Multiview Image Coding Based on Geometric Prediction

Xing San, Hua Cai, *Member, IEEE*, Jian-Guang Lou, *Member, IEEE*, and Jiang Li, *Senior Member, IEEE*

***Abstract*—Many existing multiview image/video coding techniques remove inter-viewpoint redundancy by applying disparity compensation in a conventional video coding framework, e.g., H.264/MPEG-4 AVC. However, conventional methodology works ineffectively as it ignores the special characteristics of inter-viewpoint disparity. In this paper, we propose a geometric prediction methodology for accurate disparity vector (DV) prediction, such that we can largely reduce the disparity compensation cost. Based on the new DV predictor, we design a basic framework that can be implemented in most existing multiview image/video coding schemes. We also use state-of-the-art H.264/MPEG-4 AVC as an example to illustrate how the proposed framework can be integrated with conventional video coding algorithms. Our experiments show proposed scheme can effectively tracks disparity and greatly improves coding performance. Compared with H.264/MPEG-4 AVC codec, our scheme outperforms maximally 1.5 dB when encoding some typical multiview image sequences. We also carry out an experiment to evaluate the robustness of our algorithm. The results indicate our method is robust and can be used in practical applications.**

***Index Terms*—Disparity estimation, disparity vector (DV), geometric prediction, H.264/MPEG-4 AVC, multiview image coding (MIC), multiview video coding (MVC), source coding, triangulation.**

I. Introduction

MULTIVIEW image/video provides exciting viewing experiences. It enables users to watch a scene from different viewing directions. As a brand new application, multiview video has recently received increasing attention. Generally, to provide a smooth multiperspective viewing experience, content producers need to capture the same scene with ideal quality from multiple viewpoints. A convergent multiview camera setup is shown in Fig. 1, where the cameras are positioned inward to capture the same scene from different angles. Usually, the simultaneous multiple video streams from multiview cameras are referred to as multiview video. A multiview video sequence can be also regarded as a temporal sequence of special visual effect snapshots, captured from different viewpoints at multiple times. Such a special snapshot is comprised of all the still images taken by multiple cameras at one certain time instance, so it is essentially a multiview image sequence or a *frozen moment* sequence as named in [1]. On the other hand, as a degenerated form of multiview video in one moment, multiview image can be produced without demanding multiple cameras all the time. For instance, we can generate a multiview image sequence by moving a single video camera pointing to a scene of interest along a predefined capture trajectory. Fig. 2 depicts the concepts of multiview video, multiview image, and their relationships. Fig. 3 shows some sample images from a multiview image sequence, *Dinosaur* [2].

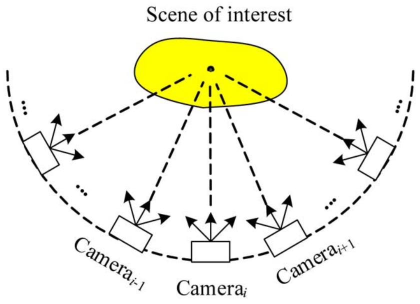


Fig. 1. Convergent multiview camera setup.

Though multiview image/video is capable of providing an exciting viewing experience, it is challenging to put it into practical applications. One major obstacle is the extremely large data required in doing so. To overcome this problem, a specially designed multiview image/video encoder becomes indispensable. For the same reason, multiview video coding (MVC) is identified by the MPEG 3-D audio and video (3 DAV) ad hoc group as one of the most challenging issues associated with such new applications as free viewpoint video [3], [4].

In general, both multiview image coding (MIC) and MVC seek to reduce the redundancy of images from different viewpoints by inter-viewpoint disparity compensation. As we know, the inter-viewpoint disparity is completely different from the temporal motion that has been widely used in traditional video coding algorithms [5]. For traditional temporal video sequences, only moving objects are displaced, while background objects often do not move or just contain global motion. Moreover, temporal motion can often be approximated as a translation model. On the contrary, the inter-viewpoint disparity is dependent on the depth of the scene and the camera setup. It cannot be simply modeled as translation. Fortunately, one advantage of the inter-viewpoint disparity over the temporal motion is that the inter-viewpoint disparity vector (DV) is highly related to the multiview geometry. In an ideal situation, the DV can be precisely predicted by the multiview geometry based on the depth information and the camera projection matrixes. We believe that both MIC and MVC can take advantage of geometric prediction.

Manuscript received January 15, 2007; revised June 1, 2007. The work presented in this paper was carried out at Microsoft Research Asia in Beijing. This paper was recommended by Guest Editor A. Smolic.

X. San is with the Department of Electrical Engineering, University of Science and Technology of China, Hefei 230027, China (e-mail: sanbo_ustc@hotmail.com).

H. Cai, J-G. Lou, and J. Li are with the Media Communication Group, Microsoft Research Asia, Beijing, 100080, China (e-mail: huacai@microsoft.com; jlou@microsoft.com; jiangli@microsoft.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2007.905382

1051-8215/$25.00 © 2007 IEEE

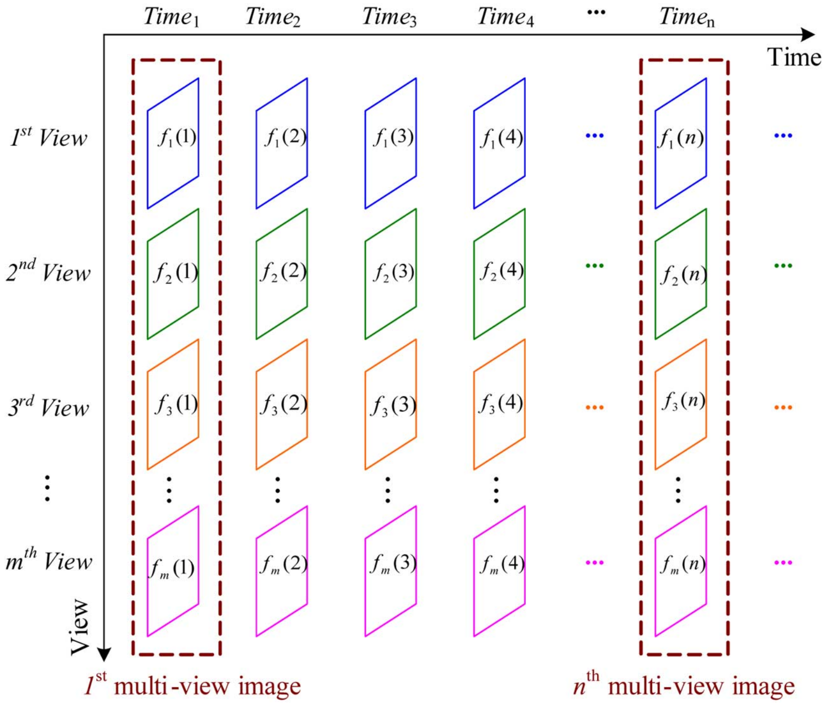

$Time_1$
$Time_2$
$Time_3$
$Time_4$
$Time_n$
Time
1^{st} View
2^{nd} View
3^{rd} View
m^{th} View
View
$f_1(1)$ $f_1(2)$ $f_1(3)$ $f_1(4)$ $f_1(n)$
$f_2(1)$ $f_2(2)$ $f_2(3)$ $f_2(4)$ $f_2(n)$
$f_3(1)$ $f_3(2)$ $f_3(3)$ $f_3(4)$ $f_3(n)$
$f_m(1)$ $f_m(2)$ $f_m(3)$ $f_m(4)$ $f_m(n)$
1^{st} multi-view image
n^{th} multi-view image

Fig. 2. Relation between multiview video and multiview image. A multiview video sequence consists of m conventional video streams captured from m viewpoints ($f_i(j)$ denotes the jth frame of the ith view), while a typical multiview image sequence can be constructed by all the still images from m viewpoint at any time instant, e.g., the first and the nth multiview image.

Fig. 3. Sample images from a multiview image sequence Dinosaur.

In this paper, we propose a new method for improving coding efficiency through geometric prediction. We discuss how to integrate the proposed geometric prediction methodology with existing video coding schemes, such as state-of-the-art H.264/MPEG-4 AVC, to achieve higher yet reliable coding performance. The main contributions of this paper can be summarized as follows.

1) We propose an effective geometric prediction methodology for accurate DV prediction.
2) We propose a basic MIC framework based on geometric prediction, which can also be implemented in most of the existing MVC methods as long as inter-viewpoint prediction is exploited.
3) We integrate the proposed framework with H.264/MPEG-4 AVC. Our experimental results show the validity of geometric prediction.
4) We also carry out an experiment to evaluate the robustness of our proposed algorithm. The results indicate that our proposed coding method is robust and can be used in practical applications.

The rest of the paper is organized as follows. We first review related works in Section II. Next, in Section III, we briefly review the property of multiview geometry since it is the theoretical foundation of this paper. We then describe the proposed multiview geometry based DV predictor in Section IV. Taking the H.264/MPEG-4 AVC codec as an example, we present a new coding framework using the proposed predictor in Section V. In Section VI, we show the experimental results including both coding efficiency and noise sensitivity. We conclude our work and discuss future research directions in Section VII.

