A Low Bit-Rate Video Codec Based on Two-Dimensional Mesh Motion Compensation with Adaptive Interpolation

Pohsiang Hsu, K. J. Ray Liu, and Tsuhan Chen

***Abstract*—Visual communication over low bandwidth channels, such as the telephone line, requires high compression. Motion estimation and compensation techniques play an important role in achieving high compression by reducing the temporal redundancy inherent in video sequences. Current video coding standards employ block-matching algorithm to perform motion estimation that produces visually annoying blocking artifacts at low bit rate. In contrast, 2-D mesh motion compensation produces blocking free prediction by generating a smooth full motion field from the set of node motion vectors using spatial interpolation. However, the assumption of a globally smooth motion field is not valid for most natural occurring scenes, where motion discontinuities exist due to moving objects. In this paper, we propose a post-processing scheme that refines the set of node motion vectors obtained by the 2-D mesh motion estimation. The approach is to selectively break the smoothness constraint of 2-D mesh for patches that contain motion boundaries by changing the interpolation patterns. Experimental results show improvements in both PSNR and perceptual quality over the 2-D mesh motion compensation and a block-based standard video-coding standard H.263.**

***Index Terms*—Motion compensation, motion estimation, video coding.**

I. Introduction

LOW BIT-RATE video coding is one of the key components to the realization of new audio-visual communication services through low bandwidth channels. For these applications to work, a high compression of the video data is a necessity. Fortunately, typical video sequences contain large amounts of redundancy along the temporal dimension that can be reduced through motion estimation and compensation to achieve considerable compression. Indeed, this is the approach used by most existing video-coding standards (MPEG-1 and 2, H.261, and H.263), where the block matching algorithm (BMA) is used for motion estimation. However, BMA is incapable of modeling nontranslational motions such as rotations and zooming, since it employs a 2-D translational block motion model. More importantly, the use of a block translational motion model in BMA is known to produce visually annoying blocking artifacts, which occurs between adjacent blocks with different motion vectors.

One promising approach to remedy this problem is to use a 2-D mesh-based motion estimation and compensation based on image warping. In this approach, the motion prediction image is formed through warping the previous frame using spatial interpolation. The use of spatial interpolations gives the 2-D mesh motion estimator the ability to model frequently occurring nontranslational motions such as rotations and zooming. More importantly, the warping process produces a smooth motion field that is devoid of blocking artifacts. 2-D mesh motion estimation and compensation for video coding have been widely studied in the literature [1]–[6], [12].

2-D mesh motion estimation can be classified as backward tracking or forward tracking [3]. In backward tracking, each new frame is covered with new mesh and its nodes are adjusted by establishing correspondence with its previous frame. On the hand, the mesh is said to be forward tracking if the mesh is propagated from the previous frame to the current frame by establishing correspondence from the previous frame to the current frame. We note that backward tracking has lower computational complexity than forward tracking. Recently, Wang and Ostermann [5] evaluated the use of mesh-based motion estimation and compensation in H.263-like coders where they reported that the coded images have lower average PSNR than H.263 at similar bit rates.

One inherent problem with 2-D mesh-based motion estimation is that a globally smooth motion field model is assumed. Clearly, this is not true for most image sequences where the motion field contains discontinuities along moving object's boundary. Recognizing this fact, Altunbasak and Tekalp [6] proposed the occlusion adaptive mesh concept. In this approach, an irregular triangular mesh was first designed for the first frame and forwardly tracked. This forward-tracking approach involves node additions and deletions in newly discovered occlusion area (covered regions and uncovered regions), where the maintenance of the mesh structure can become quite complex. Also, Ishwar and Moulin [12] proposed a hybrid approach that combines BMA with regular quadrilateral mesh motion estimation and compensation with backward tracking.

In this paper, we propose a simple approach to allow the 2-D mesh to handle occlusion in the covered regions. The framework is built upon regular quadrilateral mesh motion estimation and compensation with backward tracking. This was chosen due to computational complexity considerations. The proposed scheme performs post-processing on the set of node motion vectors obtained by 2-D mesh motion estimation in two steps. First, the smooth bilinear interpolation used by 2-D mesh is adaptively changed for patches that contain motion discontinuities by selecting from a fixed set of new interpolation patterns. Second, the set of node motion vectors is refined with respect to the new set of interpolation patterns. The proposed method improves the motion-compensated prediction image by giving the motion estimator some ability to handle motion discontinuities. In addition, the proposed approach will always outperform regular quadrilateral mesh motion estimation because the adaptive interpolation pattern selection is greedy in nature.

