Motion-Based Analysis and Segmentation of Image Sequences using 3-D Scene Models

Eckehard Steinbach, Peter Eisert, and Bernd Girod

Telecommunications Laboratory,
University of Erlangen-Nuremberg
Cauerstrasse 7, 91058 Erlangen, Germany
{steinb, eisert, girod}@nt.e-technik.uni-erlangen.de

Abstract

In this paper we present an algorithm for automatic extraction and tracking of multiple objects from a video sequence. Our approach is model-based in the sense that we first use a robust structure-from-motion algorithm to identify multiple objects and to recover initial 3-D shape models. Then, these models are used to identify and track the objects over multiple frames of the video sequence. The procedure starts with recovering a dense depth map of the scene using two frames at the beginning of the sequence, and representing the scene as a 3-D wire-frame computed from the depth map. Texture extracted from the video frames is mapped onto the model. Once the initial models are available we use a linear and low complexity algorithm to recover the motion parameters and scene structure of the objects for the subsequent frames. Combining the new estimates of depth and the initially computed 3-D models into an unstructured set of 3-D points with associated color information, we obtain updates of the 3-D scene description for each additional frame. We show that the usage of a 3-D scene model is suitable to analyze complex scenes with several objects. In our experimental results, we apply the approach presented in this paper to the problem of video sequence segmentation, object tracking, and video object plane (VOP) generation. We separate the video sequences into different layers of depth and combine the information from multiple frames to a compact and complete description of these layers.

1 Introduction

One of the objectives of MPEG-4 is to provide content-based manipulation of objects in image sequences [1, 2, 3]. For that purpose several image object representations like video object planes (VOPs), sprites or synthetic scene descriptions like SNHC [4] have been defined. However, it is not specified in MPEG-4, how this additional object information is extracted from video sequences. This task is left to the user and one of the possible approaches to solving this is addressed in this paper.

To create an object-based scene representation of a video sequence it is necessary to segment different objects in images. This segmentation can be based on motion information as initially demonstrated in the layered representation of moving images proposed in [5] and later refined in [6]. However, this approach leads to 2-D representations of objects and is limited to motion scenarios that can be described by a 2-D affine transformation. It is therefore desirable to describe the objects in the scene in terms of their 3-D shape and texture. This requires the construction of a 3-D model from 2-D images.

For content-based representation of video sequences as addressed in the MPEG-4 effort, model-based techniques seem to be particularly suited since it is very likely that the physical objects in the 3-D scene represent the content we are interested in. Model-based analysis of video sequences has been shown to be a promising approach for various applications like video coding, object tracking and object recognition. In particular, very low bit rate video coding has received considerable attention in the literature [7] ... [11]. These model-based algorithms generally extract motion and texture parameters of the objects from the video frames using three-dimensional models and synthesize the objects at the decoder using transmitted pose and texture update information.

We can basically distinguish two classes of model-based image sequence analysis algorithms. First, approaches where an initial 3-D model (generic or explicit) of the objects present in the scene is available, and second, techniques that build up and continuously refine a 3-D model of the scene using information cues such as structure-from-motion, structure-from-shading, stereo information etc.

In the first class we usually know the number and types of the objects in the scene. This represents a considerable restriction on the image sequences that can be processed. In addition, expensive equipment like a 3-D laser scanner or a range finder are often required to obtain these models. The usage of generic models is an alternative to reduce some of the before mentioned restrictions since the generic model can initially be adjusted to the object(s) present in the scene. We therefore do not need to record a 3-D model for each new sequence, but can process the same type of sequence using a single generic model.

The second class of algorithms, including the one described in this paper, can process a larger variety of image sequences. An initial model is extracted from the first couple of frames of the sequence and is continuously refined and updated along the image sequence. 3-D structure-from-motion algorithms are particularly suited to accomplish this task since the relative motion between the camera and the objects in the scene is a powerful cue for the recovery of the 3-D shape of objects from two (or more) 2-D views. Koch describes in [12] a system that automatically extracts 3-D object shape and 3-D motion from a video sequence where the model of the scene is adapted to the real scene via comparison of the visual reconstruction of the model scene with the real input sequence. For monocular image sequences the initial model is derived from 2-D object silhouettes with or without scene specific knowledge, depending on the application.

The algorithm described in this paper first extracts an initial 3-D representation of the objects in the scene using a robust 3-D motion and structure estimation algorithm that has been shown to successfully deal with multiple moving objects [13]. The initial model is continuously refined over the following frames using a linear and low complexity model-based algorithm. The recovered 3-D motion and depth information allows to manipulate and separate the objects in the scene.

The remainder of this paper is organized as follows. We first present a description of the proposed algorithm (section 2). In section 3 the underlying geometry and the constraints imposed in our approach are shown. In section 4 we discuss the robust structure-from-motion algorithm used to estimate an initial 3-D representation of the objects in the video sequence. We then describe in section 5 how this initial model is used in combination with a linear 3-D motion estimation algorithm to track the objects in the following frames. In section 6 we apply the algorithm presented in this paper to the well known *Flowergarden* sequence with the aim of automatic scene segmentation and demonstrate that the scene objects are successfully separated using the recovered depth information. Combination of motion and structure information from multiple frames finally leads to a compact and complete representation of the objects in the scene that can be used, for example as Video Object Planes (VOP), in the context of MPEG-4.

2 Building the 3-D Scene Representation

For the segmentation and manipulation of video sequences we use a model-based, three-dimensional representation of the scene. Each moving object in the scene is modeled with an unstructured set of 3-D points and associated color information. The 3-D motion of all objects is tracked individually with a low complexity model-based motion estimation algorithm. The use of 3-D motion parameters and depth information for the segmentation

into individual objects is more accurate than approaches based on 2-D affine motion models and VOPs can directly be generated from the model geometry. Additionally, we are able to reconstruct temporarily occluded areas because the models contain information about shape and texture of these object parts. The basic structure of the proposed algorithm is depicted in Fig. 1.

For the first two frames of a video sequence no prior knowledge about depth and motion in the scene is available. Therefore, we use the robust and complex algorithm described in section 4 that estimates the motion of all objects from two successive frames. From the estimated motion parameters a dense map of depth of the objects is computed. From these depth maps initial 3-D wire-frame models are constructed and texture is extracted from the images. Having obtained this information, we use a low complexity model-based algorithm described in section 5 which estimates the 3-D motion in all following frames. Using the motion parameters the structure of an object is determined for the next frame and the depth map is combined with the motion compensated 3-D points to get an updated scene model which becomes more and more complete over time. This is particularly important if a foreground object covers those parts in the scene we are interested in. Whenever an object point becomes visible in two successive frames we recover a depth and texture estimate of this point and can incorporate it into our 3-D representation of the scene.

