

Transactions Briefs

Two- Versus Three-Dimensional Object-Based Video Compression

A. Murat Tekalp, Yucel Altunbasak, and Gozde Bozdagi

Abstract— This paper compares two-dimensional (2-D) and three-dimensional (3-D) object modeling in terms of their capabilities and performance (peak signal-to-noise-ratio and visual image quality) for very low bitrate video coding. We show that 2-D object-based coding with affine/perspective transformations and triangular mesh models can simulate almost all capabilities of 3-D object-based approaches using wireframe models at a fraction of the computational cost. Furthermore, experiments indicate that a 2-D mesh-based coder–decoder performs favorably compared to the new H.263 standard in terms of visual quality.

Index Terms—Model-based coding, MPEG Phase 4, three-dimensional wireframe, two-dimensional mesh, video compression.

I. INTRODUCTION

The establishment of audio-visual conversational services over very low bit-rate channels, such as the public switched telephone network (PSTN) and wireless media, is an important emerging application for the telecommunications industry. As a short-term solution to the world-wide standardization of very low bit-rate video (less than 64 kb/s) compression/decompression, the International Telecommunications Union (ITU-T, formerly CCITT) has recently developed the draft recommendation H.263 [1]. Recommendation H.263 employs an improved version of the classical block-based motion-compensated (MC) discrete cosine transform (DCT) (hybrid) coding concept, which also forms the basis of the prior H.261, MPEG-1, and MPEG-2 standards. This hybrid MC-DCT coding strategy is, however, well known to generate blocking and mosquito artifacts at very low bit rates.

As a parallel effort, the International Standards Organization (ISO) has initiated MPEG Phase 4 in order to emphasize *content-based interactivity* and *universal accessibility* in addition to *improved compression efficiency*. Object-based video modeling is considered a promising approach to meet all of these functionalities. An object may be defined as a visually significant component of a scene. In the context of audio-visual conversational services, for example, objects are likely to be faces of speakers. Object-based representations are also relevant in content-based multimedia access and bitstream editing of compressed video, compression for interactive entertainment services, and synthetic-natural hybrid coding applications. Video objects can be represented by three-dimensional (3-D) or two-dimensional (2-D) models. A 3-D object may be described by a surface or a volume

Manuscript received November 21, 1995; revised April 9, 1996. This paper was recommended by Associate Editor K. Aizawa. This research is supported in part by a National Science Foundation IUCRC grant and a New York State Science and Technology Foundation grant to the Center for Electronic Imaging Systems at the University of Rochester.

A. M. Tekalp and Y. Altunbasak are with the Department of Electrical Engineering and Center for Electronic Imaging Systems, University of Rochester, Rochester, NY 14627 USA.

G. Bozdagi was with the Center for Electronic Imaging Systems, University of Rochester, Rochester, NY 14627 USA. She is now with the Digital Imaging Technology Center, Xerox Corporation, Webster, NY 14580 USA.

Publisher Item Identifier S 1051-8215(97)01145-2.

model that exhibits 3-D rigid or flexible motion, while a 2-D object is represented by a shape that undergoes a 2-D spatial transformation.

This paper compares 3-D versus 2-D object modeling for object-based compression. Although various modeling approaches have been reviewed in recent surveys, such as [2]–[6], a comparison of their performance in terms of peak signal-to-noise-ratio (PSNR) and visual quality has not been reported. Here, we compare a 3-D knowledge-based approach using a 3-D head-shoulders wireframe model, a 2-D triangular mesh-based method, and the new H.263 standard in terms of their motion compensation and compression performance. In Sections II and III, respectively, we review 3-D and 2-D object modeling. This analysis shows that a 2-D triangular mesh-based coder with affine motion mapping can simulate most capabilities of a 3-D wireframe-model-based approach under the orthographic projection. Further, 2-D mesh models (unlike 3-D wireframe models) can be easily designed for arbitrary scenes, and 2-D parametric motion estimation is a better-posed problem than 3-D motion and structure estimation. Comparative experimental results are provided in Section IV.

II. 3-D OBJECT MODELING

Modeling scene objects with 3-D structure and motion potentially enables accurate motion compensation, even in the presence of out-of-plane rotations and mild deformations, at the expense of increased complexity and sensitivity to deviations from the modeling assumptions. The first step in 3-D modeling is to detect the boundary (2-D silhouette) of objects of interest. Several methods have been proposed for automated detection of objects, which range from specialized algorithms, such as face detection modules [7], to general techniques, such as color and motion segmentation [8]. Three-dimensional model-based compression schemes can be classified as those that do not assume prior knowledge of the scene content (generic) and those which do (knowledge-based). Methods based on the source model of rigid 3-D objects (R3D) with 3-D motion [9] and flexible 3-D objects (F3D) with 3-D motion [10] belong to the former class. In these schemes, a 3-D triangular mesh (wireframe) model of an object of interest is reconstructed from its 2-D silhouette using the generalized cylinder approach. The flexible object model allows for deformations on the surface of the object due to local motion in addition to its global motion.

