VISUALISATION OF VIDEOCONFERENCE IMAGE SEQUENCES USING VRML 2.0

Ioannis Kompatsiaris and Michael G. Strintzis

Information Processing Laboratory
Electrical and Computer Engineering Department
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece
email: ikom@dion.ee.auth.gr
phone: (+30-31) 996-359, FAX: (+30-31) 996-398

ABSTRACT

In this paper a procedure for visualisation of videoconference image sequences using Virtual Reality Modeling Language (VRML) 2.0 is described. First, image sequence analysis is performed in order to estimate the shape and motion parameters of the person talking in front of the camera. For this purpose, we propose the K-Means with connectivity constraint algorithm as a general segmentation algorithm combining information of various types such as colour and motion. The algorithm is applied "hierarchically" in the image sequence and it is first used to separate the background from the foreground object and then to further segment the foreground object into the head and shoulders regions. Based on the above information, the 3D shape parameters are estimated for each sequence and a 3D model is automatically adapted. The rigid 3D motion is estimated next for each sub-object. Finally a VRML file is created containing all the above estimated information and can be viewed using any VRML 2.0 compliant browser.

Keywords : *virtualised reality; model-based image sequence analysis; VRML.*

1 INTRODUCTION

Modeling of 3D scenes from real 2D images has been the focus of considerable attention in literature [1, 2, 3]. While much work has been done in the signal processing community, there is a lack in interportability and commonly accepted formats. For the time being, MPEG-4 standard working towards this direction is still in the development process [4]. On the other hand, VRML file format for describing 3D worlds has been developed by the computer graphics society, to represent mainly synthetic 3D environments [5]. VRML is a widely accepted format and it is used on the World Wide Web for dynamical interaction with 3D worlds.

The incorporation of real scenes into such environments is called *Virtualised Reality* [6]. Like Virtual Reality (VR), Virtualised Reality also immerses the viewer in a virtual environment. The two differ, however, in how the virtual worlds model are constructed. VR environments are typically created using simplistic CAD models and lack fine detail, special in the texturing part. Virtualised Reality, in contrast, automatically constructs the virtual model from images of the real world, preserving the visible detail of the real-world images. Furthermore, other aspects of the real world can be estimated and be added to the virtualised environment, such as motion. In [7] 2D motion extracted from real images was used to animate synthetic 2D objects.

In this paper, a procedure for visualisation of a videoconference image sequence in VRML 2.0 is described. First, image sequence analysis is performed in order to extract the parameters of the real world. Then, these parameters are converted into VRML format and a moving 3D representation of the image sequence can be viewed by any VRML compliant browser. The visualisation offers enhanced telepresence to the viewer, since a 3D representation of the scene is created. Furthermore, the user can, through the VRML browser, interact with the scene, for example turn the 3D model in order to see the person from the side. A segmentation scheme based on the K-Means clustering algorithm is used, in order to extract the 3D shape and motion parameters [8].

Since the 3D representation derived from real images is available, it is very easy to integrate synthetic objects in this environment. For example a new background can be added to replace the old one that can be totally synthetic or an other real image. Also several aspects of the virtualised environment, such as lighting, can be directly control through VRML nodes.

The paper is organised as follows. In the following Section the K-Means with connectivity constraint algorithm is described. In Section 3, the 3D shape parameters are estimated and the final 3D model is formed. In Section 4 the 3D model is used for rigid 3D motion estimation. The visualisation process using VRML is described in Section 5.

2 THE K-MEANS WITH CONNECTIVITY CONSTRAINT ALGORITHM

Clustering based on the K-Means algorithm is a region segmentation method which is widely used [9]. The

main problem of K-Means clustering is that the regions produced are not connected and there may be parts of a region contained in an other region. In order to solve this problem, we propose an extended K-Means algorithm with connectivity constraint. During the clustering procedure, before a pixel is assigned to a specific object, it is checked whether the objects remain solid and the merging criterion is also affected by this factor. Furthermore we combine information from different sources for more efficient segmentation.

