Fast image warping using adaptive partial matching

Dong-Keun Lim
Samsung Electronics Company
Limited Digital Media Research
and Development Center
416 Maetan-3Dong
Paldal-Gu, Suwon
Kyungi-Do, 442-742
Korea
E-mail: dk2003.lim@samsung.com

Yo-Sung Ho
Kwangju Institute of Science and
Technology
Department of Information and
Communications
1 Oryong-dong, Puk-gu
Kwangju, 500-712
Korea

Abstract. The block-matching motion estimation algorithm using a translational motion model cannot provide acceptable image quality in low bit-rate coding. To improve coding performance, we can use image warping using affine transformation as a more complicated motion model than the translational motion model. However, when some node points are flipped over in the image warping method, it creates severe deformations of the mesh structure and brings image degradations. It also requires large computational complexity. We show that wide search ranges do not always improve reconstructed image quality due to the large deformation of meshes. We analyze optimal search ranges according to frame difference and decide variable search ranges adaptively to get a motion vector at each node point. Since the block with a larger error than a threshold makes a large distortion, we give higher priority during motion estimation at a certain rate. To reduce computational complexity, we also introduce an adaptive partial matching method instead of applying the hexagonal matching method on the whole image. As a result, we develop an effective image warping method in terms of computational complexity, reasonable reconstructed image quality, and fewer number of coding bits.
[DOI: 10.1117/1.1645251]

Subject terms: image warping; affine transform; motion estimation; partial matching.

Paper 030196 received Apr. 30, 2003; revised manuscript received Sep. 9, 2003; accepted for publication Sep. 9, 2003.

1 Introduction

One of the important developments for image coding[1–3] is a mathematical model describing the motion of objects. Because of the requirements for simplicity and real-time processing, translational motion models have mainly been investigated for image coding. A translational movement generates a frame-to-frame displacement of the moving object. Several algorithms have been proposed to estimate the amount of displacement between two successive frames.

A block-matching algorithm (BMA) is mainly employed for video compression. The BMA adopts a translational motion model to describe the motion of blocks in image objects. The problems of BMA, such as blocking artifacts, are caused by the insufficiency of the motion model to represent the real world. To overcome this defect, many coding methods use more sophisticated motion models. We use affine transformation as the more complicated motion model. Image warping using affine transformation divides the image frame into many triangular patches and then estimates motion vectors of the triangular patches by their deformations. The deformation of a triangular patch can be described as affine transformation.[1–6] Since there are significant deformations of mesh structure when some node points are flipped over, they bring image degradations.

We propose a new image warping method based on affine transformation. The reconstructed image is formed by geometrically transforming the previous image based on mapping a selected set of image points between the previous and current images. The main proposed scheme is an adaptive partial matching method. We develop a more effective motion estimation method than conventional image warping methods in terms of computational complexity, comparable image quality, and fewer number of sending bits. By simulating a video coder combined in the proposed scheme with a discrete cosine transform (DCT), we show that the proposed scheme is a suitable method at very low bit-rate coding. The high performance is due to the fact that affine transformation maintains the acceptable or higher level of prediction error, even when the number of node points is reduced to about half the number of blocks used in the previously developed image warping methods.[2,3,6]

2 Image Warping

2.1 *Concept of Image Warping*

In a video coding scheme with image warping, a predicted image is formed from the previous image using geometric transformations.[1–9] The transformation specifies spatial relationships between each point in the previous and predicted images. For each pixel in the predicted image, we calculate the corresponding spatial position in the previous image, and an image value at the position using the spatial relationships. Since the image value usually does not lie on the integer pixel node, we predict the value using a bilinear interpolation. The transformation parameters must be transmitted to the decoder as overhead information. Since the transformation parameters require more bits to represent

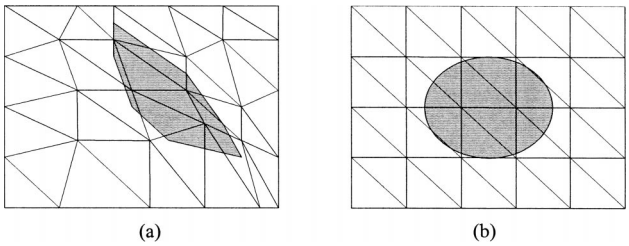

Fig. 1 Deformed Patches in Image Warping: (a) irregular patches in previous frame and (b) regular patches in current frame.

them, we transmit motion vectors for a selected subset of image points called node points or control points. These points are used as fixed points in the transformation. The displacements between pairs of corresponding node points constitute the motion prediction method.