On the other hand, knowledge-based schemes are tailored for particular types of scenes, such as head-and-shoulders scenes. They start with a predesigned 3-D wireframe model which needs to be scaled and perhaps adapted to the scene under consideration. The MBASIC method [11], based on an orthographic global motion model and cut-and-paste facial expression synthesis, was among the first systems using a 3-D wireframe model. Later, Li and Forcheimer proposed simultaneous global and local motion estimation based on a 3-D wireframe model, where local motion was modeled in terms of a number of action units based on the facial action coding system (FACS) [12]. In these methods, scaling and adaptation of the wireframe model have been performed prior to motion estimation. As a result, any wireframe model misfit affects motion estimation negatively. More recently, Bozdagi *et al.* [13] presented a method for simultaneous global/local motion estimation and 3-D wireframe (3DW) model adaptation using geometric and

photometric constraints. Facial expression analysis based on the FACS can also be easily incorporated into this formulation. Finally, a codec which automatically switched from a generic object-based mode to a knowledge-based mode, using a face detection module, was presented in [14].

In the following, we briefly review the 3DW framework proposed by Bozdagi *et al.* Later, in Section III, we show that almost all capabilities of this 3-D model can be reproduced by a generic and more robust 2-D content-based mesh model. The source model is a 3-D wireframe model (composed of N flexibly connected triangular patches), where each patch is described by the equation of a plane in 3-D given by

$$Z = p_i X + q_i Y + c_i, \quad i = 1 \cdots N. \quad (1)$$

Here p_i , q_i , and c_i , $i = 1 \cdots N$, define the normal vectors of the patches and their z -offsets, respectively, which characterize the structure of the 3-D wireframe model. The problem of simultaneously estimating the 3-D global motion parameters ω_x , ω_y , ω_z , T_x , and T_y (under the orthographic projection) and adapting the wireframe model to a particular head-and-shoulders sequence (through estimation of p_i , q_i , and c_i , $i = 1 \cdots N$) can be formulated as minimization of the squared error in the optical flow equation over all pixels within object boundaries, given by [13]

$$E = \sum_{i=1}^N \sum_{(x,y) \in \text{ith patch}} e_i^2(x,y) \quad (2)$$

where $x = X$ and $y = Y$ denote the orthographic projection of the world coordinates into the image plane, and

$$\begin{aligned} e_i(x,y) = & I_x[\omega_z y - \omega_y(p_i x + q_i y + c_i) + T_x] \\ & + I_y[-\omega_z x + \omega_x(p_i x + q_i y + c_i) + T_y] \\ & + I_t - \rho(L_x, L_y, L_z) \\ & \cdot \left\{ \frac{\left(-\frac{-\omega_y + p_i}{1 + \omega_y p_i}, -\frac{\omega_x + q_i}{1 - \omega_x q_i}, 1 \right)}{\left[\left(\frac{-\omega_y + p_i}{1 + \omega_y p_i} \right)^2 + \left(\frac{\omega_x + q_i}{1 - \omega_x q_i} \right)^2 + 1 \right]^{1/2}} \right. \\ & \left. - \frac{(-p_i, -q_i, 1)}{(p_i^2 + q_i^2 + 1)^{1/2}} \right\} \quad (3) \end{aligned}$$

with respect to ω_x , ω_y , ω_z , T_x , T_y , p_i , q_i , and c_i where (L_x, L_y, L_z) is the unit vector in the mean illuminant direction, ρ is the surface albedo, and I_x , I_y , and I_t are the partial derivatives of image intensity in x , y , and t , respectively. The last term in (3) accounts for the photometric effects (intensity variations due to 3-D motion) and can be dropped should this effect be neglected.

It is important to note that the structure parameters p_i , q_i , and c_i are not completely independent of each other (see Fig. 1). Each triangular patch is surrounded by either three or two (if it is on the boundary of the object) other triangles. The fact that two neighboring patches must intersect at a straight line and three neighboring patches must meet at a point imposes constraints on the structure parameters in the form

$$p_i X^{(ij)} + q_i Y^{(ij)} + c_i = p_j X^{(ij)} + q_j Y^{(ij)} + c_j \quad (4)$$

where p_j , q_j , and c_j denote the parameters of the j th patch, and $[X^{(ij)}, Y^{(ij)}]$ denote the coordinates of a point that lies at the intersection of the i th and j th patches. These constraints serve to

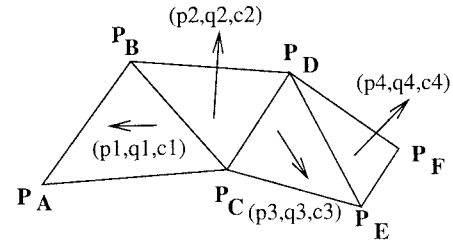


Fig. 1. Connectivity constraint for the 3-D mesh.

reduce the number of independent structure parameters and preserve the connectivity of the 3-D wireframe model. A stochastic relaxation algorithm has been used to find the best global motion and independent structure parameters. The independent structure parameters p_i , q_i , and c_i have been sequentially determined, where previously updated parameters at the same iteration cycle serve as constraints [13]. Although this formulation is quite powerful, complexity of the resulting algorithm and possible convergence to a local minimum may hamper the effectiveness of the method.

