows the object texture to slide on the object surface in order to reduce image distortions due to shape modifications. Experimental results show that combined shape and motion estimation leads to a considerable improvement in comparison to independent estimation of shape or motion.

References

[1] A. Pentland and B. Horowitz, "Recovery of Nonrigid Motion and Structure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 7, pp. 730-742, July 1991.

[2] D. Terzopoulos and D. Metaxas, "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 7, pp. 703-714, July 1991.

[3] D. DeCarlo and D. Metaxas, "The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation," *Proc. CVPR '96*, pp. 231-238, 1996.

[4] 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, no. 6, pp. 545-555, June 1993.

[5] R. Koch, "Dynamic 3-D Scene Analysis through Synthesis Feedback Control," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 6, pp. 556-568, June 1993.

[6] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp. 323-344, August 1987.

[7] B. K. P. Horn, "Robot Vision," *MIT Press*, Cambridge, 1986.

[8] H. Niemann, B. Girod, H.-P. Seidel, B. Heigl, W. Heidrich, and M. Magnor, "The sfb 603 - model based analysis and visualization of complex scenes and sensor data", *In J. Dassow and R. Kruse, editors, Informatik '98 - Informatik zwischen Bild und Sprache. 28. Jahrestagung der Gesellschaft für Informatik, Magdeburg, Germany, Springer Lecture Notes in Computer Science*, pp. 319-328, Berlin, 1998.

[9] P. Eisert, E. Steinbach, and B. Girod, "Automatic Reconstruction of 3-D Stationary Objects from Multiple Uncalibrated Camera Views," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 261-277, vol. 10, no. 2, March 2000.