Figure 1 illustrates the prediction method. Uniformly spaced node points have been selected from the current image, and their spatial matches have been searched from the previous image. The geometric transformation is applied to blocks of the previous image to make a predicted image. Uniform translational displacement of the node points produces a result close to that of the block-matching algorithm. Relative differences in the displacements of the node points can distort the mesh structure and compensate changes in an object shape. The whole prediction process can be divided into three distinct steps, such as selection of the node points, pixel matching, and geometric transformation.

2.2 *Mathematical Notation*

Motion compensation methods can be defined as techniques that divide images into local regions and estimate a set of motion parameters for each region. The procedure synthesizes the predicted image of the k'th frame $\hat{I}_k(x,y)$ from the decoded image of the previous frame $\tilde{I}_{k-1}(x',y')$. This process can be written as

$$\hat{I}_k(x,y)=\tilde{I}_{k-1}(x',y')=\tilde{I}_{k-1}[f(x,y),g(x,y)], \tag{1}$$

where the geometric relationship between $\hat{I}_k(x,y)$ and $\tilde{I}_{k-1}(x',y')$ is defined by the transformation functions $x'=f(x,y)$, $y'=g(x,y)$.

2.3 *Affine Transformation*

For affine transformation, we usually perform the following matrix operations[10,11]:

$$x=a_{11}u+a_{21}v+a_{31}=f(u,v),$$

$$y=a_{12}u+a_{22}v+a_{32}=g(u,v), \tag{2}$$

or

$$[x,y,1]=[u,v,1]\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix}. \tag{3}$$

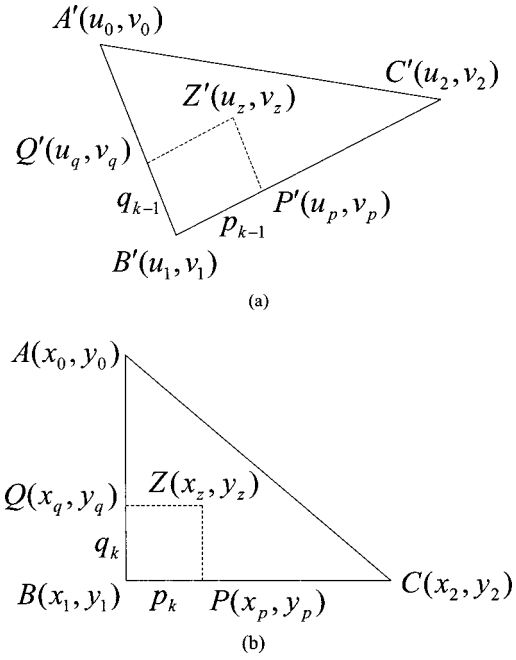

Fig. 2 Deformation of patches in image warping: (a) previous image patch ($t=k-1$) and (b) current image patch ($t=k$).

Three vertices in a triangle specify a geometrical affine mapping, as shown in Fig. 2.

The coefficients are simply determined by solving the following linear equations.

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}=\begin{bmatrix} u_0 & v_0 & 1 \\ u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \end{bmatrix}\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix}, \tag{4}$$

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix} = \frac{1}{\Delta}\begin{bmatrix} v_1-v_2 & v_2-v_0 & v_0-v_1 \\ u_2-u_1 & u_0-u_2 & u_1-u_0 \\ u_1v_2-u_2v_1 & u_2v_0-u_0v_2 & u_2v_1-u_1v_0 \end{bmatrix} \times\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}, \tag{5}$$

where

$$\Delta=u_0(v_1-v_2)-v_0(u_1-u_2)+(u_1v_1-u_2v_1). \tag{6}$$

From Eqs. (3) and (5), we can obtain motion vectors inside the triangular patch.

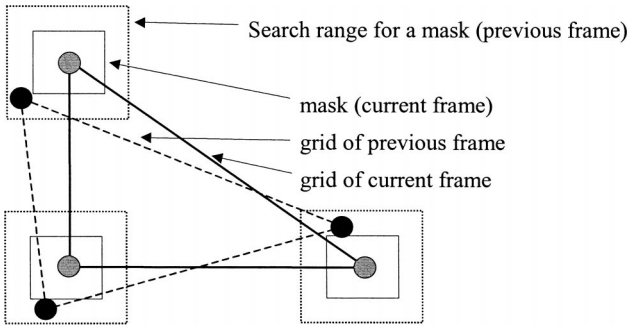

Fig. 3 Coarse motion estimation.