III. 2-D OBJECT MODELING

A variety of methods have been proposed for improved MC, including deformable block MC [15], overlapped block MC [16], region-based MC [17]–[19], and 2-D mesh-based MC [20]–[27]. These approaches can also be applied to 2-D object-based MC given the boundaries of objects of interest. Among the region-based approaches, Hoetter [17] proposed the source model of flexible 2-D objects (F2D) with 2-D motion, where the motion of an object is characterized by a subsampled dense motion field within the object boundaries. Later, Gerken [19] developed a very low bitrate codec based on this model. Mesh-based motion compensation features a continuously varying motion field which can model affine, perspective, or bilinear spatial deformations. Brusewitz [20] proposed triangle-based motion compensation, where a triangular mesh is overlaid on the image. Sullivan and Baker [21] used quadrilateral meshes for motion compensation under the name control grid interpolation. Recently, Nakaya *et al.* [23] proposed a hexagonal matching procedure for motion estimation and compensation based on a regular mesh. Huang *et al.* extended this approach using a hierarchical uniform mesh [24]. Wang *et al.* [25] developed an optimization framework for motion compensation based on an active mesh, which adapts to scene content. Altunbasak *et al.* [26] proposed practical algorithms for content-based mesh design and tracking. Two-dimensional mesh structures have been designed to fit within given object boundaries to obtain 2-D object models [27].

This paper claims that it is possible to capture most capabilities of a 3-D wireframe model-based approach using a 2-D content-based mesh model with triangular patches. The capabilities under consideration are: improved motion compensation, such as modeling out-of-plane rotations; simulation of facial expressions by action units; modeling of color/texture on uncovered sides of the wireframe model; and photometric effects. The improved motion compensation capability can be easily replicated by a 2-D mesh model, because: i) the orthographic/perspective projection of a wireframe model results in a 2-D mesh model with triangular patches, and ii) the 3-D motion of a planar patch can be described by an affine/perspective mapping in 2-D under the orthographic/perspective projection. Furthermore, estimation of mapping parameters is a better posed problem since it does not involve depth/structure estimation. Mapping parameter estimation methods which make it possible to realize this capability of 2-D meshes have only recently been proposed in [26] (by the authors),

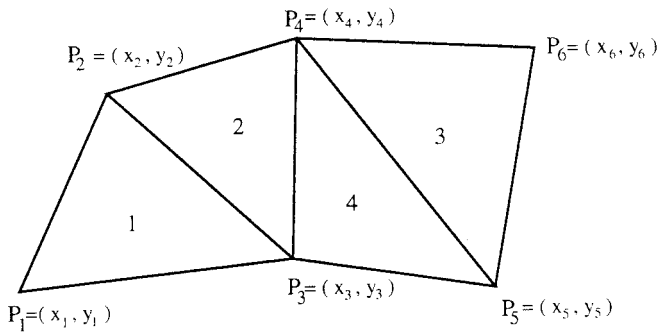


Fig. 2. Connectivity constraint for the 2-D mesh.

where closed-form expressions are derived for affine parameter estimation from a dense motion field under connectivity constraints. In the following, we show that the patch-based estimation method in [26] is a 2-D analog of the 3-D scheme discussed in Section II.

In [26], the mapping parameters are estimated to minimize the error between the measured dense flow and the predicted parametric flow, as is the case in the 3-D method [see (2) and (3)]. Because flow vectors other than those at the node points are also employed, geometrical constraints are needed to preserve the connectivity of the 2-D mesh [analogous to those in 3-D given by (4)]. Suppose that the motion of each patch is governed by a six parameter affine model, where $a_{i1} \dots a_{i6}$ denote the parameters for the i th patch. Then, the motion at pixel (x_j, y_j) within the i th patch of frame k can be modeled by

$$\begin{aligned} \tilde{u}_j &= \tilde{x}'_j - x_j = (a_{i1} - 1)x_j + a_{i2}y_j + a_{i3} \\ \tilde{v}_j &= \tilde{y}'_j - y_j = a_{i4}x_j + (a_{i5} - 1)y_j + a_{i6} \end{aligned} \quad (5)$$

where $(\tilde{x}'_j, \tilde{y}'_j)$ denotes the coordinates of the matching pixel in the $k - 1$ st frame as predicted by the model. The affine parameters for each patch are estimated by means of a constrained least squares estimation procedure given a set of optical flow estimates $u_j = x'_j - x_j$ and $v_j = y'_j - y_j$ within the respective patch. The constrained estimation procedure (which is analogous to the sequential estimation of the independent structure parameters in [13]) can be best explained by the following example.