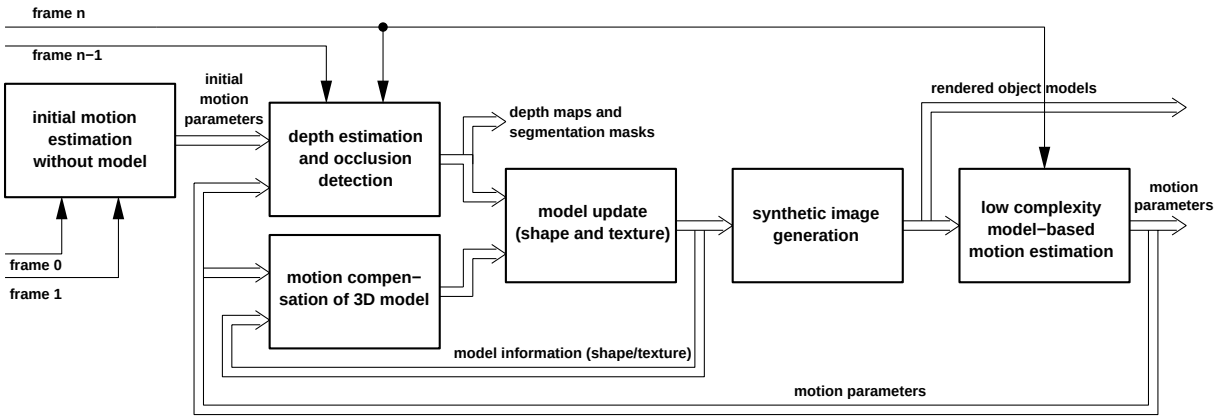

Figure 1: Basic structure of the proposed algorithm.

3 Basic Geometry

Fig. 2 illustrates the underlying geometry for the 3-D scene structure and motion estimation, with P_1, P_1' representing an object point before and after the motion, respectively.

The motion of the objects in the scene is assumed to be rigid body motion and can be

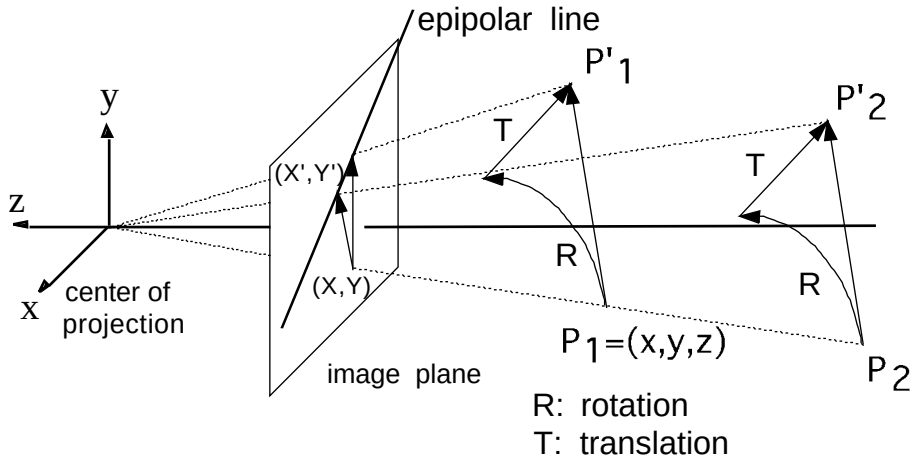

Figure 2: Basic undelying geometry illustrating perspective projection and the epipolar line

described as

$$[x'\ y'\ z']^T \ = \ [\mathbf{R}][x\ y\ z]^T + \mathbf{T} \tag{1}$$

with $[\mathbf{R}]$ the 3×3 rotation matrix containing the nine elements $r_1 \ldots r_9$ and $\mathbf{T} = [T_x\ T_y\ T_z]^T$ the translation vector. We assume perspective projection which is described by

$$\begin{aligned} X = -\frac{x}{z} \qquad & Y = -\frac{y}{z} \\ X' = -\frac{x'}{z'} \qquad & Y' = -\frac{y'}{z'}\ . \end{aligned} \tag{2}$$

From a monocular image sequence, we can expect to recover 3-D rotation, but 3-D translation only up to a common scale factor. This is illustrated in Fig. 2 for a point P_2 that is projected to the same image point (X_2, Y_2) as P_1 but to a different point (X_2', Y_2') along the epipolar line. We therefore wish to estimate the 5-dimensional motion parameter set describing the projection of the rigid body motion. In the following we will normalize the translation vector to unit length without loss of generality. Following the derivation in [14] we now combine (1) and (2) to obtain

$$\begin{aligned} X' \ &= \ -\frac{(-r_1X - r_2Y + r_3)z + T_x}{(-r_7X - r_8Y + r_9)z + T_z} \\ Y' \ &= \ -\frac{(-r_4X - r_5Y + r_6)z + T_y}{(-r_7X - r_8Y + r_9)z + T_z}\ . \end{aligned} \tag{3}$$

Modification of (3) leads to

$$\begin{aligned} z &= \frac{T_x + T_z X'}{X'(r_7 X + r_8 Y - r_9) + (r_1 X + r_2 Y - r_3)} \\ z &= \frac{T_y + T_z Y'}{Y'(r_7 X + r_8 Y - r_9) + (r_4 X + r_5 Y - r_6)} \,. \end{aligned} \tag{4}$$

Equating the right-hand sides of (4) finally results in

$$[X' \; Y' \; 1] \, \mathbf{E} \, [X \; Y \; 1]^T \; = \; 0 \tag{5}$$

with

$$\mathbf{E} = \begin{bmatrix} T_z r_4 - T_y r_7 & T_z r_5 - T_y r_8 & -T_z r_6 + T_y r_9 \\ T_x r_7 - T_z r_1 & T_x r_8 - T_z r_2 & -T_x r_9 + T_z r_3 \\ -T_y r_1 + T_x r_4 & -T_y r_2 + T_x r_5 & T_y r_3 - T_x r_6 \end{bmatrix} . \tag{6}$$

Equation (5) represents a straight line in the image plane, the epipolar line. Please note that the $\mathbf{E}$ matrix in (6) can be multiplied with any scalar and the equality in (5) still holds. This simply means that there is a common scale factor for the translation parameters that cannot be determined.

4 Building the Initial 3-D Model of the Scene

For model-based video sequence analysis we require a 3-D description of the objects (typically shape and texture). Building this model from the first couple of frames of the image sequence can be achieved using a structure-from-motion algorithm that computes relative depth values for the object points after motion estimation. Traditionally, structure-from-motion algorithms utilize a two-stage approach. First, feature points are extracted from the current image and their correspondence to features in the previous image is established. In a second step, rigid body motion parameters and depth values are computed from these feature point correspondences. There is no feedback from the computation of motion parameters and depth to the feature matching process, and, typically, the results are very sensitive to errors in feature correspondences [14, 15, 16, 17]. In [18] Steinbach *et al.* presented an algorithm that does not separate feature matching and 3-D motion recovery computation and thus overcomes the inherent limitations of the conventional two-stage approach. The algorithm is based on the geometrical observations in section 3. Please remember that feature correspondences for a given 3-D motion are constrained to lie on the epipolar line (5) in the image, and that the position along the epipolar line corresponds to different depth values.