3 Adaptive Partial Matching

3.1 *Hexagonal Matching Method*

In image warping, motion estimation of the node points consists of coarse motion estimation by a block-matching algorithm, and a refinement of the estimated motion vectors by a hexagonal matching method.[2] The block-matching algorithm in coarse motion estimation is applied on the block that includes a node point in its center, as shown in Fig. 3. A mask has the same meaning with a block in the block-matching algorithm.

In refinement, we first move a node point to an adjacent point, updating the deformation of the patches in the hexagon, as shown in Fig. 4. We synthesize the predicted image inside the hexagon by image warping. We calculate a mean absolute difference inside the hexagon between the current and predicted images, search an optimum position that minimizes this error, and register the optimum position as a new location. The refinement process is iteratively applied to the next node point. The improvement of the prediction error reduces as the iteration continues. However, the computational cost is not cheap because of the repeated matching process.

3.2 *Adaptive Partial Matching Method*

The hexagonal matching method brings a higher image quality than the block-matching algorithm and other image warping methods. It, however, requires a large amount of computational complexities. There are several possibilities that are simple but give slightly less image quality. We now propose a new scheme called an adaptive partial matching (APM) method. The main procedures are shown in Fig. 5.

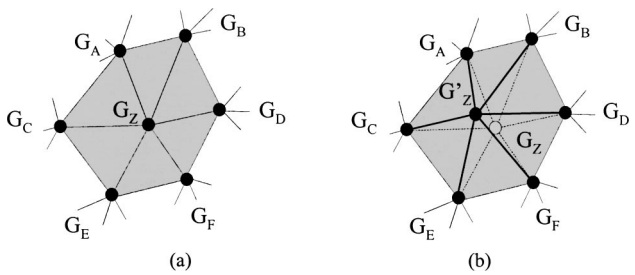

Fig. 4 Refinement process of the hexagonal matching[2]: (a) patches before processing and (b) patches after processing.

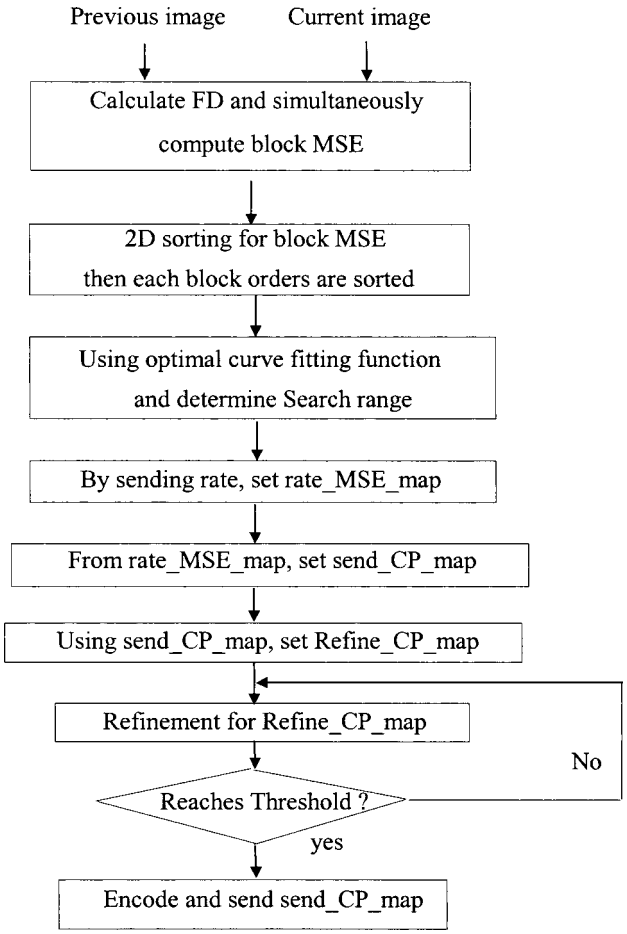

Fig. 5 An adaptive partial matching method.

We use an adaptively variable search range to obtain a motion vector of each node point at the coarse motion estimation.

3.3 *Selection of Adaptive Variable Search Range*

We first consider an effect of the search range. Although it will saturate or slightly increase, a peak signal-to-noise ratio (PSNR) of the block-matching algorithm (BMA) always increases when we increase a search range. Unlike BMA, the image warping method does not estimate motion vectors using a real prediction error. As shown in Figs. 6 and 7, we note that the PSNR of the image warping method does not always increase according to an increase of the search range. An unnecessary increase of the search range does not always improve quality. There will be a point where the maximum performance is given. By arranging this point, some optimal search ranges can be determined.