Suppose we have a mesh with four patches as depicted in Fig. 2. Let $P_A = (x_A, y_A)$ and $P'_A = (x'_A, y'_A)$ denote the coordinates of node A at times t and $t + \Delta t$, respectively. Let $P_B, P'_B, P_C, P'_C, P_D, P'_D, P_E, P'_E, P_F,$ and P'_F be defined similarly. At the first patch, the motion vectors for all three nodes have not been previously estimated. Then, we estimate the affine motion parameters for the first patch (unconstrained) by minimizing

$$\sum_{j=1}^{N_1} (I_x \tilde{u}_j + I_y \tilde{v}_j + I_t)^2 \quad (6)$$

with respect to $a_{11} \dots a_{16}$, where N_1 is the number of estimated flow vectors within the first patch. Next, we estimate the affine parameters $a_{21} \dots a_{26}$ for patch 2, under the constraint that the motion vectors for the nodes B and C are already known, which can be expressed as

$$\begin{aligned} x'_B &= a_{21}x_B + a_{22}y_B + a_{23} \\ y'_B &= a_{24}x_B + a_{25}y_B + a_{26} \end{aligned} \quad (7)$$

$$\begin{aligned} x'_C &= a_{21}x_C + a_{22}y_C + a_{23} \\ y'_C &= a_{24}x_C + a_{25}y_C + a_{26}. \end{aligned} \quad (8)$$

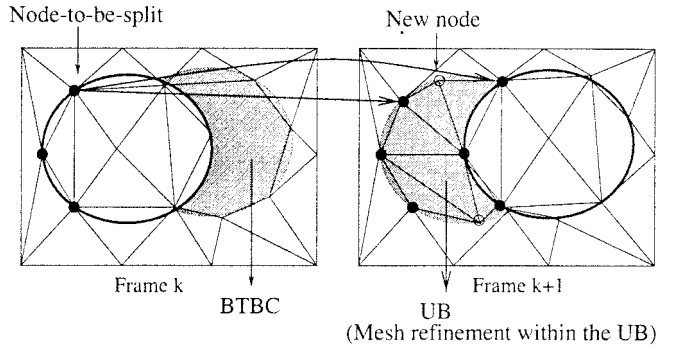


Fig. 3. Occlusion-adaptive mesh model.

Since there are four equations in six unknowns, there are only two free parameters in this case. Then, (6) is minimized with respect to the two free parameters using N_2 estimated point correspondences. Since an affine transformation maps a straight line onto another straight line, this will ensure preserving connectivity of the mesh along the line $P_B P_C$. In the case of patch 3, only the motion vector for the node D is known, so we have four free parameters. Choosing the free parameters as $a_{32}, a_{33}, a_{35},$ and a_{36} , we have

$$\begin{aligned} a_{31} &= \frac{x'_D - a_{32}y_D - a_{33}}{x_D} \\ a_{34} &= \frac{y'_D - a_{35}y_D - a_{36}}{x_D}. \end{aligned} \quad (9)$$

Then, (6) is minimized with respect to the four free parameters using N_3 estimated point correspondences. Finally, for patch 4, observe that the affine parameters can readily be estimated from the motion vectors at nodes $C, D,$ and E , which have already been estimated. The procedure is completed when all patches are visited. For best results, processing of patches should be prioritized such that patches where we have the highest confidence on the estimated flow (correspondence) vectors and the highest spatial image activity are processed first [26].

Standard 2-D mesh models [20]–[25] preserve connectivity over the entire frame. This limits our ability to synthesize certain facial actions, such as opening and closing of eyes and the mouth, using a 2-D mesh model. To this effect, we employ an occlusion adaptive 2-D mesh model, where discontinuities are allowed at occlusion and/or self-occlusion boundaries. The concept of occlusion-adaptive mesh is illustrated in Fig. 3, where new nodes are generated in the uncovered background (UB), and nodes in the background-to-be-covered (BTBC) are deleted. Following is a summary of the 2-D occlusion-adaptive mesh-based coding (2DM) algorithm.

- 1) Set $k = 0$. Encode the initial frame using an intraframe coding technique. Estimate the dense motion field from the k th to $(k+1)$ st frame. Find the BTBC in the k th frame given the dense motion field by thresholding the displaced frame difference. Contiguous regions are formed by eliminating small regions and filling in small holes. Fit a polygon about each contiguous BTBC region.
- 2) Design a 2-D content-based mesh in the k th frame, where no nodes are allowed in the BTBC, as follows.
 - a) Label all pixels, except those in the BTBC polygon(s), as “unmarked.” Select all corner points of the BTBC polygon(s) as node points.
 - b) Compute the average square displaced frame difference DFD_{2avg} (per node point) over all unmarked pixels.

- c) Compute a cost function

$$C(x, y) = \left| \frac{\partial I}{\partial x} \right| + \left| \frac{\partial I}{\partial y} \right| \quad (10)$$

associated with each unmarked pixel. Find the unmarked pixel with the highest $C(x, y)$ (so that selected node points, hence the boundaries of the patches, coincide with spatial edges) which is not closer to any other previously selected node point than a prespecified distance. Label this point as a node point.