We now describe the basic steps of the algorithm used to compute the initial estimate of the 3-D motion parameters and scene structure. Fig. 3 shows the corresponding block

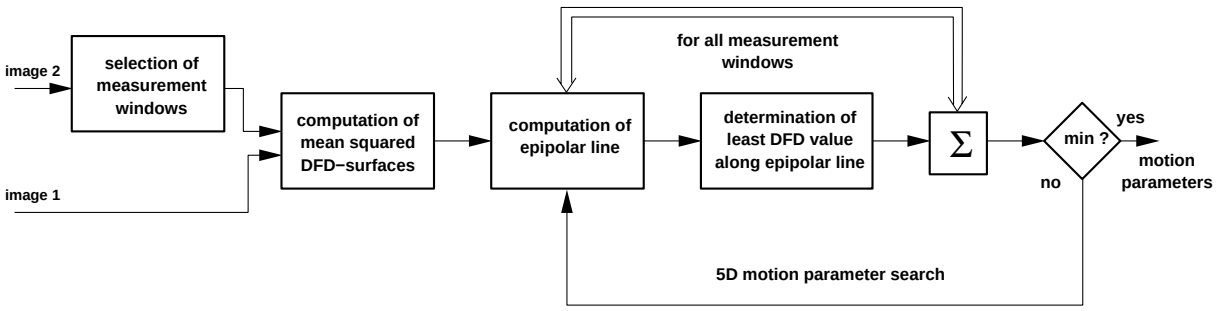

Figure 3: Block diagram of the 3-D motion estimation algorithm that is used to estimate the initial motion between the first two frames of the image sequence.

diagram. Considering the first two frames of a video sequence I_1 and I_2, we first divide I_2 into rectangular measurement windows of fixed size (typically 7×7 or 15×15 pixels) and compute mean squared *displaced frame difference* (DFD) surfaces with respect to image I_1. The DFD surface at a point (X_i, Y_i) is given by

$$DFD(X_i,Y_i;d_X,d_Y) = \sum_{(r,s)\in(N,M)} \left(I_1(X_i+d_X+r,Y_i+d_Y+s) - I_2(X_i+r,Y_i+s)\right)^2 \tag{7}$$

where I_1 and I_2 are the intensity images, (d_X, d_Y) the displacement, and N, M the size of the measurement windows in horizontal and vertical directions, respectively. Note that the computation of the DFD surfaces is very similar to the full-search *block matching*. The epipolar line can be computed as shown in (5) given the assumed motion parameter set and the image location of the measurement window. As shown in Fig. 3 we first compute the epipolar lines and then determine the smallest values along these lines on the DFD surfaces. These minima are accumulated for all measurement windows. The search for the minimum DFD value along the epipolar line is currently performed as follows. We step along the epipolar line in equidistant intervals (typically 0.25 pixels) and evaluate the bilinearly interpolated values of the DFD surface.

The cost function to be minimized for F measurement windows selected in the image is therefore given by

$$\min_{\mathbf{E}} \sum_{i=1}^{F} \min_{(X_i+d_X,Y_i+d_Y)\in L_i(\mathbf{E})} DFD(X_i, Y_i; d_X, d_Y) \tag{8}$$

with $L_i(\mathbf{E})$ the epipolar line for the *ith* measurement window. Since the search for the correspondence on the epipolar line requires the knowledge of the 3-D motion parameters

the algorithm searches the 5-dimensional motion parameter space. The motion parameter set leading to the smallest accumulation value is considered to be optimal. In our current implementation we employ a coarse-to-fine search in the 5-D motion parameter space to determine a local minimum of the cost function in (8). We assume a maximum angle for the rotation (typically $-2^o \leq R_x, R_y, R_z \leq 2^o$) and evaluate the 5-D motion parameter space on a coarse grid. The grid points leading to the smallest values in (8) are then refined in a local neighborhood with higher motion parameter resolution. An additional conjugate direction search (Powell-search [30]) is used to further refine the estimate. Please note that the search stops in a local minimum of (8). The minimization of (8) over the 5-D space leads to a large number of evaluations. In [18] Steinbach *et al.* introduced the *Epipolar Transform* reducing the evaluation of the cost function to a simple table look-up. For a detailed description of the *Epipolar Transform* and complexity considerations please refer to [18].

Now, given the motion parameters, a dense map of depth can be recovered for all points in the scene by searching for their displacement vectors (d_X, d_Y) along the corresponding epipolar lines and evaluating (4). The depth map is scaled with the same factor as the translation vector.

In general real world scenes we have to deal with several objects undergoing different motions in the 3-D space. The structure-from-motion algorithm therefore has to recover the motion of each object independently in order to be able to compute meaningful depth maps for all objects in the scene. Recently, there had been some attempts to identify and estimate the various motion components present in the scene [19, 20, 21, 22]. Let us now consider the case where there are two objects undergoing different rigid motions, say $\mathbf{E_1}$ and $\mathbf{E_2}$. Obviously, the least squares minimization of the cost function as given in equation (8) does not yield good results as the epipolar lines corresponding to these motions for a particular point could be very different. While estimating the dominant motion $\mathbf{E_1}$, the second motion field creates outliers that must be identified and rejected in order to obtain a good estimate of the dominant motion parameters. In [13] we proposed an extension to [18] that uses the *least median of squares* (LMedS) estimator [23] to differentiate or classify the motion field into constituent groups.

The LMedS estimator has been used in various computer vision applications, including motion analysis [24]. An extremely important property is that the LMedS estimator can tolerate up to 50% data contamination by outliers [23]. Here, one replaces the mean of the squared residuals by their median to achieve the robustness. Hence, we modify our cost function in (8) to estimate the motion parameters $\mathbf{E_1}$ into

$$\min_{\mathbf{E_1}} \left(med_{\forall i} \left\{ \min_{(X_i+d_X, Y_i+d_Y) \in L_i(\mathbf{E_1})} DFD(d_X, d_Y; X_i, Y_i) \right\} \right) \tag{9}$$

There is an inherent assumption in using equation (9) to estimate the motion parameters $\mathbf{E_1}$. At least half of the measurement windows used in the estimation process have to belong to the object moving with motion $\mathbf{E_1}$. In other words, the cardinality of the segmented region belonging to the first object when normalized with respect to the total size of the image must be greater than 0.5. The evaluation of the cost function in (9) is slightly more complex than the evaluation of (8). Instead of simply accumulating the minima along the epipolar lines we now have to sort them and determine the median. Please note that the evaluation of (8) or (9) has to be performed for each potential motion set during the 5-D parameter space search since no closed form solution for the minimization can be derived.