More specifically, each pixel is going to be subdivided according to its position on the image (x, y) and a function describing characteristic pixel properties such as colour and motion

$$F(x,y) = a_c \tilde{I}(x,y) + a_m \tilde{V}(x,y) \ ,$$

where $\tilde{I}$ is a normalised version of the initial image between 0 and 1 and $\tilde{V}$ is also a normalised version of the picture produced by taking the frame differences of subsequent frames. For the weighting parameters it holds that : $a_c + a_m = 1$.

Two pixels (x_1, y_1) and (x_2, y_2) are considered connected if $|x_1 - x_2| <= 1 \ AND \ |y_1 - y_2| <= 1$ (8-connectivity). A pixel (x, y) is defined as *blocked* when it is decided to be connected to sub-object s_j but there is no way finding a path of connected each other pixels that haven't been assigned to any sub-object, starting from (x, y) and ending to a pixel that has been already assigned to s_j.

First the traditional K-Means algorithm (KM) is performed :

- Step 1 For every sub-object s_i set random initial values for C_i, where C_i is the property center of each sub-object.
- Step 2 For every pixel check the difference between $F(x, y)$ and C_j, $j = 0, 1$. If $|F(x, y) - C_k| < |F(x, y) - C_j|$ for $k \neq j$, assign (x, y) to sub-object s_k.
- Step 3 According to the new subdivision recalculate C_i. If M_i elements are assigned to s_i then

$$C_i = \frac{1}{M_i} \sum_{l=0}^{M_i - 1} F(x_l, y_l)$$

- Step 4 If the new C_i is equal with the old then stop, else goto *Step 2.*

The above result is going to be improved using the K-Means with connectivity constraint (KMC) algorithm :

- Step 1 Find the pixel with property closest to the property center of sub-object s_j, C_j, $j = 0, 1$, that has been already assigned to s_j by the KM algorithm. Remove all other elements previously attached to sub-objects $\{s_0, s_1\}$. Note that C_i remain unchanged as produced by the KM.
- Step 2 For every pixel (x, y) that has not been yet assigned to a sub-object :
  - Step 2.1 If $|F(x, y) - C_k| < |F(x, y) - C_j|$ for $k \neq j \ AND$ there is a pixel of s_k connected with (x, y) then assign (x, y) to s_k, else goto *Step 2.2*
  - Step 2.2 Check if pixel (x, y) is *blocked*. If it is, assign (x, y) to sub-object other than s_k. Goto *Step 3.* If it isn't *blocked* goto *Step 3*, leaving (x, y) unsorted.
- Step 3 If all pixels have been assigned to a sub-object stop. Else goto *Step 2.*

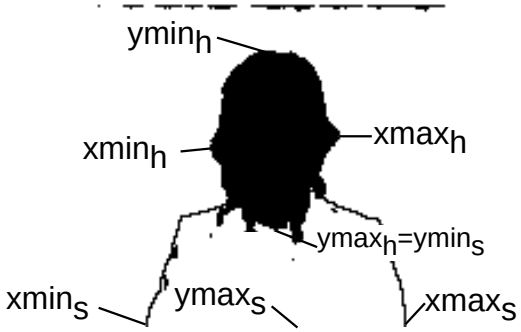

Figure 1: Points of interest on the segmentation picture

3 3D SHAPE PARAMETERS ESTIMATION

The segmentation procedure, described in the previous section, is applied to the image in a "hierarchical" manner. First the foreground is separated from the background and then the foreground object is further segmented to the head and shoulders regions. The colour and motion information $\tilde{I}(x, y)$ and $\tilde{V}(x, y)$ remain the same but the weighting parameters differ in the two segmentation procedures. The resulting segmentation mask for the "Claire" image sequence (Fig. 2) is shown in Fig. 4.