- d) Grow a region about this node point until $\sum [DFD(x, y)]^2$ in this region is greater than DFD_{2avg} . Label all pixels within this region as "marked."
- e) Go to 2b) until a desired number of node points, N , are selected.
- f) Given the selected node points, apply Delauney triangulation to obtain a content-based mesh.
- 3) Compute node-point motion vectors to minimize (6) as described, and perform motion compensation. The processing order of patches is determined according to the criterion function

$$O_k = \alpha_1 \frac{DFD_k}{N_k} + \alpha_2 \frac{1}{\sigma_k^2}$$

where α_1 and α_2 are positive scalars and DFD_k , σ_k^2 , and N_k denote the displaced frame difference, the variance, and the number of points in patch k , respectively. The patch with the smallest O_k is processed first. Prioritization avoids propagation of the erroneous motion estimates.

- 4) Propagate all node points by their computed motion vectors. Increment k by one. Find all model failure (MF) regions using the actual and reconstructed k th frames. Find the enclosing polygons for each MF region. Encode intensity within the MF polygons using 8×8 DCT coding. Delete all node points inside the enclosing polygons.
- 5) Compute the dense motion field from the k th to $(k+1)$ st frame, and find the BTBC in the k th frame. Employ the following mesh refinement algorithm in the MF region(s) (excluding the BTBC).
- Insert the corner points of the MF polygon in the new node list.
 - Apply the node point selection algorithm within the MF region (excluding the BTBC).
 - Reapply triangulation.
 - Go to step 3).

The number of bits for MF region encoding is obtained by subtracting the bits spent for motion vector encoding from the number of bits allocated for each frame. At this time, bits allocated for each frame are input to the coder externally.

Two-dimensional mesh-based MC is interrelated to MC based on the F2D model proposed by Hoetter [17] and overlapped block MC (OBMC). Two-dimensional mesh-based object modeling can be viewed as an extension of the F2D model, whereby motion vectors at the node points of the mesh constitute an irregularly sampled motion field. The 2DM method employs a parametric dense motion field that is interpolated from these node point motion vectors by an affine mapping. While 2DM method considers affine combinations of the motion vectors to construct a smooth motion field, the OBMC method employs linear combinations of image intensities pointed by several neighboring motion vectors to obtain a

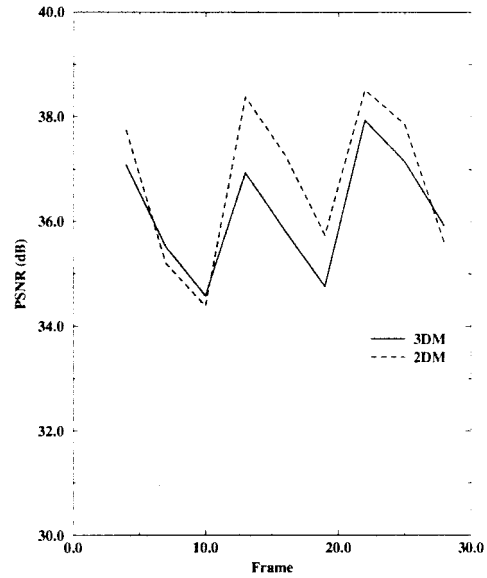


Fig. 4. Full-frame motion-compensation PSNR over frames 1–30.

smooth intensity image. Furthermore, the affine mapping (warping) originates from deterministic modeling (rigid motion of planar objects under orthographic projection), whereas OBMC may be derived by statistical modeling principles [16].

The 2-D mesh-based approach also offers a limited capability to model photometric effects and uncovered regions. Estimation of intensity variations (photometric effects) using 2-D mesh models has been discussed in [27]. Finally, texture/color mapping from a background memory in uncovered areas can be implemented by using more sophisticated data structures, such as 2-D mosaics.

IV. RESULTS

In this section, we provide two sets of experimental results using the *Miss America* sequence. The first set of experiments compares 2-D and 3-D modeling in terms of their MC performance. The second set of experiments show that 2-D mesh model based coder–decoder (codec) performs favorably compared to the new H.263 standard in terms of PSNR and visual quality.

In the first set of experiments, we have utilized the CANDIDE wireframe model [29] in the 3DW-based method and its perspective projection in the 2DM-based method. The 3-D wireframe model (with 155 nodes) is interactively fitted to each frame using 11 feature points corresponding to the top of the head, the tip of the chin and the nose, left and right cheeks—upper and lower positions, and a point midway between the right and left eyes, the center of the mouth. That is, these feature points were marked manually. The scaled wireframe model overlaid on the third frame (starting from frame 0) is depicted in Fig. 5(b). The 2DM method followed the steps described in Section III, except that the 2-D content-based mesh was replaced by the perspective projection of the CANDIDE model in this case. Ten common intermediate format (CIF) frames, from 0–27 (inclusive), skipping every two frames, are processed. Each frame is predicted from the original of the previously processed frame. The optical flow estimates were obtained by the method of Lucas–Kanade (LK) [28]. Experimental procedures follow those described in [13] and [26]. The results of comparative evaluation in terms of the average (over the ten processed frames) full-frame MC PSNR and facial MC PSNR are presented in Table I. In this case, comparison of the facial PSNR is more meaningful, since the 3-D wireframe only

