3-D Model-Based Segmentation of Videoconference Image Sequences

Ioannis Kompatsiaris, *Student Member, IEEE,* Dimitrios Tzovaras, and Michael G. Strintzis, *Senior Member, IEEE*

***Abstract*— This paper describes a three-dimensional (3-D) model-based unsupervised procedure for the segmentation of multiview image sequences using multiple sources of information. The 3-D model is initialized by accurate adaptation of a two-dimensional wireframe model to the foreground object of one of the views. The articulation procedure is based on the homogeneity of parameters, such as rigid 3-D motion, color, and depth, estimated for each subobject, which consists of a number of interconnected triangles of the 3-D model. The rigid 3-D motion of each subobject for subsequent frames is estimated using a Kalman filtering algorithm, taking into account the temporal correlation between consecutive frames. Information from all cameras is combined during the formation of the equations for the rigid 3-D motion parameters. The threshold used in the object segmentation procedure is updated at each iteration using the histogram of the subobject parameters. The parameter estimation for each subobject and the 3-D model segmentation procedures are interleaved and repeated iteratively until a satisfactory object segmentation emerges. The performance of the resulting segmentation method is evaluated experimentally.**

***Index Terms*—Multiview image sequences segmentation, rigid 3-D motion estimation, 3-D model-based analysis.**

I. Introduction

DIGITAL video is an integral part of many newly emerging multimedia applications. New video coding standards, such as MPEG-4, do not concentrate only on efficient compression methods, but also on providing better ways to represent, integrate, and exchange visual information [1]. These efforts aim to provide the user with greater flexibility for "content-based" access and manipulation of multimedia data. Model-based methods have been used to enable these functionalities.

The ability of model-based techniques to describe a scene in a structural way has opened up new areas of applications. Very low-bit-rate coding, video production, realistic computer graphics, multimedia interfaces and databases, and medical visualization are some of the applications that may benefit by exploiting the potential of model-based schemes [2]–[5]. In order to obtain a model-based representation, an input video sequence must first be segmented into an appropriate set of arbitrarily shaped regions (termed the video object planes in the MPEG-4 verification model), where each of the regions may represent a particular content of the video stream [6]. The features of each "object" such as shape, motion, and texture information can subsequently be coded into the so-called video object layer for transmission or storage. Although the standards will provide the needed functionalities in order to compose, manipulate, and transmit the "object-based" information, the production of these objects is out of the scope of the standards and is left to the content developer. Thus, the success of any object-based approach depends largely on the segmentation of the scene based on its image contents. In a videophone-type application, for example, an accurate segmentation of the facial region can serve two purposes: 1) it can allow the encoder to place more emphasis on the facial region since this area (i.e., the eyes and mouth in particular) is the focus of attention of the human visual system, and 2) it can also be used to extract features so that higher level descriptions can be generated (i.e., personal characteristics, facial expressions, and composition information). In a similar fashion, the contents of a video database can be segmented into individual objects, where the following features can be supported: 1) sophisticated query and retrieval operations, 2) advanced editing and composition, and 3) better compression ratios. These issues and objectives are currently addressed within the framework of the upcoming MPEG-4 and future MPEG-7 standards [1].

Segmentation methods for two-dimensional (2-D) images may be divided primarily into region-based and boundary-based methods [7], [8]. Region-based approaches [9] rely on the homogeneity of spatially localized features such as gray-level intensity, texture, motion, and other pixel statistics. Region-growing [10] and split-and-merge techniques [11] also belong in the same category. On the other hand, boundary-based methods primarily use gradient information to locate object boundaries. Deformable whole boundary methods [12], [13] rely on the gradient features of parts of an image near an object boundary. Other techniques include the segmentation by anisotropic diffusion introduced in [14] and mathematical morphology [15], [16] methods. Among many morphological transformations, the watershed transformation [17] has received considerable attention for image segmentation. Methods for the segmentation of image sequences have been presented in, among others, [18]–[22]. In most of these methods, region-growing and merging techniques are used, depending on the homogeneity of the 2-D or three-dimensional (3-D) motion.

Manuscript received October 31, 1997. This work was supported by the European CEC Project ACTS PANORAMA (Package for New Autostereoscopic Multiview Systems and Applications, ACTS Project 092) and the Greek Secretariat for Science and Technology Program YPER. This paper was recommended by Associate Editor T. Sikora.

The authors are with the Information Processing Laboratory, Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki 54006, Greece.

Publisher Item Identifier S 1051-8215(98)06320-4.

The present paper addresses a related, but considerably more difficult problem, which is the coherent simultaneous segmentation of two or more views of a scene. To this end, an unsupervised procedure for the segmentation of multiview image sequences using various sources of information is proposed. The segmentation procedure is performed at the triangle level of a 3-D model, as in [23] and [24], using characteristic properties of the triangle, such as rigid 3-D motion, color, and depth. Subobjects consist of collections of one or more interconnected triangles of the 3-D model, with similar parameters. The 3-D model is initialized by adapting a 2-D wireframe to the foreground object. The proposed adaptation of the 2-D wireframe is a novel procedure resulting in a very accurate representation of the foreground object. Using depth and multiview camera geometry, the 2-D wireframe is reprojected in the 3-D space, forming a consistent wireframe for all views. Information from all cameras is combined to form the equations for the extraction of the rigid 3-D motion parameters. Combined motion, color, and depth information, extracted from multiview information, is used, along with the 3-D motion fields. The rigid 3-D motion of each subobject for subsequent frames is estimated using a Kalman filtering algorithm, taking into account the temporal correlation between consecutive frames. Furthermore, in the present work, the threshold characterizing the homogeneity of subobjects is adaptively updated until a satisfactory segmentation emerges. Following region separation, and depending on the application, knowledge-based methods may be used to decide whether a separated region corresponds to a specific semantic object (e.g., the head in a videoconference image sequence). The parameter estimation for each subobject and the 3-D model segmentation procedures are interleaved and repeated iteratively until a satisfactory object segmentation emerges.

The methodology used overcomes a major obstacle in multiview video analysis, caused by the difficult problem of determining and handling coherently corresponding objects in the different views. This is achieved in this paper by defining segmentation and object articulation in the 3-D space, thus ensuring that all ensuing operations (for example, rigid 3-D motion estimation of each subobject) remain coherent for all views of the scene.

The proposed algorithm exploits fully both spatial and temporal correlation of the images since neighborhood constraints are taken into account (for rigid 3-D motion estimation of each triangle of the first of a group of frames) and Kalman filtering is used for tracking the motion in subsequent frames. This combination is shown to be both computationally efficient and highly effective.

The *a priori* information needed consists only of the depth map; all other information is extracted using only the projected images of all views. In fact, given the depth information, the hierarchical algorithm in [24] may be used to create the foreground/background segmentation mask. For depth map estimation, the algorithms presented in [18] or in [25] and [26] can be used. The purpose of the present paper is the definition of a method for the segmentation of the resulting complicated foreground object.

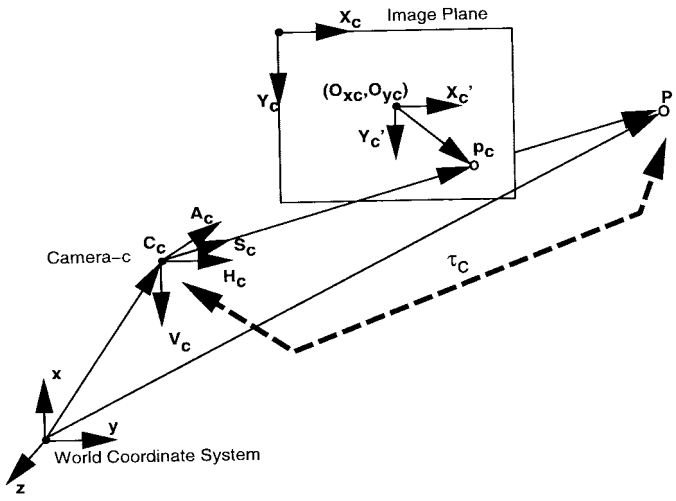


Fig. 1. CAHV camera model.

In order to evaluate the above methodology, a series of enhanced-telepresence videoconference sequences was used. In these, telepresence was sought by using multicamera setups (and auto-stereoscopic displays), and by permitting the transmission of complex scenes with gestures and motion and objects other than just heads and shoulders. The most practical of these multicamera setups uses three cameras, placed on either side and the top of the monitor [27]. All results of the present paper are derived assuming this camera geometry, and are evaluated with real multiview sequences produced by precisely such an arrangement of cameras. However, the basic ideas and results of the paper may be easily extended to monoscopic, stereoscopic, and arbitrary multiview systems using arbitrary arrangements and numbers of cameras.

The paper is organized as follows. In the following section, the three-camera geometry is described, and in Section III, the procedure for the 3-D model initialization is given. The technique used for object articulation is examined in Section IV. In Section V, the estimation of the segmentation parameters is described. The object segmentation procedure and the threshold update algorithm are presented in Section VI. In Section VII, application in segmenting monocular and stereoscopic image sequences is described. In Section VIII, experimental results are given evaluating the performance of the proposed methods. Conclusions are finally drawn in Section IX.

II. Camera Model

A camera model describes the projection of 3-D points onto a camera target. The model used here is the CAHV model introduced in [28]. This model describes extrinsic camera parameters such as position and orientation and intrinsic camera parameters such as focal length and intersection between optical axis and image plane.

As mentioned in Section I, in our multiview camera geometry, three cameras c are used: $c = left, top, right$**.** For each camera c, the model contains the following parameters shown in Fig. 1: 1) position of the camera $\boldsymbol{C}_c$, 2) optical axis $\boldsymbol{A}_c$, i.e., the viewing direction of the camera (unit vector), 3) horizontal camera target vector $\boldsymbol{H}_c$ (x axis of the camera target), and 4)

vertical camera target vector $\boldsymbol{V}_c$ (y axis of the camera target), radial distortion, and sx, sy pixel size.

In our camera model, we will assume that the radial distortion is compensated. The camera's setup is previously calibrated and maintained static. In this case, the projection of a 3-D point $\boldsymbol{P}$, with coordinates relative to world coordinate system, onto the image plane (X_c', Y_c') is [28]

$$X_c' = \frac{(\boldsymbol{P}-\boldsymbol{C}_c)^T \cdot \boldsymbol{H}_c}{(\boldsymbol{P}-\boldsymbol{C}_c)^T \cdot \boldsymbol{A}_c}, \quad Y_c' = \frac{(\boldsymbol{P}-\boldsymbol{C}_c)^T \cdot \boldsymbol{V}_c}{(\boldsymbol{P}-\boldsymbol{C}_c)^T \cdot \boldsymbol{A}_c}. \quad (1)$$

The coordinates (X_c', Y_c') are camera centered (image plane coordinate system) with the unit pel. The origin of the coordinate system is the center point of the camera. The coordinates of a point relative to the picture coordinate system (X_c, Y_c) is given by $(X_c, Y_c) = (X_c' + O_{x,c}, Y_c' + O_{y,c})$, where $(O_{x,c}, O_{y,c})$ is the center of the image plane in the picture coordinate system.

Conversely, given its position (X_c, Y_c) on the camera plane, the 3-D position of a point can be determined by

$$\boldsymbol{P} = \boldsymbol{C}_c + \tau_c \cdot \boldsymbol{S}_c(X_c, Y_c) \quad (2)$$

where $\boldsymbol{S}_c(X_c, Y_c)$ is the unit vector pointing from the camera to the point in the direction of the optical axis and τ_c is the distance between the 3-D point and the center of camera c.

III. 3-D Model Initialization

The generation of the 3-D model object is based on an initial adaptation of a 2-D wireframe model to the foreground object of one of the projected images (left, top, or right). Since the 2-D wireframe is adapted to the image, the 3-D model can be formed by using depth information (i.e., the distance of the 3-D point from the specific camera) in (2). The consistent camera geometry allows the 3-D model to be adapted to the other views as well.