Having estimated the first component of the motion, a dense map of displacement vectors for all pixels is obtained. The residual error for each feature point given by the minimum of the DFD surface along the corresponding epipolar line is sorted by their magnitude. A search is now initiated over the sorted residuals in order to locate the break point when there is a sudden large increase in magnitude. The points above the break point are outliers and should belong to the second object undergoing a different motion $\mathbf{E_2}$. Hence, we achieve an automatic segmentation of the scene based on motion parameters. The parameters $\mathbf{E_2}$ are now estimated using equation (8), but the computation is restricted to the segment that belongs to the second object.

Theoretically, the above procedure can very easily be generalized to deal with even a larger number of motion components in the field of view of the camera, provided that the cardinality of the region belonging to the next dominant motion is larger than the cardinality of the rest of the objects, by simply reiterating the procedure on the remaining region. However, recovering the 3-D motion and structure for many objects becomes difficult since the region of support for each object is typically smaller. This means that for small objects only a small number of measurement windows is available. A different formulation of this fact is that the active viewing angle per object becomes smaller which in turn makes 3-D motion estimation more difficult [16].

5 Model-based 3-D Motion Estimation

Given the previously described initial 3-D model of the objects in the scene and assuming a constant focal length we can derive a linear and low complexity algorithm used for motion estimation in the following frames [25]. Under the assumption of rigid body motion (1) and given depth the epipolar line is reduced to a single point and no search along the epipolar line has to be performed. This simplifies the task of motion estimation and makes

it possible to use a linear algorithm for the estimation that is much faster than the initial one which cannot incorporate depth information.

Our model-based algorithm for motion estimation is based on [26, 27, 12] and is extended to a hierarchical framework. Instead of using feature points or dividing the image in measuring windows the entire image part that covers the object is taken into account. For this purpose we use the well known optical flow constraint equation that can be set up at each pixel position

$$\frac{\partial I}{\partial X_p} \cdot u + \frac{\partial I}{\partial Y_p} \cdot v + \frac{\partial I}{\partial t} = 0 \tag{10}$$

where $\frac{\partial I}{\partial X_p}$, $\frac{\partial I}{\partial Y_p}$ and $\frac{\partial I}{\partial t}$ denote the partial derivatives of the intensity I in X_p, Y_p and t directions, respectively, and $[u \;\; v]^T$ the velocity vector in the image plane. In the discrete case $[u \;\; v]^T$ becomes the displacement vector that is a function of the unknown motion parameters. The smoothness constraint that is normally used for the computation of an optical flow field is here replaced by the object motion model. Adding this constraint, which takes the 3-D motion equation and the camera model into account, to the optical flow constraint equation leads to a linear system of equations that can be solved in a least-squares sense.

Since we are looking for a linear solution for the estimation problem we have to employ some linearizations that are justified by the assumption of small motion between two successive video frames in the sequence. The linearized version of the rotation matrix in equation (1) is given by

$$R \approx \begin{bmatrix} 1 & -R_z & R_y \\ R_z & 1 & -R_x \\ -R_y & R_x & 1 \end{bmatrix} \tag{11}$$

where R_x, R_y and R_z are the rotational angles around the x-, y- and z-axis. The trajectory of an object point in the 3-D space specified by equation (1) is then projected in the 2-D image plane using the camera model (2). After applying a first order approximation we obtain for the 2-dimensional displacement the following equation:

$$\begin{aligned} u = X'_p - X_p &\approx f_x \left[-R_y - R_z Y_p - \frac{T_x}{z} + X_p \left(R_x Y_p - R_y X_p - \frac{T_z}{z} \right) \right] \\ v = Y'_p - Y_p &\approx f_y \left[R_x + R_z X_n - \frac{T_y}{z} + Y_p \left(R_x Y_p - R_y X_p - \frac{T_z}{z} \right) \right], \end{aligned} \tag{12}$$

where z denotes the scaled depth obtained from the model. The factors f_x and f_y transform the image plane coordinates in (2) into pixel coordinates

$$X_p = f_x \cdot X \;, \quad Y_p = f_y \cdot Y \;\; . \tag{13}$$

Combining this description for the rigid body motion with the optical flow constraint equation (10) results in a linear equation for the six unknown motion parameters

$$a_0 R_x + a_1 R_y + a_2 R_z + a_3 T_x + a_4 T_y + a_5 T_z = -\frac{\partial I}{\partial t} \tag{14}$$

with the parameters a_0 to a_5 given by

$$\begin{aligned}
a_0 &= f_x \frac{\partial I}{\partial X_p} X_p Y_p + f_y \frac{\partial I}{\partial Y_p} \left(1 + Y_p^2\right) \\
a_1 &= -f_x \frac{\partial I}{\partial X_p} \left(1 + X_p^2\right) - f_y \frac{\partial I}{\partial Y_p} X_p Y_p \\
a_2 &= -f_x \frac{\partial I}{\partial X_p} Y_p + f_y \frac{\partial I}{\partial Y_p} X_p \\
a_3 &= -f_x \frac{\partial I}{\partial X_p} \frac{1}{z} \\
a_4 &= -f_y \frac{\partial I}{\partial Y_p} \frac{1}{z} \\
a_5 &= -f_x \frac{\partial I}{\partial X_p} \frac{X_p}{z} - f_y \frac{\partial I}{\partial Y_p} \frac{Y_p}{z}.
\end{aligned} \tag{15}$$

At each pixel position of the object we obtain one equation and the resulting over-determined linear system of equations is solved in a least-squares sense. At least six equations are necessary for the algorithm but due to the large number of object pixels, each contributing one additional equation, we can discard some possible outliers. These outliers can be detected analyzing the partial derivatives of the intensity and the motion model. The optical flow constraint equation (10) is only valid for small displacement vectors because of the linearization of the intensity values. If the estimate of the displacement vector length for the pixel at position $[X_p Y_p]$

$$\hat{d}(X_p, Y_p) = \sqrt{\frac{(\frac{\partial I}{\partial t})^2}{(\frac{\partial I}{\partial X_p})^2 + (\frac{\partial I}{\partial Y_p})^2}} \tag{16}$$

is larger than a threshold G, the pixel is classified as an outlier and not used for motion estimation.

Note that although we are able to estimate all six motion parameters with this model-based motion estimator, the translation of an object can only be estimated up to a common scale factor because the size of the object cannot be determined during the model acquisition. However, in all successive frames the same scaling factor is utilized and the length of the estimated translation vector is specified relative to the first estimation which in our case is normalized to one.