Manuscript received November 2, 1998; revised March 1, 2000. This work was supported in part by the Office of Naval Research under Grant N00014-93-10566 and the National Science Foundation under Grant MIP9457397. This paper was recommended by Associate Editor K. N. Ngan.

P. Hsu and K. J. R. Liu are with the Electrical Engineering Department and Institute for System Research, University of Maryland, College Park, MD 20742 USA (e-mail: phsu@eng.umd.edu; kjrliu@eng.umd.edu).

T. Chen is with the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: tsuhan@ece.cmu.edu).

Publisher Item Identifier S 1051-8215(01)00670-X.

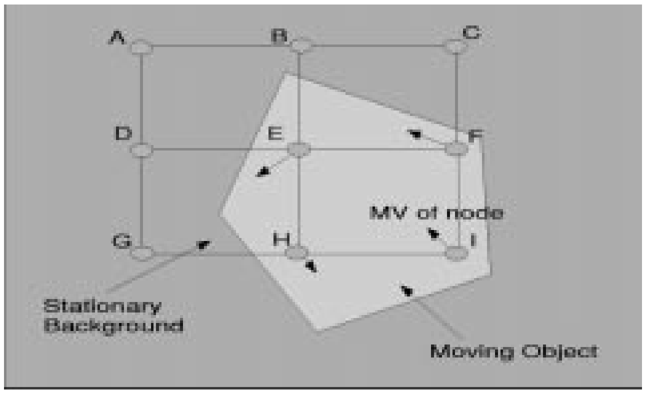


Fig. 1. Illustration of incorrect interpolation of motion field in patches containing multiple motions.

II. 2-D Mesh Motion Estimation and Compensation

In this section, we describe the 2-D mesh motion estimator with backward tracking used in our system. For each frame to be coded, we cover it and its previous frame with a regular quadrilateral mesh. The nodes of the mesh are spaced 16 pixels apart in both directions. The nodes located on the border of the image are extrapolated using the nearest internal node as in [1]. In 2-D mesh motion estimation, the mesh is deformed in the previous frame so that the contents of the two frames match with each other as closely as possible. The displacement of the nodes (i.e., node motion vector) will be used to generate the full motion field through spatial transformation. Given the set of node motion vectors, the full motion field is generated patch by patch through bilinear interpolation using the node motion vectors. Let $\mathbf{s} = (x, y)$ denote a point in the image and $\mathbf{d}(\mathbf{s}) = (d_x(\mathbf{s}), d_y(\mathbf{s}))$ denotes the motion vector for the point $\mathbf{s}$. Given the four node motion vectors of a patch $\mathbf{d}(\mathbf{s_0}), \mathbf{d}(\mathbf{s_1}), \mathbf{d}(\mathbf{s_2}), \mathbf{d}(\mathbf{s_3})$, we can obtain the motion vectors for the points inside its patch by

$$\begin{aligned} d(\mathbf{s}) = d(\mathbf{s_0})(1-u')(1-v') + d(\mathbf{s_1})u'(1-v') \\ + d(\mathbf{s_2})v'(1-u') + d(\mathbf{s_3})u'v' \end{aligned} \tag{1}$$

where u' and v' are normalized coordinates that span the patch, i.e.,

$$u' = \frac{x - x_0}{x_1 - x_0}, \quad v' = \frac{y - y_0}{y_2 - y_0}. \tag{2}$$

Since the same node motion vector is used in the interpolation process of all four adjacent patches that contain it, a connectivity constraint is placed which implies that a smooth motion field will be generated. In our system, we employ a modified version of the popular hexagonal matching algorithm (HMA) proposed by Nakaya and Harashima [2], where the node motion vectors are initialized using a three level hierarchical block matching algorithm [11]. For the three-level hierarchical block matching, we used block sizes of 32, 16, and 8 pixels with the node located in the center of the block and with a corresponding search range of $\pm 15, \pm 7$, and ± 3 pixels. Once the set of node motion vectors is initialized, they are refined in an iterative fashion to obtain the final set. In each pass of the refinement, each node in the mesh is visited once in some scanning order and perturbed in a small region while keeping its eight surrounding node motion vectors fixed. Moreover, the perturbation of the node is constrained to lie inside the region enclosed by its eight surrounding node motion vectors. The process continues until a local minimum is reached or a preset number of iterations have been reached. In our system, we adapt the raster scanning order for simplicity.