The analysis of the phenomena is as follows. The goal using spatial transformation in a mesh-based procedure is to minimize the interpolation errors from nodal values. However, the minimization of this error alone will cause certain nodes to get very close to each other. In the extreme case, as shown in Fig. 8, certain trajectories of adjacent nodes may cross each other, and elements will become flipped over. Node C destroys the mesh structure. It will introduce significant image degradations.

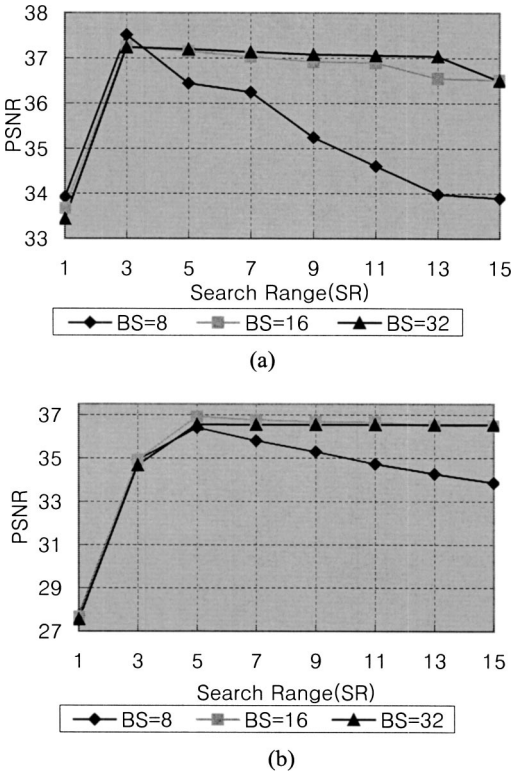

Fig. 6 PSNR versus search range (SR) and block size (BS) for Miss America: (a) Miss America 1 and 4 (FD=31.23 dB) and (b) Miss America 70 and 73 (FD=25.59 dB).

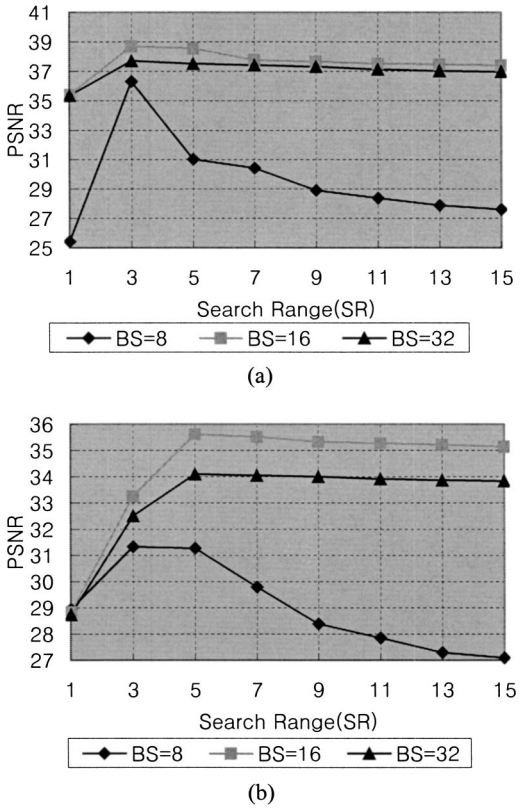

Fig. 7 PSNR versus SR and BS for Claire: (a) Claire 46 and 49 (FD=31.55 dB) and (b) Claire 1 and 4 (FD=28.53 dB).

Now we derive adaptive search ranges as a function of frame difference (FD). We first obtain distributions of the optimal search range according to several (FDs), as shown in Fig. 9. To make a mathematical model of the optimal search range, we use the least-mean square (LMS) method.

The LMS algorithm is the simplest and perhaps the most universally applicable adaptive algorithm. For the LMS algorithm, the corresponding input-output relationship is described by

$$y=\sum_{i=0}^{L} a_i x^i, \tag{7}$$

where x is the input value and y is the estimated output value using the L'th-order polynomial.