Based on the above extracted information, the 3D shape of both the head and shoulders region is going to be modeled by 3D ellipsoids with different parameters. The equation of a 3D ellipsoid is

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} + \frac{(z - z_0)^2}{c^2} = 1 \ . \qquad (1)$$

The parameters need to be calculated are : x_0, y_0, z_0, a, b, c for head (h) and shoulders (s). Parameter z_0 describes the depth of the most far away from the camera point, while $z_0 - c$ the point closest to the camera. These values can be arbitrary chosen taking into account that z_0 should be the same for the head and the shoulders and that c must be greater for the head : $z_{0h} = z_{0s}$, $c_h > c_s$.

The rest parameters are found by using points of interest as shown in Figure 1 :

$$a_h = (xmax_h - xmin_h)/2, \ x_{0h} = xmin_h + a_h,$$

$$b_h = (ymax_h - ymin_h)/2, \ y_{0h} = ymin_h + b_h,$$

$$a_s = (xmax_s - xmin_s)/2,\ x_{0s} = xmin_s + a_s,$$
$$b_s = ymax_s - ymin_s,\ y_{0s} = ymin_s + b_s.$$

As it can be seen shoulders are modeled as a half 3D ellipsoid. The resulting 3D models are shown in Fig. 5, 6, for "Claire" and "Miss America" (Fig. 3), respectively.

4 RIGID 3D MOTION ESTIMATION

A straightforward application since the 3D model is available, is to estimate the rigid 3D motion and use the small set of rigid 3D motion parameters in order to update the model in the next time instance and also reconstruct the next frame using only information from the previous one. Since there are sub-objects undergoing different rigid 3D motion, the rigid 3D motion estimation procedure is applied for each sub-object (head and shoulders).

The motion of an arbitrary point $\mathbf{P}(t)$ on sub-object s_k to its new position $\mathbf{P}(t+1)$ is described by

$$\mathbf{P}(t+1) = \mathbf{R}\mathbf{P}(t) + \mathbf{T}\ , \qquad (2)$$

where :

$$\mathbf{R} = \begin{bmatrix} 1 & -w_z & w_y \\ w_z & 1 & -w_x \\ -w_y & w_x & 1 \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} .$$

For the estimation of the model parameter vector $\mathbf{a} = (w_x, w_y, w_z, t_x, t_y, t_z)$ the rigid 3D motion of each point is projected on the 2D image plane using the camera geometry. The projected 2D motion is assumed to be equal with the 2D motion provided from a block matching algorithm and an overdetermined system for the rigid 3D motion parameters is formed. The system is solved using Least Median of Squares algorithm. In order to demonstrate the performance of the rigid 3D motion estimation algorithm the original frame difference between frames 0 and 2 is compared with the frame difference between original frame 2 and rigid motion compensated frame 2. The results for the "Claire" sequence are shown in Fig. 7, 8.

5 VISUALISATION WITH VRML 2.0

The recent version of VRML, VRML 2.0, provides enhanced static worlds together with interaction, animation and scripting. Animation can be obtained by nodes called interpolators. Similarly, the script nodes allow animation of objects through events they generate. In this work, we used the **IndexedFaceSet**, **Transform**, **PositionInterpolator**, **OrientationInterpolator**, **TimeSensor** and **ROUTE** nodes in order to visualise all the information extracted from the image sequence analysis. The **IndexedFaceSet** node is used to describe the geometry of our 3D model, consisting of 3D points and their triangular connections. The **IndexedFaceSet** node also contains an image for texture mapping onto the shape. The texture mapping is controlled by the texture coordinates, which take value in the range [0.0,1.0], and are specified in the two dimensional texture space (s,t), for each vertex of the shape. The interpolator nodes enable incorporating animation into the VRML 2.0 scene. An interpolator node takes a set of key values, and generates an interpolated value at the specified time instant using those key values. The **TimeSensor**, being a sensor node, keeps track of time and generates events as time passes. Typically, it is used for animations, periodic utilities, or timed events. Each of the two objects, head and shoulders, are grouped under a different **Transform** node, which controls their position in the 3D space. The rigid motion parameters of each sub-object are stored as key values in the **PositionInterpolator** and **OrientationInterpolator** nodes and using the **TimeSensor** node and the **ROUTE** node, those values are applied at regular intervals to the head and shoulders objects through the **Transform** node. The **PositionInterpolator** node holds the translation parameters, while the **OrientationInterpolator** node holds the rotation parameters.