III. Problem Formulation

In a typical video scene, the true motion field will normally contain discontinuities due to moving objects. These motion boundaries can not be handled properly under the 2-D mesh framework because a continuity constraint has been placed on it. More specifically, the 2-D mesh motion estimator can only produce a continuous motion field inside each patch due to the parametric modeling. This implies that for patches containing multiple motions, the performance of the 2-D mesh motion compensation will most likely be unacceptable, since we will be attempting to model the motions of different objects by a continuous motion field. Moreover, the node motion vectors of these patches will be adjusted so that a compromise in terms of prediction error is reached which means that these motion vectors are no longer true motion vectors. This gives rise to error propagation in the iterative process of node motion vector refinement.

To illustrate the points made so far, let us consider the scenario shown in Fig. 1, where we have a rotating object on a still, inhomogeneous background. We focus our attention on patch D-E-G-H, where there are two distinct motions, one for the rotating object, and the other corresponds to the still background. This means that the true motion field inside this patch is discontinuous with a motion boundary along the moving object. In this case, the 2-D mesh motion estimator will attempt to find node motion vectors D, E, G, and H that forms a good prediction inside this patch. Because the actual discontinuous motion field cannot be generated through bilinear transformation using the node motion vectors, the 2-D mesh motion estimator will often generate unreliable node motion vectors as a compromise, which results in bad prediction.

Given the true node motion vectors for patch D-E-G-H, the motion field inside this patch should be ideally generated as follows. The moving object should be predicted using a combination of node motion vectors E and H, while node motion vectors D and G will be used to generate the background portion of the patch. The adaptive usage of the node motion vectors described above can also be seen as changing the interpolation pattern used in generating the motion field inside the patch. More specifically, we are changing from the scene-independent fixed bilinear interpolation using the four node motion vectors to an adaptively defined interpolation pattern that utilizes the node motion vectors to adjust to the scene content. This is also similar to performing a motion field segmentation [7] of this patch. However, an exact adaptive interpolation pattern requires the knowledge of the object's contour to be known at the decoder, which must be sent as side information or estimated at the decoder side using previous frames.

To balance the tradeoff between accurate boundary description (leading to lower distortion) versus the side information rate needed for description of the contours, we make the following two assumptions. First, each patch contains at most two moving