The technique utilized by the LMS algorithm to update coefficients is based on the method of steepest decent. This can be described in algorithmic form using vector notation as follows:

$$A_{k+1}=A_k-\mu\nabla_k, \tag{8}$$

where the coefficient vector is defined as

$$A_k=[a_0(k)\ldots a_L(k)]^T, \tag{9}$$

and gradient vector can be formulated as

$$\nabla_k=\frac{\partial E[\varepsilon_k^2(n)]}{\partial A_k},\quad \varepsilon_k(n)=[x(n)-y_k(n)],\quad n=1,2,\ldots N$$
$$=\left\{\frac{\partial E[\varepsilon_k^2(n)]}{\partial a_0(k)}\cdots\frac{\partial E[\varepsilon_k^2(n)]}{\partial a_L(k)}\right\}^T. \tag{10}$$

In these equations, μ is a parameter that controls the rate of convergence. The μ must be selected according to the resolution of input and output pair, such as (x,y) in Eq. (7). In this work, we use x as FD and y as search range (SR). The tolerance of the estimated SR is set to ± 1. To guarantee the convergence, we take care to use μ. Thus, a large μ could result in an adaptive process that never converges. Conversely, if μ is too small, the coefficient vector adaptation is very slow. The range of μ can be changed according the order of L in Eq. (7). N number of samples $x(n)$ are used for a polynomial curve fitting. MSE $E[\varepsilon_k^2(n)]$ decreases from step k to step $k+1$. For more convergence, each process can be iterated.

Using the LMS method, we can obtain polynomial curve fittings with different orders for the adaptively variable search range, as shown in Fig. 9. Among them, some nonmonotonic functions are not available. The polynomial curve fitting with the order of 1 or 2 will give a sufficient approximation. The coefficients for the curve are arranged at the x axis from the highest order. As an example, for curve fitting with the order of 2, we obtain the following function of FD.

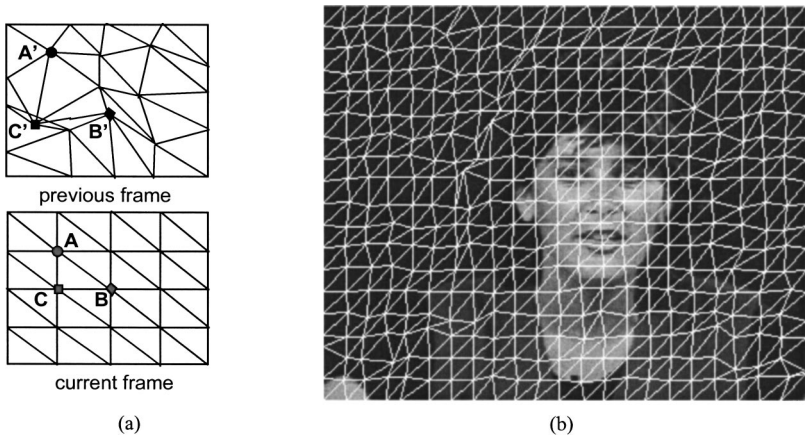

Fig. 8 Deformation of mesh structure: (a) flipped-over mechanism and (b) actual flip over.

$$SR = 0.003898FD^2 - 0.672984FD + 20.867619. \quad (11)$$

In this work, we use the order of 2. To obtain a correct curve fitting, we should use the smaller value of μ than 1e-4. We use the μ of 1e-6.

3.4 *Distribution of Block Mean Square Error and Map Design*

From distributions of the block mean square error (BMSE), as shown in Fig. 10, we know that a large number of BMSE