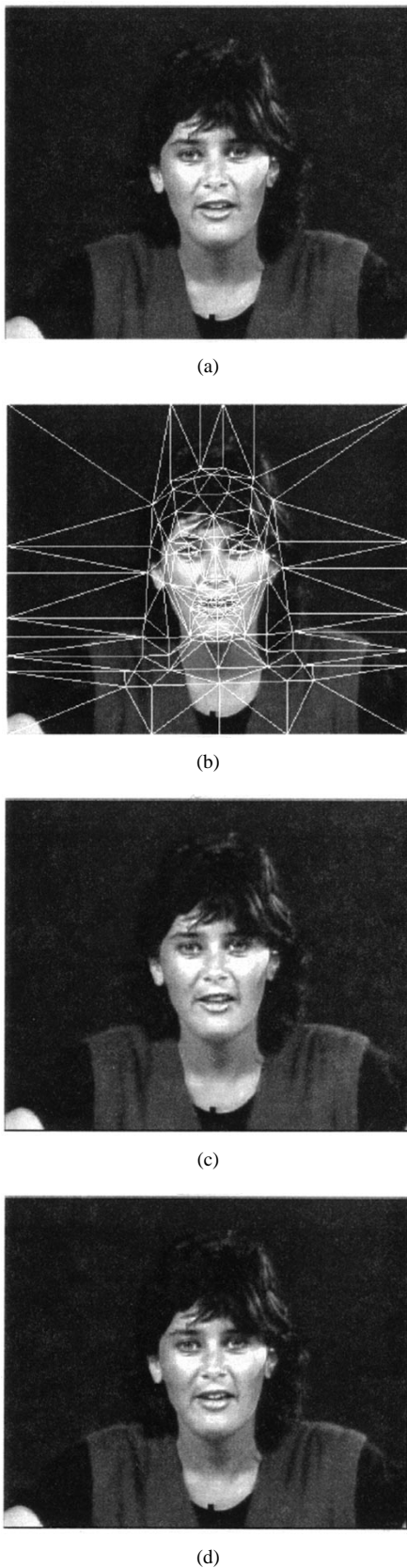


Fig. 5. (a) Original third frame. (b) CANDIDE model overlaid on the third frame. Motion-compensated third frame using (c) 3DW method and (d) 2DM method.

models the facial area. Fig. 4 depicts the full-frame MC PSNR for each processed frame of the sequence. Fig. 5(a), (c), and (d) show

TABLE I
MOTION COMPENSATION PSNR FOR 2-D MESH
VERSUS 3-D WIREFRAME-BASED METHODS

Method	Average Facial PSNR (dB)	Average Frame PSNR (dB)
2DM	33.04	36.74
3DW	33.01	36.19



Fig. 6. The content-based mesh overlaid on the zeroth frame.

TABLE II
COMPARISON OF A 2-D MESH-BASED
METHOD (2DM) WITH THE H.263 STANDARD

Frame	2DM		TMN5 v1.6		TMN5 v1.6 adv.pred.	
	PSNR (dB)	Bits	PSNR (dB)	Bits	PSNR (dB)	Bits
0	37.68	10632	37.68	10632	37.68	10632
3	37.64	1264	37.31	1320	37.32	1264
6	37.43	1240	36.92	1312	37.06	1240
9	37.17	624	36.70	672	36.76	624
12	37.12	600	36.75	568	36.73	600
15	36.85	384	36.42	360	36.47	384
18	36.51	704	35.85	712	36.05	704
21	36.36	632	35.57	664	35.80	632
24	36.19	464	35.50	432	35.77	464
27	35.92	392	35.48	400	35.54	392

the original and motion compensated third frames using the 3DW and 2DM methods, respectively. Inspection of the results indicate that the performance of the two algorithms are very similar. However, the 2-D approach requires about 30 s per frame, whereas the 3-D approach requires 4–5 h per frame on a SparcStation 20M. (It is possible to use a gradient-based optimization scheme with the 3-D approach which may reduce the computation time to several minutes per frame at the expense of quality of the results.)

The purpose of the second set of results is to compare the performance of the 2DM method with that of the new H.263 standard. To this effect, we implement a 2-D mesh-based codec, including an entropy coder, where a fixed amount of bits are allocated for each frame [26]. A content-based (adaptive) mesh is generated for the zeroth frame (shown in Fig. 6) which is then tracked frame-to-frame using the algorithm described in Section III. The experiments were performed at 16 kb/s and 10 Hz using a quarter common intermediate format (QCIF) version of the *Miss America* sequence. The version 1.6 of the TMN5 by Telenor is used with the advanced prediction mode (including overlapped motion compensation) ON and OFF, respectively, for the H.263 anchor. TMN5 has a buffer control algorithm which allocates a variable number of bits per frame. Because a sophisticated rate control scheme has not been



(a)



(b)



(c)

Fig. 7. (a) Original 26th frame. Reconstructed 26th frame by (b) H263 and (c) 2DM methods.

developed for the 2DM method, we have forced all three methods, 2DM, TMN5 v1.6, and TMN5 v1.6 advanced prediction mode, to process every third frame. Furthermore, the 2DM method allocated bits for each frame to match the bit allocation regime of the TMN5 v1.6 advanced prediction mode for the sake of comparison of PSNR values. The resulting decompression PSNR (full-frame) and actual bits per frame used for all algorithms are listed in Table II. It is important to observe that the 2DM method outperforms TMN5 (both baseline and advanced prediction modes). This is in spite of the fact that our software is still in development and has not been optimized in any way. Fig. 7(a)–(d) show the original and reconstructed 26th frames using the H.263 (advanced mode OFF) and the 2DM methods, respectively. Visual evaluation of motion rendition also favors the 2DM method, because the affine mappings accommodate rotation and scaling in addition to translation, and the continuity of the motion field alleviates blocking artifacts at very low bitrates.

