Multiview Video Sequence Analysis, Compression, and Virtual Viewpoint Synthesis

Ru-Shang Wang and Yao Wang, *Senior Member, IEEE*

***Abstract*—This paper considers the problem of structure and motion estimation in multiview teleconferencing-type sequences and its application for video-sequence compression and intermediate-view generation. First, we introduce a new approach for structure estimation from a stereo pair acquired by two parallel cameras. It is based on a 2-D mesh representation of both views of the imaged scene and a parameterization of the structure information by the disparity between corresponding nodes in the image pair. Next, we describe a novel image alignment approach which can convert images captured using nonparallel cameras to coplanar-like images. This approach greatly eases the computational burden incurred by the nonparallel camera geometry, where one must consider both horizontal and vertical disparities. Finally, we present a coder for multiview sequences, which exploits the proposed alignment and structure estimation algorithm. By extracting the foreground objects and estimating the disparity field between a selected view and a reference view, the coder can compress the image pair very efficiently. In the meantime, by using the coded structure information, the decoder can generate virtual viewpoints between decoded views, which can be very helpful for telepresence applications.**

***Index Terms*—Convergent cameras, disparity estimation, intermediate view synthesis, non-coplanar image alignment, stereoscopic and multiview sequence coding.**

I. Introduction

STEREOSCOPIC, or in general multiview video, can provide more vivid and accurate information about the scene structure than monoview video. However, one major obstacle for using multiview video is the extremely large amount of data associated with them. To be able to transmit or store these sequences, substantial compression of the data must be accomplished. Instead of presenting to the users the acquired views as is, a desirable feature for any multiview system is that the displayed views can be adjusted either based on users requests or automatically, as required in virtual-reality applications. This requires that the compressed stream contain information about the 3-D structure of the imaged object. In contrast to the traditional waveform-based coding approach, a 3-D object-based coding system enables user to manipulate the imaged scene by using 3-D parameters embedded in the coded data. Such capability is the key to many emerging interactive multimedia applications.

Presently, the most mature technique for stereoscopic sequence compression is the block-based stereoscopic coding (BBSC) method defined in the MPEG-2 multiview profile [1], which is a straightforward extension of the main profile of MPEG-2 for monoview video. With this approach, the coder first compresses, say, the left view with a monoscopic video coding algorithm. To code the right view, each macroblock is predicted both from the left view using disparity-compensated prediction (DCP), and from the previous frame of the right view using motion-compensated prediction (MCP). Depending on which gives smaller prediction error, either or both are used and the prediction error is then coded. To make use of existing coders for monosequences, the disparity vector can be estimated in the same way as for motion estimation, i.e., assuming the disparity is blockwise constant and finding the best matching macroblock in the left view. One advantage of this approach is that it can be implemented using the temporal scalability mode of the MPEG-2 standard [2]. Although the above approach offers a readily available solution for compatible 3-D TV, the use of the block-based disparity model for disparity estimation and compensation limits its compression gain. Because the estimated disparity field is usually discontinuous and does not provide a one-to-one mapping between left and right views, it could be difficult and less precise to interpolate intermediate views, and the interpolated images usually have visible artifacts. In [3], several constraints on the disparity field derived from the stereo imaging geometry and the relation between disparity and motion have been exploited for more accurate disparity estimation.

Instead of deriving 2-D motion and disparity for performing MCP and DCP, a potentially more effective approach is to segment the imaged scene into separate objects and estimate the 3-D structure and motion of each object. Until now, such approaches have had limited success with arbitrary scenes as encountered in 3-D-TV applications. However, for applications such as videoconferencing, where the imaged scene usually contains a stationary background and one or more foreground objects (usually people), promising results have begun to emerge. In [4], certain feature points are first selected in one view. The 3-D positions and motion vectors of these points are then determined by disparity analysis between two views and 2-D motion estimation in individual views. The global motion parameters are then determined by a robust regression method known as least-median-of squares, which can suppress the impact of outlier points to the estimated parameters. In the presence of mul-

Manuscript received March 15, 1999; revised September 30, 1999. An earlier version of this work was presented at the IEEE Workshop on Multimedia Signal Processing, Dec. 1998, Los Angeles, CA, and also at the International Picture Coding Symposium, April 1999, Portland, OR. This paper was recommended by Guest Editor K. Ngan.

R.-S. Wang was with the Department of Electical Engineering, Polytechnic University, Brooklyn, NY 11201. He is now with Ezenia Inc., Burlington, MA 01803 USA (e-mail: rwang@videoserver.com).

Y. Wang is with the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY 11201 USA (e-mail: yao@vision.poly.edu).

Publisher Item Identifier S 1051-8215(00)02792-0.

tiple objects, this algorithm can be applied recursively, each time yielding the motion parameters corresponding to one object. In [5], a 3-D wireframe model was used to represent the foreground object structure. The 3-D positions of nodes are obtained through a disparity analysis between corresponding nodes in the left and right views. First, disparities and, hence, depths for selected nodes are estimated using a block-matching procedure. The depths at other nodes are then obtained using least-squares surface fitting. The motions of the nodes are estimated using a globally rigid plus locally deformable motion model. In [6], a multiocular system with multiple parallel cameras was used for disparity and occlusion estimation. In the case of three cameras, two views (left and right) were coded by either two separate MPEG-2 monoview coders or by the BBSC scheme described above, whereas the intermediate view is generated from the two decoded views based on the estimated disparity and occlusion information. The latter is obtained by a dynamic programming method [7].

The above schemes all assume the stereo or multiviews are obtained with parallel cameras, in which case only horizontal disparities exist. In the multiview environment, many cameras are used to capture different parts of the scene from different angles and the resulting videos have vertical disparities as well. This not only increases the computational complexity but also decreases the accuracy of disparity estimation. In [8], the authors considered the compression of multiviews obtained by convergent cameras. But they assume that camera parameters are known, and apply the epipolar constraint to limit their search along the epipolar lines, but allowing additional vertical disparity. In [9], image rectification was used to convert a non-coplanar image pair to virtual parallel planes and then a hierarchical block matching approach is employed for disparity estimation in the virtual planes. To perform rectification, the camera parameters must be known or estimated through camera calibration using the scheme developed by Tsai [10], which requires access to images of certain test patterns acquired by the same camera configuration. In practice, such images may not be available together with the sequences to be compressed. Furthermore, the camera configuration may change over time.

In our approach, we approximate each view of a 3-D scene by a 2-D mesh, where each element corresponds to a 3-D surface patch. Each node in the mesh corresponds to a 3-D position, which can be deduced from the disparity between this node and its corresponding node in another view. Instead of using a triangular mesh as in [4] and [5], by which the surface is represented by planar patches, we use a quadrilateral mesh so that the surface patch corresponding to each mesh element can be curved. Unlike most previous works, which assume the camera setup is parallel or the camera parameters are known, we do not require knowledge about the camera geometry. Instead, we perform image alignment through estimating the fundamental matrix that relates the corresponding epipolar lines in a given image pair, both corresponding to the same epipolar plane in 3-D. The aligned images are such that all the epipolar lines are horizontal. To estimate horizontal disparities at nodal points in the aligned images, we minimize a matching error that takes into account of the fact that the disparity field within each element is bilinearly interpolated from nodal disparities. This is in contrast to prior approaches, which assume the disparity is constant in a block surrounding each node. A hierarchical exhaustive search algorithm and a fast search algorithm are developed.

We have integrated the above alignment scheme and mesh-based disparity estimation method in a multiview coding system. It chooses a center view as the reference and codes it independently. Each of the other views is coded with respect to the reference view using disparity/motion-compensated prediction. The input views are first aligned so that their epipolar lines become horizontal. Mesh-based disparity estimation is then applied to the aligned images. Disparity compensated prediction is then accomplished based on nodal displacements mapped back to the original image coordinate. As part of the coder, we also developed a scheme for foreground object extraction and apply the above procedures to the foreground objects only. For three test sequences, two stereoscopic sequences and one with three views, we have achieved better compression performances than the BBSC coder, especially at relatively low bit rates. In addition to enabling good compression performance, the proposed mesh-based disparity representation also facilitates the synthesis of intermediate views easily, which is becoming increasingly important for virtual reality applications.

In the remainder of this paper, we first describe the proposed mesh-based disparity estimation approach and the method for disparity compensated prediction and view synthesis, all for coplanar camera setup. We then present our image alignment scheme and describe how to perform prediction and view synthesis in a convergent camera system. Finally we introduce the proposed coding system and show simulation results.

II. Disparity Estimation for Coplanar Image Pairs

As described in Section I, we approximate the surface of an imaged object by a mesh structure, and determine the nodal disparities between two projected views by minimizing the matching errors over corresponding elements. Although the general formulation applies to an arbitrary camera configuration, we focus on the special case when the given image pair is parallel for which special simplifications are possible. In this section, we first describe the mesh-based representation for the object surface and the associated disparity estimation problem. We then formulate the disparity estimation problem as a minimization problem, and introduce two algorithms for estimating nodal disparities: a hierarchical exhaustive search scheme and a fast algorithm. Finally, we describe how to perform DCP and view synthesis based on estimated nodal disparities.