II. Related Work

In the past decade, many different approaches have been proposed for coding multiview image or video. Most of them focus on how to use the disparity/depth map to improve video coding efficiency.[1] In some approaches [6], [7], a block based disparity-compensated prediction is used to exploit inter-viewpoint disparity information. Similar to the motion compensation (MC) for coding temporal video sequences, disparity compensation (DC) can be used to describe where each image block of the current view comes from in a previous view. The encoder can

[1] Although our paper focuses on multiview image which contain more than three views, we also include researches of stereo image/video coding in this section as these technologies are closely related.

switch between DC and MC to minimize the prediction error. For example, Luo *et al.* [8] adopt the combination of DC and MC to code stereo video sequences. Their coding structure is similar to the MVP structure of MPEG-2 [6], where one view sequence is referred to as a reference stream and the other view sequence is coded as a target stream. Li *et al.* [9] claim that the unsymmetrical coding structure in MVP of MPEG-2 will bring very high complexity when coding more than two views. They present a symmetrical coding scheme with the combination of DC and MC. Authors in the paper [10] describe a hierarchical scheme based on DC to compress a dense light field. They use vector quantization for compressing prediction residual to facilitate random view access. Woo *et al.* [11] propose an improved overlapped-block-matching (OBM) technique and a disparity estimation/compensation approach using adaptive windows with variable shapes. In a recent research [12], disparity-compensated lifting that incorporates DC into the discrete wavelet transform (DWT) using a lifting structure is proposed to pursue the benefits of wavelet coding, such as scalability in all dimensions and superior compression performance.

Besides disparity-compensated prediction, some new prediction algorithms have also been proposed. For example, both the furthest left and right images can be used as reference images to predict intermediate images based on a linear subspace projection method [5], [13], [14], where each intermediate image block can be compensated by a linear combination of two corresponding blocks on the left and right images. This algorithm assumes that the multiview image is captured by some parallel coplanar cameras. However, in real applications, many multiview systems have more complex camera setups (e.g., a centripetal camera array is used in [1] to provide interactive multiview video service). In paper [15], a multiscale recurrent patterns based algorithm is presented as a more general framework. In this algorithm, a new image block can be approximated using dilated, contracted, displaced, or deformed versions of blocks already processed. However, it is too complex to be used for coding multiview image/video, especially when the number of views increases.

3-D reconstruction [16], [17] and view interpolation [18], [19] based on multiview image have been extensively studied in the community of computer vision and computer graphics. It is natural to apply these technologies in MVC and MIC. For example, a model-aided predictive coding algorithm is proposed in [20], in which the authors use a 3-D reconstruction algorithm to recover the geometric model of the scene, and then use the model to aid DC and occlusion detection between views. However, their work is only tested on some synthesis datasets. It is known that constructing a precise 3-D geometric model from real world multiview image is often a difficult task, especially when there is a complex lighting condition. On the other hand, unlike the 3-D reconstruction, the view interpolation is a relative lightweight algorithm in general [19]. In [21], Martinian *et al.* propose a novel extension of the H.264/MPEG-4 AVC codec for MVC. Their algorithm exploits the inter-viewpoint correlation by a view interpolation technique, and extends the H.264/MPEG-4 AVC's buffer management method to allow disparity-compensated view synthesis prediction. However, the decoder has to synchronize the depth maps with the encoder in order to apply the view synthesis prediction. The synchronization not only costs 5–10% of the total bit rate, but also increases the coding complexity.

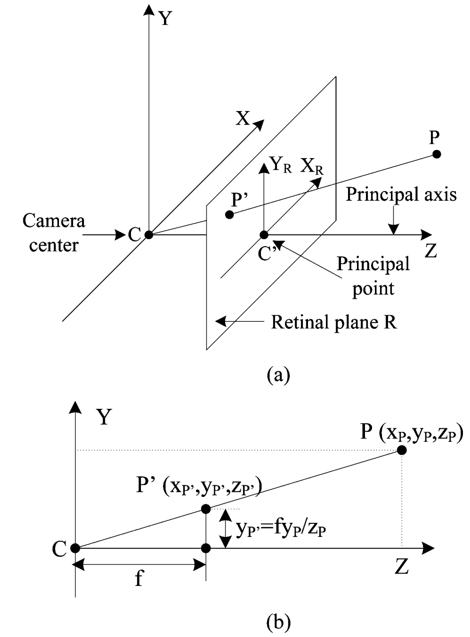


Fig. 4. Geometry of pinhole camera model. (a) Gometric relation between a point P in 3-D space and its projective point P' in the retinal plane. (b) Coordinate relation between P and P'.

In this paper, we propose a geometric prediction methodology for disparity-compensated coding. Different from the existing DC methods mentioned above, our focus is to use the multiview geometry in DV prediction to largely reduce the matching cost. The proposed methodology can also work with many existing DC methods to jointly improve coding efficiency.

III. Basic Multiview Geometry

A. Camera Model

In computer vision and optical measurement, the well known pinhole camera model [16] (Fig. 4) is often used to describe the geometric relationship of image formation. In the pinhole model, the geometric process for image formation is completely determined by a perspective *projection center* and a *retinal plane*. The projection of a scene point can be modeled as:

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M \begin{bmatrix} x_P \\ y_P \\ z_p \\ 1 \end{bmatrix} \tag{1}$$

where M is a 3×4 matrix, namely *projection matrix*; $[x_P \ \ y_P \ \ z_P \ \ 1]^T$ is the homogeneous coordinates of a 3-D point P; $[u \ \ v \ \ 1]^T$ is the homogeneous coordinates of the image point; and z is P's depth. Usually, the projection matrix M is determined by the camera's intrinsic parameters (e.g., focal length) and its posture in the world coordinate. It can be estimated by camera calibration [16].

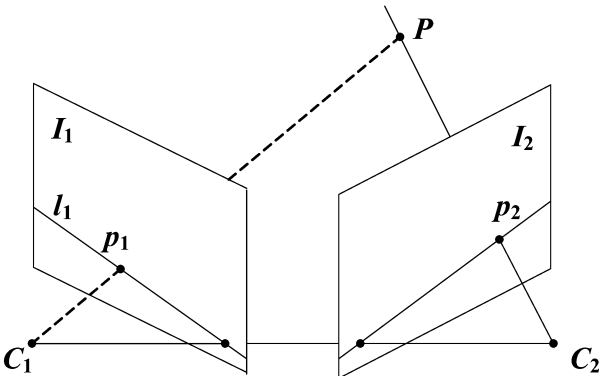


Fig. 5. Epipolar geometry.

B. Epipolar Geometry

Epipolar geometry is the geometry constraint between a stereo pair of images [16], [22], which has been extensively studied in computer vision. Fig. 5 depicts the epipolar geometry, where C_1 and C_2 are the optical centers of the first and the second cameras, and the plane I_1 and I_2 are the first and the second image planes. Given a point P in a 3-D space, let us denote its projection on the second image plane as P_2. According to the epipolar geometry, its corresponding point P1 in the first image is constrained to lie on line l_1. This line is called the epipolar line of P_2. The epipolar constraint can be formulated as

$$\tilde{P}_1^T \cdot F \cdot \tilde{P}_2 = \tilde{P}_1^T \cdot l_1 = 0 \tag{2}$$

where $\tilde{P}_1^T$ and $\tilde{P}_2$ are the homogeneous coordinates of P_1 and P_2, and F is a 3×3 matrix called the *fundamental matrix* (FM). From (2), it is clear that once FM is available, the equation of epipolar line l_1 can be computed. Actually there are many ways to determine FM. A good review of existing techniques for estimating FM is presented in [22]. If cameras are calibrated, FM can be easily calculated from the camera projection matrices.

Epipolar geometry is a very useful tool for MVC. In our previous paper [23], we utilize the epipolar geometry to significantly speed up the block-based motion estimation process in MVC.

C. Reconstruction of Spatial Points

3-D reconstruction from multiple images is one of the most important research topics in the area of stereo vision. There are lots of 3-D reconstruction algorithms proposed in the last decade. In this section, we only briefly introduce the triangulation calculation that will be used later in our algorithm. As depicted in Fig. 5, the point P_1 and P_2 are both the projections of the same spatial point $P.P_1$ and P_2 are called the *corresponding point pair*. Given the projection matrixes of two cameras, we have the following equations:

$$\begin{aligned} Z_{c1}\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} &= M_1 \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} m_{11}^1 & m_{12}^1 & m_{13}^1 & m_{14}^1 \\ m_{21}^1 & m_{22}^1 & m_{23}^1 & m_{24}^1 \\ m_{31}^1 & m_{32}^1 & m_{33}^1 & m_{34}^1 \end{bmatrix} \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \end{aligned} \tag{3}$$

$$\begin{aligned} Z_{c2}\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} &= M_2 \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} m_{11}^2 & m_{12}^2 & m_{13}^2 & m_{14}^2 \\ m_{21}^2 & m_{22}^2 & m_{23}^2 & m_{24}^2 \\ m_{31}^2 & m_{32}^2 & m_{33}^2 & m_{34}^2 \end{bmatrix} \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix} \end{aligned} \tag{4}$$

where M_1 and M_2 are the projection matrixes of two cameras; $(u_1, v_1, 1)$ and $(u_2, v_2, 1)$ are the homogeneous coordinates of P_1 and P_2. With elimination of Z_{c1} and Z_{c2}, the (3) and (4) become:

$$\begin{aligned} &\left(u_1 m_{31}^1 - m_{11}^1\right) x_P \\ &\quad + \left(u_1 m_{32}^1 - m_{12}^1\right) y_P + \left(u_1 m_{33}^1 - m_{13}^1\right) z_P \\ &\quad = m_{14}^1 - u_1 m_{34}^1 \\ &\left(v_1 m_{31}^1 - m_{21}^1\right) x_P \\ &\quad + \left(v_1 m_{32}^1 - m_{22}^1\right) y_P + \left(v_1 m_{33}^1 - m_{23}^1\right) z_P \\ &\quad = m_{24}^1 - v_1 m_{34}^1 \end{aligned} \tag{5}$$

$$\begin{aligned} &\left(u_2 m_{31}^2 - m_{11}^2\right) x_P \\ &\quad + \left(u_2 m_{32}^2 - m_{12}^2\right) y_P + \left(u_2 m_{33}^2 - m_{13}^2\right) z_P \\ &\quad = m_{14}^2 - u_2 m_{34}^2 \\ &\left(v_2 m_{31}^2 - m_{21}^2\right) x_P + \left(v_2 m_{32}^2 - m_{22}^2\right) y_P \\ &\quad + \left(v_2 m_{33}^2 - m_{23}^2\right) z_P \\ &\quad = m_{24}^2 - v_2 m_{34}^2. \end{aligned} \tag{6}$$