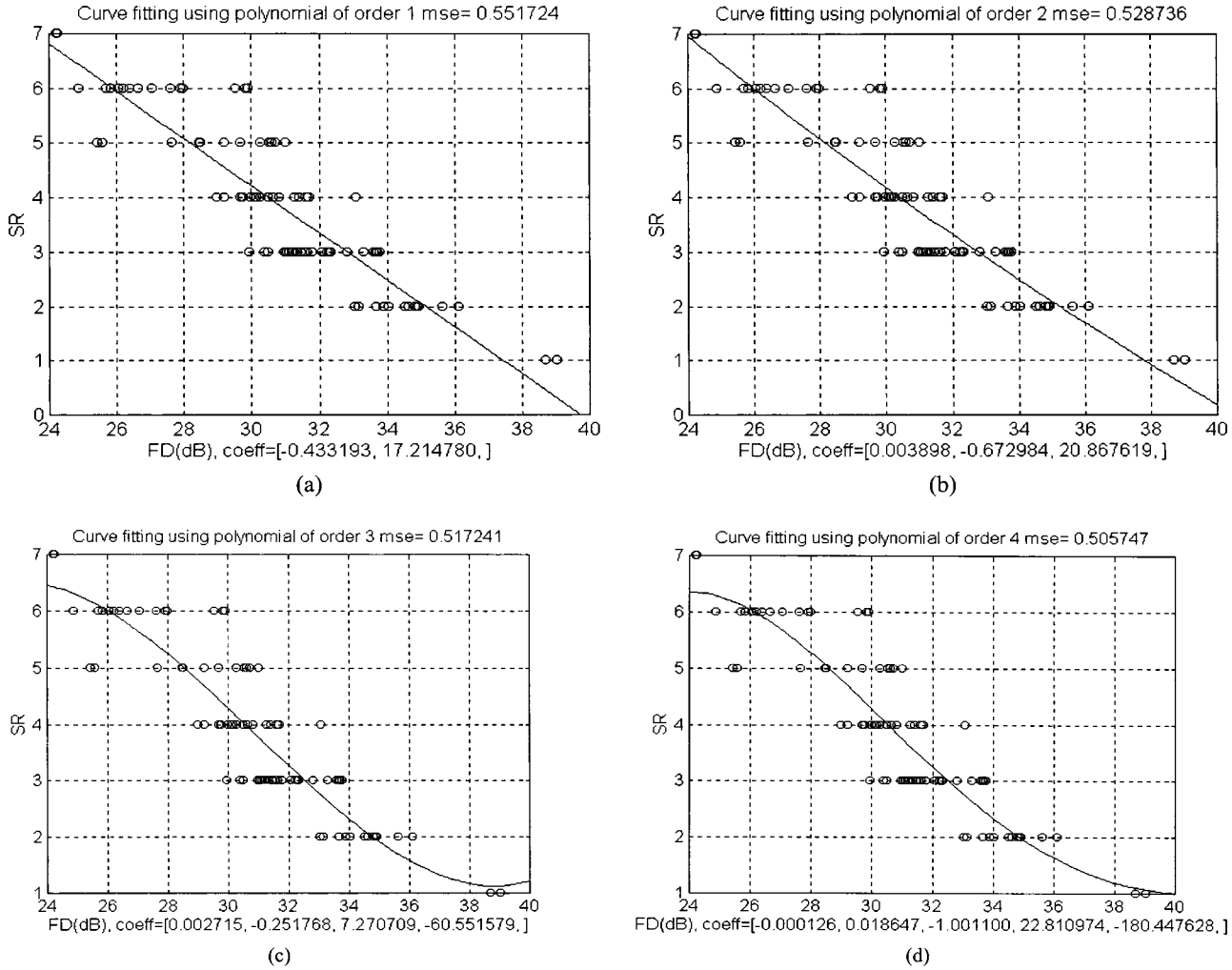

Fig. 9 Polynomial curve fittings for the adaptive search ranges: curve fitting with the order of 1(a), 2(b), 3(c), and 4(d).

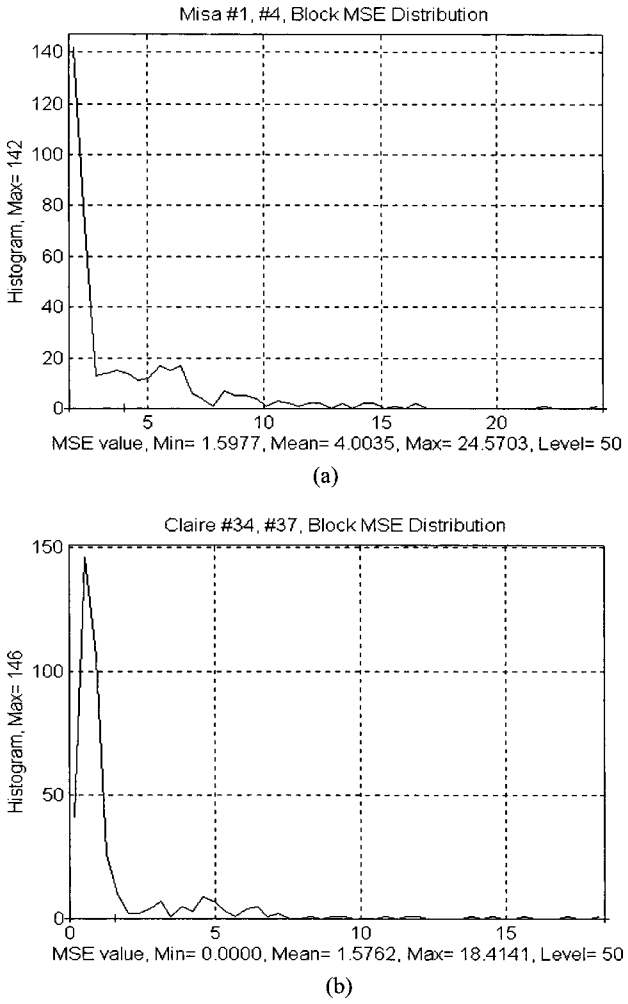

Fig. 10 Distributions of block MSE: difference values for (a) Miss America and (b) Claire.

is distributed below the mean BMSE notified on the x axis. The upper part of the mean BMSE may make a large error in a motion prediction. On the other hand, since the lower part of the mean BMSE will generate smaller errors, human eyes cannot distinguish difference. For that reason, we only perform a motion prediction for the larger error blocks within a sending rate.