A. Mesh-Based Disparity Representation

The proposed disparity estimation method is based on a representation of the imaged object surface by a 3-D mesh, where each element corresponds to a small surface patch, as illustrated in Fig. 1. In this illustration and in our simulations, we have used a quadrilateral mesh structure. But triangular meshes can also be used. When imaged from different views, this 3-D mesh leads to different projected 2-D meshes. For disparity estimation, we do not know the nodal positions in the 3-D mesh. Rather, we start with a 2-D mesh in a reference view, estimate the corresponding positions of nodes in this mesh in the other view, and

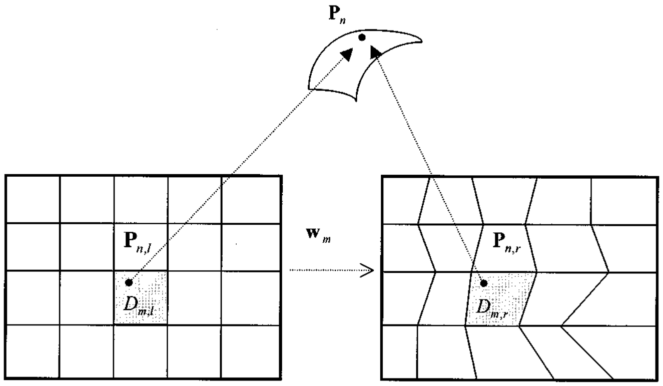


Fig. 1. Illustration of the 3-D mesh structure and mapping functions between corresponding elements in a stereo pair.

finally reconstruct the 3-D positions of all the nodes based on the estimated displacements between corresponding nodes, which are called nodal disparities. For the 3-D mesh to approximate the object surface well, ideally the 2-D mesh in the reference view should be adapted to the object shape so that the surface patch corresponding to each element is smooth. This however requires the knowledge of the depth distribution of the object surface, which is what we try to estimate. Ideally, an iterative scheme is required. In the first iteration, a regular mesh or a mesh that is adapted to edges in the reference view is used to yield an initial depth estimation. In the next iteration, the reference mesh is refined, and the depth distribution is re-estimated. In our implementation, for reduced computations, we simply use a regular mesh for the reference view. The following notations are used to describe a 3-D mesh and its projected left and right 2-D mesh at time t: N and M represent the numbers of nodes and elements in the mesh; N_m contains the indices of nodes defining element m, M_n includes the indices of elements attached to node n; $\boldsymbol{p}_n = [x_n, y_n, z_n]$, $n = 1, 2, \cdots, N$, represent 3-D nodal positions, which constitute the structure parameters; $\boldsymbol{P}_{n,v} = [X_{n,v}, Y_{n,v}]$, $v = l, r$, represent nodal positions projected in view v; $\boldsymbol{D}_n = \boldsymbol{P}_{n,l} - \boldsymbol{P}_{n,r}$ describe 2-D nodal displacements between left and right views; $D_{m,v}$, $m = 1, 2, \cdots, M$, represent elements in view v; $\boldsymbol{w}_m(\boldsymbol{P})$ represent the 2-D mapping function of element m, which specifies the corresponding point in the right view given a point P in $D_{m,l}$. The mapping function is related to the nodal displacements by

$$\boldsymbol{w}_m(\boldsymbol{P}) = \boldsymbol{P} + \sum_{n \in N_m} \phi_{m,n}(\boldsymbol{P})\boldsymbol{D}_n, \qquad \boldsymbol{P} \in D_{m,l} \tag{1}$$

where the interpolation function (also known as the shape function) $\phi_{m,n}(\boldsymbol{P})$ describes the weight of node n in interpolating the displacement at $\boldsymbol{P}$ in element m (cf. Fig. 2).

B. Disparity Estimation for Coplanar Case: A Hierarchical Exhaustive Search Algorithm and a Fast Algorithm

Given a pair of stereo views at frame time t, and assuming that a 2-D mesh has been established in, say, the left view, i.e., $\boldsymbol{P}_{n,l}$ are known, the disparity estimation problem is to estimate $\boldsymbol{D}_n$, the nodal displacement between left and right views. When the two views are parallel, only horizontal disparity exists so that the 2-D variable $\boldsymbol{D}_n$becomes a 1-D variable, to be denoted by D_n. Similarly, $\boldsymbol{w}_m(\boldsymbol{P})$ becomes a 1-D function, to be denoted by $w_m(\boldsymbol{P})$. Let $f^l(\boldsymbol{P}, t)$ and $f^r(\boldsymbol{P}, t)$ represent the left and right image views. To estimate D_n, we minimize the following DCP error:

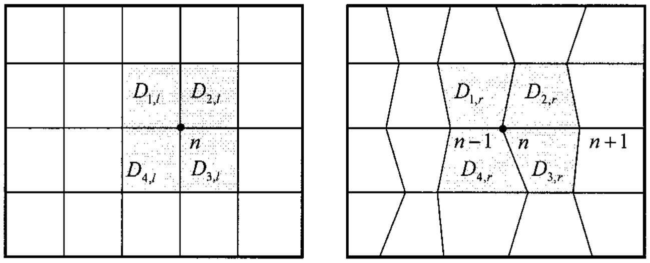


Fig. 2. Node n and its associated four mesh elements and eight adjacent nodes.

$$E_{\text{DCP}} = \sum_{m \in M} \int_{\boldsymbol{P} \in D_{m,l}} e\left(f^l(\boldsymbol{P}, t), f^r(w_m(\boldsymbol{P}), t)\right) d\boldsymbol{P}. \tag{2}$$

In (2), $e(f_1, f_2)$ represents an error measure between two image values. One can use either the magnitude or magnitude square of the difference $f_1 - f_2$.

Finding the optimal set of D_n that minimizes the error in (2) is a nonquadratic optimization problem, and no closed-form solution exists. We have developed two iterative schemes: a hierarchical exhaustive search method and a fast search algorithm. Note that with the quadrilateral mesh structure, each node is surrounded by four elements and eight nodes, as shown in Fig. 2. Given the positions of its surrounding eight nodes, a locally optimal criterion for determining the position of the center node is to minimize the DCP error accumulated over the four elements. This criterion forms the basis for both estimation algorithms. In both cases, we update one node at a time, while fixing the other nodes. Starting from the top-left node, we update each node successively, until reaching the bottom-right node. Then the process repeats from the top-left node. Each stage of updating all the nodes is considered as one iteration. The two search schemes differ in their ways for determining the update for a given node.

With the hierarchical exhaustive search method, we produce multiresolution representations of both the left and right views. The search at the higher resolution is initialized by the solutions found at the previous lower resolution. At each resolution, we search one node at a time, while fixing the neighboring nodes at their positions determined previously. Specifically, we move node n in the horizontal direction within the range between nodes $n - 1$ and $n + 1$. For each candidate position, we compute the DCP error over the four mesh elements attached to node n and find the position that yields the minimal error. In our implementation, the number of resolutions is 2, the mesh element size is 32 × 16 in each resolution, and the search step size is 1 pixel. In each resolution, three iterations are used. Because the number of nodes and the search range are large, this exhaustive search algorithm takes quite significant amount of time.

The fast algorithm uses the gradient information to update each node and can give good results with proper initialization. Using the square error for $e(f_1, f_2)$ in (2), the minimum is achieved when

$$\frac{\partial E_{\mathrm{DCP}}}{\partial D_n} = -2 \sum_{m \in M_n} \int_{\boldsymbol{P} \in D_{m,l}} \left[f^l(\boldsymbol{P}, t) - f^r(w_m(\boldsymbol{P}), t) \right] \cdot g_r(w_m(\boldsymbol{P})) \phi_{m,n}(\boldsymbol{P}) d\boldsymbol{P} = 0 \quad (3)$$

where $g_r(\boldsymbol{P}) = (\partial f^r / \partial X)|_{\boldsymbol{P}}$.

Let D_n^i represent the solution at ith iteration, and denote $D_n^{i+1} = D_n^i + \Delta D_n$, then by using the first order Taylor expansion, we obtain

$$f^r\left(w_m^{i+1}(\boldsymbol{P}), t\right) = f^r\left(w_m^i(\boldsymbol{P}), t\right) + g_r\left(w_m^i(\boldsymbol{P})\right) \sum_{k \in N_m} \phi_{m,k}(\boldsymbol{P}) \Delta D_k. \quad (4)$$

Substituting the above approximation into (3), we arrive at a system of linear equations in terms of ΔD_k, $k \in N_m$

$$\sum_{m \in M_n} \sum_{k \in N_m} \left(\int_{\boldsymbol{P} \in D_{m,l}} \phi_{m,n}(\boldsymbol{P}) \phi_{m,k}(\boldsymbol{P}) g_r^2\left(w_m^i(\boldsymbol{P})\right) d\boldsymbol{P} \right) \Delta D_k = \sum_{m \in M_n} \int_{\boldsymbol{P} \in D_{m,l}} e_m^i(\boldsymbol{P}) \phi_{m,n}(\boldsymbol{P}) g_r\left(w_m^i(\boldsymbol{P})\right) d\boldsymbol{P} \quad (5)$$

where

$$e_m^i(\boldsymbol{P}) = f^l(\boldsymbol{P}, t) - f^r\left(w_m^i(\boldsymbol{P}), t\right).$$

Ideally, we need to determine ΔD_n for all nodes simultaneously by solving (5). To further reduce the computation, we solve ΔD_n for one node at a time, while fixing the neighboring nodes at their positions determined previously. The equation for ΔD_n is simply (cf. Fig. 2)

$$\sum_{m=1}^{4} \left[\int_{\boldsymbol{P} \in D_{m,l}} \phi_{m,n}^2(\boldsymbol{P}) g_r^2\left(w_m^i(\boldsymbol{P})\right) d\boldsymbol{P} \right] \Delta D_n = \sum_{m=1}^{4} \int_{\boldsymbol{P} \in D_{m,l}} e_m^i(\boldsymbol{P}) \phi_{m,n}(\boldsymbol{P}) g_r\left(w_m^i(\boldsymbol{P})\right) d\boldsymbol{P}. \quad (6)$$

Within each iteration, the above equation is solved for all nodes successively. The position of a node is immediately updated based on the solution of (6) before proceeding to the next node. The update given by (6) for a particular node may lead to flip-over of adjacent nodes. If this happens, we will move the node by a smaller amount (the original amount attenuated by 2/3). If the resulting position still causes a flip-over, we will discard the update and retain the original nodal position. We also calculate the DCP errors associated with the old position and the new position. If the new error is larger, we will keep the old position. To guarantee the above fast algorithm to converge to the correct solution, the hierarchical exhaustive searching method is employed for an initial frame to derive a good initial solution. For the following frames the solution obtained from the previous frame is used as the initial solution for the new frame. In our implementation, a sequence is divided into groups of pictures (GOP) of length 15 frames, and the hierarchical search algorithm is employed at the first and eighth frame of each GOP. With this parameter setting, three iterations of the fast algorithm are sufficient for arriving at a result similar to that of the exhaustive search algorithm. Note that the nodal displacements found by the fast algorithm are nonintegers in general. We quantize them to the nearest integers at the end of the final iteration.