In the present paper, very few assumptions are made regarding the content of the scene and the motion of its objects. The *a priori* information needed consists only of the depth map; all other information is extracted using only the projected images of all views. In fact, given the depth information, the hierarchical algorithm in [24] may be used to create the foreground/background segmentation mask. For depth map estimation, the algorithms presented in [18] or in [25] and [26] can be used. Without loss of generality, to simplify the subsequent discussion, we will assume that only one object exists in the foreground. However, it is emphasized that the methodology of the paper remains valid and applicable in the presence of more than one such objects. For example, if two distinct foreground objects exist, the same techniques may be used to segment separately each foreground object.

The adaptation of the 2-D wireframe to the foreground object is a coarse-to-fine procedure consisting of the following steps.

- A regular grid is first created, covering the full size of the 2-D image.
- Only triangles overlapping with the foreground object or with the boundary of the foreground object are retained. The remaining triangles are discarded.
- Nodes of the triangles lying on either side of the boundary of the foreground object are forced to meet this boundary. If a node inside the boundary moves to exactly the same position as a point outside the boundary, then only the node outside the boundary is moved.
- The "force" applied for the movement of the nodes is propagated to the remaining nodes so as not to drastically alter the regularity of the wireframe.

These steps are described in more detail in the sequel. The initial regular grid is of the form

$$p = (iDX, jDY)$$

where DX, DY are the horizontal and vertical distances of the nodes of the wireframe, $i = 0, \cdots, (SX/DX)$ and $j = 0, \cdots, (SY/DY)$. SX, SY are the horizontal and vertical sizes of the image, respectively. The value of DX, DY defines the detail of the segmentation mask. In cases where a coarse segmentation mask is needed, a 3-D model consisting of large triangles may be used, making the segmentation procedure much faster, whereas for highly detailed masks, a finer mesh may be used. Such a triangulation covering the full size of the middle view can be seen in Fig. 4(a).

In the next step, a background/foreground segmentation mask is used to separate and discard the triangles that do not overlap with the foreground object. Such a mask is particularly easily calculated if the foreground consists of a moving object in front of a static background without texture (such as in videotelephony applications). A simple edge-detection algorithm suffices then to provide the information on the foreground object silhouette.

There exist many simple methods for the adaptation of 2-D wire grids to segmented objects [29]. In order to avoid creating very small triangles on the object boundary, and also to avoid disrupting the regularity of the wireframe, the following method was used. A triangle is retained as part of the wireframe when at least one of its vertices lies on the foreground object. All nodes of triangles overlapping with the boundary of the object are attracted to the boundary by "forces" proportional to their distance from it. Specifically, each such node is assumed to have eight degrees of freedom: $\{f_1, \cdots, f_8\}$ as shown in Fig. 2. Following each of the eight directions, a point on the object boundary p' is found and a force $\vec{F_i}$ is defined, having the same direction as f_i and magnitude equal to the Euclidean distance between node p and point p_i':

$$F_i = \|p - p_i'\|. \quad (3)$$

If no point p_i' on the boundary of the foreground object can be found in the direction f_i, then F_i is set to infinity. The force applied to node p in order to meet the object boundary is chosen to have magnitude F_k, where

$$k = \arg \min_{i \in \{1, \cdots, 8\}} F_i$$

and the direction is the one of the corresponding direction f_k.

In the special case where a triangle has two of its nodes lying on the object boundary, forcing the third node on the

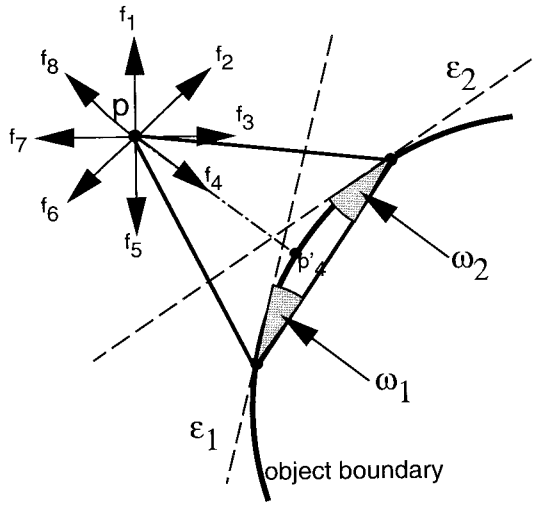


Fig. 2. Special case where a triangle may be discarded from the wireframe.

boundary would create a very small triangle (Fig. 2) which must be discarded. Specifically, the triangle is discarded when

$$\omega_1 + \omega_2 < t$$

where ω_1, ω_2 are the angles between the triangle vertex within the object and the tangents $\varepsilon_1, \varepsilon_2$ to the object boundary at the two nodes on the boundary, and t is a threshold. In case of concavities, some triangles overlapping with the background may be created as a result of the above procedure. These are discarded in a final postprocessing procedure.

The above procedure results in an accurate adaptation of the wireframe to the foreground object boundary. To avoid the creation of very small triangles and disruption of the regularity of the wireframe, the "force" F applied to each node is used for the adaptation of the remaining nodes of the wireframe. Specifically, the force $\vec{F}$ with direction f and magnitude F defined by (3) is propagated according to a Gaussian model, taking into account the distance between the node p and all other mesh nodes, following the approach in [30]. If $\vec{F}$ is the force applied to node p, then the force $\vec{F}_q$ applied to a node q with distance d from node p has the same direction as $\vec{F}$ and magnitude:

$$F_q = \frac{1}{\sqrt{2\pi}\sigma}\, e^{(d-\mu)^2/2\sigma^2}.$$

This means that node q is moved to a new position q' in the direction f_q of $\vec{F}_q$ and with distance from its previous position $||q - q'|| = F_q$. The mean value μ and standard deviation σ are selected so as to harmonize the resulting wireframe mesh.

The result of the above procedure is a very accurately adapted wireframe over the foreground object in one of the views [Fig. 4(c)]. By applying (2) to all nodes $p = (X, Y)$, the 3-D points $\boldsymbol{P} = (x, y, z)$ are found and the 3-D model is formed [Fig. 4(d)]. Due to the consistency of the camera geometry, the projection of the 3-D model to the image planes formed by other cameras also leads to 2-D wireframes adapted to the foreground object of the other views.

IV. Object Articulation

A novel subdivision method based on characteristic descriptors of each triangle of the 3-D model will be proposed for the articulation of the foreground object. The model initialization procedure described above results in a set of interconnecting triangles in the 3-D space: $\{T_k, k = 1, \cdots, K\}$, where K is the number of triangles of the 3-D model. In the following, $\mathcal{S}^{(i)}$ will denote an articulation of the 3-D model at iteration i of the articulation algorithm, consisting of $\{s_l^{(i)}, l = 1, \cdots, L^{(i)}\}$ subobjects. Each subobject consists of $\{T_m^{(s_l^{(i)})}, m = 1, \cdots, M^{(s_l^{(i)})}\}$ triangles.

The object articulation procedure exploits the homogeneity of a set of characteristic descriptors based on known and estimated parameters, such as rigid 3-D motion, color, and depth, for each subobject. The total descriptor value for each subobject will, in general, be defined as

$$p^{(s_l^{(i)})} = \sum_{j=0}^{P-1} a_j \overline{x}_j$$

where $\overline{x}$ is the value of a specific descriptor, normalized so as to lie between 0 and 1, P is the number of characteristic descriptors used, and

$$\sum_{j=0}^{P} a_j = 1.$$

Two subobjects $s_l^{(i)}, s_m^{(i)}$ will be merged if

$$||p^{(s_l^{(i)})} - p^{(s_m^{(i)})}|| \le th^{(i)}.$$

The selection of the appropriate threshold $th^{(i)}$ characterizing the similarity between different subobjects is critical for the definition of the object articulation procedure. At each iteration i, this threshold $th^{(i)}$ is updated according to the histogram of the parameters $p^{(s_l^{(i)})}$, as will be explained in Section VI-A.

The proposed iterative object articulation procedure is composed of the following steps.

Step 1: Set $i = 0$. Let an initial segmentation $\mathcal{S}^{(0)} = \{s_l^{(0)}, l = 1, \cdots, L^{(0)}\}$, with $L^{(0)} = K, T_1^{s_l^{(0)}} = T_l$ and $M^{(s_l^{(0)})} = 1$. In this initialization step, each subobject consists of one triangle of the 3-D model. Set an initial threshold value: $th^{(0)} \ll 1$.

Step 2: Apply the segmentation parameters estimation algorithm to each subobject $s_l^{(i)}$ to find $p^{(s_l^{(i)})}$ (Section V).

Step 3: Execute the object segmentation procedure that subdivides the initial object into $L^{(i)}$ subobjects, i.e., $\mathcal{S}^{(i)} = \{s_l^{(i)}, l = 1, \cdots, L^{(i)}\}$ (Section VI).

Step 4: Use the histogram of $p^{(s_l^{(i)})}$ to define the new threshold $th^{(i+1)}$ (Section VI-A).

Step 5: If $||th^{(i)} - th^{(i+1)}| \le \epsilon$, where ϵ is a threshold affecting the number of subobjects created, then stop. Else set $i = i + 1$ and go to Step 2.

The procedures for segmentation parameter estimation, object segmentation, and the threshold value update algorithm are described in the following sections.

V. Parameter Estimation for Each Subobject

The subdivision criterion is based on the homogeneity of characteristic descriptors of each triangle of the 3-D model. Although many such descriptors can be used for efficient segmentation, in this paper, we focus on those based on the rigid 3-D motion parameters, color and depth information.

A. Rigid 3-D Motion Equation

For each subobject, the rigid 3-D motion parameters are estimated for a number of frames. In order to exploit temporal correlation between consecutive frames, a Kalman filtering approach is used. The system of equations describing the rigid 3-D motion parameters is formed, and is used to determine the Kalman filter for the estimation and tracking of the rigid 3-D motion.

In the following, for the sake of notational simplicity, the subobject $s_l^{(i)}$ will be simply denoted by s. This means that the procedure to be described is applied to all subobjects at any iteration step of the object articulation algorithm. The rigid motion of each subobject s is modeled using three rotation and three translation parameters [31]:

$$\boldsymbol{P}_{t+1} = \boldsymbol{R}^{(s)}\boldsymbol{P}_t + \boldsymbol{T}^{(s)} \tag{4}$$

with $\boldsymbol{R}^{(s)}$ and $\boldsymbol{T}^{(s)}$ being of the form

$$\boldsymbol{R}^{(s)} = \begin{bmatrix} 1 & -w_z^{(s)} & w_y^{(s)} \\ w_z^{(s)} & 1 & -w_x^{(s)} \\ -w_y^{(s)} & w_x^{(s)} & 1 \end{bmatrix} \tag{5}$$

$$\boldsymbol{T}^{(s)} = \begin{bmatrix} \tau_x^{(s)} \\ \tau_y^{(s)} \\ \tau_z^{(s)} \end{bmatrix} \tag{6}$$

where $\boldsymbol{P}_t = (x_t, y_t, z_t)$ is a 3-D point on the 3-D planes defined by the triangles $T_m^{(s)}$ of subobject s. The rigid 3-D motion parameters vector for subobject s is $\boldsymbol{a}^{(s)} = (w_x^{(s)}, w_y^{(s)}, w_z^{(s)}, \tau_x^{(s)}, \tau_y^{(s)}, \tau_z^{(s)})$.