In Fig. 11, we explain how to make several maps such as rate_MSE_map, send_CP_map, and refine_CP_map. Numbers of the MSE_map are arranged as ascending orders with the MSE of each block. According to a sending rate, we set the rate_MSE_map using the MSE_map, which gives a priority. In this figure, a filled point means a node point used for more block compensation, and the other empty point means a node point used for one block compensation. The mark O means a control point (CP) located in four corners of the set block in the rate_MSE_map, and the other mark X means a CP needed only for the refinement, which is known by the send_CP_map. The send_CP_map will be transmitted to a decoder.

4 Experimental Results

4.1 *Experimental Environment*

The simulations were performed using monochrome test sequences Miss America, Claire, and Salesman for each 88 frames, which originally consist of 352×288 luminance pixels. The frame rate, which is originally 30 Hz, was set at 10 Hz (the 1st, 4th, 7th,..., 88th frames were used) in the simulations. In the simulations of motion compensation, the original image of the previous frame was used as the reference frame to synthesize the predicted image. Image quality was evaluated by peak signal-to-noise ratio (PSNR) defined as

$$\mathrm{MSE}=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[I_k(m,n)-\tilde{I}_k(m,n)]^2, \tag{12}$$

$$\mathrm{PSNR}=10\log_{10}\frac{255^2}{\mathrm{MSE}}\ [\mathrm{dB}], \tag{13}$$

where we use $M=352$, $N=288$, and block size (BS) $=16$.

There are 357 node points for 16×16-segmented blocks in an image frame. Node points along the frame borders makes little influence on prediction quality, unless the scene is moving along the borders. In our experiments, we do not estimate the motion vector of node points along the frame borders. No motion vectors are transmitted for the points lying on image borders. The motion of the border node points is restricted to be along the border with only one degree of freedom. The relevant components of the closest point inside the picture are used as a 1-D motion vector.

4.2 *Performances of Coarse Motion Estimation Using Adaptively Variable Search Ranges*

When we employ the adaptively variable search range (SR), performances of the coarse motion estimation can be

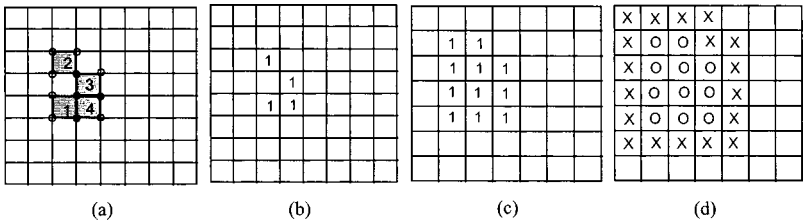

Fig. 11 An example of map design: (a) MSE map, (b) rate_MSE_map, (c) send_CP_map, and (d) Refine_CP_map.

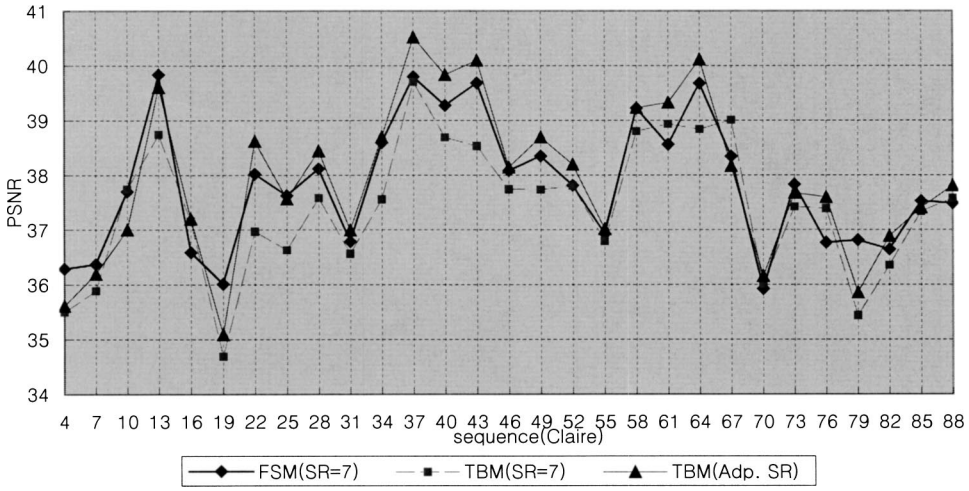

Fig. 12 Coarse motion estimation using adaptively variable search ranges (Claire).