TABLE I
PROCESSING TIME REQUIRED BY THE HIERARCHICAL EXHAUSTIVE SEARCH METHOD AND THE FAST ALGORITHM

Sequence/Method	Hierarchical Exhaustive Search (Two layers each with 3 iterations)	Fast Algorithm (3 Iterations)
MAN (384x192)	102 seconds	8 seconds
ANNE (720x288)	231 seconds	17 seconds
GWEN (720x288)	97 seconds (Three layers, 16x16)	25 seconds

Processing time is defined as the CPU time required for 1 image pair on a Pentium Pro 200–MHz processor running Linux.

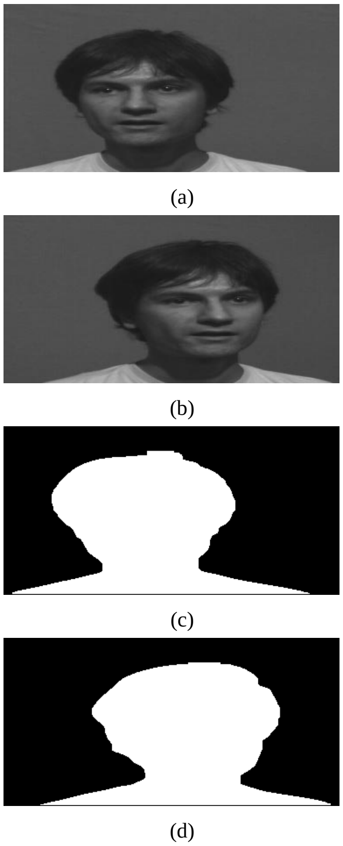
Table I compares the computation times required by the hierarchical exhaustive search method and the fast algorithm for different sized images. For "MAN" and "ANNE," a mesh element size of 32 × 16 is used, and the fast algorithm reduces the processing time by at least a factor of 12. For the sequence "GWEN," we also use 32 × 16 elements with the fast algorithm. But for the hierarchical exhaustive search we use mesh element size of 16 × 16 in each layer to speed up the process. The final results are then resampled to produce nodal disparities on a mesh with element size 32 × 16.

In general, solutions obtained by both schemes will depend on the ordering of the nodes within each iteration. We have tried two ordering methods: one uses the conventional raster order, another uses an interleaved order. Our simulations have shown that they lead to similar performance. Therefore, we have chosen to use the simple raster ordering.

C. Global Disparity Estimation

In most videoconferencing sequences, the foreground object (the human head and shoulder) dominates most of the image. When the scene was taken by a parallel camera setup with a large baseline, there is a global disparity associated with the object. Fig. 3 shows the left and right images and the extracted foreground maps for a selected field in test sequence "MAN," which was taken with a coplanar (parallel) camera setup with 50-cm baseline and about 2 m between the person and the cameras. The image size is 384 × 192.[1] We can see that the images of the main object in the left and right views are separated by a large distance. We refer to this distance as *global disparity*. For the case where the background is homogenous, we estimate the global disparity by minimizing the absolute error between the left image and the shifted right image. If the sequence contains a complicated background, then taking the absolute error over the entire image does not work well. In this case, we compute

[1] This sequence, as well as the other two sequences introduced later, are in the interlaced format with 60 fields/s. In our simulations, we treat each field as a progressive frame. The actual field size of sequence "MAN" is 386 × 193, but in order to use the MPEG standard for coding, we reduced the size to 384 × 192.



(a)

(b)

(c)

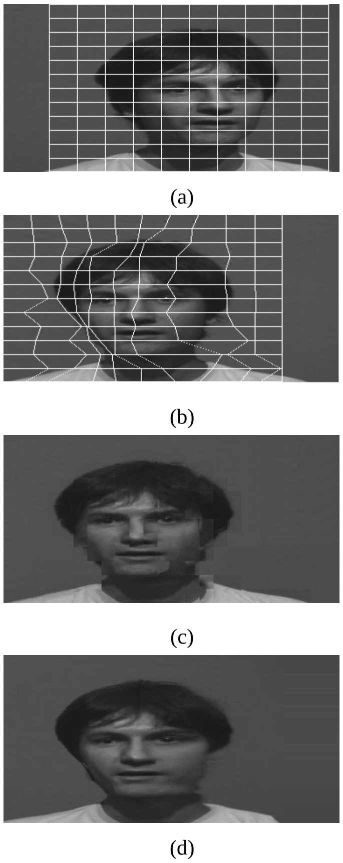
(d)

Fig. 3. Test sequence “MAN”. Displayed images are for field 0. (a) Original right image. (b) Original left image. (c) Right foreground map. (d) Left foreground map.

the error only over pixels in the foreground region. Fig. 4(a) and (b) show the regular mesh overlaid on the left image with global disparity compensation and the mesh found for the right image using the nodal disparity estimation scheme described in Section II-B. For this example, the global disparity is 52 pixels. Estimation of the global disparity initially not only helps to improve the accuracy of nodal disparity estimation, but also speeds up the process. Note that for most sequences obtained by convergent camera systems, there is no need for global disparity compensation, because all the cameras are focusing on the object already.

D. Disparity Compensated Prediction and Intermediate View Generation

In this section, we describe the process for DCP and virtual viewpoint synthesis, given estimated nodal disparities. We assume that only horizontal disparity is present. In this case, each rectangle in the predicted view corresponds to a trapezoid in the reference view. To perform DCP, for each pixel in the predicted view, we need to find its corresponding pixel in the reference view. Although this can be accomplished by using bilinear mapping between every two corresponding elements, a simpler approach is to first find the horizontal disparity values for all the points on the vertical grid lines in the mesh of the reference view by applying piecewise linear interpolation vertically within each element, and then applying piecewise linear interpolation within each element horizontally, line by line.



(a)

(b)

(c)

(d)

Fig. 4. Disparity compensated prediction for “MAN.” Displayed images are for field 0. (a) Regular mesh overlaid on the left image with global disparity compensation. (b) Corresponding mesh on the right image determined by the proposed mesh-based disparity estimation scheme. (c) Predicted right view using block matching (16 × 16 block, search range ±100 pixels; PSNR is 32.03 dB over the foreground). (d) Predicted right view based on nodal correspondences (PSNR is 27.48 dB over the foreground).

Fig. 4(c) and (d) show the predicted right image from the left one based on the estimated disparity field. Fig. 4(c) is obtained by a block matching algorithm using a block size of 16 × 16 and a search range of ±100 pixels and Fig. 4(d) is obtained by the proposed disparity estimation method with initial global disparity estimation. The dB values given in the figure captions are the peak signal-to-noise ratios (PSNR) of predicted images measured over the foreground region. Although block matching leads to a higher PSNR, the proposed estimation scheme yields visually more accurate prediction. Fig. 5(a)–(d) show the results for another test sequence “ANNE.” This sequence was taken by a convergent camera setup and then artificially converted to the coplanar geometry. It is also an interlaced sequence with a field size of 720 × 288. The camera baseline is 50 cm and the object was about 2 m from the cameras. The original left and right images (resized to 360 × 288) are shown later in Fig. 8. In this image pair, a large portion of the right side of the face is not visible from the left view and therefore is not predictable. The block matching algorithms found completely wrong matching blocks whereas the proposed scheme, with the help of the foreground map, assigned a disparity value of zero to these regions. There are also very severe artifacts in the background and around the object boundary by the block-matching

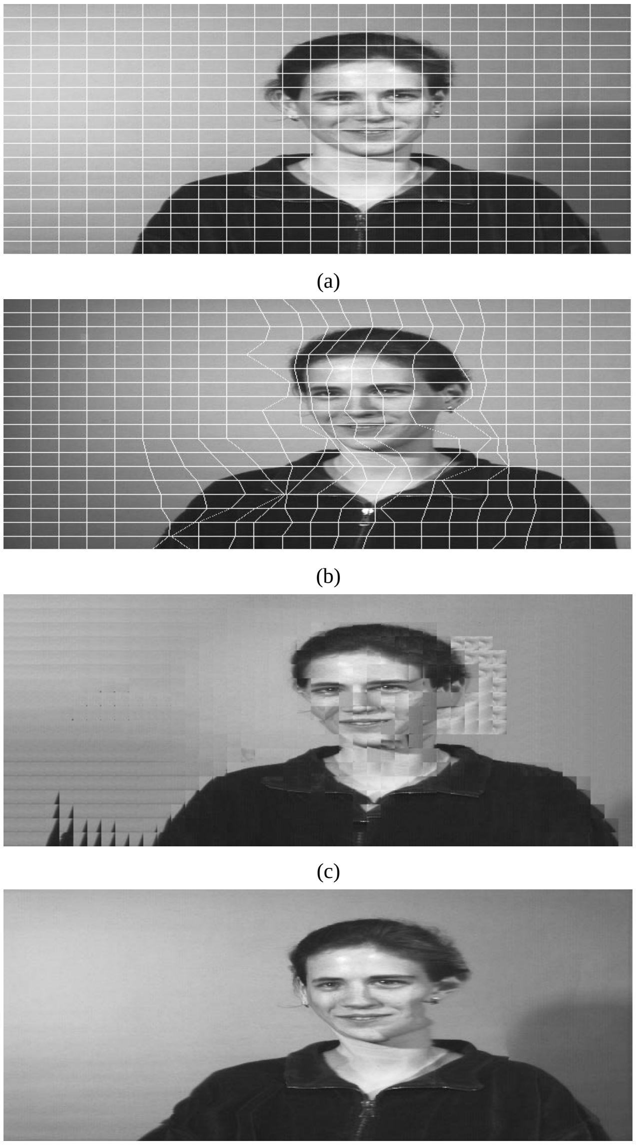


Fig. 5. Disparity-compensated prediction for "ANNE." Displayed images are for field 0. (a) Regular mesh overlaid on the left image. (b) Corresponding mesh on the right image determined by the proposed mesh-based disparity estimation scheme. (c) Predicted right view using block matching (16 × 16 block, search range ±100 pixels; PSNR is 27.25 dB over the foreground). (d) Predicted right view based on nodal correspondences shown between Fig. 5(a)and (b) (PSNR is 23.76 dB over the foreground).