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

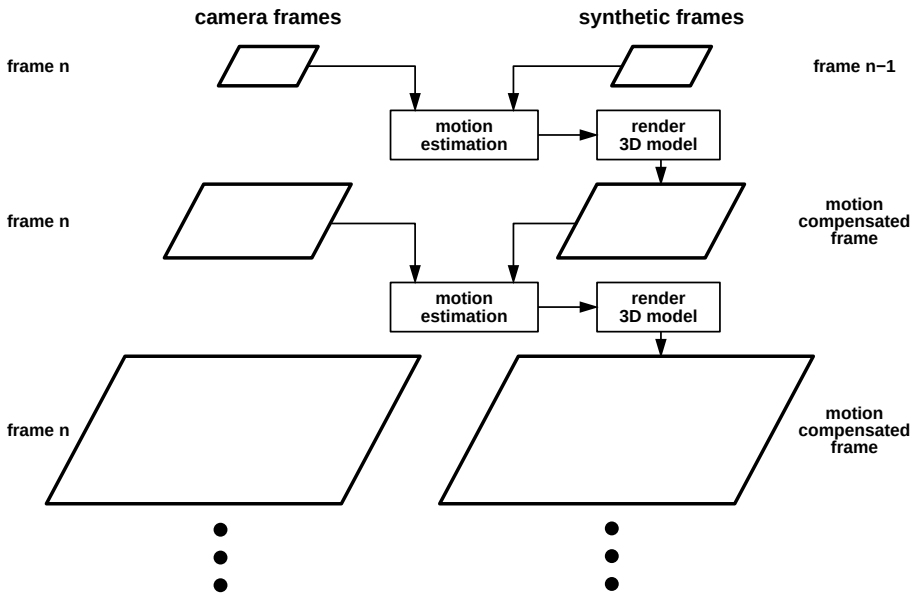

Figure 4: Image pyramid of the hierarchical motion estimation scheme.

vectors between two successive video frames. To overcome this limitation a hierarchical scheme is used for the motion estimation as shown in Fig. 4. 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. For the sub-sampling, simple moving average filters are used to reduce aliasing effects. The resulting images are then further filtered by a Gauss-filter to smooth the edges before estimating the motion. 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 getting a more accurate motion parameter set. In our current implementation we use four levels of resolution starting from 44 x 30 pixels. For each new level the resolution is doubled in both directions leading to a final resolution of 352 x 240 pixels (CIF). At the same time the threshold G used to detect outliers for the motion estimation is reduced from 5 (first level) to 0.5 (highest resolution) which means that at higher levels more pixels are classified as outliers. Experiments with this hierarchical scheme showed that it is able to estimate displacements of up to 30 pixels between two frames.

6 Experimental Results

In this section we apply the algorithms described above to the problem of scene segmentation or VOP generation. We use 24 frames of the *Flowergarden* sequence. The aim is to segment the scene into different layers using the 3-D motion information of the scene. In other words, the scene objects are separated using only depth information. Please note that we process only every second frame in the following. In a first step we use the method described in section 4 to estimate the motion and an initial structure of the scene. Fig. 5 a) shows frames 0 and 2 of the Flowergarden sequence. We estimate the motion

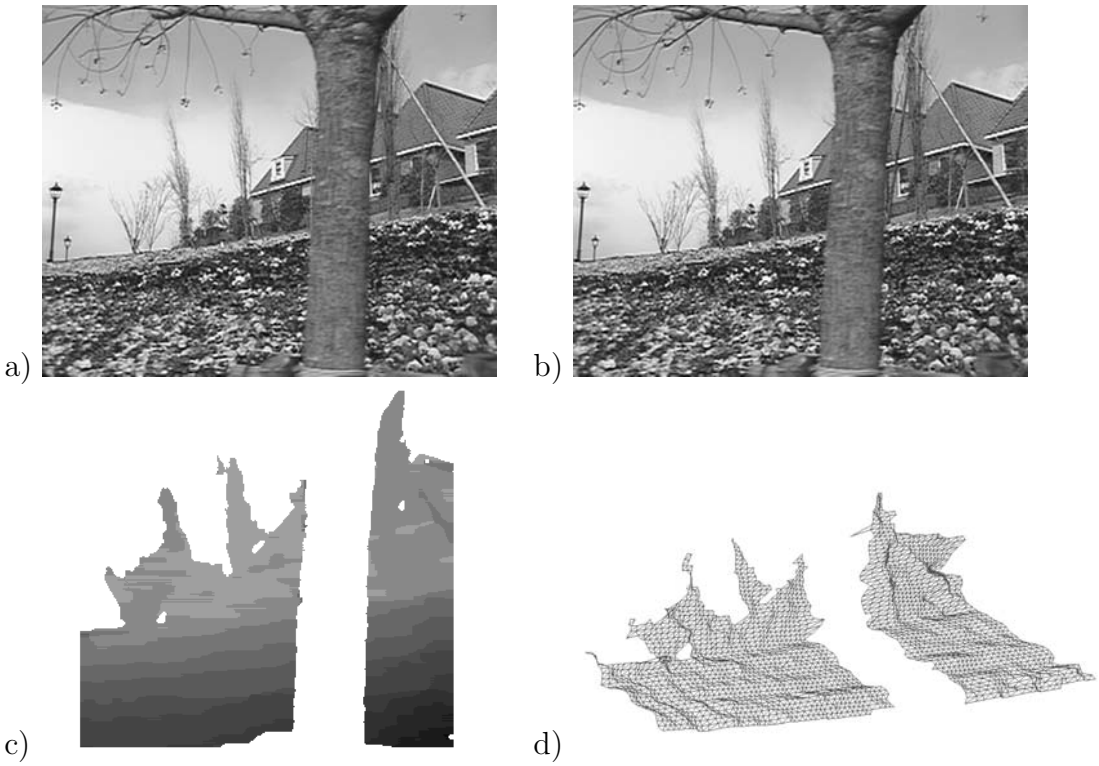

Figure 5: a) Frame 0 b) Frame 2 of the Flowergarden sequence, c) the recovered depth map using the motion estimate obtained from the algorithm in section 4, d) wire-frame representation computed from the depth map in c).

parameters to be $R_x = 0.012^o$, $R_y = -0.43^o$, $R_z = 0.048^o$, $T_x = -0.9977$, $T_y = -0.0287$, and $T_z = -0.0608$. Given the motion parameters we recover the dense map of depth using (4). For explicit occlusion detection and the imposition of neighborhood constraints we use a modified version of the stereo depth estimation algorithm by Falkenhagen [28]. This modification includes the extension of stereo constraints (neighboring and ordering, smoothness, occlusion etc.) to arbitrary epipolar geometry [29], and adaptive cost func-

tions for the explicit determination of occlusions. Fig. 5c shows the resulting dense map of depth for frame 0.