At time t, each point $\boldsymbol{P}_t$ on s is projected to points $(X_{c,t}, Y_{c,t}), c = l, t, r$ on the planes of the three cameras. Using (1) and (4), the projected 2-D motion vector $\boldsymbol{d}_c(X_c, Y_c)$ is determined by

$$\begin{aligned} d_{xc}(X_{c,t}, Y_{c,t}) &= X_{c,t+1} - X_{c,t} \\ &= \frac{(\boldsymbol{R}^{(s)}\boldsymbol{P}_t + \boldsymbol{T}^{(s)} - \boldsymbol{C}_c)^T \cdot \boldsymbol{H}_c}{(\boldsymbol{R}^{(s)}\boldsymbol{P}_t + \boldsymbol{T}^{(s)} - \boldsymbol{C}_c)^T \cdot \boldsymbol{A}_c} \\ &\quad - \frac{(\boldsymbol{P}_t - \boldsymbol{C}_c)^T \cdot \boldsymbol{H}_c}{(\boldsymbol{P}_t - \boldsymbol{C}_c)^T \cdot \boldsymbol{A}_c} \end{aligned} \tag{7}$$

$$\begin{aligned} d_{yc}(X_{c,t}, Y_{c,t}) &= Y_{c,t+1} - Y_{c,t} \\ &= \frac{(\boldsymbol{R}^{(s)}\boldsymbol{P}_t + \boldsymbol{T}^{(s)} - \boldsymbol{C}_c)^T \cdot \boldsymbol{V}_c}{(\boldsymbol{R}^{(s)}\boldsymbol{P}_t + \boldsymbol{T}^{(s)} - \boldsymbol{C}_c)^T \cdot \boldsymbol{A}_c} \\ &\quad - \frac{(\boldsymbol{P}_t - \boldsymbol{C}_c)^T \cdot \boldsymbol{V}_c}{(\boldsymbol{P}_t - \boldsymbol{C}_c)^T \cdot \boldsymbol{A}_c} \end{aligned} \tag{8}$$

where $\boldsymbol{d}_c(X_c, Y_c) = (d_{xc}(X_{c,t}, Y_{c,t}), d_{yc}(X_{c,t}, Y_{c,t}))$.

Using the initial 2-D motion vectors, estimated by applying a block-matching algorithm to the images corresponding to the left, top, and right cameras, and also using (7) and (8), a linear system of equations for the rigid motion parameter vector $\boldsymbol{a}^{(s)}$ for subobject s between time t and $t+1$ is formed

$$\boldsymbol{b}^{(s)} = \boldsymbol{D}^{(s)}\boldsymbol{a}^{(s)}. \tag{9}$$

Omitting, for notational simplicity, dependence on time t, subobject s, and camera c, $\boldsymbol{D}$ is equal to

$$\boldsymbol{D} = \begin{bmatrix} d_{x,0}[0] & d_{x,0}[1] & d_{x,0}[2] & d_{x,0}[3] & d_{x,0}[4] & d_{x,0}[5] \\ d_{y,0}[0] & d_{y,0}[1] & d_{y,0}[2] & d_{y,0}[3] & d_{y,0}[4] & d_{y,0}[5] \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{x,L}[0] & d_{x,L}[1] & d_{x,L}[2] & d_{x,L}[3] & d_{x,L}[4] & d_{x,L}[5] \\ d_{y,L}[0] & d_{y,L}[1] & d_{y,L}[2] & d_{y,L}[3] & d_{y,L}[4] & d_{y,L}[5] \end{bmatrix} \tag{10}$$

where

$$\begin{aligned} d_{x,l}[0] &= -H_y z_l + H_z y_l + q_{x,l} A_y z_l - q_{x,l} A_z y_l \\ d_{x,l}[1] &= H_x z_l - H_z x_l - q_{x,l} A_x z_l + q_{x,l} A_z x_l, \\ d_{x,l}[2] &= -H_x y_l + H_x x_l + q_{x,l} A_x y_l - q_{x,l} A_y x_l \\ d_{x,l}[3] &= H_x - q_{x,l} A_x, d_{x,l}[4] = H_y - q_{x,l} A_y \\ d_{x,l}[5] &= H_z - q_{x,l} A_z \end{aligned}$$

and

$$q_{x,l} = d_{x,l} + \frac{(\boldsymbol{P}_l - \boldsymbol{C})^T \cdot \boldsymbol{H}}{(\boldsymbol{P}_l - \boldsymbol{C})^T \cdot \boldsymbol{A}}.$$

For the y coordinates

$$\begin{aligned} d_{y,l}[0] &= -V_y z_l + V_z y_l + q_{y,l} A_y z_l - q_{y,l} A_z y_l \\ d_{y,l}[1] &= V_x z_l - V_z x_l - q_{y,l} A_x z_l + q_{y,l} A_z x_l \\ d_{y,l}[2] &= -V_x y_l + V_x x_l + q_{y,l} A_x y_l - q_{y,l} A_y x_l \\ d_{y,l}[3] &= V_x - q_{y,l} A_x, d_{y,l}[4] = V_y - q_{y,l} A_y \\ d_{y,l}[5] &= V_z - q_{y,l} A_z \end{aligned}$$

and

$$q_{y,l} = d_{y,l} + \frac{(\boldsymbol{P}_l - \boldsymbol{C})^T \cdot \boldsymbol{V}}{(\boldsymbol{P}_l - \boldsymbol{C})^T \cdot \boldsymbol{A}}.$$

Also

$$\boldsymbol{b} = \begin{bmatrix} q_{x,0} \cdot (\boldsymbol{P}_0 - \boldsymbol{C})^T \cdot \boldsymbol{A} - (\boldsymbol{P}_0 - \boldsymbol{C})^T \cdot \boldsymbol{H} \\ q_{y,0} \cdot (\boldsymbol{P}_0 - \boldsymbol{C})^T \cdot \boldsymbol{A} - (\boldsymbol{P}_0 - \boldsymbol{C})^T \cdot \boldsymbol{V} \\ \cdots \\ q_{x,L} \cdot (\boldsymbol{P}_L - \boldsymbol{C})^T \cdot \boldsymbol{A} - (\boldsymbol{P}_L - \boldsymbol{C})^T \cdot \boldsymbol{H} \\ q_{y,L} \cdot (\boldsymbol{P}_L - \boldsymbol{C})^T \cdot \boldsymbol{A} - (\boldsymbol{P}_L - \boldsymbol{C})^T \cdot \boldsymbol{V} \end{bmatrix}. \tag{11}$$

In the above $l = 0, \cdots, L$, where L is the number of 3-D points $\boldsymbol{P}_l = [x_l, y_l, z_l]^T$ contained in s (the set of all 3-D points contained in triangles $T_m^{(s)}$ of s), and $\boldsymbol{C}, \boldsymbol{A} = [A_x, A_y, A_z]^T$, $\boldsymbol{H} = [H_x, H_y, H_z]^T$, and $\boldsymbol{V} = [V_x, V_y, V_z]^T$ are the multiview camera parameters. The 2-D motion vectors $[d_{x,l}, d_{y,l}]^T$ correspond to the projected 3-D point $\boldsymbol{P}_l$. The set of 3-D points in each triangle is found by 3-D backprojection of the set of discrete 2-D points contained in the corresponding 2-D triangle.

Equations (9)–(11) define a system of $2 \times 3 \times L$ equations with six unknowns, where L is the number of 3-D points contained in s since, for each 3-D point $\boldsymbol{P}_t$, two equations are formed for the X and Y coordinates for each of the three cameras. This system combines the rigid 3-D motion

parameters, with the camera parameters and the points of the 3-D model, taking as initial values the 2-D motion, using available information from *all* cameras simultaneously.

Whenever the value of L falls below a predefined threshold, neighboring triangles are also used in order to enhance the stability and efficiency of the rigid 3-D motion estimation procedure. In this case, the additional 3-D points contained in triangles neighboring those of s, i.e., triangles sharing at least two common nodes with any triangle of s, are used in (9). This is normally necessary only in the initial step of the articulation procedure where each subobject consists of only one triangle.

B. 3-D Motion Tracking Using Kalman Filtering

In order to exploit the temporal correlation between consecutive frames, a Kalman filter [32], [33] is applied for the calculation of the 3-D rigid motion parameters at every time instant. In this way, the computationally complicated solution of (9) is needed only for the first of a sequence of F frames. In subsequent frames, the estimation of the motion parameters is based on the initial frame estimation improved by additional observations as additional frames arrive. Omitting, for the sake of further notational simplification, the explicit dependence of the motion parameters to the subobject $s_l^{(i)}$, thus writing $\boldsymbol{a}_t, \boldsymbol{b}_t, \boldsymbol{C}_t$ instead of $\boldsymbol{a}_t^{(s_l^{(i)})}, \boldsymbol{b}_t^{(s_l^{(i)})}, \boldsymbol{C}_t^{(s_l^{(i)})}$, the dynamics of the system are described as follows:

$$\boldsymbol{a}_{t+1} = \boldsymbol{a}_t + w \cdot \boldsymbol{e}_{t+1} \tag{12}$$

$$\boldsymbol{b}_{t+1} = \boldsymbol{D}_{t+1}\boldsymbol{a}_{t+1} + \boldsymbol{v}_{t+1} \tag{13}$$

where $\boldsymbol{a}$ is the rigid 3-D motion vector of each subobject and $\boldsymbol{e}_t$ is a unit-variance white random sequence. The term $w \cdot \boldsymbol{e}_{t+1}$ describes the difference of consecutive frames, and a high value of w implies a small correlation between consecutive frames, and can be used to describe fast-changing scenes, whereas a low value of w may be used when the motion is relatively slow and the temporal correlation is high. The term $\boldsymbol{v}_{t+1}$ represents the random error of the formation of the system (9), and is modeled as white zero-mean Gaussian noise, with $E\{v_n \cdot v_{n'}\} = \mathcal{R}_v \delta(n - n')$, where v_n is the nth element of $\boldsymbol{v}$.

The equations giving the estimated value of $\hat{\boldsymbol{a}}_{t+1}$ in terms of $\hat{\boldsymbol{a}}_t$ are [34], [35]

$$\hat{\boldsymbol{a}}_{t+1} = \hat{\boldsymbol{a}}_t + \boldsymbol{K}_{t+1} \cdot (\boldsymbol{b}_{t+1} - \boldsymbol{D}_{t+1} \cdot \hat{\boldsymbol{a}}_t) \tag{14}$$

$$\boldsymbol{K}_{t+1} = \left(\boldsymbol{R}_t + w^2\boldsymbol{I}\right) \cdot \boldsymbol{D}_{t+1}^T \cdot \boldsymbol{k}^{-1} \tag{15}$$

$$\boldsymbol{k} = \boldsymbol{D}_{t+1} \cdot \boldsymbol{R}_t \cdot \boldsymbol{D}_{t+1}^T + \boldsymbol{D}_{t+1} \cdot w^2\boldsymbol{I} \cdot \boldsymbol{D}_{t+1}^T + \mathcal{R}_v \tag{16}$$

$$\boldsymbol{R}_{t+1} = (\boldsymbol{I} - \boldsymbol{K}_{t+1} \cdot \boldsymbol{D}_{t+1}) \cdot \left(\boldsymbol{R}_t + w^2\boldsymbol{I}\right) \tag{17}$$

where $\hat{\boldsymbol{a}}_{t+1}$ and $\hat{\boldsymbol{a}}_t$ are the predictions of the unknown motion parameters corresponding to the $t+1$th and tth frame, respectively, $\boldsymbol{K}_{t+1}$ represents the correction matrix, and $\boldsymbol{R}_t$ and $\boldsymbol{R}_{t+1}$ describe the covariance matrix of the estimation error $\boldsymbol{E}_t$ and $\boldsymbol{E}_{t+1}$, respectively

$$\boldsymbol{E}_t = (\boldsymbol{a}_t - \hat{\boldsymbol{a}}_t), \quad \boldsymbol{R}_t = E\{\boldsymbol{E}_t \cdot \boldsymbol{E}_t^T\}$$

$$\boldsymbol{E}_{t+1} = (\boldsymbol{a}_{t+1} - \hat{\boldsymbol{a}}_{t+1}), \quad \boldsymbol{R}_{t+1} = E\{\boldsymbol{E}_{t+1} \cdot \boldsymbol{E}_{t+1}^T\}.$$

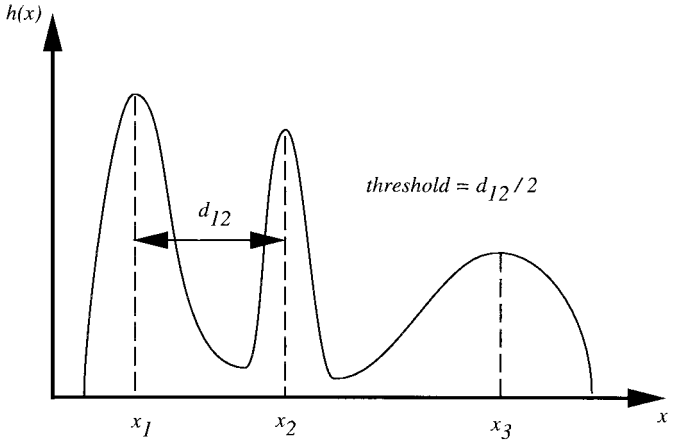


Fig. 3. Update of the threshold based on the histogram of the subobject's parameters.