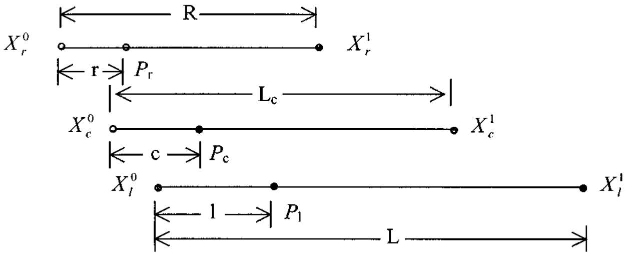


Fig. 6. Illustration of disparity compensated prediction and intermediate view synthesis along each scan line, when only horizontal disparities exist.

algorithm, which is caused by the very different shadow patterns captured in the left and right views.

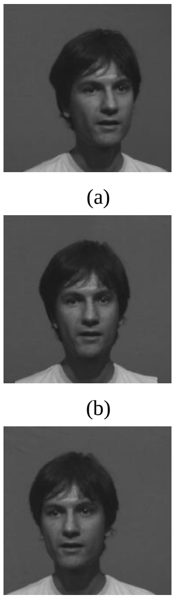


Fig. 7. Intermediate view synthesis for "MAN." Displayed images are for field 0. (a) Left view. (b) Synthesized center view. (c) Right view.

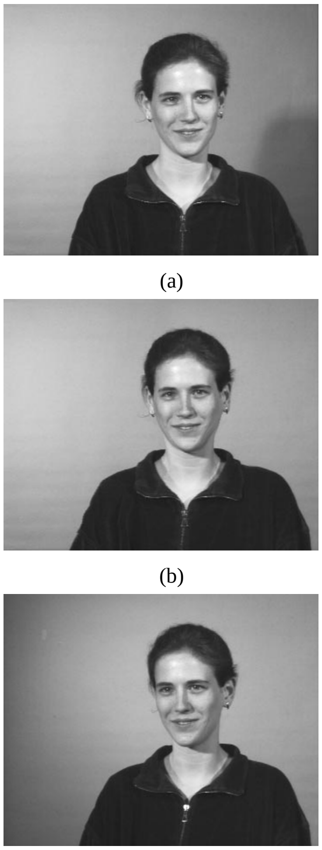


Fig. 8. Intermediate view synthesis for "ANNE." Displayed images are for field 0. (a) Left view. (b) Synthesized center view. (c) Right view.

For intermediate view generation, we need to find, for a given pixel to be generated in the synthesized view, its corresponding positions in the left and right views. First for each pair of corresponding nodes in the left and right views, we find its corresponding position in the synthesized view based on the distance

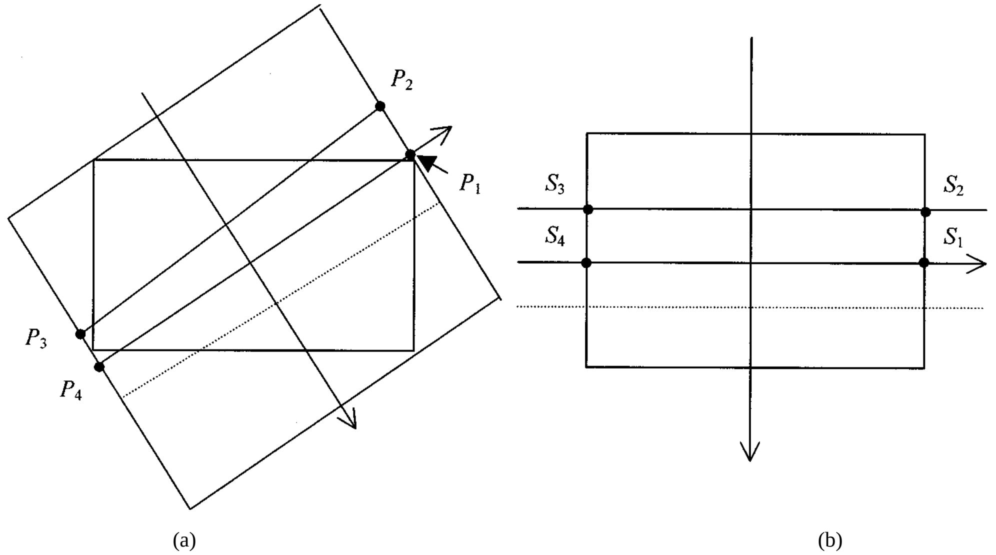


Fig. 9. Illustration of coordinates warping for image alignment. (a) A pair of epipolar lines in the original image coordinate. (b) Warped horizontal epipolar lines in the aligned image coordinate.

of the synthesized view to the left and right views, respectively. The result is quantized to the nearest integer. From each group of four nodes forming a quadrangle, we determine the pixels on the left and right borders, also quantized to the nearest integer. Then we perform view synthesis within each element line-by-line. Consider the example given in Fig. 6, which shows two pairs of corresponding nodes in the left and right images, X_l^0 and X_r^0, X_l^1 and X_r^1, along a horizontal line. First, we determine the corresponding end points in the center view to be synthesized, X_c^0 and X_c^1. In this case, $X_c^{0,1} = (1-k)X_l^{0,1} + kX_r^{0,1}$ where $k = B'/B$ (B' is the baseline distance from left camera to the virtual camera and B the baseline distance between left and right cameras). For any point P_c between X_c^0 and X_c^1, we find its corresponding positions P_r and P_l according to

$$\frac{P_c - X_c^0}{L_c} = \frac{P_l - X_l^0}{L} = \frac{P_r - X_r^0}{R}. \tag{7}$$

In general, image value at P_c can be synthesized from a weighted average of image values at P_r and P_l, using $f_c(P_c) = w_l \cdot f_l(P_l) + w_r \cdot f_r(P_r)$ with $w_l + w_r = 1$. As described in [13], the "head and shoulder" foreground object in the scene is convex. In this case, the left-hand side of the person's head can be found with more accuracy in the left view, and vice versa, the right-hand side should better be taken from the right-view image. Motivated by this observation, we find a vertical center axis for the foreground object and split the foreground region into three sections. If P_c is in the center section, it is synthesized from both images, with w_l being proportional to the distance of P_c to the center axis. If P_c is on the left side, then $w_l = 1, w_r = 0$. Finally, if it is on the right side, then $w_l = 0$, $w_r = 1$.

The synthesized center views for "MAN" and "ANNE" (resized to 192 × 192 and 360 × 288) are shown in Figs. 7 and 8, respectively. Both images were synthesized from original image pairs. We can see that correct eye contact is produced by the synthesized view. This is very important in televideoconferencing applications, where the synthesized virtual view can give the illusion of personal contact.

III. Alignment of Images Obtained by Convergent Cameras

In Section II, we assumed that a given image pair is obtained by two coplanar cameras. When the given image pair is from two convergent cameras, the mesh-based disparity estimation algorithm described in Section II-B cannot be applied directly. In [9], images obtained by convergent cameras are rectified to produce two synthesized images, which approximate the images that would have been acquired by two parallel cameras. A limitation of this approach is that it requires accurate knowledge of the intrinsic and extrinsic camera parameters. When images of special calibration patterns are available, the technique proposed by Tsai [10] can be employed to estimate the camera parameters. However such images are usually not available together with the actual video sequence. Also, the camera configurations may not stay exactly the same over time. To overcome the above problems, we developed an image-alignment scheme, which depends only on detectable features in the given image pair. We do not reproject the image pair such that they could be treated as if they were captured from a parallel camera setup. Instead, we find two epipolar lines on the original two images which belong to the same epipolar plane, and warp the image pair such that all the epipolar lines will be in the horizontal direction. The aligned images are then used as input to the disparity estimation algorithm described in Section II. A remarkable feature of this approach is that, to determine the epipolar lines in a noncoplanar image pair, we do not need to know the actual camera parameters. Rather, we only need to know the fundamental matrix, which can be estimated from a set of corresponding features between the given two images. In this section, we first describe the algorithm for estimating the fundamental matrix, which is adapted from [14]. We then describe our image alignment scheme.

A. *Estimation of the Fundamental Matrix*

The epipolar constraint is well known in stereovision: for each point $\boldsymbol{m}$ in the first image, its corresponding point $\boldsymbol{m}'$ lies on its epipolar line I'_m in the second image. Let $\boldsymbol{m}$ and $\boldsymbol{m}'$ be represented in the homogeneous coordinate, i.e., $\boldsymbol{m} = [u \quad v \quad 1]^T$ and $\boldsymbol{m}' = [u' \quad v' \quad 1]^T$. Under the epipolar constraint, the following equation must be satisfied:

$$\boldsymbol{m}^T\boldsymbol{F}\boldsymbol{m}' = 0 \tag{8}$$

where the 3 × 3 matrix $\boldsymbol{F}$ is known as *fundamental matrix*.