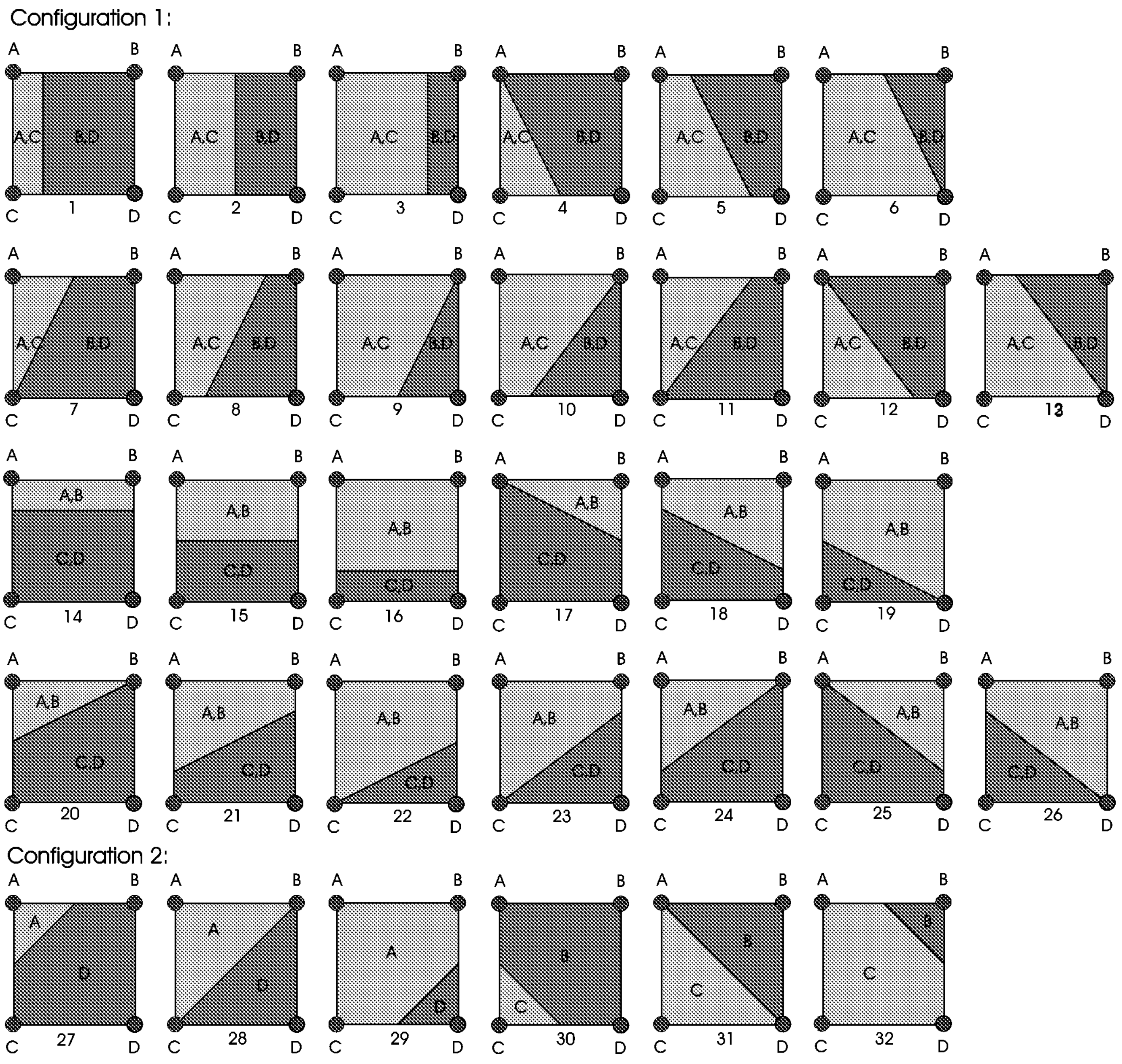


Fig. 2. Set of interpolation patterns.

objects. Second, a straight line can approximate the division between the two moving objects. Moreover, the number of straight line divisions is constrained to lie in a limited set of straight lines resulting in further reduction of rate necessary for description. In summary, we propose to construct a small set of interpolation patterns in addition to the fixed bilinear interpolation pattern. The set of interpolation patterns will then be used in a motion field post-processing scheme to allow adaptive utilization the node motion vectors for patches that contain multiple motions.

IV. Adaptive Interpolation Patterns

In this section, we describe the set of interpolation patterns used to approximate the motion boundary inside a patch. In addition to the default bilinear interpolation used in 2-D mesh, 32 new interpolation patterns are introduced and shown in Fig. 2. Each interpolation pattern specifies how the motion field will be generated for a 16×16 patch given its four node motion vectors. The interpolation patterns are constrained to be straight line divisions only where straight line separates each interpolation pattern into two regions. The interpolation patterns are classified into two types of configurations. In the first configuration, the interpolation pattern consists of two regions where each region contains two node motion vectors. In the second configuration, the interpolation pattern consists of two regions where one region contains three node motion vectors, while the other region contains only one.

Let the node motion vectors of a patch be denoted by $\mathbf{d}_A, \mathbf{d}_B, \mathbf{d}_C, \mathbf{d}_D$ in the order shown in Fig. 2. The interpolation patterns in Fig. 2 specify the set of node motion vectors used to generate the motion field inside the region, as indicated by the letters inside the region. For instance, the first interpolation pattern on the first row in Fig. 2 belongs to the second configuration. It divides the region into two vertical strips, where node motion vectors $\mathbf{d}_A$ and $\mathbf{d}_C$ are used for the narrow strip, while node motion vectors $\mathbf{d}_B$ and $\mathbf{d}_D$ are used for the wide strip.

For the first configuration, the motion field inside each region is interpolated using its two node motion vectors. The following rules are used for interpolation in the region containing:

1) node motion vectors $\mathbf{d}_A$ and $\mathbf{d}_B$: $\mathbf{d}(x,y) = \mathbf{d}_B * (x/16) + \mathbf{d}_A * (1-(x/16))$.
2) Node motion vectors $\mathbf{d}_A$ and $\mathbf{d}_C$: $\mathbf{d}(x,y) = \mathbf{d}_C * (y/16) + \mathbf{d}_A * (1-(y/16))$
3) Node motion vectors $\mathbf{d}_C$ and $\mathbf{d}_D$: $\mathbf{d}(x,y) = \mathbf{d}_D * (x/16) + \mathbf{d}_C * (1-(x/16))$
4) Node motion vectors $\mathbf{d}_B$ and $\mathbf{d}_D$: $\mathbf{d}(x,y) = \mathbf{d}_D * (y/16) + \mathbf{d}_B * (1-(y/16))$

where (x,y) denotes the distance of the pixel from the top left corner of the patch.