The initial value $\hat{\boldsymbol{a}}_0$ of the filter (first frame or $t = 0$) is found by directly solving (9). More specifically, since $2 \times 3 \times L \geq 6$ for all subobjects (in the worst case, s is composed of a single triangle with $L = 3$ and $2 \times 3 \times 3 = 18$), (9) is overdetermined and can be solved by the robust least median of squares motion estimation algorithm described in detail in [36]. Erroneous initial 2-D estimates, produced by the block-matching algorithm, will be discarded by the least median of squares motion estimation algorithm.

The initial correlation matrix $\boldsymbol{R}_0$ is

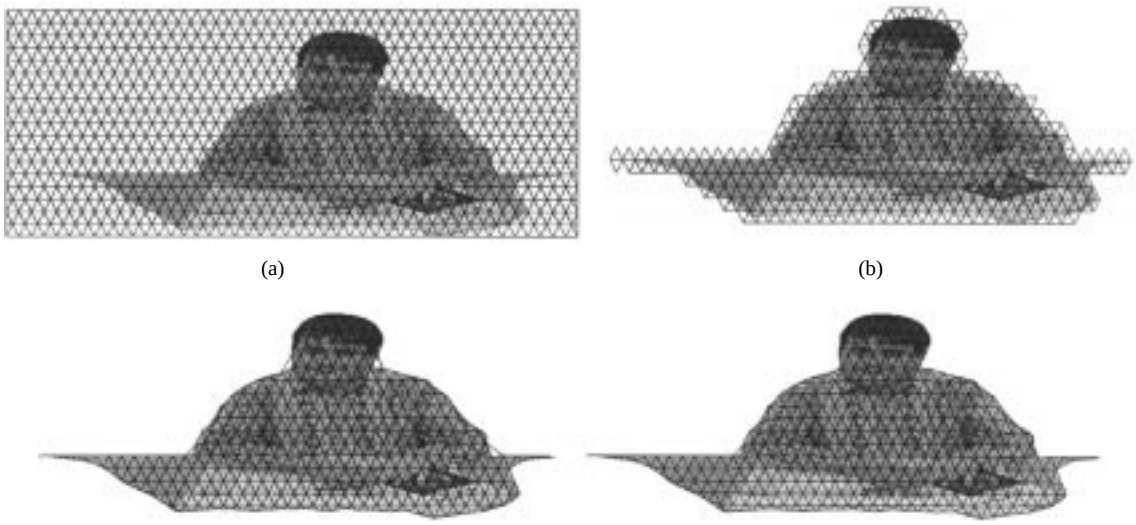
$$\boldsymbol{R}_0 = E\{\boldsymbol{a}_0 \cdot \boldsymbol{a}_0^T\}.$$

In the above, w and $\boldsymbol{v}$ are assumed to be the same for the whole mesh, and hence independent of the subobject s. Notice that (9) is solved only once in order to provide the initial values for the Kalman filtering. During the next frames, $\boldsymbol{D}$ and $\boldsymbol{b}$ are only formed for use in the Kalman filter procedure.

The final rigid 3-D motion descriptor characterizing each subobject s is the sum of the rigid 3-D motion parameters for frames $t = 0, \cdots, F-1$, where F is the total number of frames used. More specifically, we define for a rigid 3-D motion vector $\boldsymbol{a}_t$ at time t the matrix

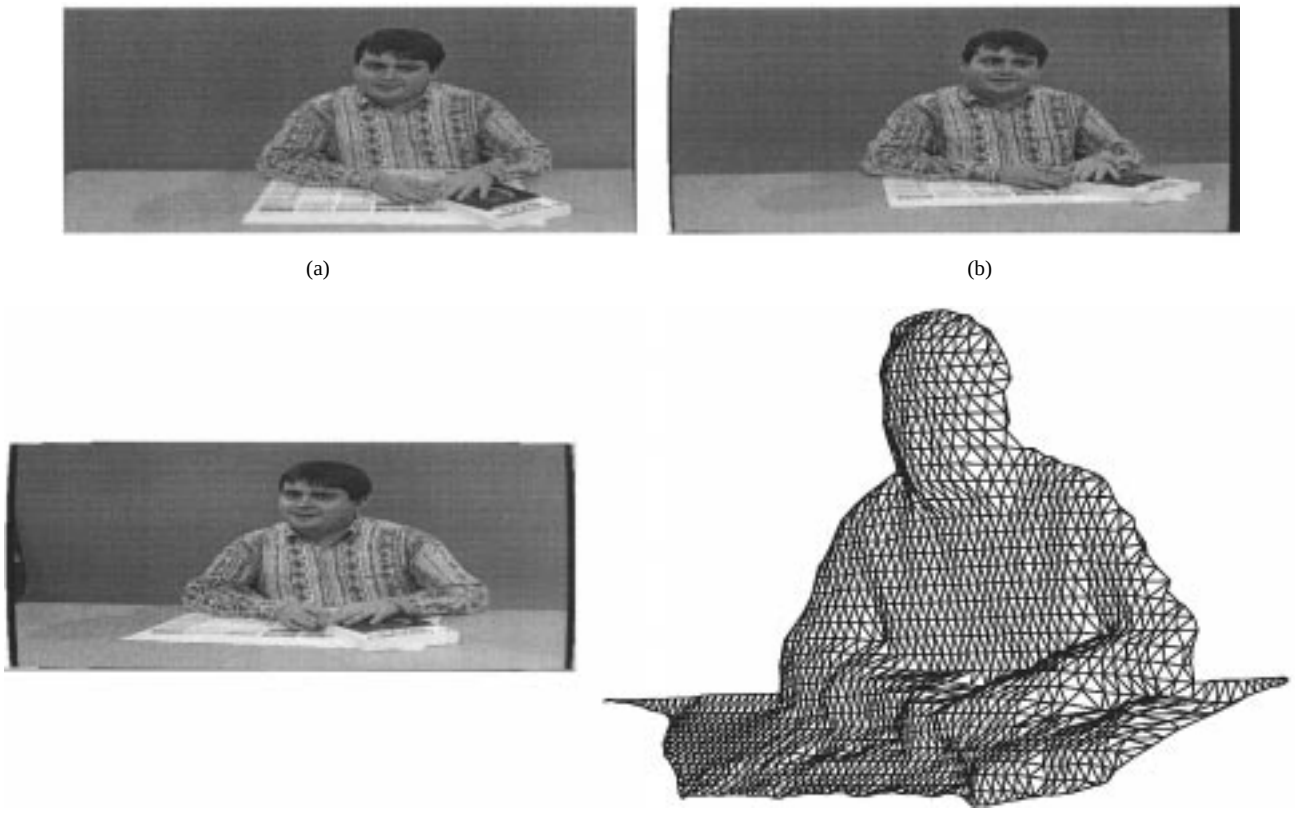
$$\boldsymbol{M}_t = \begin{bmatrix} 1 & -w_{z_t} & w_{y_t} & \tau_{x_t} \\ w_{z_t} & 1 & -w_{x_t} & \tau_{y_t} \\ -w_{y_t} & w_{x_t} & 1 & \tau_{z_t} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{18}$$

The total rigid 3-D motion for a number of frames is



(a) (b) (c) (d)

Fig. 4. (a) Regular wireframe covering the full size of the middle view. (b) Coarse adaptation with triangles covering only the foreground object. (c) Fine adaptation to the foreground object. (d) 3-D model produced by reprojecting the 2-D wireframe to the 3-D space.



(a) (b) (c) (d)

Fig. 5. "Ludo3" sequence. (a) Top view. (b) Left view. (c) Right view. (d) 3-D model produced by the model initialization procedure.

described by

$$\boldsymbol{M} = \prod_{t=0}^{F-1} \boldsymbol{M}_t \tag{19}$$

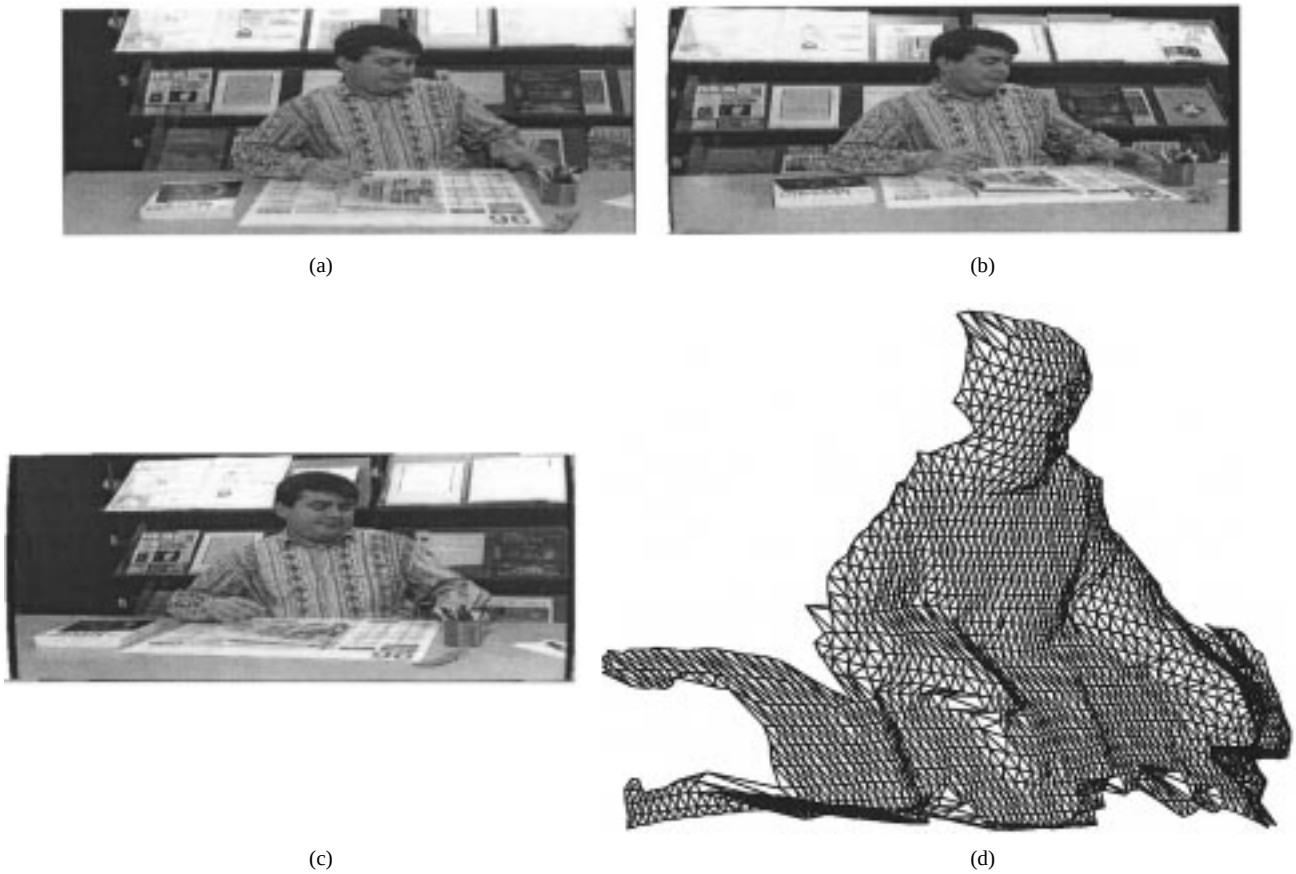
and the rigid 3-D motion parameters for descriptor $\boldsymbol{a}^{(s)}$ are extracted from $\boldsymbol{M}$, which is of the form of $\boldsymbol{M}_t$. i.e., the motion descriptor