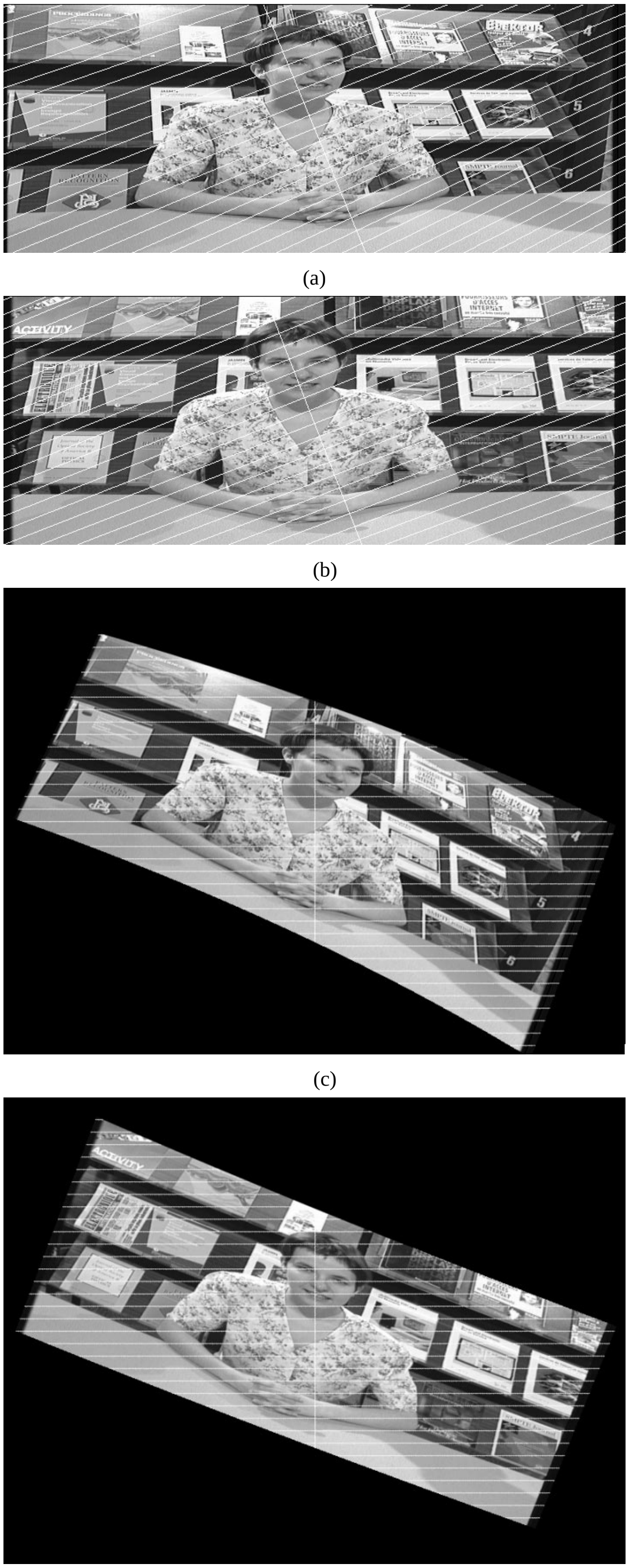
To estimate the fundamental matrix for a given stereo pair, we employed the scheme presented in [14]. It consists of three steps: first, a corner detector [15] is applied to the image pair to select certain feature points from each image, then a correlation measure is applied to identify corresponding feature points between the two images, lastly the least median of squares (LMedS) method [16] is used to obtain the fundamental matrix associated with this image pair.

B. *Image Alignment*

Given the fundamental matrix for an image pair, we find a set of epipolar line pairs each belonging to the same epipolar plane and then warp each image so that all the epipolar lines on the image are mapped to equally spaced horizontal lines. Take a left-top image pair as an example, the alignment process proceeds as follows.

1) Find the fundamental matrix for the image pair, using the algorithm described in Section III-A.
2) From the center point $C(c_x, c_y)$ in the left image, find line l'_c in the top image. Then find a point $C'(c'_x, c'_y)$ in the epipolar line l'_c with $c'_x = c_x$. Next, for the point $C'(c'_x, c'_y)$, find the line $l_{c'}$ in the left image.
3) Find two lines that are perpendicular to l'_c and $l_{c'}$, respectively, and label them l'_{cp} and $l_{c'p}$. These two lines are used to search for additional epipolar lines.
4) Use a predefined line distance y_d, find another point along $l_{c'p}$. Use the same process as in steps 2 and 3 to find a set of epipolar line pairs. Repeat this process to find all the epipolar lines associated with points on $l_{c'p}$ with distance y_d apart.
5) Every two adjacent epipolar lines, for example, l'_c and the line above it in the top image, and $l_{c'}$ and the line above it on the left image, will form a set of points $[p_1 \quad p_2 \quad p_3 \quad p_4]$ which is shown in Fig. 9(a). To make the epipolar lines horizontal, we warp pixels in the quadrilateral $[p_1 \quad p_2 \quad p_3 \quad p_4]$ to a rectangle $[s_1 \quad s_2 \quad s_3 \quad s_4]$ in the aligned image [Fig. 9(b)]. This is done by bilinear interpolation for both left and top images.
6) Repeat step 5 until all the epipolar lines inside the images are completed.

Fig. 10 shows the original left and top images and the aligned image pair for field 0 of the test sequence "GWEN." This sequence was prepared by the Centre Commun d'Etudes de Télédiffusion et Télécommunications (CCETT) using a trinocular camera system. The baseline between left and right camera is 80 cm and the top camera is 30 cm above the baseline. The sequence is interlaced with a field size of 720 × 288. All three cameras are focusing on the human object about 2-m away. The lines overlaid on top of the images are the corresponding epipolar lines. We can see that all the epipolar lines on the aligned images are in the horizontal direction. Furthermore, for



(a)

(b)

(c)

(d)

Fig. 10. Image alignment for "GWEN." Lines overlaid on images are epipolar lines. (a) Original left view. (b) Original top view. (c) Aligned left view. (d) Aligned top view.

any point on an epipolar line in the left image, its matching point lies on its corresponding epipolar line on the top image, both in the original image pair or the aligned image pair.

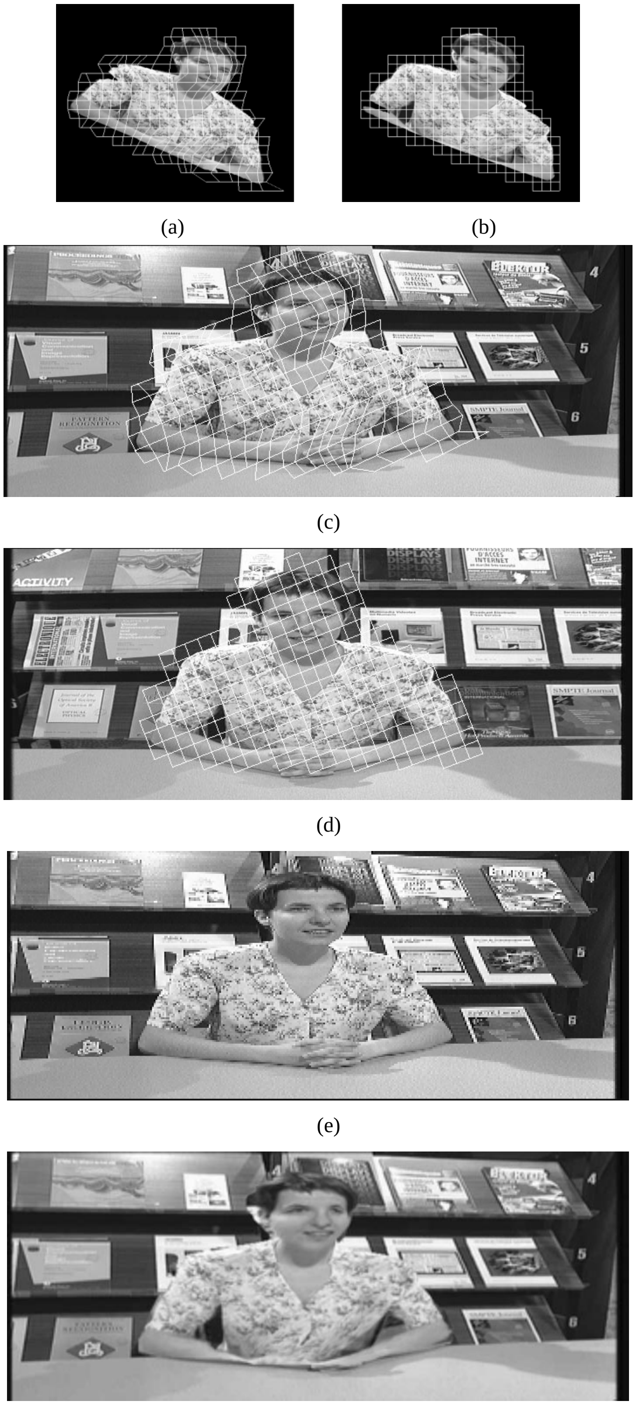
C. Disparity Compensated Prediction and Intermediate View Generation on Nonparallel Images

The process described in Section III-B will provide the aligned image pair as the input to our mesh based disparity compensated prediction algorithm described in Section II. The resulting meshes (only applied to the foreground region) for field 1 of sequence "GWEN" are shown in Fig. 11(a) and (b). To predict the left view from the top view, a straight forward approach is to first obtain the predicted view in the aligned coordinate, based on the estimated nodal disparities, and then map the predicted view back to the original coordinate. We have found, however, when the alignment and realignment process is applied to the same image in tandem, a significant error would result, with a PSNR of around 30 dB only. In order to avoid degradation due to resampling, we first remap the nodal points in the aligned images back to the original domain by an inverse alignment (realignment) process. We then use the inverse bilinear mapping method described in [12], which enables the warping from one quadrangle to another quadrangle, to predict the left image from the top image. Fig. 11(c) and (d) show the mesh structures mapped from those in Fig. 11(a) and (b). Fig. 11(e) and (f) show the original left view and that predicted from the top view.