Therefore, $[x_P \ y_P \ z_P \ 1]^T$ (the spatial coordinates of P) is the solution of the above four linear equations. We can use least-square method to estimate the position of point P, if the two camera's projective matrixes and the corresponding points P_1 and P_2 are all known.

IV. Multiview Geometry Based DV Predictor

Motion compensation is widely used in video coding schemes for removing the temporal correlation of successive frames. In state-of-the-art coding schemes such as H.264/MPEG-4 AVC, motion compensation performance (in terms of coding efficiency) is optimized by minimizing a Lagrangian cost function [24]

$$J(v) = D(v) + \lambda \cdot R(v). \tag{7}$$

In the above, the optimal motion vector (MV) is determined by the block matching distortion $D(v)$ and the matching cost $R(v)$ (i.e., number of bits for representing the block offset v). Similarly, the optimization criterion is also valid for disparity compensation in MIC, where the inter-viewpoint correlation needs to be removed. In general, two kinds of efforts can be devoted for high efficient coding of multiview image: (1) reducing the matching distortion by either considering better reference images or applying view interpolation technology [21]; and (2) reducing the matching cost.

As a key module in video coding schemes, vector predictor (either for MV or DV) reduces the matching cost by predicting the value of each input vector. The objective of the predictor

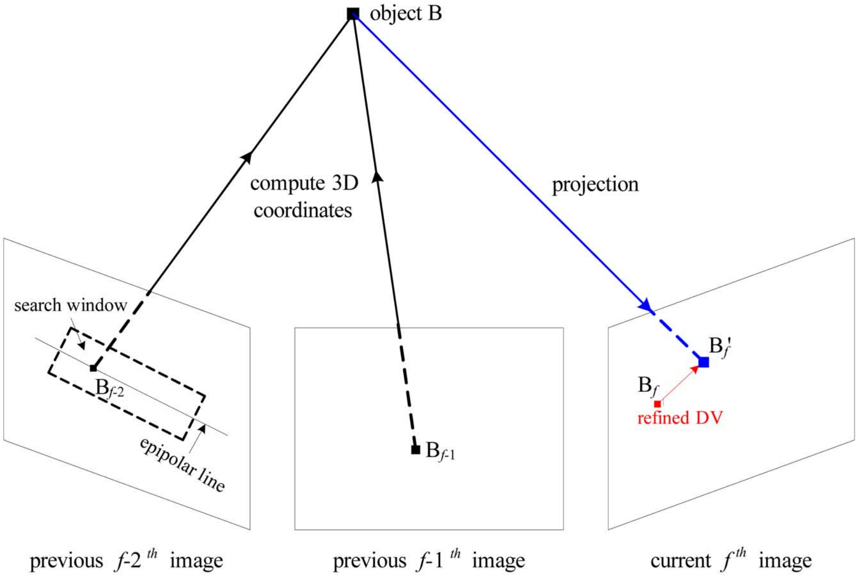


Fig. 6. Proposed geometric prediction methodology. B_{f-1}—a block in the $f-1$th image; B_{f-2}—the corresponding block of B_{f-1} in the $f-2$th image; B—the corresponding object in the 3-D scene; B'_f—the projected result in the fth image (i.e., the current image being coded); B_f—the real corresponding block of B_{f-1} in the fth image.

in MIC can be described as follows: assuming that the f th multiview image is predicted from a previously encoded multiview image (say, the $f-1$th image) with block-based disparity compensation, its DVs are represented by $V_f(i,j)$ for the block with coordinates (i,j). Now the question is how to find a good DV predictor such that the energy of the prediction residual, $\sum_i \sum_j |V_f(i,j) - \tilde{V}_f(i,j)|$ (where $\tilde{V}_f(i,j)$ is the predicted vector of $V_f(i,j)$), can be minimized.

It is known that MVs have strong spatial correlation as the majority of temporal motion is translation. As a result, median prediction, which is adopted in many advanced video coding scheme such as H.264/MPEG-4 AVC, can efficiently remove the MVs' correlation by predicting from the neighboring MVs [24]. However, the median predictor becomes ineffective even more in the multiview case. The reasons are, first, the inter-viewpoint correlation is very different from the temporal correlation as the view displacement is highly dependent on depth and viewing angle. It cannot be modeled as simple object translation and the translation model is not valid any more. Hence, the DVs' spatial correlation is not as strong as that of MVs. Second, the displacement may be very large compared to MVs. In particular, even the background might be of large displacement due to camera rotation (e.g., the convergent multiview camera setup as shown in Fig. 1), whereas in the conventional video the background is often static or contains a global motion. Third yet most importantly, the view displacement in the multiview case is structured and predictable [14]. On the contrary, though the temporal motions can be predicted through median prediction, they are not absolutely predictable.

We believe that it is possible to find a more effective way for DV prediction in MIC. Unlike the median predictor that relies on simple spatial correlation of MVs, our proposed multiview geometry based DV predictor can accurately track the disparity and can greatly reduce the matching cost. Specifically, the proposed DV predictor calculates $\tilde{V}_f(i,j)$ in three steps (Fig. 6): (1) searching for corresponding block pairs from two encoded images; (2) projecting corresponding block pairs to the f th image; and (3) obtaining $\tilde{V}_f(i,j)$ through DV fusion.

A. *Step 1: Searching for Corresponding Block Pairs*

The purpose of this step is to find the corresponding object blocks of the same scene but from two different viewing images. We name each group consisting of two corresponding blocks as a *corresponding block pair*. Once the corresponding block pair is available, its corresponding 3-D coordinates in the 3-D scene can be calculated.

To find the corresponding block pairs, we choose two images from the previously encoded images for processing. The search is carried out upon reconstructed images (or, the decoded images obtained at the decoder side). The reason is that we need to repeat the search process at the decoder side, whereas at the decoder side only the reconstructed images are available. Actually, any two encoded images in a multiview image sequence can be used. However, to improve the search accuracy, we choose the two nearest neighboring images in our scheme (Fig. 6), because they are the closest ones among all the encoded images with respect to the current image. These two selected images are less influenced by occlusion and thus can provide a better DV prediction accuracy.

Based on the epipolar geometry introduced in Section III-B, we can constrain the search range in a narrow window around the epipolar line (Fig. 7). Searching along the epipolar line can remarkably reduce the block mismatching probability, especially in the homogeneous regions. The detailed search algorithm is: we first partition the $f-1$th image into rectangular blocks. The block size is identical to the minimum coding block size that a codec can support. For example, when this algorithm is integrated with H.264/MPEG-4 AVC, the block size is set to 4×4. For each block $B_{f-1}(x_{f-1}, y_{f-1})$ with coordinates (x_{f-1}, y_{f-1}) (here the block coordinates are represented by

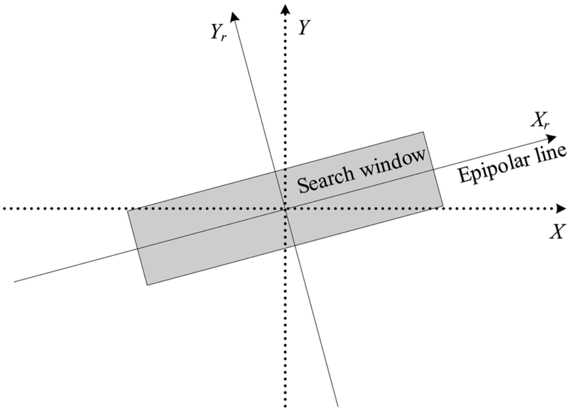


Fig. 7. Corresponding block pair search along epipolar line.

the centroid coordinates[2] of the block), we can obtain its corresponding epipolar line through (2). A full search is then performed within the narrow search window. At last, the optimal corresponding block $B_{f-2}(x_{f-1}+\Delta x_{f-2}, y_{f-1}+\Delta y_{f-2})$ of $B_{f-1}(x_{f-1}, y_{f-1})$, where $(\Delta x_{f-2}, \Delta y_{f-2})$ denote the block offset from (x_{f-1}, y_{f-1}), can be obtained using a minimal mean-squared-error (MSE) criterion between the two blocks.