$$\boldsymbol{a}^{(s)} = \left(w_x^{(s)}, w_y^{(s)}, w_z^{(s)}, \tau_x^{(s)}, \tau_y^{(s)}, \tau_z^{(s)}\right)$$

is found from

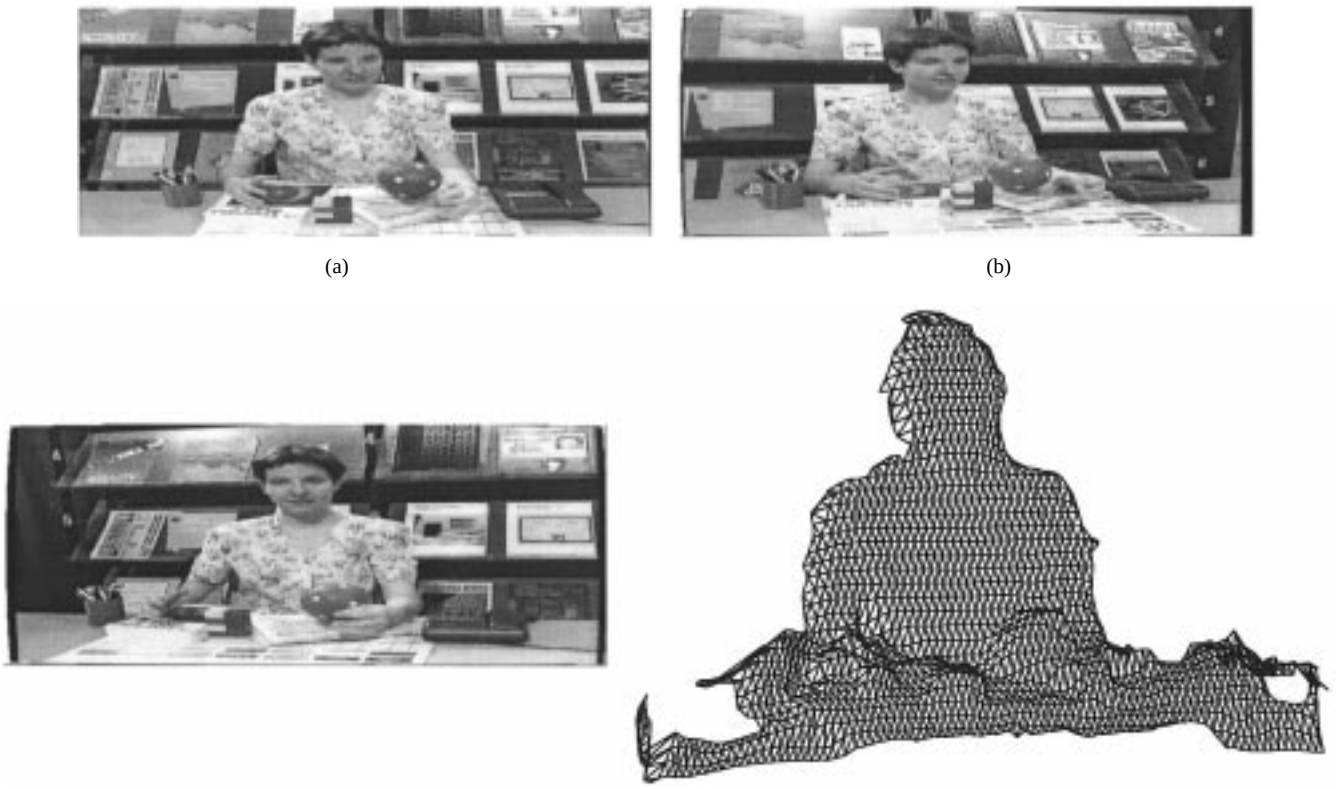
$$\boldsymbol{M} = \begin{bmatrix} 1 & -w_z^{(s)} & w_y^{(s)} & \tau_x^{(s)} \\ w_z^{(s)} & 1 & -w_x^{(s)} & \tau_y^{(s)} \\ -w_y^{(s)} & w_x^{(s)} & 1 & \tau_z^{(s)} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\boldsymbol{M}$ is determined using (18) and (19).



(a) (b) (c) (d)

Fig. 6. "Ludo8" sequence. (a) Top view. (b) Left view. (c) Right view. (d) 3-D model produced by the model initialization procedure.



(a) (b) (c) (d)

Fig. 7. "Gwen9" Sequence. (a) Top view. (b) Left view. (c) Right view. (d) 3-D model produced by the model initialization procedure.

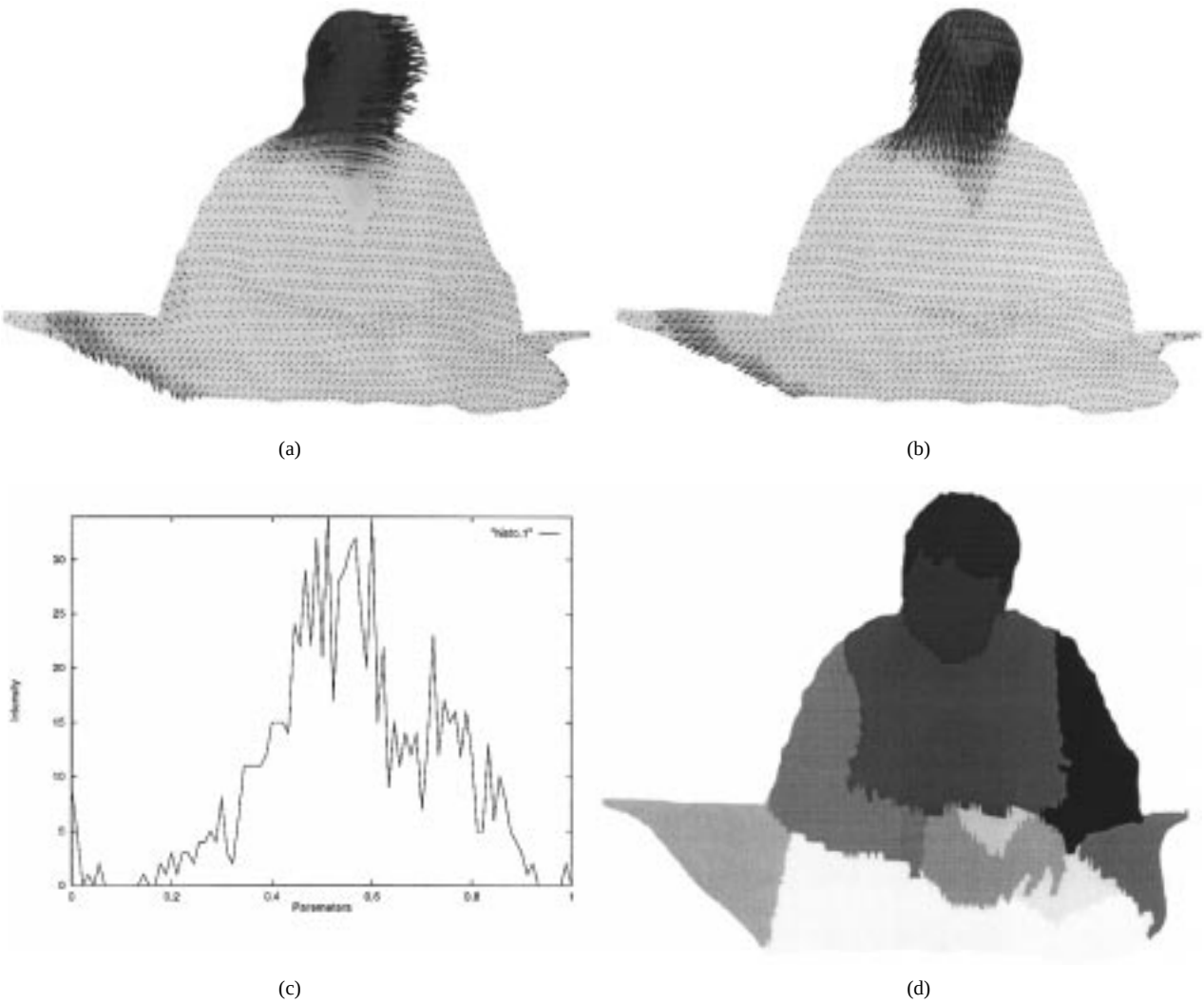


Fig. 8. "Ludo3" sequence: (a) 3-D translation parameters of each triangle. (b) 3-D rotation parameters of each triangle. (c) Histogram of the subobject parameters after some few iterations. (d) Final segmentation of the 3-D model.

C. Color and Depth Descriptor Estimation

In addition to the rigid 3-D motion descriptor of each triangle, other observations can be used for efficient segmentation. More specifically a color descriptor can be assigned to each subobject s. For each triangle $T_m^{(s)}$ contained in s, let

$$y^{(T_m^{(s)})} = \frac{1}{3} \sum_{c=0}^{2} \left(\frac{1}{N_c^{(T_m^{(s)})}} \sum_{i=1}^{N_c^{(T_m^{(s)})}} I_c(x_i, y_i) \right) \tag{20}$$

where $c = 0, 1, 2$ corresponds to $c = left, top, right$ cameras, $N_c^{(T_m^{(s)})}$ is the number of 2-D projected points for each triangle for each view c, and $I_c(x_i, y_i)$ is the intensity value of each projected view. The color descriptor for subobject s is then defined by