As it can been seen, all the parameters needed in order to represent the videoconference scene in model-based from; 3D model, motion and texture, may be stored and viewed in VRML format. This provides interportability, thus solving a major problem with model-based schemes. Since for each model-based scheme a specific decoder and viewer is required the use of such schemes remain constrained. Using VRML not only a standardised file format and a general purpose viewer can be used, but the coding of the extracted parameters is also solved, since soon a standard VRML compressed format will be announced.

6 ACKNOWLEDGEMENTS

This work was supported by YPER project of the Greek Secretariat of Science and Technology.

References

[1] D. Tzovaras, N. Grammalidis, and M. G. Strintzis, "Object - Based Coding of Stereo Image Sequences using Joint 3-D Motion/Disparity Compensation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, Apr. 1997.

[2] K. Aizawa, H. Harashima, and T. Saito, "Model-based analysis-synthesis image coding (MBASIC) system for a person's face," *Signal Processing : Image Communication*, vol. 1, pp. 139–152, Oct. 1989.

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

[4] "Overview of the MPEG-4 Standard," tech. rep., ISO/IEC JTC1/SC29/WG11 N1730, Stockholm Jul. 1997.

[5] VRML 2.0 Specification, http://vrml.sgi.com/moving-worlds.

[6] T. Kanade and P. J. Narayanan, "Virtualised reality : Constructing virtual worlds from real scenes," *IEEE Multimedia*, pp. 34–46, Jan.-March 1997.

[7] P. E. Eren, C. Toklu, and M. Tekalp, "Object-based video manipulation and composition using 2d meshes in VRML," in *IEEE Workshop on Multimedia Signal Processing*, (Princeton, New Jersey, USA), pp. 257–261, June 1997.

[8] I. Kompatsiaris and M. G. Strintzis, "Automatic 3D Model Construction for Rigid 3D Motion Estimation of Monocular Videoconference Image Sequences," in *International Workshop on Synthetic Natural Hybrid Coding and 3D Imaging*, (Rhodes, Greece), pp. 44–47, Sept. 1997.

[9] S. Sakaida, Y. Shishikui, Y. Tanaka, and I. Yuyama, "Image segmentation by integration approcah using initial dependence of k-means algorithm," in *Picture Coding Symposium 97*, (Berlin, Germany), pp. 265–269, September 1997.

Figure 2: Original image "Claire".

Figure 3: Original image "Miss America".

Figure 4: Segmentation of "Claire".

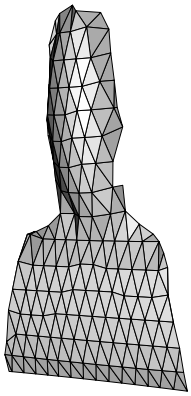

Figure 5: The resulting 3D model for "Claire"

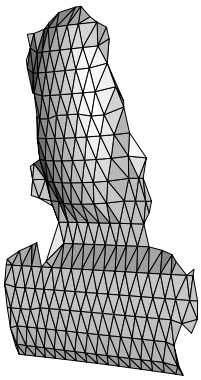

Figure 6: The resulting 3D model for "Miss America"

Figure 7: Frame difference between original frames 0 and 2 for "Claire".

Figure 8: Frame difference between original frame 2 and rigid motion compensated frame 2.