It should be mentioned that the above approach was just a brute force implementation for searching the corresponding block pairs. Of course, some fast search algorithms [23] other than the full search can be used within the search window to further speed up the search process. Also, we believe that applying more sophisticated search algorithms [25] such as the multiscale searching [26] can facilitate the improvement of the search accuracy. This is out of the scope of this paper.

B. Step 2: Projecting Corresponding Block Pairs to the fth Image

After the first step, we obtain a corresponding block pair $B_{f-1}(x_{f-1}, y_{f-1})$ and $B_{f-2}(x_{f-1}+\Delta x_{f-2}, y_{f-1}+\Delta y_{f-2})$ for each block $B_{f-1}(x_{f-1}, y_{f-1})$ of the $f-1$th image. Now our objective is to find its projection position in the fth image.

Based on the 3-D reconstruction theory, the 3-D coordinates of each corresponding pair can usually be recovered by a simple triangulation algorithm [16]. The coordinates are then projected to the fth image to calculate the pair's projection position. Specifically, given the projection matrixes M_{f-2} and M_{f-1}, and the blocks' coordinates $(x_{f-1}+\Delta x_{f-2}, y_{f-1}+\Delta y_{f-2})$ and (x_{f-1}, y_{f-1}), the corresponding 3-D coordinates can be computed with (5) and (6) by a least-square method. Then, with the projection matrix M_f of the current camera and the calculated 3-D coordinates, we can calculate the projection position on the current image fth by (1). Note that in Section II the calculation is point based, whereas in this step we perform the calculation in block basis. This is because that our target is to predict the DV of a block for the coding purpose. The point based calculation can also be regarded as a special case of the block based calculation (with 1 × 1 block size).

After the projection, the offset between the corresponding block pair's position in the $f-1$th image and that in the fth image can be obtained:

$$v_f(x_{f-1}, y_{f-1}) = (x_f(B_{f-1}) - x_{f-1}, y_f(B_{f-1}) - y_{f-1}) \quad (8)$$

where $x_f(B_{f-1})$ and $y_f(B_{f-1})$ are the projected coordinates of the block $B_{f-1}(x_{f-1}, y_{f-1})$ in the fth image. This offset is regarded as a DV candidate, which will be used in the next step to obtain $\tilde{V}_f(i,j)$ through DV fusion.

In addition to the above-mentioned method, there are some other methods for finding the projection position. In one method, the trifocal tensor transfer of three images is used for projection computing [27], [28]. In fact, the above method is equivalent to the trifocal tensor transfer. Another method is to use the pair-wise epipolar relations, in which the projection position in the fth image is determined as the intersection of the two epipolar lines [16]. However, this computation is not always very well conditioned. When the point locates on the trifocal plane (i.e., the plane going through the three projection centers), the projection position is completely undetermined as the epipolar lines become parallel.

C. Step 3: Obtaining $\tilde{V}_f(i,j)$ Through DV Fusion

After the second step, we get many DV candidates. Each of them represents the displacement when a block of the $f-1$th image is projected to the fth image. Now our objective is to find the predicted DV $\tilde{V}_f(i,j)$ from these DV candidates for each coding block. However, it is difficult to get $\tilde{V}_f(i,j)$ directly from the candidates. There are two reasons. First, each coding block in the fth image locates in the rectangular grid (say, 8 × 8 or 16 × 16), whereas the DV candidates are often not on the grid. Second, due to occlusion and calculating errors, the DV candidates may distribute non-uniformly in the fth image. Hence, a DV fusion step is required to merge similar candidates into one predicted DV.

In our implementation, the value of $\tilde{V}_f(i,j)$ is calculated with three different cases according to the number n of DV candidates located in a coding block:

First, if there is no DV candidate in the coding block (i.e., $n = 0$), the block is regarded as unpredictable and $\tilde{V}_f(i,j)$ is set to zero. In other words, the proposed DV predictor will be disabled in this case due to occlusion or inaccurate calculation.

Second, if there is only one DV candidate (i.e., $n = 1$), then $\tilde{V}_f(i,j)$ will be directly set to that candidate.

Last, if there are multiple DV candidates in the coding block (i.e., $n \geq 2$), $\tilde{V}_f(i,j)$ will be fused by averaging all the candidates if the candidates satisfy

$$\frac{1}{n-1}\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n}(|x_i - x_j| + |y_i - y_j|) < \mathrm{Thres}_{\mathrm{DV}}. \quad (9)$$

Otherwise, the block is regarded as unpredictable and $\tilde{V}_f(i,j)$ is set to zero. In the above, (x_i, y_i) are the coordinates of the ith DV candidate and $\mathrm{Thres}_{\mathrm{DV}}$ is a fixed threshold, which equals 4.0 in our implementation.

[2] We propose this centroid-based epipolar line calculation in that the resulting epipolar line computed from the centroid rather than the top-left corner of the block can more precisely reflect the average epipolar constraint for a group of pixels.

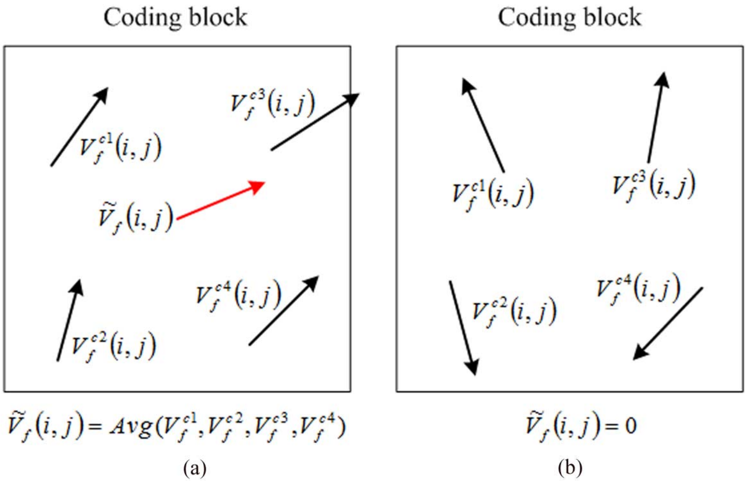


Fig. 8. Examples of DV fusion. (a) Four DV candidates can be merged into one predicted DV. (b) Four DV candidates cannot be merged into one predicted DV.

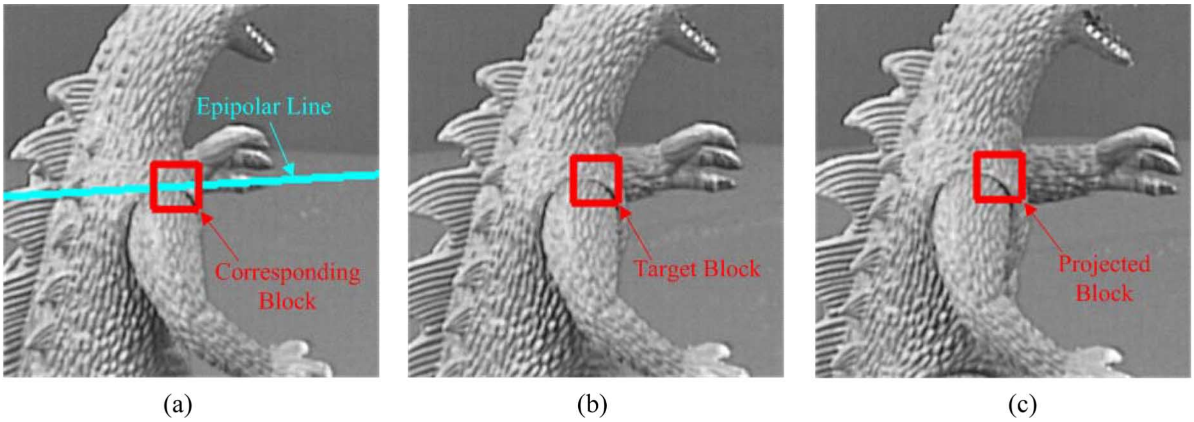


Fig. 9. Example of geometric prediction. (a) $f-2$th image. (b) $f-1$th image. (c) fth image (the current encoding image).

Fig. 8 shows two examples of DV fusion. Fig. 9 shows an example of our proposed geometric prediction procedure, where Fig. 9(a)–(c) are three consecutive images (the $f-2$th, $f-1$th, and fth image) from different viewpoints. We chose an image block on the shoulder of dinosaur of the $f-1$th image as the experimental target block [Fig. 9(b)]. Its corresponding block on the $f-2$th image can be obtained by searching around its epipolar line [Fig. 9(a)]. These two blocks together forms a corresponding block pair. We can then predict the projection position of this block pair on the fth image [Fig. 9(c)]. These three blocks are well matched, which indicates that the inter-viewpoint motion is predictable based on the multiview geometry.