$$y^{(s)} = \frac{1}{M^{(s)}} \sum_{m=1}^{M^{(s)}} y^{(T_m^{(s)})}.$$

Similarly, in order to assign a depth descriptor to each triangle, we define

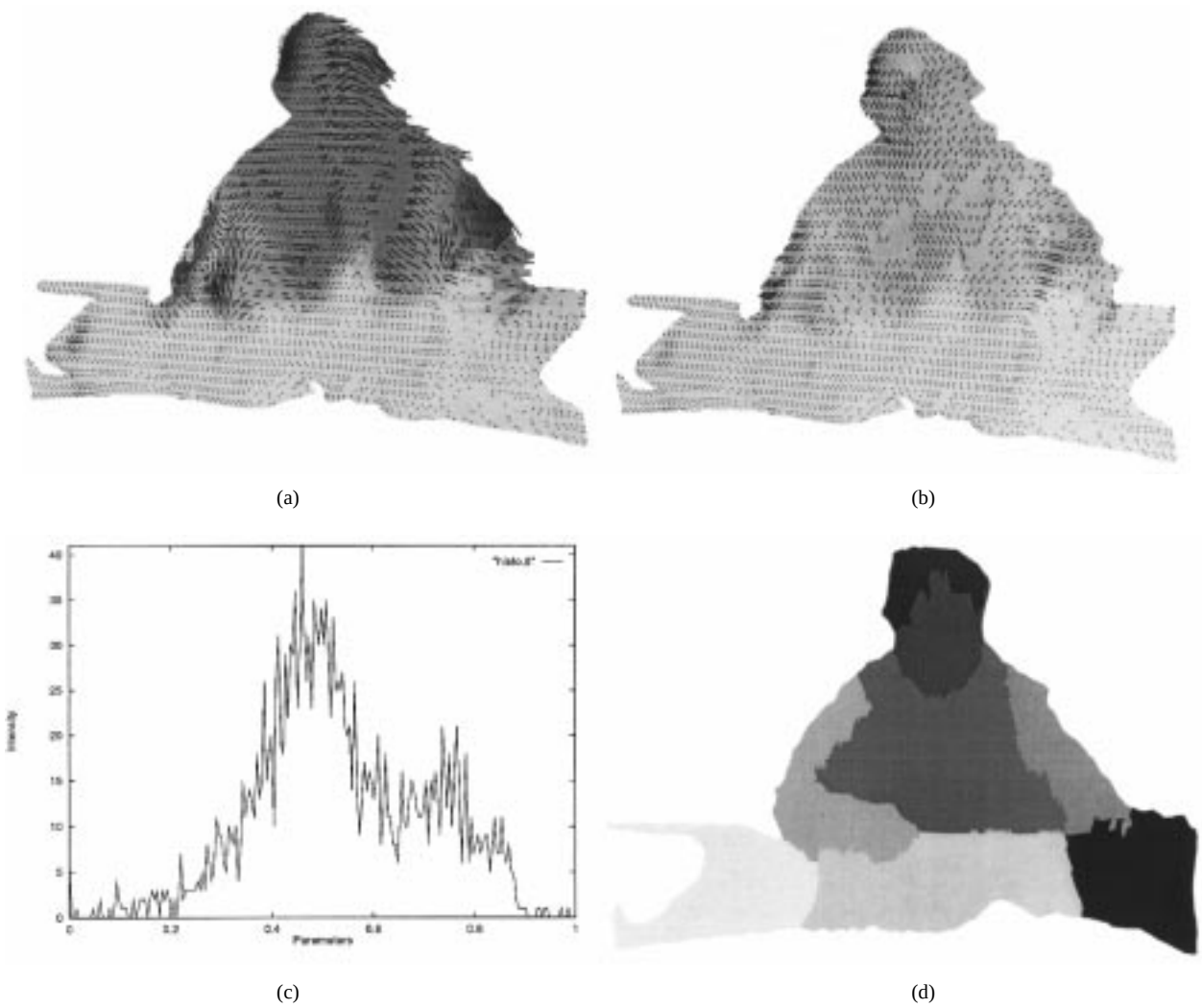
$$d^{(T_m^{(s)})} = \frac{1}{3} \sum_{c=0}^{2} \left(\frac{1}{N_c^{(T_m^{(s)})}} \sum_{i=1}^{N_c^{(T_m^{(s)})}} d_c(x_i, y_i) \right) \tag{21}$$

where $c = 0, 1, 2$ corresponds to $c = left, top, right$ cameras, $N_c^{(T_m^{(s)})}$ is the number of 2-D projected points for each triangle for each view c, and $d_c(x_i, y_i)$ are the projected depth maps to each view. For each subobject s, the depth descriptor is then defined by

$$d^{(s)} = \frac{1}{M^{(s)}} \sum_{m=1}^{M^{(s)}} d^{(T_m^{(s)})}.$$

VI. Object Segmentation

At each iteration of the object articulation method, the homogeneity of the descriptor of each subobject is exploited. For the motion descriptors, the rigidity constraint imposed on each rigid object component is exploited. This constraint



(a) (b) (c) (d)

Fig. 9. "Ludo8" sequence: (a) 3-D translation parameters of each triangle. (b) 3-D rotation parameters of each triangle. (c) Histogram of the subobject parameters after some few iterations. (d) Final segmentation of the 3-D model.

requires that the distance between any pair of points of a rigid object component must remain constant at all times and configurations. Thus, the motion of a rigid model object component represented by a mesh of triangles can be completely described by using the same six motion parameters. Therefore, to achieve object articulation, neighboring subobjects which exhibit similar 3-D motion parameters are joined together. In the same way, neighboring subobjects which exhibit similar color and depth descriptors are merged to the same subobject.

For each triangle, the following total descriptor value is estimated:

$$p^{(s)} = a_0\overline{w}_x^{(s)} + a_1\overline{w}_y^{(s)} + a_2\overline{w}_z^{(s)} + a_3\overline{t}_x^{(s)} + a_4\overline{t}_y^{(s)} + a_5\overline{t}_z^{(s)} + a_6\overline{y}^{(s)} + a_7\overline{d}^{(s)} \tag{22}$$

where $\overline{x}$ is the normalized value of x between 0 and 1 and

$$\sum_{i=0}^{7} a_i = 1.$$

The choice of values for a_i depends on the weight given to each specific descriptor in each specific application. For example, if rigidly moving components must be found, the weights corresponding to the rigid 3-D motion descriptors should be higher than the others.

In the region-growing algorithm described next, the similarity of descriptors of subobjects is checked only between neighboring subobjects. Two subobjects are considered to be neighbors if at least one triangle of one of the subobjects shares two common vertices with a triangle from the other subobject. Of course, if these triangle exist, they must lie on the "boundary" of each subobject.

Thus, the following iterative algorithm is proposed.

Step 1: Set $l = 1$.

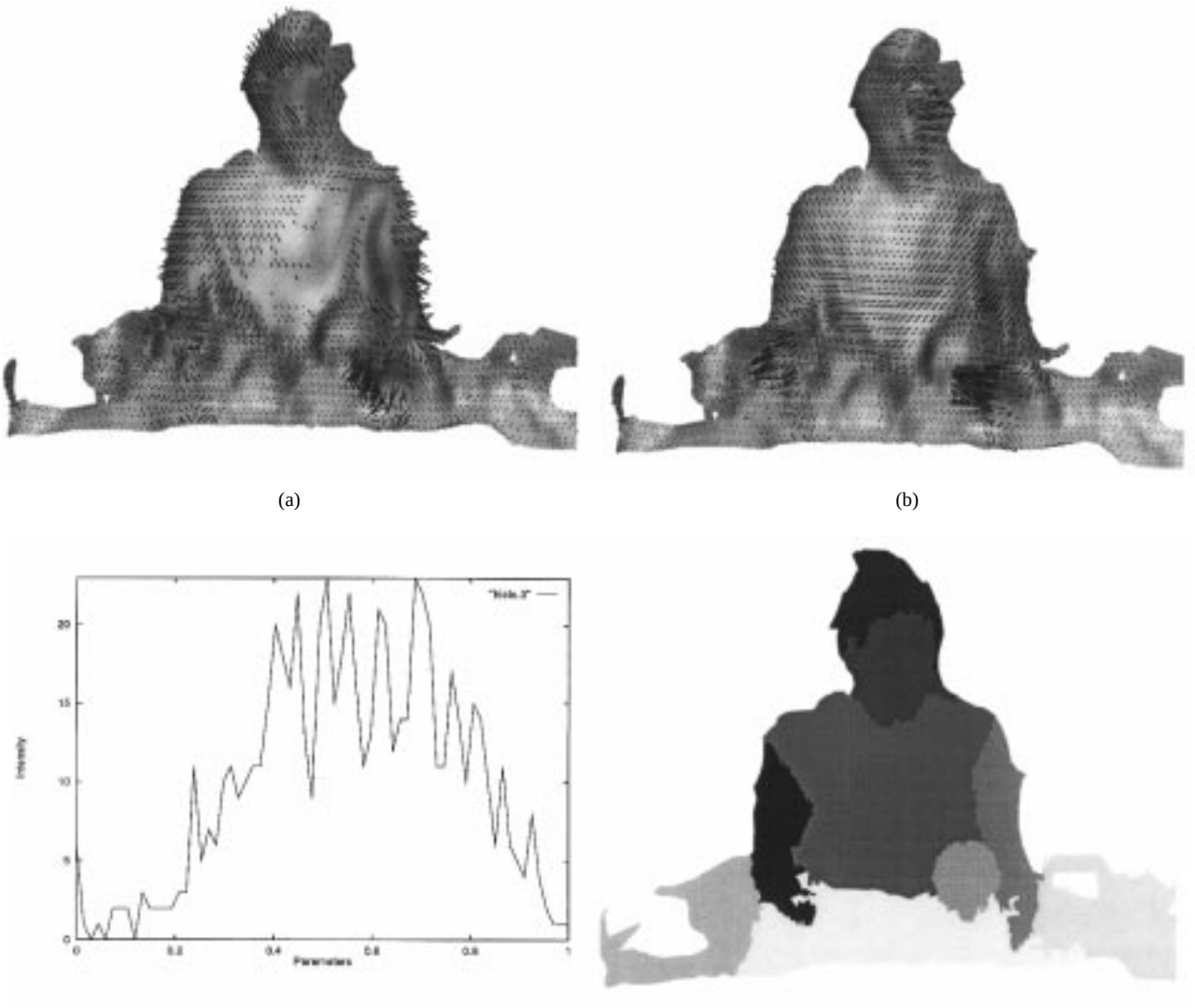
Step 2: Find subobject $s_m^{(i)}$ that is neighboring $s_l^{(i)}$. If

$$|p^{(s_l^{(i)})} - p^{(s_m^{(i)})}| \leq th^{(i)}$$

then $s_l^{(i)} = s_l^{(i)} \cup s_m^{(i)}, L^{(i)} = L^{(i)} - 1$ and go to Step 2. Else, go to Step 2 and search for another neighbor $s_l^{(i)}$. If no such neighbor exists, go to Step 3.

Step 3: Set $l = l + 1$. If $l \leq L^{(i)}$, go to Step 2. Else stop.

The above algorithm is a region-growing procedure based on the triangles of the 3-D model.



(a) (b) (c) (d)

Fig. 10. "Gwen9" sequence. (a) 3-D translation parameters of each triangle. (b) 3-D rotation parameters of each triangle. (c) Histogram of the subobject parameters after some few iterations. (d) Final segmentation of the 3-D model.

A. Threshold Update Procedure

A critical parameter for the convergence and efficiency of the algorithm is the choice of the threshold value $th^{(i)}$. This value is calculated on the basis of the histogram function of the parameters of each subobject. More specifically, the histogram function is given by

$$c = h(x)$$

where c is the number of subobjects with $p^{(s_l^{(i)})} = x$.

At each iteration i of the algorithm, the maxima of this function are estimated, and the threshold is found as the half distance between the two maxima closest to each other. For the estimation of the maxima, the method presented in [37] is used, following a smoothing procedure using a median filter. In other words, if the maxima occur at $x_i, i = 1, \cdots, M$ and $d_{ij} = ||x_i - x_j||$ is the minimum distance between any (x_i, x_j), then $th = (d_{ij}/2)$ (Fig. 3). For $i = 0$, the initial value of threshold is not found using the histogram because each subobject consists of only one triangle and the parameter estimation procedure is not expected to be very reliable. For this first iteration, a small value is given instead to th, and subobjects with very similar descriptors are first created, thus making the descriptor estimation procedure more reliable in the following iterations.