V. CONCLUSION

This paper shows that a 2-D object-based codec with a triangular mesh model can capture most capabilities of 3-D object-based codecs using a wireframe model. Furthermore, we have compared the performance of a 2-D mesh-based codec with that of the new H.263 standard for very low bit-rate coding in terms of PSNR and visual quality. The results indicate that adaptive mesh-based methods compare favorably against the H.263 standard, although some more work is needed to improve the mesh-based codec. Although 2-D object modeling has been found sufficient for motion-compensated coding applications, 3-D object modeling may be valuable for other applications including synthetic-natural hybrid coding, facial animations, and virtual reality.

REFERENCES

- [1] ITU-T Recommendation H.263, "Video coding for low bitrate communication (TMN5)," July 1995.
- [2] H. G. Musmann, M. Hotter, and J. Osterman, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Commun.*, vol. 1, pp. 117–138, Oct. 1989.
- [3] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A review," *IEEE Trans. Image Processing*, vol. 3, pp. 589–609, Sept. 1994.
- [4] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate applications," *Proc. IEEE*, vol. 83, pp. 259–271, Feb. 1995.
- [5] D. E. Pearson, "Developments in model-based video coding," *Proc. IEEE*, vol. 83, pp. 892–906, June 1995.
- [6] T. Ebrahimi, E. Reusens, and W. Li, "New trends in very low bitrate video coding," *Proc. IEEE*, vol. 83, pp. 877–891, June 1995.
- [7] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, pp. 705–740, May 1995.
- [8] J. Y. A. Wang and E. Adelson, "Representing moving images with layers," *IEEE Trans. Image Processing*, vol. 3, pp. 625–638, Sept. 1994.
- [9] J. Ostermann, "Object-oriented analysis-synthesis coding based on the source model of moving rigid 3D object," *Signal Processing: Image Commun.*, vol. 6, May 1994.
- [10] —, "Object-based analysis-synthesis coding based on the source model of moving flexible 3-D objects," *IEEE Trans. Image Processing*, vol. 3, pp. 705–711, Sept. 1994.
- [11] 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.
- [12] H. Li, P. Roivainen, and R. Forchheimer, "3-D motion estimation in model-based facial image coding," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 545–555, June 1993.
- [13] G. Bozdađı, A. M. Tekalp, and L. Onural, "3-D motion estimation and wireframe adaptation including photometric effects for model-based coding of facial image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 246–256, Sept. 1994.
- [14] J. Ostermann and M. Kampmann, "Automatic adaptation of a facial mask in an analysis-synthesis coder based on moving flexible 3D objects," in *Int. Workshop Coding Tech. for Very Low Bitrate Video*, Colchester, UK, Apr. 1994, paper no. 2.4.
- [15] V. Seferidis and M. Ghanbari, "General approach to block-matching motion estimation," *Opt. Eng.*, vol. 32, pp. 1464–1474, July 1993.
- [16] M. Orchard and G. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Trans. Image Processing*, vol. 3, pp. 693–699, Sept. 1994.
- [17] M. Hoetter, "Object oriented analysis-synthesis coding based on moving two-dimensional objects," *Signal Processing: Image Commun.*, vol. 2, pp. 409–428, Dec. 1990.
- [18] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Processing: Image Commun.*, vol. 3, pp. 23–56, 1991.
- [19] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 3, pp. 228–236, Sept. 1994.
- [20] H. Bruswitz, "Motion compensation with triangles," presented at Int. Conf. 64-kb Coding of Moving Video, Rotterdam, Netherlands, Sept. 1990.

- [21] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proc. ICASSP'91*, Toronto, Canada, 1991, pp. 2713–2716.
- [22] J. Niewegłowski, T. G. Campbell, and P. Haavisto, "A novel video coding scheme based on temporal prediction using digital image warping," *IEEE Trans. Consumer Electron.*, vol. 39, pp. 141–150, Aug. 1993.
- [23] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 339–356, June 1994.
- [24] C. L. Huang and C. Y. Hsu, "A new motion compensation method for image sequence coding using hierarchical grid interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 1, pp. 72–85, 1994.
- [25] Y. Wang and O. Lee, "Active mesh—A feature seeking and tracking image sequence representation scheme," *IEEE Trans. Image Processing*, vol. 3, pp. 610–624, Sept. 1994.
- [26] Y. Altunbasak, A. M. Tekalp, and G. Bozdagi, "2-D object based coding using a content-based mesh and affine motion parameterization," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, Oct. 1995, pp. 394–397.
- [27] C. Toklu, A. T. Erdem, M. I. Sezan, and A. M. Tekalp, "Tracking motion and intensity variations using hierarchical 2D mesh modeling for synthetic object transfiguration," *Graphical Models, Image Processing*, vol. 58, no. 6, pp. 553–573, Nov. 1996.
- [28] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA Image Understanding Workshop*, 1981, pp. 121–130.
- [29] W. J. Welsh, "Model-based coding of videophone images," *Electron. Commun. Eng. J.*, pp. 29–36, Feb. 1991.