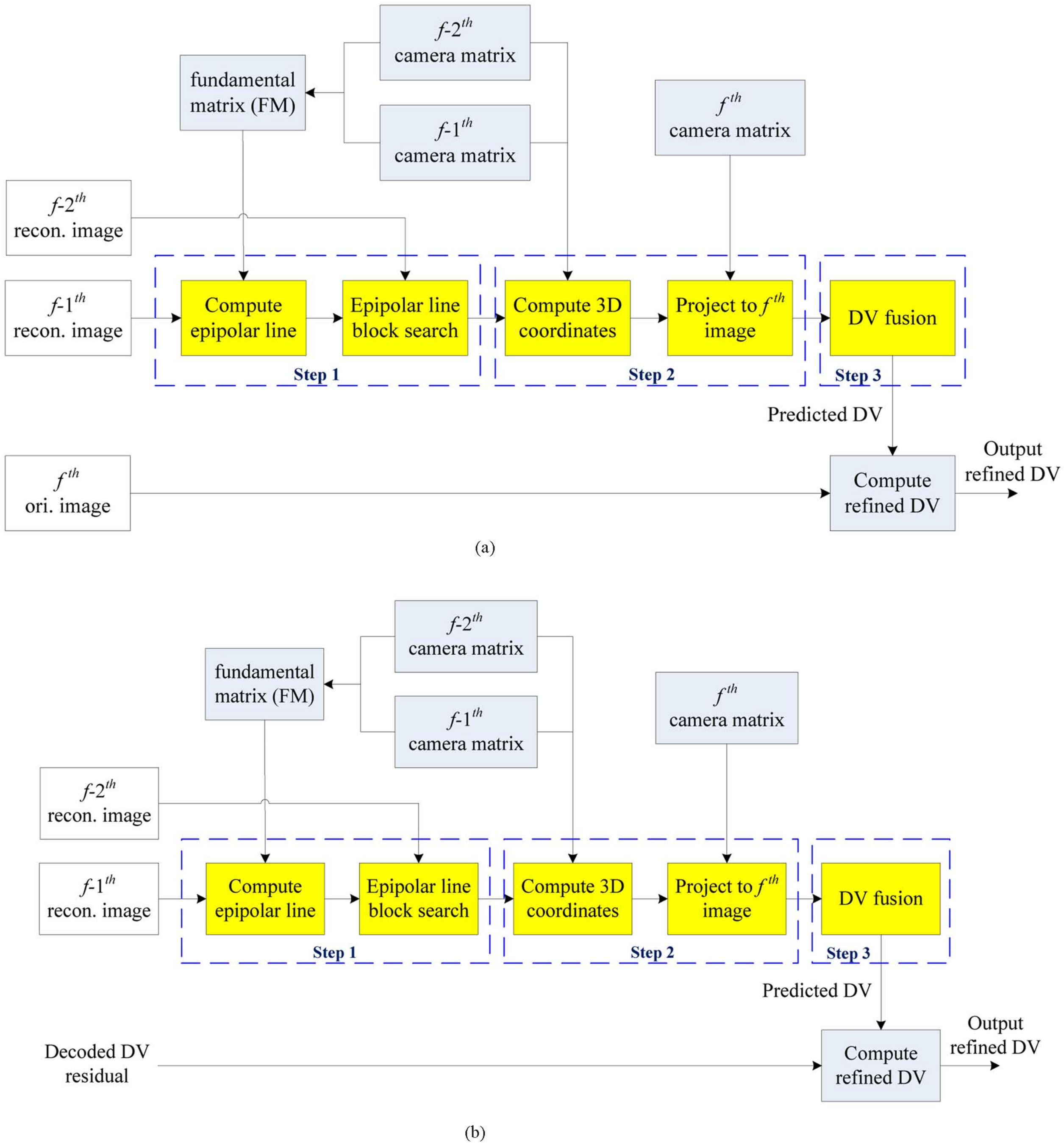
It is known that the projective geometric relations between views are independent of the scene structure. Therefore, the proposed DV predictor only relies on cameras' internal parameters and postures, which are uniquely defined by the projection matrices of the views. Moreover, the projection matrices can be computed offline through camera calibration [16].

V. Proposed Coding Framework and Integration With H.264/MPEG-4 AVC

Following the description of the new DV predictor given in Section IV, in this section, we present how to design a new framework for coding multiview image. We then use H.264/MPEG-4 AVC as an example to illustrate how the proposed framework can be integrated with the conventional video coding algorithms.

We design a new coding mode, the GP (geometric prediction) mode, to utilize the proposed DV predictor. As depicted in Fig. 10, in the GP mode, the proposed DV predictor is first used to calculate the geometric predicted DV $\tilde{V}_f(i,j)$ when a block $B_f(i,j)$ of the fth image is being processed. Then, a DV refinement process is performed in a square area centered at $\tilde{V}_f(i,j)$ to find the optimal DV. Subpixel accuracy compensation can also be performed. This is because $\tilde{V}_f(i,j)$ is usually not the optimal DV from the coding efficiency perspective due to calibration noise, geometric estimation noise, and varying lighting conditions. At each search step, the R-D optimization as illustrated in (7) is adopted, where $D(v)$ is counted as the sum of absolute differences (SAD) and $R(v)$ is counted as the number of bits representing the residual vector after being subtracted by $\tilde{V}_f(i,j)$. This refinement process is similar to the conventional motion estimation in which the median predicted MV is often used as the search starting point (or center of the search area). Finally, the residual vector between the searched optimal DV and $\tilde{V}_f(i,j)$ is encoded.

Incorporating the GP mode with the conventional INTER mode and INTRA mode, we design a new framework for coding multiview image. In the new framework, the first image is encoded as an I-frame using the INTRA mode. The second image



(a)

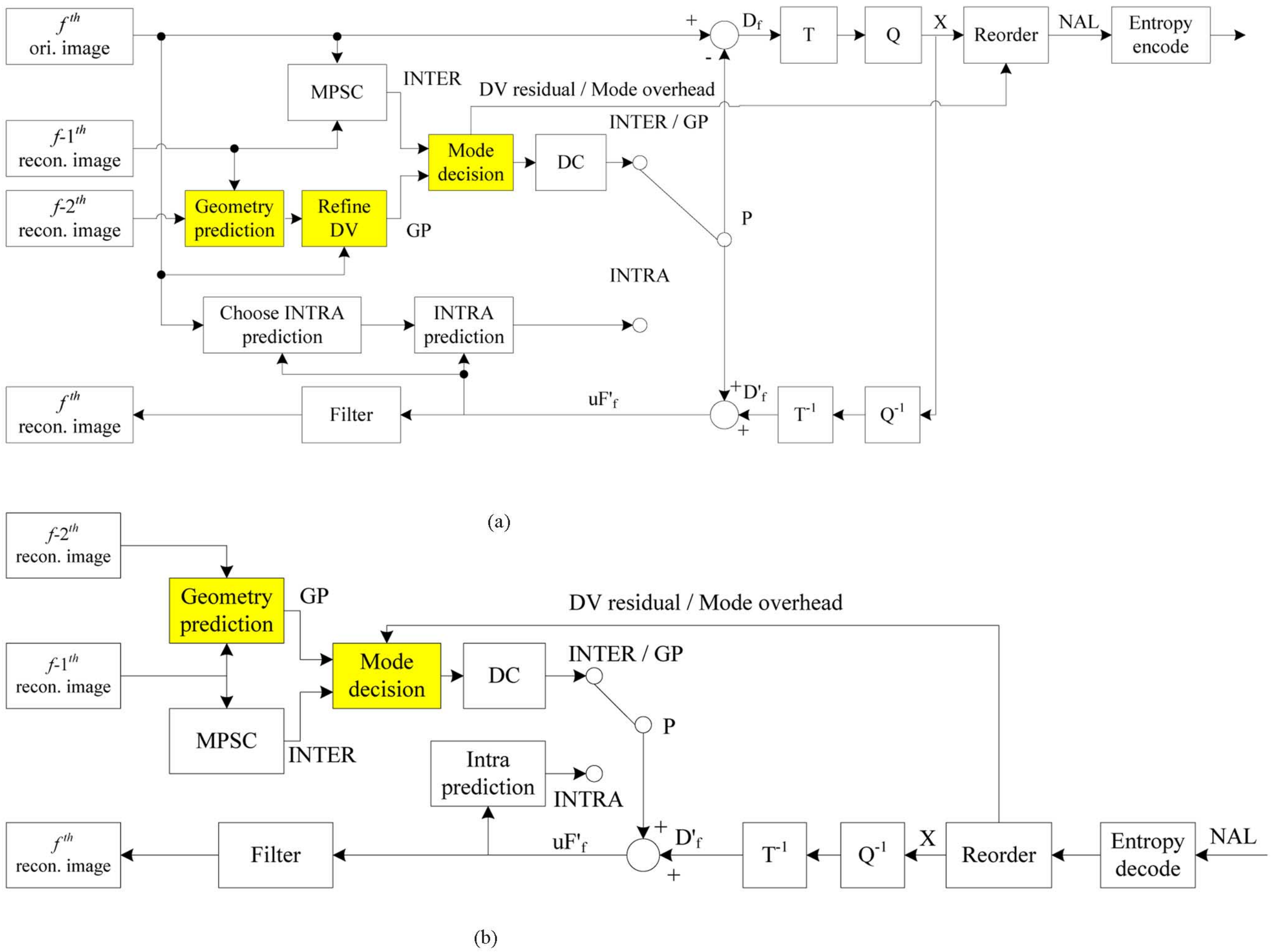
(b)

Fig. 10. The proposed GP coding mode: (a) the GP mode at encoder and (b) the GP mode at decoder.

is encoded as a P-frame using both the INTER and INTRA modes. Starting from the third image, all the three modes can be used. Taking the H.264/MPEG-4 AVC codec as one example, as shown in Fig. 11(a), we use a mode decision module to decide the optimal coding mode with the rate-distortion (R-D) criteria. After the mode decision, a one-bit overhead is generated for each block (partition) to signal whether the GP mode is used if the block (partition) is not coded in the INTRA mode. Disparity compensation is then performed to generate residual signals which will be further coded.