The histogram information is also used for the selection of the values of a_i in (22). More specifically, the weights for each descriptor are chosen in proportion to the maximum distance between maxima in the histogram of this specific descriptor. For example, if the value of a specific descriptor is constant over all subobjects, then the histogram produced based on this descriptor is a delta function, and the maxima extraction procedure will fail. So if the weight a_i associated with this descriptor is set to be very small, other descriptors with more characteristic distributions over the subobjects will dominate, and a more "segmentable" histogram will be produced. More specifically, the set of parameters a_i are determined so as to

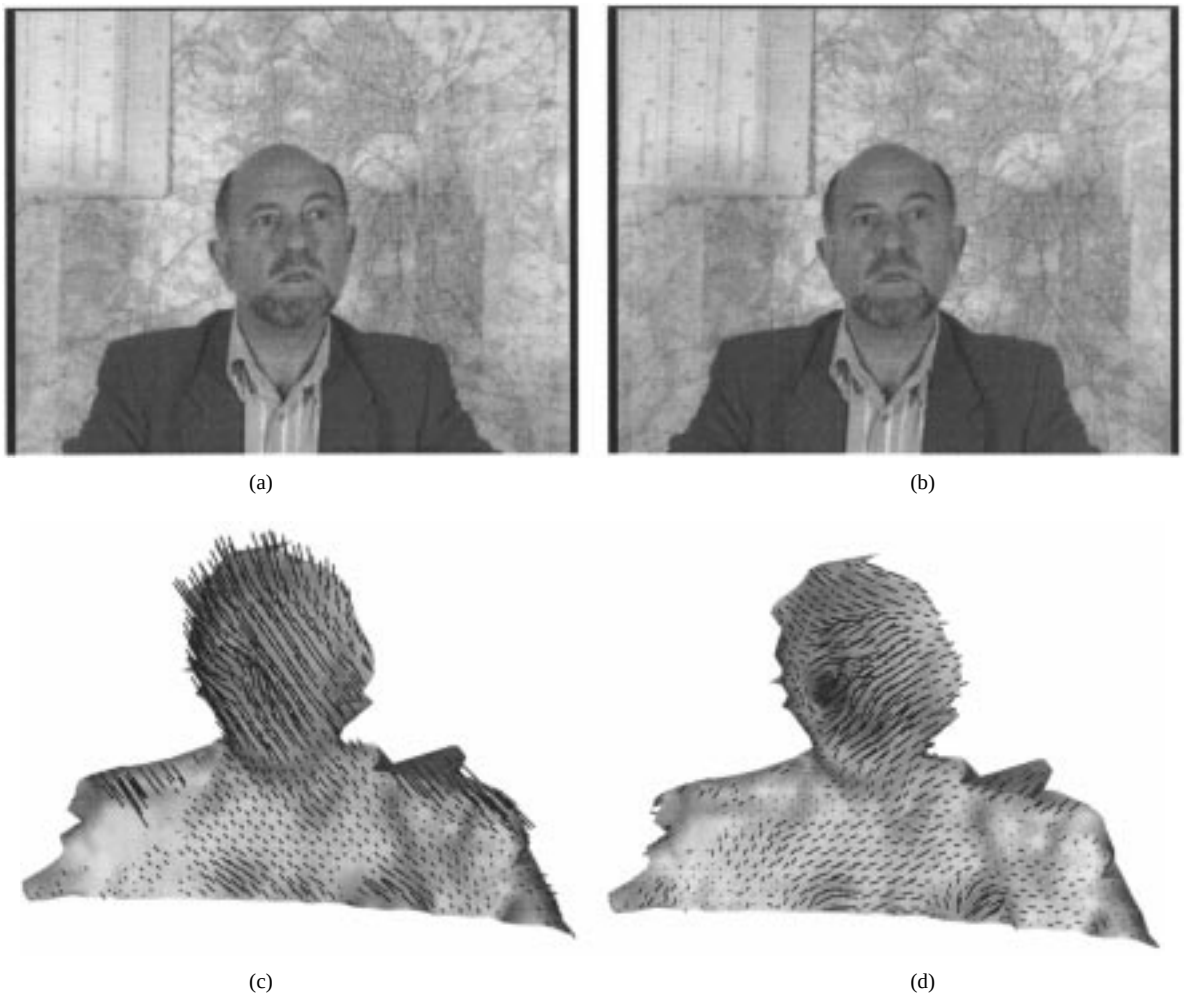


Fig. 11. "Claude" sequence. (a) Left channel of the "Claude" stereoscopic image sequence. (b) Right channel of the "Claude" stereoscopic image sequence. (c) 3-D translation parameters of each triangle. (d) 3-D rotation parameters of each triangle.

maximize the following function:

$$\sum_i \sum_j ||x_i - x_j|| \rightarrow \max .$$

VII. Applications in Monocular, Stereoscopic, and General Multiview Image Sequences

The multiview image sequence segmentation method presented may be modified so as to also be applied in monoscopic and stereoscopic image sequences if a 3-D model describing the views is produced. In monoscopic image sequences, many techniques may be used to adapt a 3-D model to the foreground object in videoconference image sequences. Knowledge-based methods [38] use explicit human head models for this purpose. Analysis-by-synthesis methods [2], [22], [39] are suitable for coding more general classes of images since the model creation procedure is based on extracted features such as shape or motion. In stereoscopic vision, depth information can be found as in the multiview case, using disparity estimation and stereoscopic camera geometry [18]. The model can be initialized as in the multiview case using the technique described in Section III.

Once a 3-D model becomes available, the segmentation procedure of the present paper can be applied directly, with obvious simple modifications. More specifically, (7) and (8) are calculated for $c = 0$ or $c = 0, 1$ in the monoscopic and stereoscopic case, respectively. All other operations remain the same since they are directly defined in the 3-D space. The only change is the dimension of the system (9), which becomes, respectively, $2 \times L$ and $2 \times 2 \times L$ in the monoscopic and stereoscopic cases. This system remains overdetermined in these cases also since a triangle usually consists of more than three points (in the worst monocular case, $2 \times 3 = 6$). Again, if the number of L falls below a predefined threshold, the neighborhood expansion technique described in the last paragraph of Section V-A is applied. Equations (20) and (21) are also calculated for $c = 0$ or $c = 0, 1$ and the sums are divided by 1 or 2, respectively.

The camera geometry also remains the same since the CAHV model is a general format that can describe various camera arrangements. For example, the simple perspective camera geometry, usually used in monocular image systems, can be described in CAHV format as follows. In perspective projection, the 3-D point is projected on the 2-D plane using $X = f(x/z), Y = f(y/z)$, where (x, y, z) is the 3-D point, (X, Y) is the 2-D projection, and f is the focal length. In CAHV format, this simply assigns the values $\boldsymbol{C} =$

$[0\ 0\ 0]^T, \boldsymbol{H} = [1\ 0\ 0]^T$, $\boldsymbol{V} = [0\ 1\ 0]^T$, and $\boldsymbol{A} = [0\ 0\ (1/f)]^T$ to the parameters of (1).

The segmentation procedure is trivially extendable so as to apply for an arbitrary number of cameras and corresponding projected views C. As explained above, (7) and (8) are calculated for $c = 0, \cdots, C-1$. The dimension of the system (9) becomes $2 \times C \times L$, and remains overdetermined even in the worst case where only one or two 3-D points are defined. Equations (20) and (21) are calculated for $c = 0, \cdots, C-1$ with the sums (20) and (21) divided by C.

VIII. Experimental Results

The proposed 3-D model-based segmentation algorithm was evaluated for the segmentation of the 3-D model created from real multiview image sequences. The interlaced multiview videoconference sequences of "Ludo3," "Ludo8," and "Gwen9" of size 720×576[1] were used. All experiments were performed at the top field of the interlaced sequences, thus using images of size 720×288.

The 3-D model was formed using the techniques described in Section III. A regular wireframe was first created covering the full size of the top view image, as shown in Fig. 4(a) for the "Ludo3" sequence. Only triangles lying on the foreground object were then retained, providing a coarse adaptation to the foreground object [Fig. 4(b)]. The wireframe was then finely adapted to the foreground object [Fig. 4(c)], and the force applied to each node was propagated all over the wireframe in order not to alter the regularity of the mesh near the boundary [Fig. 4(d)]. Using depth information, the 2-D wireframe was reprojected to the 3-D space giving the 3-D model [Fig. 5(d)]. The same procedure was followed for the other multiview image sequences. The 3-D model for the sequences "Ludo8" and "Gwen9" are shown in Figs. 6(d) and 7(d), respectively.

The rigid 3-D motion for each subobject was estimated using the methods described in Section V-B. The rigid 3-D motion for each triangle (iteration $i = 0$ of the algorithm, where each subobject consists of a single triangle) for the "Ludo3" sequence is shown in Fig. 8(a) and (b). More specifically, in Fig. 8(a), the translation parameters for each triangle are shown. The color of the triangle shows the magnitude of the translation vector, with darker areas corresponding to regions with greater motion. The vector shows the direction of the translation. In Fig. 8(b), the rotation parameters are depicted. The color of the triangle corresponds to the value of the angle of rotation (with darker areas corresponding to larger angles) and the vector shows the axis of rotation. The motion of the "Ludo3" sequence is a rotation of the head toward the right with almost no movement of the rest of the body. As can be seen, the estimated motion approximates the real motion quite accurately.

The histogram used in order to provide the decision for the threshold after the initial iteration is shown in Fig. 8(c). The final articulation of the 3-D model is shown in Fig. 8(d). As can be seen, all characteristic subobjects appear segmented.

[1]These sequences were prepared by the CNET Rennes (former CCETT) for use in the PANORAMA ACTS Project.

Fig. 12. Final segmentation of the 3-D model for "Claude."

The table is separated from the body even though neither object moves. In these areas, the motion descriptor is zero, and the dominant descriptors are the color and depth ones, which are efficiently used for the separation of the body from the table and the arms from the rest of the body.

The visualization of the rigid 3-D motion of each subobject at iteration $i = 0$ for "Ludo8" is shown in Fig. 9(a) and (b). The histogram after a few iterations is shown in Fig. 9(c). "Ludo8" is moving both his head and body toward the right, with greater motion in the area of the arms. The final segmentation is shown in Fig. 9(d), where it can be seen that characteristic parts of the body and the environment (table) are correctly recognized and separated using the combined motion, color, and depth information.

The visualization of the rigid 3-D motion of each subobject at iteration $i = 0$ for "Gwen9" is shown in Fig. 9(a) and (b). The histogram after a few iterations is shown in Fig. 10(c). "Gwen9" is moving toward the viewer while raising her hands. Again, the face was successfully separated from the body and from the table. Parts of the table with different color characteristics were separated into different subobjects.

The segmentation procedure was also applied to stereoscopic and monocular image sequences using the approach described in Section VII. The left and right channels of the interlaced stereoscopic videoconference sequence[2] "Claude" of size 360×288 are shown in Fig. 11(a) and (b), respectively. Depth was estimated using the algorithm described in [18]. The visualized 3-D motion is shown in Fig. 11(c) and (d). "Claude" is moving his head back to the left as is shown. The resulting segmentation is shown in Fig. 12.

The algorithm was also tested on the monocular videoconference image sequence "Claire" of size 176×144 [Fig. 13(a)]. The visualized 3-D motion shown in Fig. 13(b) and (c), approximates the real motion quite accurately since "Claire" is raising her head while turning it to the back. The model was created using the technique in [22]. The resulting segmentation is shown in Fig. 13(d).

All of the above articulations and rigid 3-D motion visualizations are available in both VRML 2.0 and OFF files format, for direct viewing in the 3-D space, at the WWW server http://uranus.ee.auth.gr/segmentation.

[2]This sequence was prepared by the Thompson Broadcasting Systems for use in the DISTIMA RACE project.