Edge Detection of Color Images Using Directional Operators

J. Scharcanski and A. N. Venetsanopoulos

Abstract—This paper discusses an approach for detecting edges in color images. A color image is represented by a vector field, and the color image edges are detected as differences in the local vector statistics. These statistical differences can include local variations in color or spatial image properties. The proposed approach can easily accommodate concepts, such as multiscale edge detection, as well as the latest developments in vector order statistics for color image processing. A distinction between the proposed approach and previous approaches for color edge detection using vector order statistics is that, besides the edge magnitude, the local edge direction is also provided. Note that edge direction information is a relevant feature to a variety of image analysis tasks (e.g., texture analysis).

Index Terms—Canny edge detector, color gradient, color image processing, edge detection, edge linking.

I. INTRODUCTION

The latest advances in color edge detection apply vector order statistics to spatially locate edges in color images [1]. The idea of applying vector order statistics to detect color image edges, and their spatial orientations, has also been proposed in the context of color image analysis [2]. In fact, the direction of color image edges can be utilized as a feature in a variety of image analysis tasks [3]. This work presents a class of directional operators designed to detect the location and orientation of edges in color images. The latest concepts

Manuscript received November 3, 1995. This paper was recommended by Associate Editor D. Anastassiou.

J. Scharcanski is with the Institute of Electrical Engineering, Federal School of Engineering of Itajuba, Itajuba, Brazil.

A. N. Venetsanopoulos is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 1A4, Canada.

Publisher Item Identifier S 1051-8215(97)02559-7.

in color image processing using vector order statistics can be easily incorporated to this approach. Also, multiscale edge detection can be obtained by more complex versions of these operators, namely *compound operators*, as will be detailed later.

Edges occur when there are local statistical differences in the distribution of scenic elements [4]. The statistical differences can include local variations in color, structure, or both. Various operators have been proposed for edge detection in monochrome images [5]. The *Prewitt operator* is well known and has its 3×3 row and column impulse response arrays as follows:

$$\Delta H = \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\Delta V = \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \quad (1)$$

A limitation of this type of operator is its inability to detect accurately edges in high-noise environments. However, this problem can be alleviated by extending the neighborhoods over which the differential gradients are computed.

Color edge detection can use concepts similar to those of the *Prewitt operator* for monochrome images. Two directional operators, orthogonal to each other, can be convolved with the color image, and the magnitude and direction of the image gradients be calculated based on the outputs of these operators. These directional operators may be implemented in various sizes, and the detected discontinuities (edges) can be associated with different ranges of spatial frequencies (e.g., slower spatial changes are associated with smaller spatial frequencies). Next, we present a formulation for the color edge detection approach used in this work.

II. DIRECTIONAL OPERATORS FOR EDGE DETECTION IN COLOR IMAGES

In this approach, a color $c(r, g, b)$ is represented by a vector \vec{c} in color space. Similar to (1), the row and column directional operators (i.e., in the horizontal and vertical directions) each have one positive and one negative component. For operators of size $(2w+1) \times (2w+1)$ the configuration is the following:

$$\Delta \vec{H} = [\vec{H}_- \quad \vec{0} \quad \vec{H}_+], \quad \Delta \vec{V} = \begin{bmatrix} \vec{V}_- \\ \vec{0} \\ \vec{V}_+ \end{bmatrix} \quad (2)$$

where the parameter w is a positive integer. These positive and negative components are *convolution kernels*, denoted by \vec{V}_- , \vec{V}_+ , \vec{H}_- , and \vec{H}_+ , whose outputs are vectors corresponding to the local *average colors*. In order to estimate the color gradient at the pixel (x_o, y_o) , the outputs of these components are calculated as follows:

$$\vec{H}_+(x_o, y_o) = \frac{1}{w(2w+1)} \sum_{y=y_o-w}^{y=y_o+w} \sum_{x=x_o+1}^{x=x_o+w} \vec{c}(x, y)$$

$$\vec{H}_-(x_o, y_o) = \frac{1}{w(2w+1)} \sum_{y=y_o-w}^{y=y_o+w} \sum_{x=x_o-1}^{x=x_o-w} \vec{c}(x, y)$$

$$\vec{V}_+(x_o, y_o) = \frac{1}{w(2w+1)} \sum_{y=y_o+1}^{y=y_o+w} \sum_{x=x_o-w}^{x=x_o+w} \vec{c}(x, y)$$

$$\vec{V}_-(x_o, y_o) = \frac{1}{w(2w+1)} \sum_{y=y_o-1}^{y=y_o+w} \sum_{x=x_o-w}^{x=x_o+w} \vec{c}(x, y) \quad (3)$$