At the decoder side, an inverse process is performed [Fig. 11(b)]. The multiview geometry based DV predictor is repeated to get the same $\tilde{V}_f(i,j)$ as that at the encoder side. According to the overhead bit of each block, the decoder can identify whether the GP mode is used for coding the current block (partition). Then the actual DV is obtained by adding the refined vector onto the predicted DV (either from the proposed predictor or from the median predictor, according to its coding mode). Then, after the disparity compensation, the image is reconstructed at the decoder side.

Finally, we would like to discuss the decoding complexity issue resulting from the proposed coding framework. Among the three steps of the proposed predictor, the first step, searching for corresponding block pairs, is the most time consuming one.



(a)

(b)

Fig. 11. Integration with the H.264/MPEG-4 AVC framework: (a) encoder (b) decoder.

Fortunately, its search area is a narrow window and its search complexity can also be greatly reduced. A recent research on fast disparity estimation [23] can be used to speed up the search process.

VI. Experimental Results and Discussions

We select two public multiview image sequences, *Dinosaur* (720 × 576, 36 viewpoints) [2] and *House* (768 × 576, 10 viewpoints) [29], and two multiview video sequences, *Breakdancer* (1024 × 768, 8 viewpoints) [30] and *Ballroom* (640 × 480, 8 viewpoints) [31], for extensively evaluating the performance of our proposed coding algorithm. Since we focus on MIC in this paper, for each multiview video sequence, we only select the first image of each viewpoint for test. As mentioned before, these images from different viewpoints at one certain time instance can also be regarded as a multiview image sequence. Hence, we have four multiview image sequences, *Dinosaur, House, Breakdancer*, and *Ballroom*, with different number of images (or, viewpoints), 36, 10, 8, and 8, respectively.

The H.264/MPEG-4 AVC reference software JM version 10.2 [32] is used in our experiments. Baseline profile with integer-pixel compensation plus CABAC entropy encoding is used to configure the encoder. R-D optimization is enabled. During the coding process, the first image is encoded as an I-frame, the second image is encoded as a P-frame, and all the rest images in a multiview image sequence are encoded with the proposed method. For our proposed method, horizontal search range (HSR) is set to the same value as the search range for P-frame coding in H.264/MPEG-4 AVC, while VSR is reduced to 4.

A. DV Predictor Efficiency Evaluation

In our first experiment, we evaluate the performance of the proposed DV predictor. Table I lists the percentages of different numbers of DV candidates per coding block before DV fusion at different coding block sizes. We are more interested in the block percentage when there is only one DV candidate per coding block, because this reflects the DV projection accuracy in some sense. We can see that more than 54% of blocks have only one DV candidate, and this value decreases when the block size is reduced. The reason is that searching with a smaller block size is less accurate to find the true corresponding block. A sample image is also shown in Fig. 12, where only the blocks with one DV candidate are displayed while the rest blocks with no DV candidate or multiple DV candidates are filled with black. From the figure, we can find that most blocks on the foreground can be accurately predicted. One the other hand, we also observe that many blocks with no DV candidate or multiple DV

TABLE I
BLOCK PERCENTAGES OF DIFFERENT NUMBERS OF DV CANDIDATES PER CODING BLOCK ($\mathrm{QP} = 40$)

Sequence	Coding block size	Percentages of different numbers of DV candidates		
		DV Num = 0	DV Num = 1	DV Num > 1
Dinosaur	16×16	19.94%	68.46%	11.60%
	8×8	23.77%	62.65%	13.58%
	4×4	24.91%	61.86%	13.23%
House	16×16	19.73%	68.81%	11.46%
	8×8	28.23%	56.31%	15.47%
	4×4	34.85%	50.55%	14.60%
Break dancer	16×16	27.70%	54.79%	17.51%
	8×8	33.10%	47.40%	19.51%
	4×4	35.93%	45.05%	19.02%
Ballroom	16×16	26.17%	67.58%	6.25%
	8×8	25.67%	60.63%	13.71%
	4×4	30.64%	53.97%	15.39%

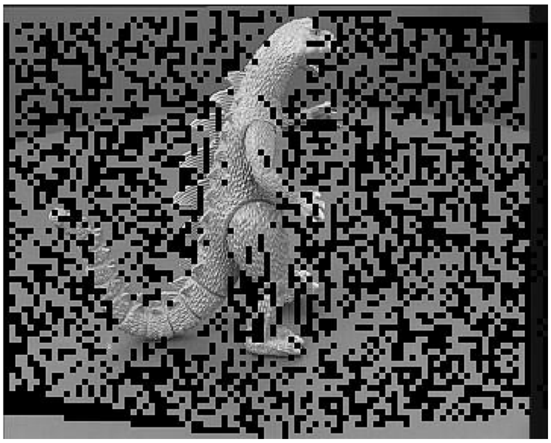


Fig. 12. Sample image before DV fusion. Black blocks indicate the blocks that have no DV candidate or multiple DV candidates.

candidates locate not only in the occlusion region but also in the background. The main reason is that the background region lacks of texture information, and thus it is difficult for a simple searching algorithm to find the true corresponding block just using the minimum MSE criterion. Although the prediction accuracy at less-textured regions can be improved by choosing other searching algorithms [25], the impact might not be that significant since these regions can be easily coded.

We then analyze the length of the residual DV after DV prediction. In general, shorter DV will result in fewer bits after entropy coding. Therefore, we can roughly evaluate the performance of the proposed predictor by comparing the total length of the residual DV outputting respectively from the median predictor and from the proposed predictor. The total length of the residual DV is calculated by

$$L = \sum (|\Delta V_x| + |\Delta V_y|) \tag{10}$$

TABLE II
COMPARISON THE LENGTH OF RESIDUAL DV AFTER MEDIAN PREDICTION AND GEOMETRIC PREDICTION (THE THIRD IMAGE OF EACH SEQUENCE, $\mathrm{QP} = 40$)

Sequence	HSR	Median prediction	Geometric prediction	Gain
Dinosaur	16	5362	3796	29.21%
House	64	63784	49301	22.71%
Break dancer	16	29854	22368	25.08%
Ballroom	16	18559	15013	19.11%

where ΔV_x and ΔV_y denote the residual DVs in the horizontal axis and vertical axis. As shown in Table II, the total length after the geometric prediction is less than that after the median prediction.

B. R-D Performance Evaluation

In this experiment, we compare the R-D performance of our proposed coding algorithm with H.264/MPEG-4 AVC. As shown in Fig. 13, our proposed coding algorithm significantly outperforms H.264/MPEG-4 AVC with a maximum 1.5-dB improvement and a 0.9-dB improvement on average in the low bit rate. It can also be seen that the gain diminishes when bit rate increases. The reason is that, in high bit rate coding the residual information produces the most bits of the outputting bit stream. In the comparison, all the overhead bits have been counted in, including bits signaling whether the proposed new coding mode is used for a certain block and bits representing the projection matrixes. The overhead costs 2% $\sim$ 4% of the total length in the low bit rate case. We also evaluate the visual quality of different coding methods. As shown in Fig. 14, our proposed algorithm gives better visual quality than H.264/MPEG-4 AVC at low bit rate.

We also evaluate our algorithm under different viewing angle intervals. In our experiment, different viewing angle intervals are attained by sampling an available multiview image sequence. For example, given the *Dinosaur* sequence whose original viewing angle interval is 10°, a 20° interval sequence can be attained by skipping one image for every two images. As listed in Table III, the coding efficiency improvement of the proposed algorithm decreases when the viewing angle interval increases. For instance, for a given QP value 50, the bit rate reduction of our algorithm under 10°, 20°, and 30° interval is 20.5%, 6.5%, and 6.0%, respectively. One of the reasons is that, when the viewing angle interval increases, more blocks are coded in the INTRA mode since block-based disparity compensation becomes less effective (Fig. 15). This can also explain why our proposed algorithm can achieve larger gain for the sequences of *Breakdancer* and *Ballroom* than that for the sequences of *Dinosaur* and *House*, because the viewing angle intervals of the former two sequences are less than those of the latter two sequences. Of course, the viewing angle interval for typical multiview image and video sequences is often not larger than 10° (3°–5° or less is even more popular in current available multiview video sequences [1]). Hence, our proposed algorithm can guarantee a significant improvement in most cases.