For the second configuration, constant interpolation is employed for both regions using the single node motion vector specified in the patterns. For instance, motion vectors in regions with node motion vector $\mathbf{d}_A$ are set to be equal to node motion vector $\mathbf{d}_A$.

V. Adaptive Interpolation Pattern Selection

In this section, we address the problem of how to choose the interpolation pattern adaptively for each patch given the set of node motion vectors from 2-D mesh motion estimation. To reduce the amount of computation, a patch is considered as a candidate for adaptive interpolation only if its prediction error is greater than a given threshold. This is done to identify the patches that the 2-D mesh motion estimator could not handle. For these candidate patches, the actual decision is based on a simplified rate distortion framework [8].

Given a candidate patch, we compute $D(i), i = 1,\ldots,32$ and D_b, where $D(i)$ denotes the resulting prediction error variance of the patch when interpolation pattern i is used and D_b denotes the resulting prediction error of the patch when the default bilinear interpolation is used. Next, we find the interpolation pattern that results in the minimum prediction error $i^* = \arg\min_{i=1,\ldots,32} D(i)$. Then, we will compare $D(i^*)$ with D_b. If D_{i^*} is greater than D_b, then we keep the default interpolation pattern. Otherwise, we encode the resulting residuals from using pattern i^* and the default bilinear interpolation pattern to obtain the number of bits needed for these residuals. Let R_{i^*} and R_b denote the bits used to represent the residual for interpolation pattern i^* and the default bilinear interpolation pattern respectively. Then, the decision rule is if $R_b > R_{i^*} + R_s$ then choose adaptive interpolation using pattern i^* else choose bilinear interpolation, where R_s is an estimate of the additional number of bits required to encode the adaptive interpolation pattern information.

Theoretically, we should use the actual number of bits needed to represent the adaptive interpolation pattern information.

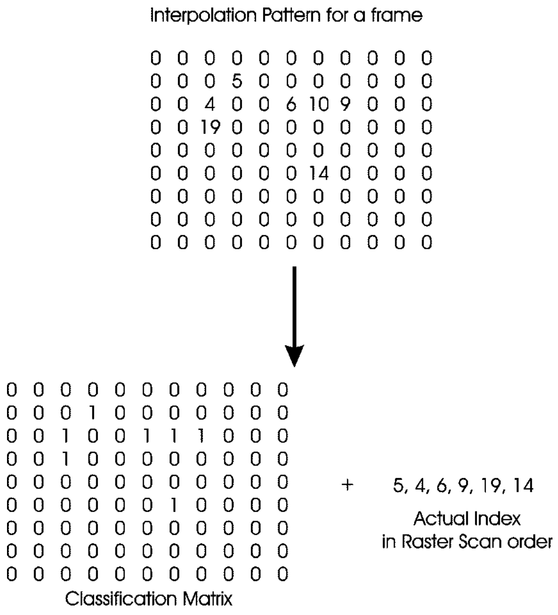


Fig. 3. Coding of interpolation patterns.

However, we do not know the incremental bit cost of choosing the adaptive pattern for a given patch due to the way that the interpolation pattern information will be encoded. Therefore, we will need to obtain an estimate of R_S experimentally. In our system, the interpolation pattern information is organized into a classification matrix followed by a string of indices to notify the decoder of the actual interpolation patterns introduced by the encoder, as shown in Fig. 3. The classification matrix $\mathbf{C}$ serves the purpose of indicating the locations where new interpolation patterns have been used and it is defined as in (3), shown at the bottom of the page, where the indices (i,j) denote that the patch is located on the jth patch row and ith patch column.

In addition, the decoder needs to be informed of which adaptive interpolation pattern is to be used for those patches where $\mathbf{C}(i,j) = 1$. Toward this end, we employed a fixed raster scan of the classification matrix in the encoder and the decoder to obtain an order of visit for the adaptive interpolated patches. The actual interpolation patterns associated with the nonzero components of C is sent in order of visit along the fixed scanning path using a 5-bit fixed length code, since there are only 32 patterns.