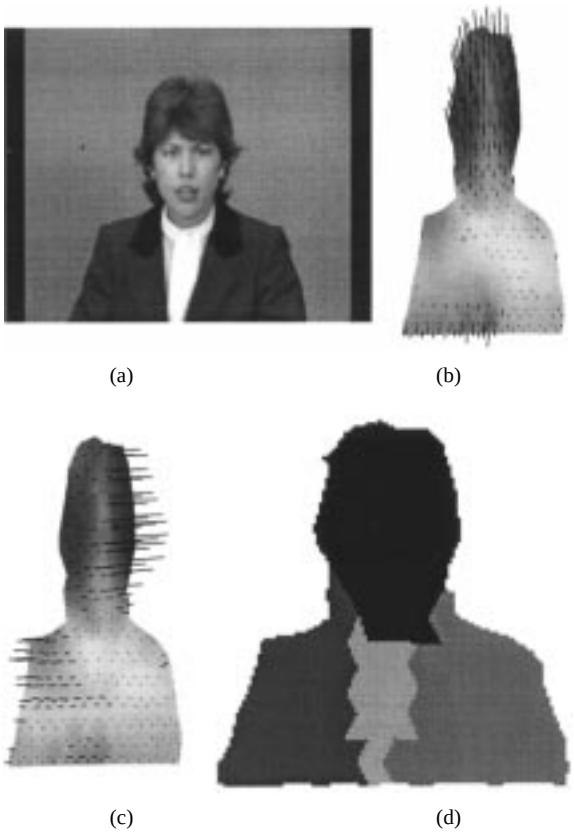


Fig. 13. "Claire" sequence: (a) First frame of "Claire." (b) 3-D translation parameters of each triangle. (c) 3-D rotation parameters of each triangle. (d) Final segmentation of the 3-D model.

IX. Conclusions

In this paper, an unsupervised procedure for the segmentation of multiview image sequences using multiple sources of information was presented. The 3-D model is initialized by adapting a 2-D wireframe to the foreground object. Using depth and multiview camera geometry, the 2-D wireframe is reprojected in the 3-D space, forming a consistent wireframe for all views. The articulation is based on the homogeneity of parameters estimated for each subobject, which consists of a number of interconnected triangles of the 3-D model. The rigid 3-D motion of each subobject for subsequent frames is estimated using a Kalman filtering algorithm. Information from all cameras is combined during the formation of the equations for the rigid 3-D motion parameters. The threshold used in the object segmentation procedure is updated at each iteration using the histogram of the subobject descriptors.

The methodology used overcomes a major obstacle in multiview video analysis, caused by the difficult problem of determining and handling coherently corresponding objects in various views. This is achieved in this paper by defining segmentation and object articulation in the 3-D space, thus ensuring that all ensuing operations (for example, rigid 3-D motion estimation of each subobject) remain coherent for all views of the scene.

A further advantage of the algorithm is that the segmentation is defined at the triangle level, thus making it possible to define the detail of the segmentation mask. In cases where a coarse segmentation mask is needed, a 3-D model consisting of large triangles may be used, making the segmentation procedure much faster, whereas for highly detailed masks, a finer mesh may be used. This is not possible with segmentation algorithms working at the pixel level.

The algorithm combines an arbitrary number of subobject descriptors. Based on the available information and the type of application, different sources with different weights can be used.

The important connectivity constraint for each subobject produced is implicitly imposed in the segmentation algorithm since each subobject is merged with only neighboring subobjects. Thus, no special postprocessing is needed in order to fill "holes" in the resulting subobjects. In fact, the only postprocessing procedure necessary is the merging of very small regions with larger ones.

Possible applications of the algorithm (apart from segmentation) include *rigid 3-D motion estimation* in model-based coding. The subobjects defined in the algorithm, along with their estimated rigid 3-D motion parameters, can be used to update the model in the next time instance [24]. The only parameters that need to be transmitted are the rigid motion parameters since the 3-D model is transmitted only at the beginning. In this manner, significant bit-rate savings may potentially be achieved. The rigid 3-D motion of each triangle, used in iteration $i = 0$ of the algorithm, can be used in a manner similar to that in [40] for *nonrigid* or *flexible 3-D motion estimation* of each node of the wireframe. A flexible 3-D motion can be assigned to each node by taking into account the rigid 3-D motion of all triangles having as a vertex the specific node.

References

[1] L. Chiariglione, "MPEG and multimedia communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 5–18, Feb. 1997.

[2] H. G. Musmann, M. Hotter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Commun.*, vol. 1, pp. 117–138, Oct. 1989.

[3] M. Hotter, "Object-oriented analysis-synthesis coding based on moving two-dimensional objects," *Signal Processing: Image Commun.*, vol. 2, pp. 409–428, Dec. 1990.

[4] S. Malassiotis and M. G. Strintzis, "Model based joint motion and structure estimation from stereo images," *Comput. Vision Image Understanding*, vol. 64, Nov. 1996.

[5] J.-R. Ohm and E. Izquierdo, "An object-based system for stereoscopic viewpoint synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, Aug. 1997.

[6] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 19–31, Feb. 1997.

[7] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, pp. 3–16, 1981.

[8] R. M. Haralick and L. G. Sapiro, "Image segmentation techniques," *Comput. Vision, Graphics, Image Processing*, vol. 29, pp. 100–132, 1985.

[9] D. Geiger and A. Yuille, "A common framework for image segmentation," *Int. J. Comput. Vision*, vol. 6, pp. 227–243, 1991.

[10] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a tree traversal algorithm," *J. ACM*, vol. 23, pp. 368–388, Apr. 1976.

[11] R. Leonardi, "Adaptive segmentation for image coding, Ph.D. dissertation, EPFL no. 691, Lausanne, Switzerland, 1987.

[12] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision*, vol. 1, pp. 312–331, 1988.

[13] L. H. Staib and J. S. Duncan, "Boundary finding with parametric deformable models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 161–175, 1992.

[14] P. Perona and J. Malik, "Scale space and edge-detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 629–639, July 1990.
[15] M. Matheron, *Random Sets and Integral Geometry*. New Work: Wiley, 1975.
[16] J. Serra, *Image Analysis and Mathematical Morphology*. London: Academic, 1982.
[17] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 583–598, June 1991.
[18] D. Tzovaras, N. Grammalidis, and M. G. Strintzis, "Object-based coding of stereo image sequences using joint 3-D motion/disparity compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, Apr. 1997.
[19] M. Hötter, R. Mester, and F. Müller, "Detection and descreption of moving objects by stochastic modeling and analusis of complex scenes," *Signal Processing: Image Commun.*, vol. 8, pp. 281–293, 1996.
[20] L. Wu, J. Benois-Pineau, P. Delagnes, and D. Barba, "Spatio-temporal segmentation of image sequences for object-oriented low bitrate image coding," *Signal Processing: Image Commun.*, vol. 8, pp. 513–543, 1996.
[21] N. Grammalidis, S. Malassiotis, D. Tzovaras, and M. G. Strintzis, "Stereo image sequence coding based on 3-D motion estimation and compensation," *Signal Processing: Image Commun.*, vol. 7, pp. 129–145, Aug. 1995.
[22] I. Kompatsiaris and M. G. Strintzis, "Automatic 3D model construction for rigid 3D motion estimation of monocular videoconference image sequences," in *Proc. Int. Workshop Synthetic Natural Hybrid Coding and 3-D Imaging*, Rhodes, Greece, Sept. 1997, pp. 44–47.
[23] G. Martinez, "Automatic analysis of flexibly connected rigid 3D objects for object-based analysis-synthesis coding (OBASC)," in *Proc. Picture Coding Symp. (PCS'94)*, Sacramento, CA, Sept. 1994, pp. 21–23.
[24] D. Tzovaras, I. Kompatsiaris, and M. G. Strintzis, "3D Object articulation and motion estimation in model-based stereoscopic videoconference image sequence coding," *Signal Processing: Image Commun.*, to be published.
[25] L. Falkenhagen, "Depth estimation from Stereoscopic Image Pairs Assuming Piecewise Continuous Surfaces," in *Proc. European Workshop Combined Real and Synthetic Image Processing for Broadcast and Video Productions*, Nov. 1994, pp. 23–24, Hamburg, Germany, pp. 23–24.
[26] E. Izquierdo and M. Ernst, "Motion/disparity analysis for 3D-videoconference applications," in *Proc. Int. Workshop Stereoscopic and 3-D Imaging*, M. G. Strintzis *et al.*, Eds., Santorini, Greece, Sept. 1995, pp. 180–186.
[27] F. Pedersini, D. Pelle, A. Sarti, and S. Tubaro, "Calibration and self-calibration of multi-ocular camera systems," in *Proc. Int. Workshop Synthetic Natural Hybrid Coding and 3D Imaging*, M. G. Strintzis *et al.*, Eds., Rhodes, Greece, Sept. 1997, pp. 81–84.
[28] Y. Yakimovski and R. Cunningham, "A system for extracting 3-D measurements from a stereo pair of TV cameras," *CGVIP*, vol. 7, no. 2, pp. 195–210, 1978.
[29] O. D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Approach.* Cambridge, MA: M.I.T. Press, 1993.
[30] D. Burr, "Elastic matching of line drawings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 6, pp. 708–713, 1981.
[31] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, pp. 384–401, July 1985.
[32] S. Lee and Y. Kay, "A Kalman filter approach for accurate 3-D motion estimation from a sequence of stereo images," *CVGIP: Image Understanding*, vol. 54, pp. 244–258, Sept. 1991.
[33] J. Kim and J. W. Woods, "3-D Kalman filter for image motion estimation," *IEEE Trans. Image Processing*, vol. 7, pp. 42–52, Jan. 1998.
[34] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice Hall, 1986, pp. 304–306.
[35] A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communications and Control.* New York: McGraw Hill, 1971.
[36] S. S. Sinha and B. G. Schunck, "A two-stage algorithm for discontinuity-preserving surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, Jan. 1992.
[37] R. K. R. Jain and B. Schunck, *Machine Vision*. New York: McGraw-Hill, 1995.
[38] K. Aizawa, H. Harashima, and T. Saito, "Model-based analysis-synthesis image coding (MBASIC) system for a person's face," *Signal Processing: Image Commun.*, vol. 1, pp. 139–152, Oct. 1989.
[39] E. Steinbach, B. Girod, and S. Chaudduri, "Robust estimation of multi-component motion in image sequences using the epipolar constraint," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 4, Munich, Germany, Apr. 1997, p. 2689.
[40] I. Kompatsiaris, D. Tzovaras, and M. G. Strintzis, "Flexible 3D motion estimation and tracking for multiview image sequence coding," *Signal Processing: Image Commun. (Special Issue on 3-D Video Technology)*, First Quarter 1998.

Ioannis Kompatsiaris (S'94) was born in Thessaloniki, Greece in 1973. He received the Electrical Engineering degree from the Electrical Engineering Department, Aristotle University of Thessaloniki, (AUTH), Thessaloniki, Greece, in 1996. He is currently working toward the Ph.D. degree at the Electrical and Computer Engineering Department, AUTH.

His research interests include image processing, computer vision, model-based monoscopic and multiview image sequence analysis and coding, and video coding standards.

Mr. Kompatsiaris is a member of the Technical Chamber of Greece.

Dimitrios Tzovaras was born in Ioannina, Greece, in 1970. He received the B.S. and Ph.D. degrees from the Electrical and Computer Engineering Department of the Aristotle University of Thessaloniki, (AUTH), Thessaloniki, Greece in 1992 and 1997, respectively.

Since September 1992 he has been a Research Assistant at the Information Processing Laboratory in AUTH. His research interests include image compression, monoscopic and stereoscopic image sequence analysis and coding, telemedicine, and multirate signal processing.

Dr. Tzovaras is a member of the Technical Chamber of Greece and SPIE.

Michael G. Strintzis (S'68–M'70–SM'79) received the Diploma in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1967, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1969 and 1970, respectively.

He then joined the Electrical Engineering Department, University of Pittsburgh, Pittsburgh, PA, where he served as an Assistant (1970–1976) and Associate (1976–1980) Professor. Since 1980, he has been a Professor of Electrical and Computer Engineering at the University of Thessaloniki, Thessaloniki, Greece. His current research interests include image coding, image processing, biomedical signal and image processing, and educational technology.

In 1984, Dr. Strintzis was awarded one of the Centennial Medals of the IEEE.