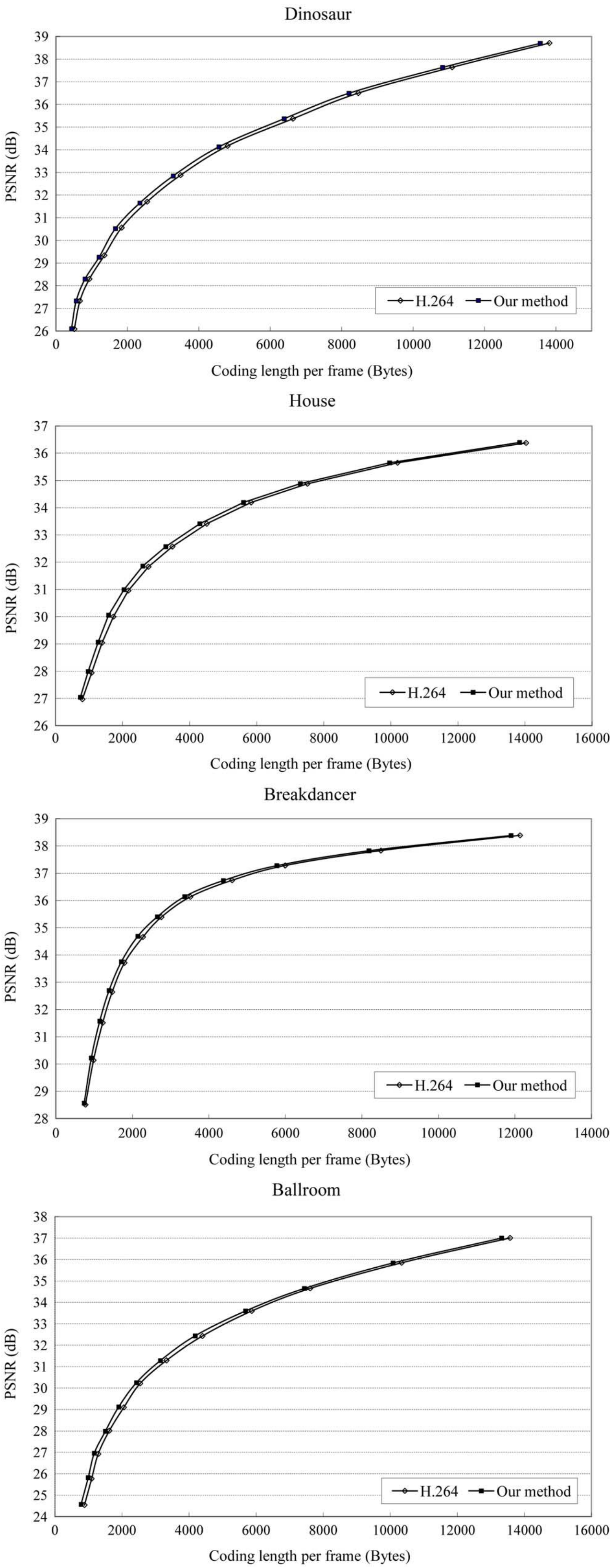


Fig. 13. R-D performance evaluation.

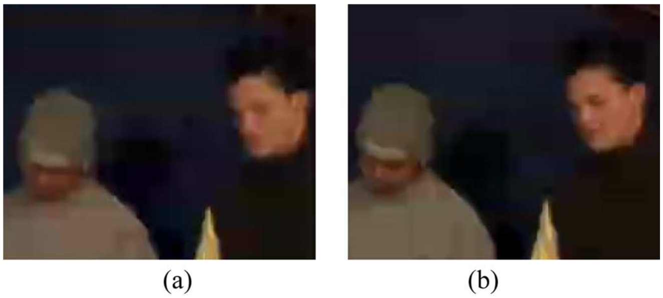


Fig. 14. Visual quality comparison—Breakdancer (encoded at 500 kbps). (a) H.264. (b) Our method.

TABLE III
PERFORMANCE EVALUATION UNDER DIFFERENT VIEWING ANGLES (DINOSAUR SEQUENCE)

Viewing angle	QP	H.264/MPEG-4 AVC		Our method	
		Length (K bytes)	PSNR (dB)	Length (K bytes)	PSNR (dB)
10°	50	3.52	26.11	2.92	26.14
	48	4.62	27.38	3.81	27.31
	46	6.40	28.31	5.50	28.28
	44	9.38	29.29	8.36	29.28
	42	13.03	30.51	11.65	30.48
	40	18.49	31.65	16.81	31.62
	38	25.56	32.80	23.59	32.78
20°	50	4.77	26.28	4.48	26.29
	48	6.43	27.58	5.88	27.59
	46	9.09	28.54	8.52	28.50
	44	13.33	29.60	12.51	29.52
	42	17.97	30.84	17.31	30.79
	40	25.06	32.00	24.41	31.98
	38	33.62	33.13	32.87	33.13
30°	50	5.61	26.41	5.29	26.45
	48	7.60	27.78	7.06	27.69
	46	10.68	28.69	10.22	28.66
	44	15.23	29.76	14.85	29.73
	42	20.33	30.97	20.12	31.01
	40	27.92	32.16	27.58	32.16
	38	36.79	33.27	36.63	33.28

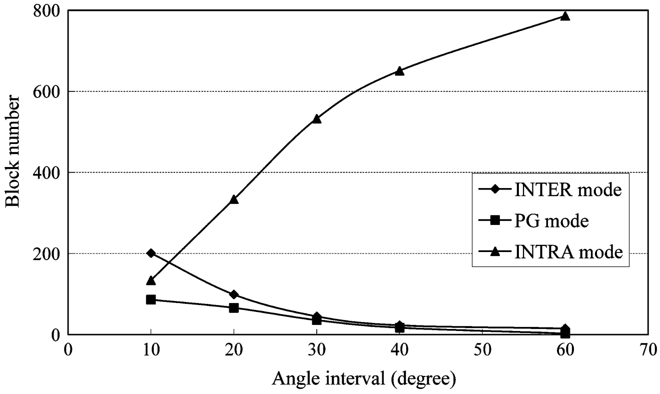


Fig. 15. Numbers of blocks for different coding modes when viewing angle increases (Dinosaur sequence, $\mathrm{QP} = 49$).

C. Complexity Evaluation

The proposed algorithm improves coding efficiency, but it also brings additional computation to both encoder and decoder, especially at the decoder side. Of all the increased complexity, block matching in Step 1 is the most computation intensive operation. We tested the complexity of our algorithm and found that, when the full search is used for block matching, the encoding complexity increases 30% and the decoding complexity

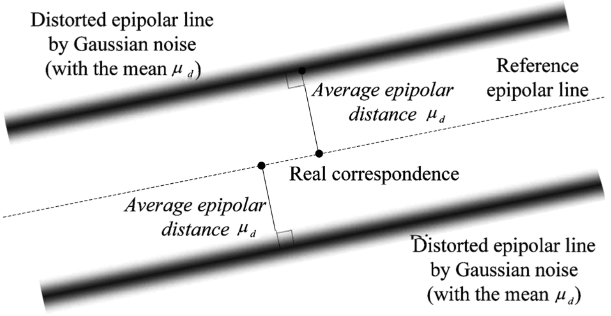


Fig. 16. Average epipolar distance and distorted epipolar lines by Gaussian noise.

increases over 1000%. On the other hand, when a fast searching algorithm (e.g., our previous work in [23]) is used, the decoding complexity just increases 100% while the coding efficiency only decreases less than 0.1 dB comparing with the full search.

D. Noise Sensitivity Evaluation

Although the techniques of camera calibration are not within the scope of this paper, it is important to examine the robustness of the proposed method to calibration noise. We believe a noise-insensitive property is highly desired to guarantee the advantages of the proposed algorithm, in case that camera geometry is not known and need to be estimated.

The robustness of our algorithm mainly depends on the correspondence matching accuracy, which is determined by the correctness of FM. Therefore, to evaluate the robustness, we test our proposed algorithm under different designed noise levels in FM estimation. Since the calibrated camera projection matrixes are available for all the sequences adopted in this paper, we can precisely derive the FM from them, and the epipolar lines calculated from these FM can be regarded as solid references. Considering that average epipolar distance (the average distance of a point to its corresponding epipolar line, denoted as μ_d hereinafter) is one of the common metrics adopted to calibrate the precision of the estimated FM [22], we distort the reference epipolar lines by imposing Gaussian noise with variances ranging from 1 to 4 pixels (Fig. 16).

It can be found from Fig. 17 that the degradation of coding efficiency due to a very noisy FM estimation $(\mu_d = 4)$ is limited and the performance is still better than that of the H.264/MPEG-4 AVC codec. In a good review of FM estimation techniques [22], we can find that even a suboptimal FM estimation algorithm can achieve a high estimation precision [i.e., a small average epipolar distance $(\mu_d \leq 1)$]. This fact and the above results indicate that our proposed coding algorithm is insensitive to typical FM estimation noise. Our algorithm can be applied to multiview image and video coding even if the camera geometry is not known and need to be estimated.

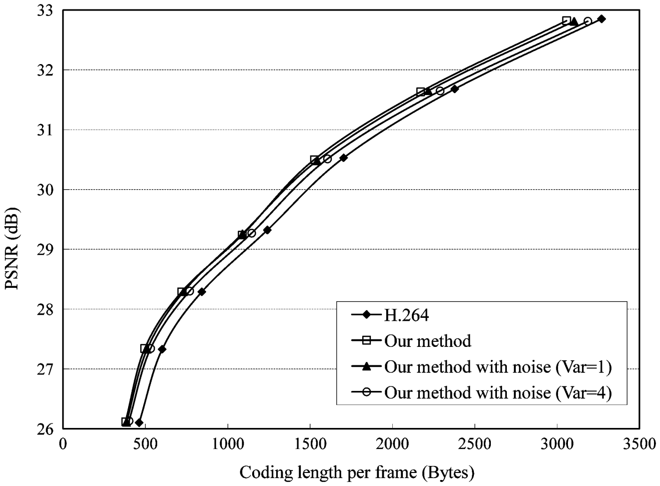


Fig. 17. R-D performance evaluation under different noise levels (dinosaur sequence).