The classification matrix is compressed using a modified modified READ (MMR) code, which is a 2-D run-length coding scheme used in compressing fax documents [10]. Therefore, adding an interpolation pattern will require 5 bits to signal the actual pattern, plus the additional bits introduced by setting the corresponding entry in the classification matrix to one. Due to the predictive nature of MMR, thhis makes it difficult to analyze the number of bits for each symbol without explicitly coding all the interpolation pattern information. Thus, a conservative estimate was made for R_s that favors

$$\mathbf{C}(i,j) = \begin{cases} 1, & \text{patch } (i,j) \text{ is interpolated adaptively.} \\ 0, & \text{patch } (i,j) \text{ is interpolated using the default bilinear interpolation.} \end{cases} \tag{3}$$

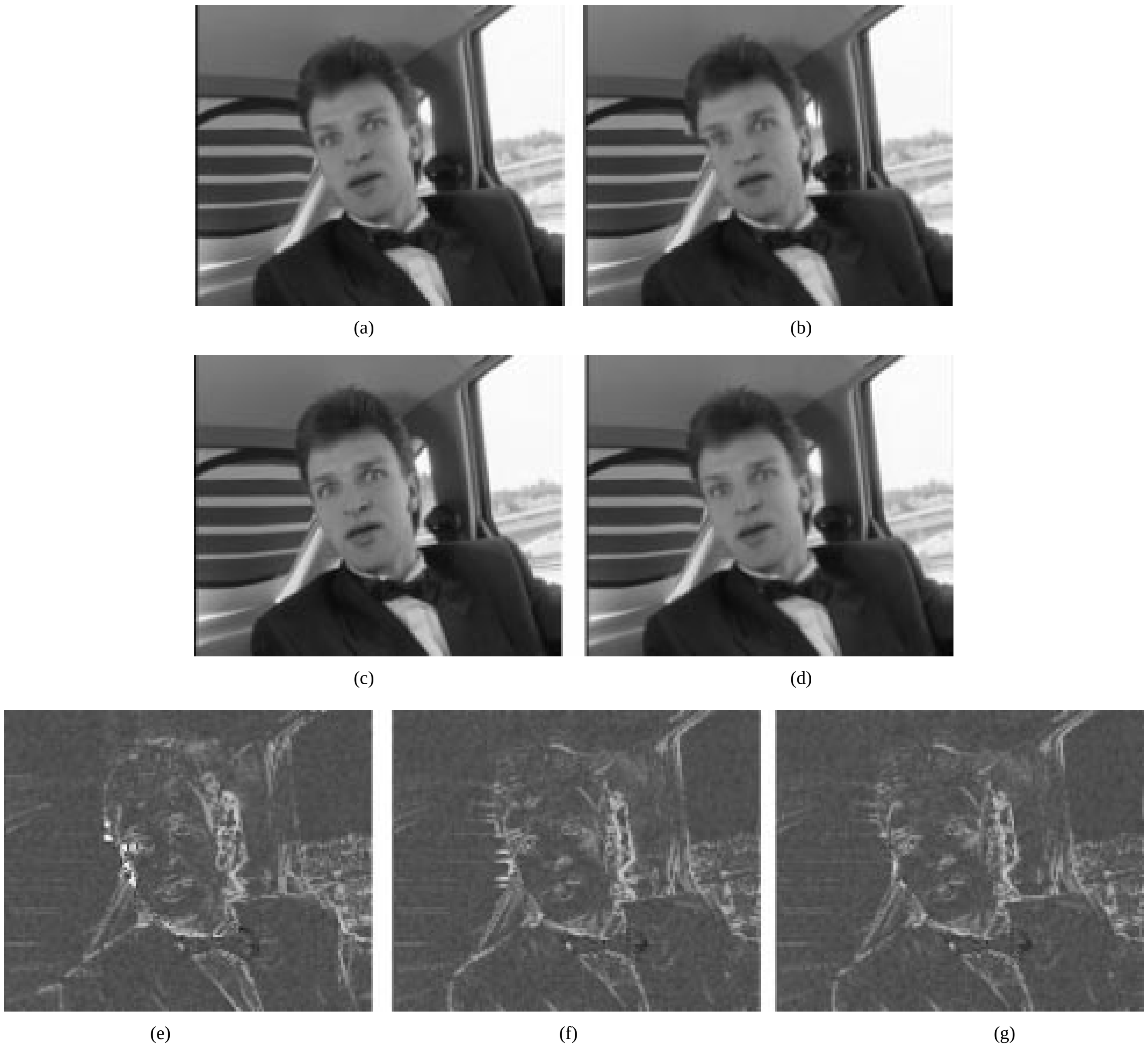


Fig. 4. Motion-compensated results (half-pixel accuracy used for all three methods). (a) Original frame 33. (b) BMA compensated frame 33. (c) 2-D mesh compensated frame 33. (d) Compensated frame 33 using proposed scheme. (e) Displaced frame difference for BMA. (f) Displaced frame difference for 2-D mesh. (g) Displaced frame difference for proposed scheme.

the selection of the default pattern. The estimate is chosen experimentally and we found that 5 bits is a good approximation to the average number of bits needed for a wide range of interpolation maps, which is how we decided on $R_s = 10$.