To synthesize an intermediate view, say the center view between the top and right views, we can first determine the nodal positions in the center view by the average of corresponding nodes in the top and right views. Then, for each quadrilateral element in the center view, we can take a weighted average of the two synthesized versions obtained by warping from its corresponding elements in the top and left images. This process, however, requires two inverse bilinear mapping. As the requirement for the quality of the synthesized view is not very stringent, we can also perform synthesis in the aligned coordinate, using the approach previously described in Section II-D, and then map it back to the original coordinate. Fig. 12 shows the synthesized center view (resized to 360 × 288) between top and right view using this second method.

IV. Coding Schemes for Multiview and Stereoscopic Sequences

Given two or more video sequences captured from different view locations or angles, the underlying idea of the proposed coder is to use one view as reference, compress this view using a monocular sequence compression technique (MPEG-2 main profile in our case), and then code each of the remaining views with respect to the reference view by both disparity and motion compensation. For stereoscopic sequences, either the left or right view can be used as reference. For multiview sequences, the reference view should be chosen such that it shares the most common regions with each of the remaining views. Given three views, left, center, and right, a natural choice is to select the center view. For data with more than three views, it may be necessary to use more than one reference views. We have not, however, experimented with this case. In the following, we first give an overview of the entire coding scheme, and then describe some components in more detail. Finally, we present simulation results.


(a) (b)

(c)

(d)

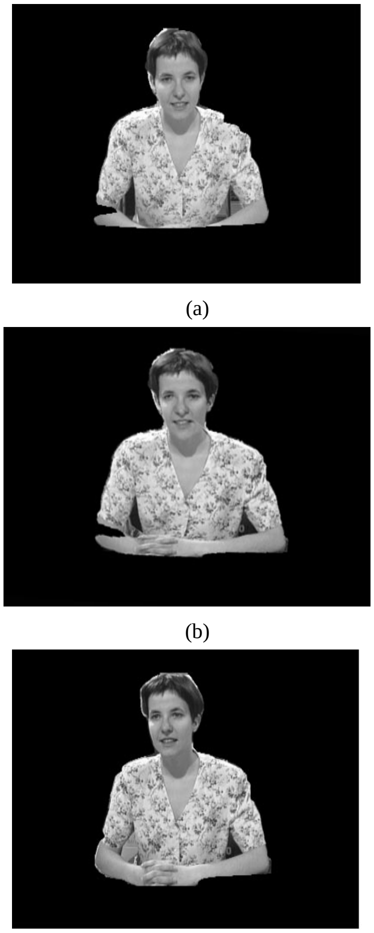
(e)

(f)

Fig. 11. Disparity compensated prediction on nonparallel images. (a) and (b): Corresponding meshes overlaid on the left and top (reference) views in the aligned coordinate. (c) and (d): Corresponding meshes mapped back to the original coordinate. (e): Original left view. (f): Predicted left view based on nodal correspondences shown in Fig. 11(c) and (d).

A. Overview of the Proposed Coder

Fig. 13 shows the block diagram of the encoder, for coding one view with respect to the reference view. The same process



(a)

(b)

(c)

Fig. 12. Synthesized center view of "GWEN" between top and right views. (a) Top view. (b) Synthesized center view. (c) Right view.

is used to code the other views. Essentially, the encoder consists of the following steps.

1) Preprocessing: Extract the foreground objects in both images and estimate the global disparity between them. If the input sequences are not coplanar, perform image alignment on the image pair using the approach described in Section III-B. The foreground extraction method can be found in [17]. To remove boundary errors which only occur over a short time, a length-5 temporal median filter was applied to the resulting foreground map.

2) Disparity Compensated Prediction: We choose 15 frames as a group of pictures (GOP). For each frame, we initialize a regular mesh in the reference view based on estimated global disparity, identify nodes falling on the foreground regions, and use the disparity estimation schemes described in Section II to determine nodal disparities between the two images. For frames 0 and 8, we start with the same regular mesh on the predicted view, and use the hierarchical exhaustive search method to find the horizontal shift of each node in the predicted view to minimize the DCP error. For the other frames, we apply the fast algorithm, starting with node positions obtained in the previous frame. The use of the hierarchical exhaustive search in the beginning and middle frame of each GOP is to stop the propagation of disparity estimation error. If the input views are not coplanar, we map the nodal positions in the aligned images to their original coordinates and predict the view to be coded from the decoded reference view, using the approach of Section III-C.

3) Coding: The reference view is coded using the MPEG-2 (TM5) encoder [18]. For the other view, we use either disparity compensated prediction or motion compensated prediction and different modes are employed. The coding schemes for various parameters are described in Section IV-B.

B. Coding Schemes for Different Types of Information

The coding schemes for different types of information are described as below.

1) Side Information: The side information includes image size, the estimated fundamental matrix, and the estimated global disparity. These parameters are coded losslessly using arithmetic coding. They are coded only once for a sequence. In addition, the extracted foreground map for the predicted view is coded for each frame, using chain coding. This is not necessary for frame 0 in each GOP, if that frame is coded as an I frame (see below).

2) Disparity Information: This includes nodal horizontal disparities in the aligned image for each frame. They are quantized to integer pixels and coded losslessly using arithmetic coding.

3) Texture Information: For frame 0 in each GOP, first we code the error between the original view and the predicted view using 8 × 8 block DCT (discrete cosine transformation). DCT coefficients are quantized using a flat quantization table with a quantization stepsize 16. If this error correction requires too many bits, we will code this frame as an I frame, i.e., every original block is coded using DCT with a nonflat quantization table. This is done following the JPEG standard.

For the other frames, we only code the foreground region. The background is duplicated from the previous frame. We divide the image into macroblocks (MB) of size 16 × 16, and code each MB in the foreground and associated four 8 × 8 blocks using one of the following modes.

Mode 1—Skip_DCP: Use the predicted image from the reference view based on the estimated disparity information. No additional bits are needed.

Mode 2—Skip_MCP: Use the predicted image from the previous decoded frame in this view based on the estimated motion information. Motion estimation is limited to ±0.5 pixel in the vertical direction only. Code the motion information losslessly (arithmetic coding).

Mode 3—DCP Error Correction: Apply DCT to the error between the original image block and the predicted block using disparity compensation. Quantize DCT coefficients using a flat quantization matrix, code the quantized coefficients using arithmetic coding. The quantization stepsize is determined based on the desired bit rate.

Mode 4—I Mode: Apply DCT to the original image block directly. Quantize DCT coefficients with the default quantization matrix in the MPEG standard. Code the quantized coefficients using arithmetic coding.

Because different modes use similar quantization stepsizes, they will lead to similar distortions. To decide which mode to use, ideally, we should perform the coding in all possible modes

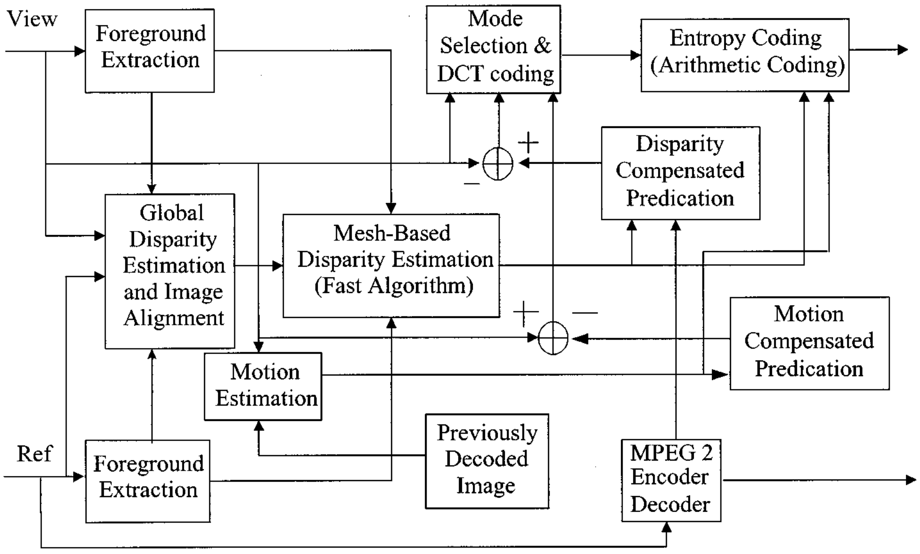

View
Foreground Extraction
Mode Selection & DCT coding
Entropy Coding (Arithmetic Coding)
+
−
Disparity Compensated Predication
Global Disparity Estimation and Image Alignment
Mesh-Based Disparity Estimation (Fast Algorithm)
+
−
Motion Compensated Predication
Motion Estimation
Previously Decoded Image
Ref
Foreground Extraction
MPEG 2 Encoder Decoder

Fig. 13. Block diagram for the proposed multiview encoder. Only the processing stage for coding one view with respect to a reference view is shown. The input views are assumed noncoplanar.