VII. CONCLUSION

A multiview encoder has been proposed in this paper, which exploits the inter-viewpoint correlation by a multiview geometry based DV predictor. The proposed encoder can accurately track the inter-viewpoint disparity and thus largely reduce the DV coding cost in a disparity compensated coding scheme. Experimental results show that the proposed method outperforms the state-of-the-art H.264/MPEG-4 AVC codec by significant margins, while keeping high robustness in real applications when the camera geometry is unknown.

In addition to MIC, the proposed idea is also valid for MVC, as long as the inter-viewpoint correlation is exploited. Moreover, while our focus in this paper is to design an efficient multiview encoder through the multiview geometry based DV predictor, we believe that the idea can also be combined with some existing disparity-compensated coding methods to achieve better compression performance.

ACKNOWLEDGMENT

The authors would like to thank their colleague D. Daniels for his proofreading of this paper.

REFERENCES

[1] J.-G. Lou, H. Cai, and J. Li, "A real-time interactive multiview video system," in *Proc. 13th ACM Int. Conf. Multimedia (ACMMM 2005)*, Singapore, Nov. 2005, pp. 161–170.

[2] Dinosaur Sequence from University of Hannover [Online]. Available: http://www.robots.ox.ac.uk/~vgg/data.html 2007

[3] A. Smolić and D. McCutchen, "3 DAV exploration of video-based rendering technology in MPEG," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 348–356, Mar. 2004.

[4] A. Smolić and P. Kauff, "Interactive 3-D video representation and coding technologies," *Proc.IEEE, Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. 1, pp. 98–110, Jan. 2005.

[5] H. Aydinoglu and M. H. Hayes, "Stereo image coding: A projection approach," *IEEE Trans. Image Process.*, vol. 7, no. 4, pp. 506–516, Apr. 1998.

[6] *Proposed Draft Amendment No. 3 to 13818 (Multiview Profile)*, International Organization for Standardization, ISO/IEC JTC1/SC29/WG11, N1088, Nov. 1995.

[7] S. Li, M. Yu, G. Jiang, T.-Y. Choi, and Y.-D. Kim, "Approach to H.264-based stereoscopic video coding," in *Proc. ICIG*, Dec. 2004, pp. 365–368.

[8] Y. Luo, Z. Zhang, and P. An, "Stereo video coding based on frame estimation and interpolation," *IEEE Trans. Broadcast.*, vol. 49, no. 1, pp. 14–21, Mar. 2003.

[9] G. Li and Y. He, "A novel multiview video coding scheme based on H.264," *Proc. ICICS*, vol. 1, pp. 493–497, Dec. 2003.

[10] X. Tong and R. M. Gray, "Coding of multiview images for immersive viewing," in *Proc. IEEE \Int. Conf. Acoustic, Speech Signal Process. (ICASSP)*, Istanbul, Turkey, Jun. 2000, vol. 4, pp. 1879–1882.

[11] W. Woo and A. Ortega, "Overlapped block disparity compensation with adaptive windows for stereo image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 2, pp. 194–200, Mar. 2000.

[12] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Light field compression using disparity-compensated lifting and shape adaptation," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 793–806, Apr. 2006.

[13] H. Aydinoglu and M. H. Hayes, "Stereo image coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, Apr. 1995, vol. I, pp. 247–250.

[14] H. Aydinoglu and M. H. Hayes, "Compression of multiview images," in *Proc. Int. Conf. Image Process.*, Nov. 13–16, 1994, vol. 2, pp. 385–389.

[15] M. H. V. Duarte, M. B. Carvalho, E. A. B. da Silva, C. L. Pagliari, and G. V. Mendonca, "Multiscale recurrent patterns applied to stereo image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1434–1447, Nov. 2005.

[16] R. Hartley and A. Zisserman*, Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[17] M. Spetsakis and J. Aloimonos, "Structure from motion using line correspondences," *Int. J. Comput. Vis.*, vol. 4, no. 3, pp. 171–183, 1990.

[18] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video interpolation using a layered representation," *Proc. SIGGRAPH*, pp. 600–608, 2004.

[19] S. E. Chen and L. Williams, "View interpolation for image synthesis," *Proc. SIGGRAPH*, pp. 279–288, 1993.

[20] M. magnor, P. Ramanathan, and B. Girod, "Multiview coding for image-based rendering using 3-D scene geometry," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1092–1106, Nov. 2003.

[21] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun, "Extensions of H.264/AVC for multiview video compression," in *Proc. Int. Conf. Image Process.*, Oct. 8–11, 2006, pp. 2981–2984.

[22] Z. Zhang, "Determine the epipolar geometry and its uncertainty: a review," *Int. J. Comput. Vis.*, vol. 27, no. 2, pp. 161–195, Mar. 1998.

[23] J. Lu, H. Cai, J.-G. Lou, and J. Li, "An epipolar geometry-based fast disparity estimation algorithm for multiview image and video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 737–750, Jun. 2007.

[24] *Advanced Video Coding for Generic Audio-Visual Services*, Int. Telecommun. Union-Telecommun. (ITU-T) and Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), Recommendation H.264 and ISO/IEC 14996-10 AVC, 2003.

[25] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1-3, pp. 7–42, Jun. 2002.

[26] M. Ouali, D. Ziou, and C. Laurgeau, "A cooperative multiscale phase-based disparity algorithm," in *Proc. Int. Conf. Image Process.*, Kobe, Japan, Oct. 1999, vol. 3, pp. 145–149.

[27] P. Torr and A. Zisserman, "Robust parameterization and computation of the trifocal tensor," *Image Vis. Comput.*, vol. 15, pp. 591–605, 1997.

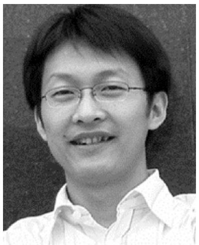
[28] P. Torr and A. Zisserman, "Robust computation and parameterization of multiple view relations," in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 727–732.

[29] House Sequence: [Online]. Available: http://www.robots.ox.ac.uk/~vgg/data.html

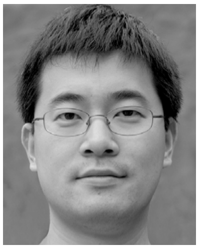
[30] Breakdancer Sequence: [Online]. Available: http://research.microsoft.com/vision/InteractiveVisualMediaGroup/3 DVideoDownload/

[31] Ballroom Sequence [Online]. Available: ftp://ftp.merl.com/pub/avetro/mvc-testseq

[32] JM Reference Software Ver. 10.2. [Online]. Available: http://iphome.hhi.de/suehring/tml/download/jm10.2.zip



Xing San received the B.S. degree and the Ph.D. degree from University of Science and Technology of China (USTC), Hefei, China, in 2002 and 2007, respectively, both in electrical engineering. In 2004 and 2006, he was a visiting student at Microsoft Research Asia, Beijing, China. He joined Omnivision Technologies, Inc., in 2007. His research interests include image/video coding, image processing, and computer vision.



Hua Cai (M'04) received the B.S. degree from the Shanghai Jiaotong University, Shanghai, China, in 1999, and the Ph.D. degree from the Hong Kong University of Science and Technology (HKUST) in 2003, both in electrical and electronic engineering.

He joined Microsoft Research Asia, Beijing, China, in December 2003 and is currently a Researcher with the Media Communication Group. His main research interests include signal processing, source coding, multiview video coding and transmission, multiview video system, and mobile media computing.

Jian-Guang Lou (M'04) received the B.S. degree and the M.S. degree in automation from Zhejiang University, Hangzhou, China, in 1997 and 2000, respectively, and the Ph.D. degree from Institution of Automation, Chinese Academy of Science, Beijing, China, in 2003.

He joined Microsoft Research Asia, Beijing, China, in 2003. His main research interests include computer vision, image processing, multiview video, and multimedia systems.

Jiang Li (SM'04) received B.S. degrees in applied physics and applied mathematics from Tsinghua University, China, in 1989, the M.S. degree in optics from the Physics Department, Zhejiang University, Hangzhou, China, in 1992, and the Ph.D. degree in applied mathematics from the State Key Laboratory of Computer Aided Design and Computer Graphics, Zhejiang University, in 1998.

He joined the Visual Computing Group, Microsoft Research Asia, Beijing, China, as Researcher in January 1999. He became Lead Researcher of the Internet Media Group in 2002 and Research Manager of the Media Communication Group in 2004. He invented Bi-level video, Portrait video, and Watercolor video, which are suitable to very low-bandwidth networks. He released Microsoft Portrait, the first video communication software prototype on Pocket PC and Smartphone. He is now leading the Media Communication Group in the research of mobile video communication, multiparty conferencing, multiview video, and peer-to-peer streaming. Before joining Microsoft, he was an Associate Professor with the Physics Department, Zhejiang University.