Once the set of interpolation patterns is selected, the set of node motion vectors is refined with respect to this new set of interpolation patterns. This is done because the selection of interpolation pattern is performed using the set of node motion vectors after standard 2-D mesh motion estimation that has been optimized for bilinear interpolation. This means that the set of node motion vectors used for adaptive interpolation with the new set of interpolation patterns may not be the best one. Thus, we propose to perform refinement similar to that used in the HMA to readjust the node motion vectors. The differences between the refinement process are that new interpolation patterns are used instead of the default bilinear interpolation.

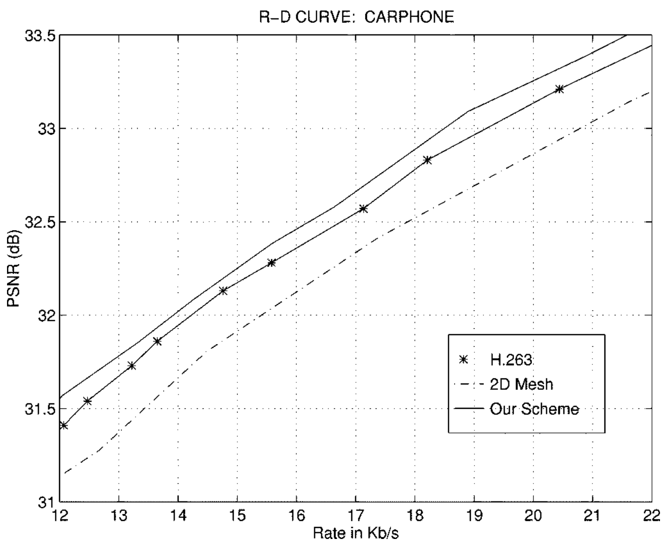
VI. Simulation Results

In this section, simulation results are presented to show the effectiveness of the proposed method. First, we demonstrate the advantage of the proposed method in motion compensated prediction over BMA and 2-D mesh. The original frames 30 and 33 of the sequence “Carphone” are used in this experiment. These two frames show the person’s head tilting and moving to the left over a striped background. Using these two frames, we performed motion estimation using the three methods to produce motion-compensated prediction images, and the results are shown in Fig. 4.

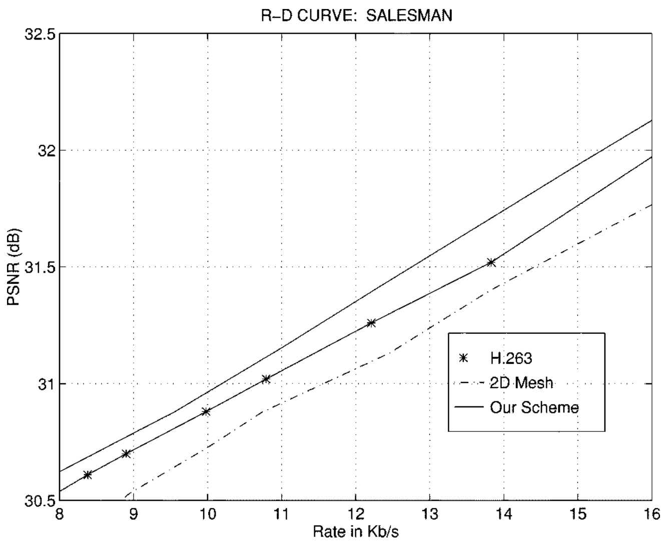
Let us focus our attention on the left part of the person’s face. As expected, the motion-compensated prediction from BMA shown in Fig. 4(b) is blocky, and moreover, part of the person’s face was chopped off. On the other hand, the motion-compensated prediction from the 2-D mesh shown in Fig. 4(c) is devoid of blocking artifacts, but part of the background on the left side of the face was warped. This illustrates the point that the 2-D mesh is not capable of handling multiple motions inside a single patch. The motion-compensated prediction produced by our proposed scheme is shown in Fig. 4(d). The displaced frame differences for the three methods are shown in Fig. 4(e) and (f). As we can see, our proposed scheme adds more ability to the 2-D mesh to handle motion boundaries inside the patch, and the background portion of the prediction image is no longer warped.