The depth map shown in Fig. 5c is slightly smaller than the original frame to simplify the correspondence search at the image borders. Please note that the tree is not included in the 3-D model since our aim is to segment the scene into different layers of depth. In this particular experiment we wish to recover the *Field and House* layer and therefore exclude the tree by setting a depth threshold. We now build a 3-D wire-frame representation (Fig. 5d) of the scene and motion compensate it together with the color information towards frame 2. Given this motion-compensated 3-D model we use the linear algorithm described in section 5 to estimate the motion parameters between frames 2 and 4. In our implementation the actual 3-D representation of the scene that is continously updated consists of an unstructured set of 3-D points plus corresponding color information. The advantage of updating an unstructured 3-D point set in comparison to updating a 3-D wire-frame is that adding new scene content results in simple padding of the new 3-D points to the list. We use the wire-frame only for motion estimation and therefore do not have to update it continously. In other words, for each depth map we compute the corresponding wire-frame but do not combine successive wire-frames.

Given the motion information from frames 2 to 4 we recover a dense map of depth for frame 2. We then motion compensate the 3-D points from frame 0 and combine them with the recovered depth map and color information of frame 2 into one set of 3-D points. Using the additional information we obtain a more complete 3-D representation of the *Field and House* layer since image points occluded by the tree in frame 0 became visible in frames 2 and 4. We now repeat this procedure with the subsequent frames. As a result we obtain a complete description of the *Field and House* layer since the tree uncovers all image parts at least once in the sequence and we therefore can include those parts in our 3-D model.

Rendering the set of 3-D points at a given time instant and viewing direction gives an impression about the temporal evolution of our 3-D scene model. This is illustrated in Fig. 6. Please note that the camera positions used to render the scene model for frames 8 and 16 do not coincide with the original viewing position at these frames. Since our aim is to reproduce the original sequence without the tree, we have to adopt the virtual camera to the actual viewing positions using the estimated motion parameters. The finally recovered image sequence for the *Field and House* layer excluding the tree is shown in Fig. 7 and 8. We show the original frame, the recovered depth map and the rendered scene model for frames 0, 8, 16, and 24 of the sequence.

Using this complete scene description for the particular layer of interest we can track the same scene part in the following frames and continuously refine our 3-D model and

Figure 6: Temporal evolution of the 3-D scene model at frame 0, 8, and 16.

the corresponding texture information.

So far the same motion parameters apply for the entire scene. In a second experiment we use a scene with 2 objects which have different relative motions with respect to the camera. Fig. 9 shows two successive frames of a person (object 1) moving in front of a static background (object 2). When we estimate the motion in the scene, the background pixels create outliers for the estimation of the motion of the foreground object and vice versa. We use again the robust algorithm described in section 4 to determine the motion of the objects in the scene. Since the person covers more than 50% of the image points, the algorithm estimates the motion of the person first. Fig. 9c shows the recovered depth map for the front object. In Fig. 9d we show the wire-frame representation of object 1 computed from the depth map in Fig. 9c. In Fig. 9e we finally show the video object plane obtained for the person. As in the previous experiment, no a priori segmentation is required. The motion information in combination with the robust 3-D motion and structure estimation algorithm in section 4 leads to automatic object separation.

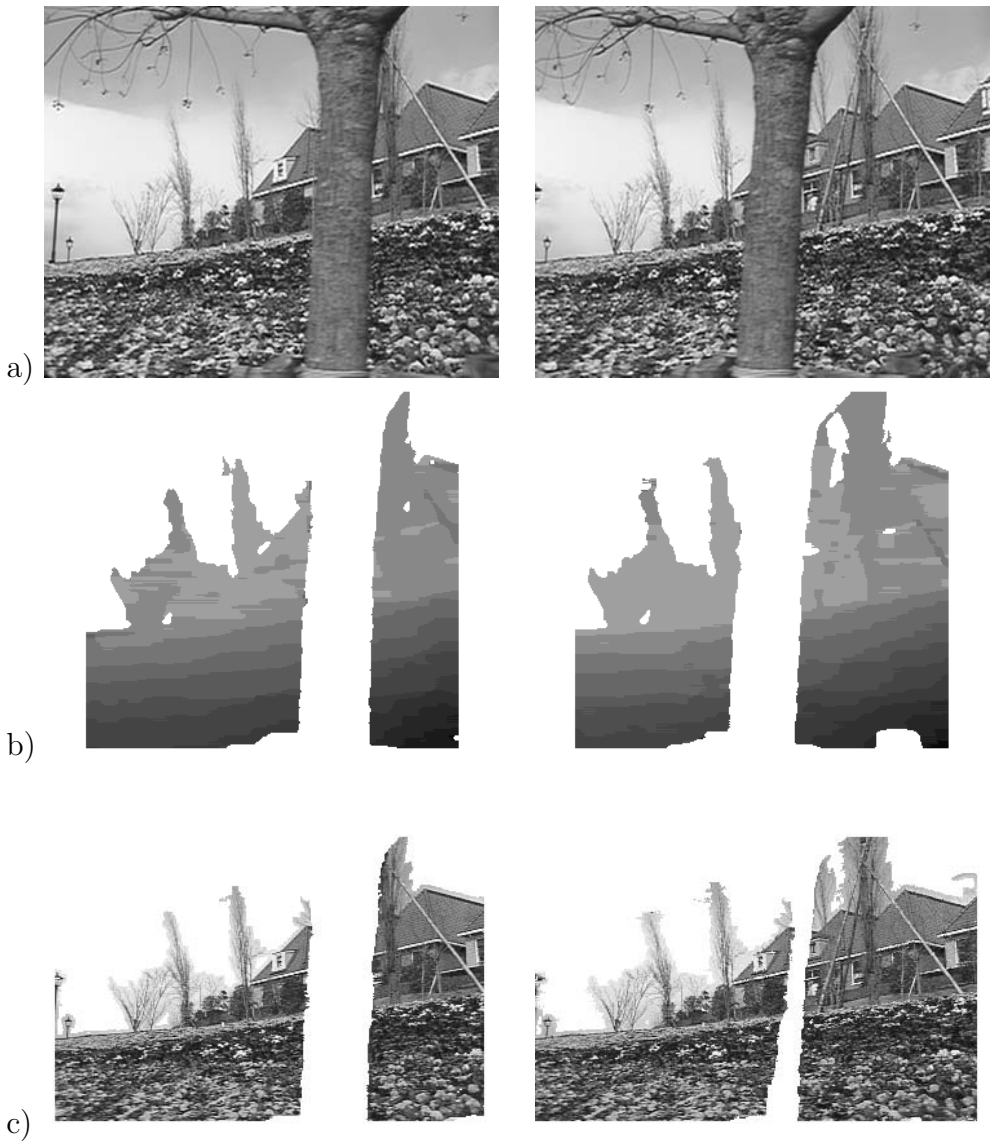

Figure 7: a) Frames 0 and 8 of the Flowergarden sequence, b) the recovered depth maps using the motion estimate obtained from the model-based algorithm in section 5, c) rendered scene model (set of 3-D points) obtained after combination of the motion compensated scene model from the previous frame and the current depth map in b).