TABLE II
CODING RESULTS FOR "GWEN" AT 2 MB/S PER PREDICTED VIEW

GWEN Left	Average PSNR (+1.12 dB)	Total Bytes for 60 fields	GWEN Right	Average PSNR (+1.2 dB)	Total Bytes for 60 fields
Ours	27.25 dB	256021	Ours	27.00 dB	256178
BBSC	26.13 dB	258359	BBSC	25.79 dB	255212

TABLE III
CODING RESULTS FOR "GWEN" AT 1.4 MB/S PER PREDICTED VIEW

GWEN Left	Average PSNR (+1.53 dB)	Total Bytes for 60 fields	GWEN Right	Average PSNR (+ 1.40 dB)	Total Bytes for 60 fields
Ours	25.91 dB	173663	Ours	25.68 dB	175863
BBSC	24.38 dB	175178	BBSC	24.28 dB	176310

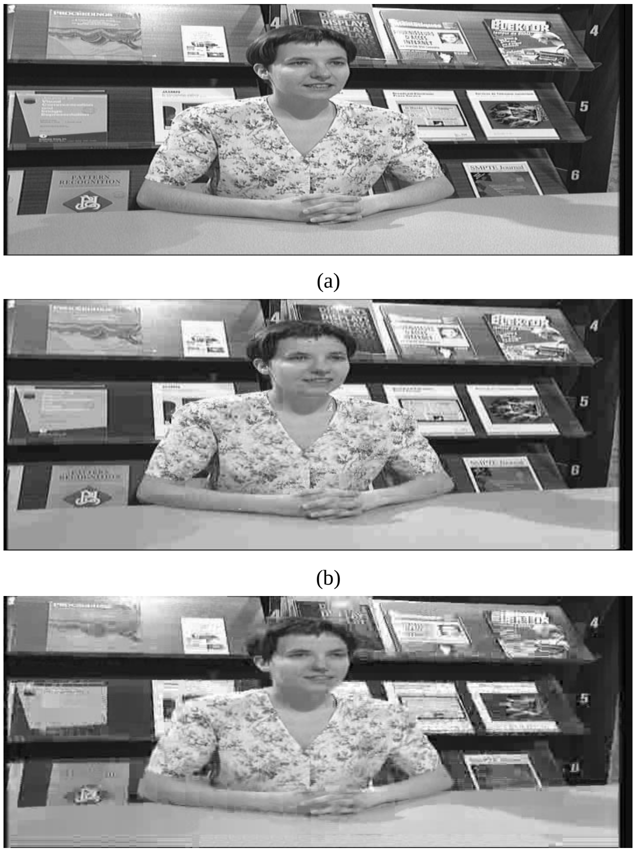
and choose the mode that requires the least number of bits. This will guarantee optimality in a rate-distortion sense. However, for reduced computations, we select the coding mode based on following easily computable measures: $S_{\rm DCP}$, the PSNR of the predicted MB using DCP, $S_{\rm MCP}$, the PSNR of the motion compensated MB, and $\tilde{S}_{\rm DCP}$, the PSNR of an 8×8 block of the predicted MB using DCP. Let $T_{\rm skip}$, T_E, and T_I be predefined thresholds, which satisfy $T_{\rm skip} > T_E > T_I$. The procedure for deciding which mode to use for each MB is as follows.

```
If ($S_{\rm DCP} > Tskip$) { Mode 1 }
Else if ($S_{\rm MCP} > Tskip$) { Mode 2}
  Else { for every block in the MB {
    If ($\tilde{S}_{\rm DCP} > T_E$) { Mode 1 }
    Elseif ($\tilde{S}_{\rm DCP} < T_I$) { Mode 4 }
    Else { Mode 3}
    }
  }.
```

4) Uncovered Region: Often due to the motion of the foreground object, uncovered background regions will appear. In our approach, we first detect the uncovered regions by using the difference between the foreground maps for two successive frames and then assign these regions in the current frame as foreground, so that the texture information in these regions can be coded using the procedure described above. Therefore, the foreground map transmitted is actually the original foreground map plus the uncovered regions. In most cases, blocks in the uncovered regions are coded in the I mode, which will contribute a significant portion of bits used for the overall foreground texture.

C. Simulation Results

The coder presented in Section IV-B has been applied to three test sequences. As described before, these sequences are interlaced and we treat each field as a progressive frame. Therefore, each GOP actually contains 15 fields. For each sequence, only first 60 fields are coded. For the reference view, we use the MP@ML mode of MPEG-2 TM5 encoder [18], with GOP $= 15$, and I/P distance $M = 3$. For the three-view sequence "GWEN," the top view is coded using MPEG-2 at 4 Mb/s. The average PSNR is 29.8 dB. Left and right sequences are coded with respect to the decoded top view. Tables II and III show the coding results obtained by our coder and by BBSC [2]. The latter is implemented by modifying the MPEG2 TM5 coder. The reference view is treated as base layer, the left/right view as

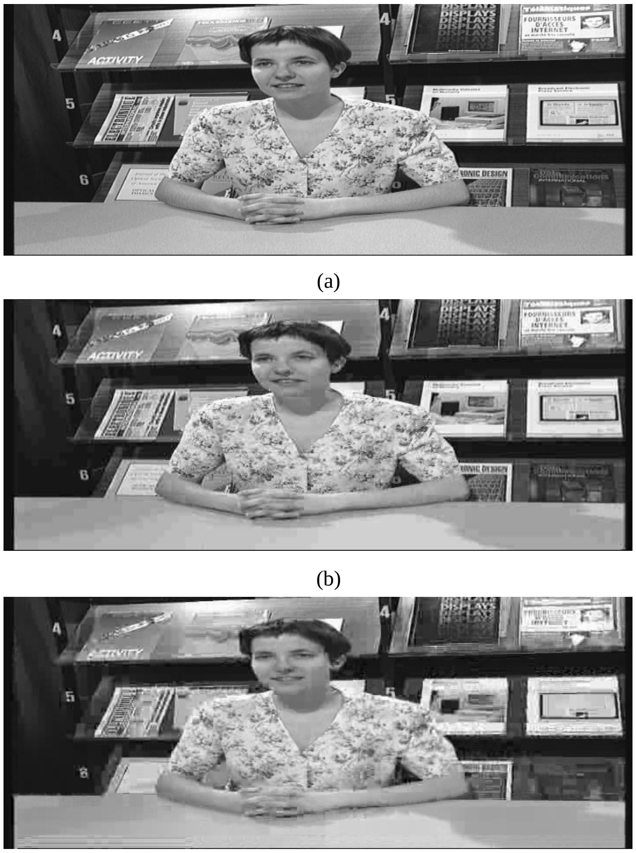


(a)

(b)

(c)

Fig. 14. Decoded left views at 1.4 Mb/s using decoded top view as references. (a) Original left image (field 1). (b) Proposed coder: 26.9 dB over the entire image and 24.8 dB over the foreground. (c) BBSC: 24.2 dB over the entire image and 22.78 dB over the foreground.



(a)

(b)

(c)

Fig. 15. Decoded right views at 1.4 Mb/s using decoded top view as references. (a) Original right image (field 1). (b) Proposed coder: 26.44 dB over the entire image and 24.53 dB over the foreground. (c) BBSC: 24.25 dB over the entire image and 22.42 dB over the foreground.

the enhancement layer. The first frame in each GOP is coded in P-mode, with the previous frame set to the same frame in the reference view. Each of the remaining frames is coded as a B-frame with the future frame set to the same frame in the reference view. The search range for disparity estimation is ± 60 pixels horizontally and ± 15 pixels vertically. Two set of results are given: one with the left and right views each coded at 2 Mb/s, the other at 1.4 Mb/s. As we can see from Table II, at 2 Mb/s our approach outperforms the BBSC by at least 1 dB. At an even lower bit rate (Table III), our approach shows more significant improvement over the BBSC.

Figs. 14 and 15 and show decoded left and right views, by the proposed coder and BBSC. One noticeable improvement by our coder is in the background. Because our coder spends more bits on the first frame in each GOP (including both background and foreground) than BBSC, the background is rendered more accurately in this frame, which is then duplicated in following frames. With BBSC, the bits are more evenly spread between the first frame (P-frame) and the remaining frames (B-frames) in a GOP, so that the texture in both the background and foreground in the first frame is not coded as accurately as with the proposed coder. Similarly, for the foreground, because the proposed coder codes the first frame more accurately, the edges and textures are sharper than those by BBSC. However, there are some artifacts that are visible in the images by the proposed coder.

In the proposed coder, the foreground map and nodal disparities take the same amount of bits, regardless the specified total rates. The bits used for texture information are adjusted manually, by varying the scale factors for the quantization matrix as well as the thresholds for switching between different modes. For the BBSC coder, these are determined by the rate-control mechanism built in the TM-5 coder. Table IV shows the bit allocation of the proposed coder for this sequence. We can see that nodal disparities, foreground map, and other side information take from 8% to 12% of the total bits. When the specified total bit rate is very low, our coder can choose to only code the side information, nodal disparities, and texture in the first frame of a clip (60 fields) in a sequence. This would reduce the bit rate to 0.28 Mb/s, with an average PSNR of 21.5 dB. The quality would still be acceptable. With the rate-control mechanism built in the TM5 coder, the lowest rate we can obtain with BBSC is about 1.3 Mb/s. At this bit rate, the BBSC produces very severe blocking artifacts.

Tables V and VI show the coding results for the right views of the two stereoscopic sequences, “MAN” and “ANNE.” We compare the results obtained using our coder and the results from BBSC. The motion search range is set to ± 60 horizontal and ± 2 vertical, with BBSC. In our simulations, the left view for “MAN” is coded at 2 Mb/s, with a PSNR of 44.51 dB. For “ANNE,” the bit rate for the left view is 4 Mb/s and the PSNR is

TABLE IV
BIT ALLOCATION OF THE PROPOSED CODER FOR "GWEN" AT 2 MB/S AND 1.4 MB/S

Bit-rate	2 Mbps		1.4 Mbps	
	Left	Right	Left	Right
Nodal disparities, foreground map and other side information	8.10%	6.63%	11.95%	9.66%
Texture in frame 0	38.72%	38.85%	33.32%	33.43%
Texture in other frames (foreground DCP error + Uncovered Background)	53.18%	54.52%	54.73%	56.91%

TABLE V
CODING RESULT FOR THE RIGHT VIEW OF SEQUENCE "MAN"

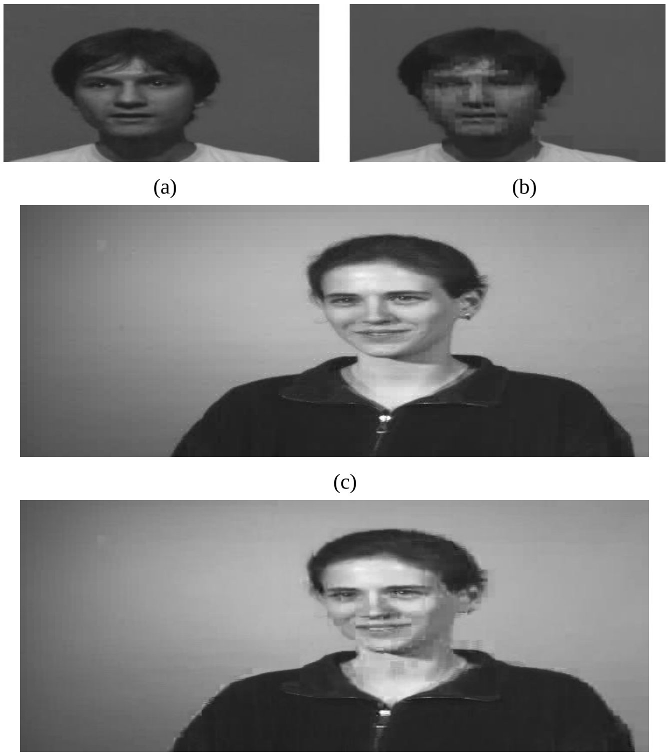
"MAN" (384x192)	Average PSNR	Total Bytes (0.5 Mbps)	Average PSNR	Total Bytes (0.4 Mbps)
Ours	40.02 dB	63725	39.72 dB	55275
BBSC	39.06 dB	63740	35.95 dB	61139*

* These are the lowest rates achievable with the highest quantization factor.