Since the proposed approach requires extra side information, a fair comparison with the 2-D mesh can only be obtained by comparing the rate-distortion performance. Toward this end, we constructed two coders to compare the encoded bit rate versus the distortion for the two schemes. We also made comparison to TMN5's implementation of H.263 with advanced prediction mode on.

For both the 2-D mesh and the proposed scheme, the motion-compensated residual signals are encoded using DCT and the node motion vectors are differentially encoded using the median predictor as specified in the H.263 standards. The simulations were conducted with the first 90 frames of the QCIF color sequences "Carphone" and "Salesman". The first frame was intra-coded using a fixed quantizer step size of 10 in all three coders. Then, we encoded every third frame of each sequence using a fixed quantizer step size *QP* to generate a single rate distortion point. We vary *QP* to obtain the rate distortion curves. The average percentage of the blocks that were chosen to be adaptively interpolated was around thirteen percent for the two test sequences. In Fig. 5, we showed the rate-distortion curves for the two sequences. As we can see, the proposed approach is better than the 2-D mesh, indicating that the bits spent coding the interpolation patterns is less than the bits saved in encoding the residual signals. Moreover, we see that the proposed approach is better than H.263.



(a)



(b)

Fig. 5. Rate-distortion curves. (a) Carphone. (b) Salesman.

VII. Conclusion

In this paper, we have presented an improvement to the 2-D backward mesh motion estimation. The proposed method performs a post-processing of the motion field generated by the 2-D mesh aimed at improved prediction for patches that contains motion discontinuities. The method is based on the observation that the assumption of globally smooth motion field is not valid for most natural scenes. This prompted the proposed approach to selectively break the smoothness constraint through changing the interpolation patterns in the motion field generation phase.

In order to balance the tradeoffs between rate and distortion, a fixed number of interpolation patterns was introduced and a criterion for selection was established. The interpolation selection criterion guarantees that the proposed approach will perform better or equal to 2-D mesh motion estimator. The amount of gains will be dependent on the contents of the actual video sequence. As the simulation results show, the 2-D mesh coder does not perform better than H.263 with advance prediction in terms of PSNR at a fixed rate. However, the proposed methods improves the prediction efficiency of the 2-D mesh and shows that it can perform better than H.263 with advance prediction with about a 0.1-dB gain.

References

[1] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proc. ICASSP*, 1991, pp. 2713–2716.

[2] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 339–356, June 1994.

[3] Y. Wang and O. Lee, "Use of two-dimensional deformable mesh structures for video coding, part I—The synthesis problem: Mesh-based function approximation and mapping," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 636–646, Dec. 1996.

[4] Y. Wang, O. Lee, and A. Vetro, "Use of two-dimensional deformable mesh structures for video coding, part II—The analysis problem and a region-based coder employing an active mesh representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 647–659, Dec. 1996.

[5] Y. Wang and J. Ostermann, "Evaluation of mesh-based motion estimation in H.263-like coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 243–252, June 1996.

[6] Y. Altunbasak and A. Tekalp, "Occlusion-adaptive content-based 2-D mesh design and tracking for object-based coding," *IEEE Trans. Image Processing*, vol. 6, pp. 1270–1280, Sept. 1997.

[7] M. T. Orchard, "Predictive motion-field segmentation for image sequence coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 54–70, Feb. 1994.
[8] G. M. Schuster and A. K. Katsaggelos, "A video compression scheme with optimal bit allocation between displacement vector field and displaced frame difference," in *Proc. ICASSP*, Atlanta, GA, May 1996.
[9] Telenor Research. [Online]. Available: ftp://bonde.nta.no/pub/tmn, 1996.
[10] K. McConnell, D. Bodson, and R. Schaphorst, *FAX: Digital Facsimile Technology and Applications*. Norwood, MA: Artech, 1992.
[11] M. Bierling, "Displacement estimation by hierarchical block-matching," *Proc. Visual Communication and Image Processing, SPIE*, vol. 1001, pp. 942–951, 1988.
[12] P. Ishwar and P. Moulin, "Switched control grid interpolation for motion compensated video coding," in *Proc. ICIP*, Santa Barbara, CA, Oct. 97.