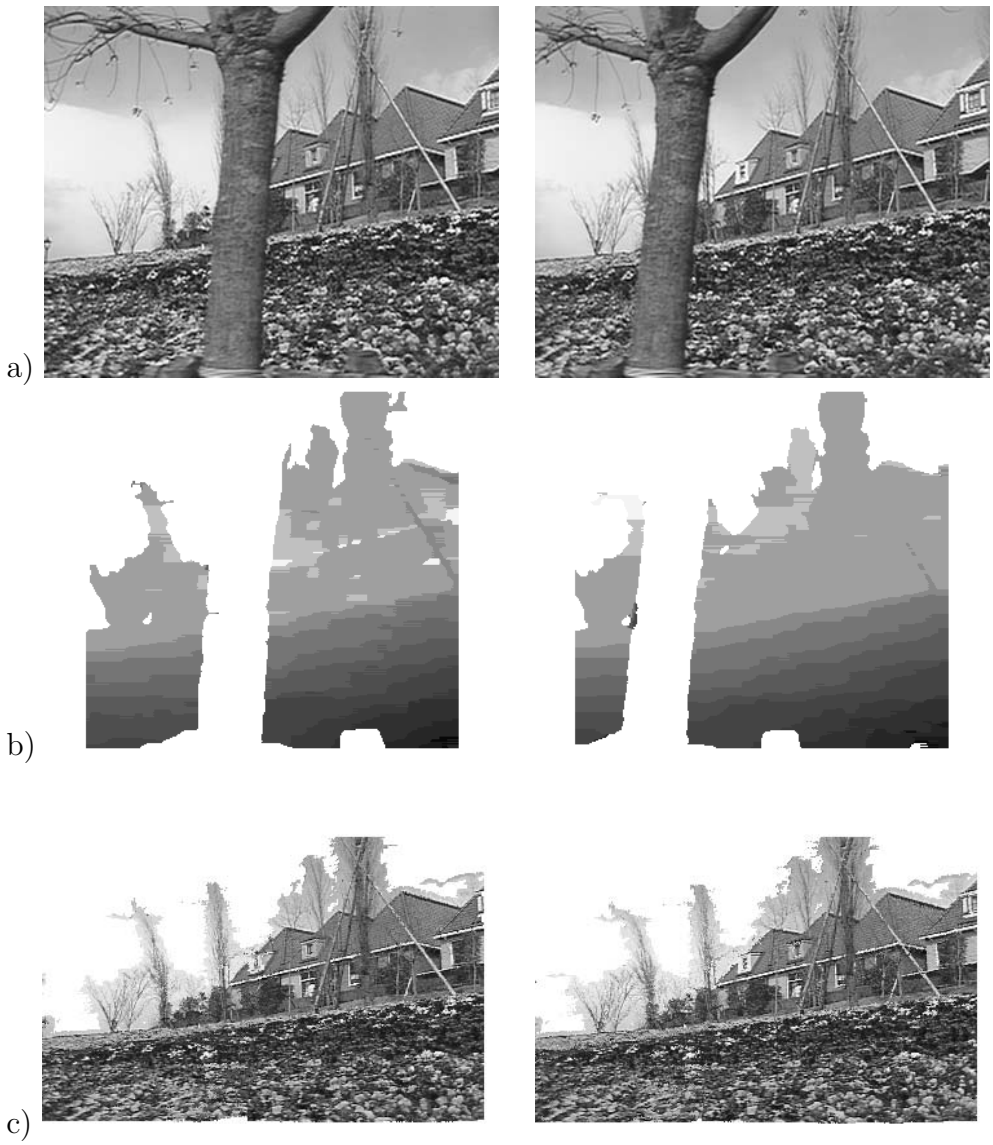

Figure 8: a) Frames 16 and 24 of the Flowergarden sequence, b) the recovered depth maps using the motion estimate obtained from the model-based algorithm in section 5, c) rendered scene model (set of 3-D points) obtained after combination of the motion compensated scene model from the previous frame and the current depth map in b).