TABLE VI
CODING RESULT FOR THE RIGHT VIEW OF SEQUENCE "ANNE"

"ANNE" (720x288)	Average PSNR	Total Bytes (2 Mbps)	Average PSNR	Total Bytes (1.3 Mbps)
Ours	39.75 dB	252099	38.39 dB	167474
BBSC	39.20 dB	252667	34.82 dB	185549*

* These are the lowest rates achievable with the highest quantization factor.


(a)

(b)

(c)

(d)

Fig. 16. Decoded images (field 1) for "MAN" using: (a) the proposed coder (41.51 dB, 0.4 Mb/s) and (b) BBSC (34.67 dB 0.48 Mb/s). Decoded images (field 1) for "ANNE" using: (c) the proposed coder (38.56 dB, 1.3 Mb/s) and (d) BBSC (33.96 dB, 1.4 Mb/s).

40.95 dB. From Tables V and VI, we can see that our approach yields significant improvement over BBSC when the bit rates are below a certain critical point. Fig. 16 presents the decoded images for "MAN" and "ANNE" at 0.4 and 1.3 Mb/s, respectively. At these low rates, images obtained from BBSC suffer from very severe blocking artifacts, whereas the proposed coder still yielded visually very satisfactory results. The blocking artifacts produced by the BBSC approach are in part because we force the coder to code at 60 fields/s, even though using a lower frame rate may be more appropriate at the specified rates.

V. SUMMARY AND CONCLUDING REMARKS

In this paper, we presented a new scheme for coding multiview sequences. The method is based on segmentation of foreground objects, image alignment, and mesh-based disparity estimation for the foreground region. The mesh-based disparity representation guarantees a smoothly varying disparity field. It also enables the interpolation of intermediate views. Image alignment serves to warp epipolar lines in a given image pair to horizontal lines so that only horizontal disparities need to be estimated. The coder does not require the knowledge of camera parameters; rather, the epipolar geometry is determined through estimating the fundamental matrix based on features detected in the input images. At an intermediate bit rate range, the proposed coder outperformed the BBSC coder by a noticeable margin. At even lower bit rates, when the BBSC coder suffers from severe blocking artifacts, the proposed coder can still yield visually acceptable results. With the coded foreground map, the fundamental matrix, and the nodal displacements, the decoder can not only reconstruct the input views, but also interpolate intermediate views. This is not feasible with block-based coders, because block-based disparity representation is discontinuous and physically incorrect. The proposed scheme can be used in a wide variety of applications, such as videoconferencing, virtual reality, and synthetic and natural hybrid coding. It is also appropriate for MPEG-4 scene compositor, where user can modify the view angle in the multiview sequence.

The proposed coder uses a structure representation that is in-between 2-D and 3-D. A truly 3-D object-based coder would attempt to construct an adaptive 3-D mesh that fits well with the object surface and code the mesh structure (3-D nodal positions in an initial frame) and motion (nodal displacements in 3-D over time). This is, e.g., the approach taken in [5]. Such a coder would, however, require very sophisticated processing. Many initial frames are also required before an accurate mesh can be obtained. By resetting the 2-D mesh to a regular mesh in the reference view at every new frame, we do not seek to construct a 3-D surface model, nor aim to track the same 3-D patch over time. Rather, we use the mesh structure merely to facilitate disparity estimation and intermediate view generation. Using a regular mesh in the reference view requires significantly less computation, compared to using an adaptive mesh, and yet can yield quite accurate disparity estimation. From our simulation results, the proposed coder can provide a good tradeoff between complexity and coding efficiency. We also expect that the proposed coder be effective for sequences with more complicated object structures than that can be handled by fully 3-D based approaches. In the current coder, we mainly rely on DCP for coding the nonreference view. MCP is employed with very limited search range and is only applied when it provides a good prediction that does not require further correction. As a future work, MCP with a more extensive search range, and switching between DCP and MCP based on prediction errors, may be more efficient.

Acknowledgment

The authors would like to thank L. Falkenhagen, from the University of Hannover, for explaining his image-rectification scheme and clarifying some stereo imaging problems. They would also like to thank J.-R. Ohm, from Heinrich Hertz Institut, Germany, and D. Pele, from CCETT, for providing sequences used in the simulation.

References

[1] *Proposed draft amendment no. 3 to 13 818 (multiview profile)*, International Organization for Standardization, ISO/IEC JTC1/SC29/WG11, N1088, Nov. 1995.

[2] A. Puri, R. V. Kollarits, and B. G. Haskell, "Stereoscopic video compression using temporal scalability," *SPIE Conf. Visual Communications and Image Processing (VCIP'95)*, vol. SPIE-2501, pp. 745–756, 1995.

[3] A. Tamtaoui and C. Labit, "Constrained disparity and motion estimators for 3DTV image sequence coding," *Signal Processing: Image Commun.*, vol. 4, pp. 45–54, 1991.

[4] D. V. Papadimitriou and T. J. Dennis, "Three dimensional parameter estimation from stereo image sequences for model-based image coding," *Signal Processing: Image Commun.*, vol. 7, pp. 471–487, 1995.

[5] S. Malassiotis and M. G. Strintzis, "Coding of video-conference stereo image sequences using 3D models," *Signal Processing, Image Commun.*, vol. 9, pp. 125–135, 1997.

[6] N. Grammalidis and M. G. Strintzis, "Disparity and occlusion estimation in mutiocular systems and their coding for the communication of multiview image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 328–344, June 1998.

[7] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 139–154, Mar. 1985.

[8] D. Tzovaras, N. Grammalidis, and M. Strintzis, "Object-based coding of stereo image sequences using joint 3-D motion/disparity compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 312–327, Apr. 1997.

[9] L. Falkenhagen, "Block-based depth estimation from image triples with unrestricted camera setup," presented at the IEEE Workshop Multimedia Signal Processing, Princeton, NJ, June 1997.

[10] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR'86)*, Miami Beach, FL, June 1986, pp. 364–374.

[11] O. Faugeras, *Three-Dimensional Computer Vision: A Geometrical Viewpoint*. Cambridge, MA: MIT Press, 1993.

[12] Y. Wang and O. Lee, "Use of 2D deformable mesh structures for video compression. Part I—The synthesis problem: Mesh based function approximation and mapping," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 636–646, Dec. 1996.

[13] J. R. Ohm, "An object-based system for stereoscopic viewpoint synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 801–811, Oct. 1997.

[14] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," INRIA, Res. Rep. 2927, July 1996.

[15] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Conf.*, Manchester, U.K., Aug. 1988, pp. 147–151.

[16] Z. Zhang, R. Deriche, Q.-T. Luong, and O. Faugeras, "A robust approach to image matching: Recovery of the epipolar geometry," in *Proc. Int. Symp. Young Investigators on Information, Computers and Control*, Beijing, China, 1994, pp. 7–28.

[17] R.-S. Wang and Y. Wang, "Stereo sequence analysis, compression, and virtual viewpoint synthesis," in *1998 IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, Los Angeles, CA, Dec. 1998, pp. 492–497.

[18] MPEG. (1999) MPEG Software Simulation Group (MSSG). [Online]. Available: http://www.mpeg.org/MPEG/MSSG/tm5

Ru-Shang Wang received the B.S. degree in electronic engineering from Tamkang University, Tamsui, Taiwan, in 1985, and the M.S. degree from the Chung Yuan Christian University, Chung Li, Taiwan, in 1989. He received the Ph.D. degree in electrical engineering from the Polytechnic University, Brooklyn, NY, in 1999.

Since then, he has been with Ezenia, Inc. (formerly VideoServer), Burlington, MA. His research interests include image/video processing, stereo imaging, multimedia application, and videoconferencing.

Yao Wang (M'90–SM'98) was born in Zhejiang, China, in 1962. She received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D degree in electrical engineering from University of California at Santa Barbara in 1990.

Since 1990, she has been on the faculty of Polytechnic University, Brooklyn, NY, and is presently an Associate Professor of Electrical Engineering. From 1992 to 1996, she was a part-time Consultant with AT&T Bell Laboratories, Holmdel, NJ, and since 1997, with AT&T Labs—Research, Red Bank, NJ. She was on sabbatical leave at Princeton University, Princeton, NJ, in 1998. Her current research interests include image and video compression for unreliable networks, motion estimation, object-oriented video coding, signal processing using multimodal information, and image reconstruction problems in medical imaging.

Dr. Wang is presently an Associate Editor for IEEE Transactions on Multimedia and for the *Journal of Visual Communications and Image Representation*. She has previously served as an Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology. She is a member of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society and the Technical Committee on Visual Signal Processing and Communications of the IEEE Circuits and Systems Society. She has served on the organizing/technical committees of several international conferences and workshops.