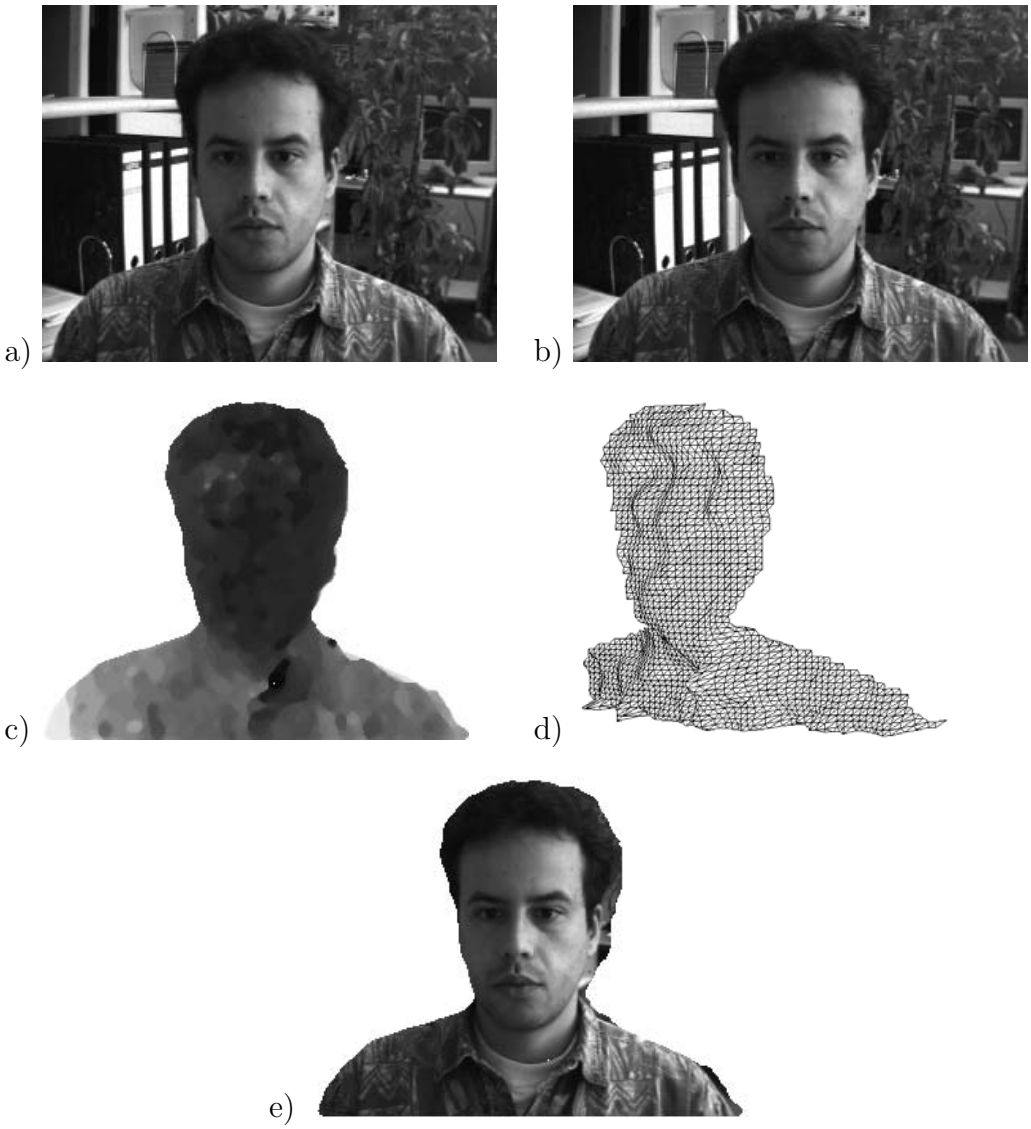

Figure 9: a) Frame 0 and b) Frame 1 of a laboratory scene consisting of 2 objects, c) recovered depth map for the dominant object (person in the foreground), d) wire-frame representation of the person computed from the depth map in c) rendered at a different viewing position, e) VOP generated from the 3-D model of the person.

7 Conclusions

In this paper we combine a robust structure-from-motion algorithm with a linear and low complexity model-based 3-D motion estimation algorithm for the purpose of automatic scene segmentation into different layers of depth. The structure-from-motion algorithm is used to initially estimate motion and structure from two frames of the scene. The depth estimate is then converted into a 3-D wire-frame representation and is used in the following as a 3-D model for model-based motion estimation. The new information (depth and texture) from the following frames is incorporated into one single 3-D model which consists of an unstructured set of 3-D points. With an increasing number of frames, the model becomes more complete since occluded background is uncovered with time. In our experimental results we recovered a connected and complete representation of the *Field and House* layer of the Flowergarden sequence using only motion and structure information. For a multiple object scene we recover the motion and structure of all objects in the scene. The automatic segmentation allows to define video object planes suitable for the content-based manipulation defined in MPEG-4. The motion and position of the VOP is tracked in the following frames using the same algorithms employed for model construction.

References

[1] "Special Issue on MPEG-4," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 1, February 1997.

[2] "MPEG-4 Requirements," *Document ISO-IEC JTC/SC29/WG11 N1495, ISO/MPEG*, Maceio, November 1996.

[3] L. Torres and M. Kunt, "Video coding: the second generation approach," *Kluwer Academic Publishers*, Englewood Cliffs, 1996.

[4] SNHC, "SNHC Systems Verification Model 4.0," *ISO/IEC JTC1/SC29/WG11 N1666*, Bristol, April 1997.

[5] J.Y.A. Wang and E.H. Adelson, "Representing moving images with layers," *IEEE Trans. on Image Processing*, 3(5), pp. 625-638, September 1994.

[6] L. Torres, D. Garcia, and A. Mates, "On the Use of Layers for Video Coding and Object Manipulation," *Proc. 2nd Erlangen Symposium on Advances in Digital Image Communication*, pp. 65-73, April 25th, Erlangen, 1997.

[7] W.J. Welsh, S. Searsby, and J.B. Waite, "Model-based image coding," *British Telecom Technology Journal*, 8(3), pp. 94-106, July 1990.

[8] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A Review," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 589-609, September 1994.

[9] B. Girod, "Image sequence coding using 3D scene models," *SPIE Symposium on Visual Communications and Image Processing*, September 1994.

[10] K. Aizawa and T. S. Huang, "Model-Based Image Coding: Advanced Video Coding Techniques for Very Low Bit-Rate Applications," *Proc. IEEE*, 83(2), pp. 259-271, February 1995.

[11] D. E. Pearson, "Developments in model-based video coding," *Proc. IEEE*, 83(6), pp. 892-906, June 1995.

[12] R. Koch, "Dynamic 3-D Scene Analysis through Synthesis Feedback Control," *IEEE Trans. PAMI*, vol. 15, no. 6, June 1993.

[13] E. Steinbach, S. Chaudhuri, and B. Girod, "Robust Estimation of Multi-Component Motion in Image Sequences Using the Epipolar Constraint," *Proceedings ICASSP '97*, Munich, April 1997.

[14] 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, no. 1, pp. 13-27, 1984.

[15] J.K. Aggarwal and N. Nandhakumar, "On the Computation of Motion from Sequences of Images - A Review," *Proc. IEEE*, vol. 7b, no. 8, pp. 917-935, August 1988.

[16] J. Weng, N. Ahuja, T.S. Huang, "Optimal Motion and Structure Estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 9, pp. 864-884, September 1993.

[17] B. Girod and P. Wagner, "Displacement Estimation with a Rigid Body Motion Constraint," *Proc. International Picture Coding Symposium*, Cambridge, Mass., USA, March 1990.

[18] E. Steinbach and B. Girod, "Estimation of Rigid Body Motion and Scene Structure from Image Sequences Using a Novel Epipolar Transform," *Proc. ICASSP '96*, pp. 1911-1914, Atlanta, 1996.

[19] A. Rognone, M. Campani and A. Verri, "Identifying Multiple Motions from Optical Flow," *Proc. 2nd ECCV*, pp 258-266, Santa Margherita Ligure, Italy, May 1992.

[20] S. Ayer, "Sequential and Competitive Methods for Estimation of Multiple Motions," *Doctoral Dissertation*, École Polytechnique Fédérale de Lausanne, 1995.

[21] M.J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75-104, January 1996.

[22] W. Wang and J.H. Duncan, "Recovering the Three-Dimensional Motion and Structure of Multiple Moving Objects from Binocular Image Flows," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 430-446, May 1996.

[23] P.J. Rousseeuw and A.M. Leroy, "Robust Regression and Outlier Detection," John Wiley, New York, 1987.

[24] S. Chaudhuri, S. Sharma, and S. Chatterjee, "Recursive Estimation of Motion Parameters," *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 434-442, November 1996.

[25] P. Eisert and B. Girod, "Model-based 3-D Motion Estimation with Illumination Compensation," *Proceedings IPA '97*, pp. 194-198, Dublin, 1997.

[26] A. N. Netravali and J. Salz, "Algorithms for Estimation of Three-Dimensional Motion," *AT&T Technical Journal*, vol. 64 no. 2, pp. 335-346, 1985

[27] 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.

[28] L. Falkenhagen, "Depth Estimation from Stereoscopic Image Pairs Assuming Piecewise Continuous Surfaces," *Paker, Y. and Wilbur, S. (Ed.), Image Processing for Broadcast and Video Production*, Hamburg 1994, pp. 115-127, Springer series on Workshops in Computing, Springer Great Britain, 1994.

[29] Olivier Faugeras, "Three-Dimensional Computer Vision, a Geometrical Viewpoint," The MIT Press, Cambridge, Massachusetts, Second printing 1996.

[30] "Numerical Recipies in C, The Art of Scientific Computing," Second Edition, Cambridge University Press, 1992.