depicted as Fig. 12. Triangular mesh-based motion estimation using the adaptively variable SR triangular mesh-based motion estimation (TBM) (adp. SR) is superior to BMA full search motion estimation (FSM) and triangular mesh-based motion estimation using a fixed SR of 7. For more analysis, we can see the original motion vector field obtained by FSM in Fig. 13. It shows a distribution of motions in one block. The motions are distributed linearly in a global view.

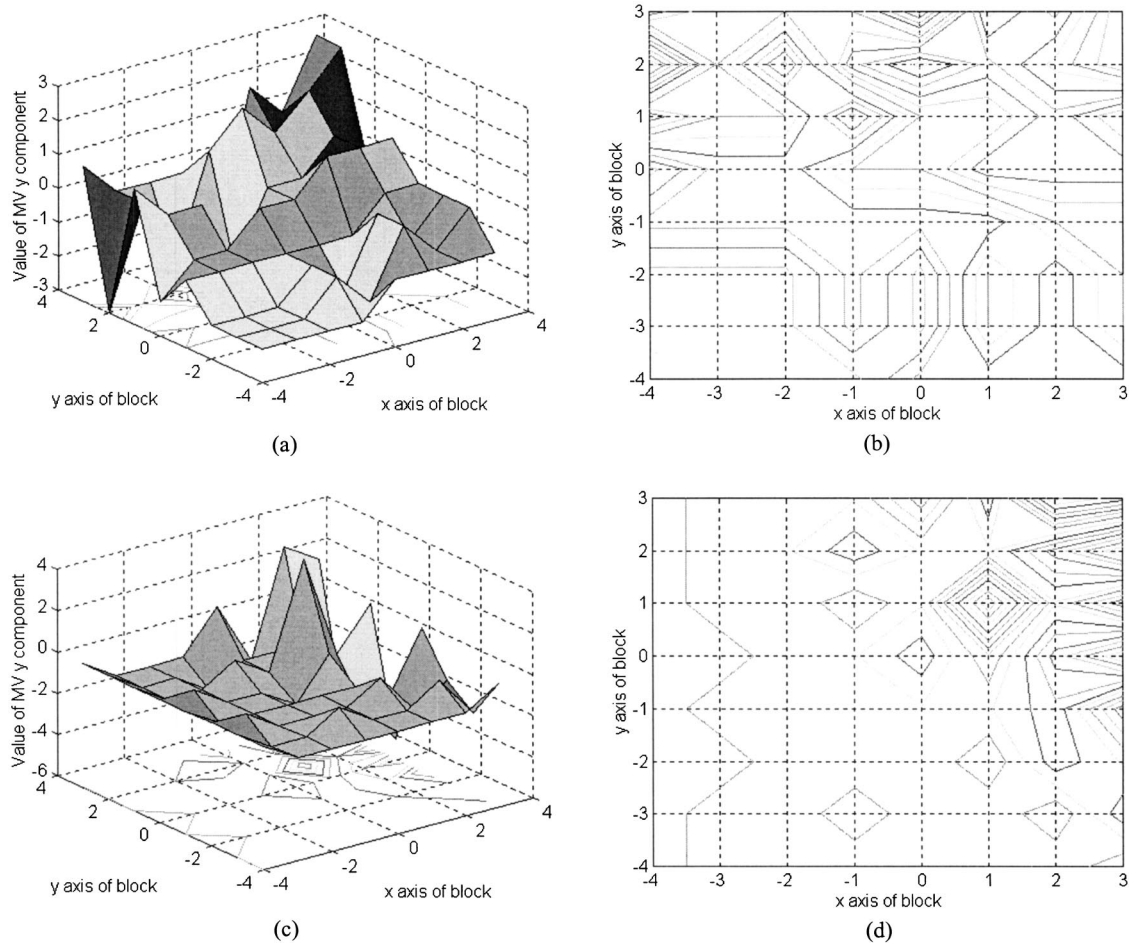

Fig. 13 Original motion vector field at block location (17,21) (Miss America 1 and 4): 3-D view for (a) *x* and (c) *y* component of motion, and contours for (b) *x* and (d) *y* component of motion.

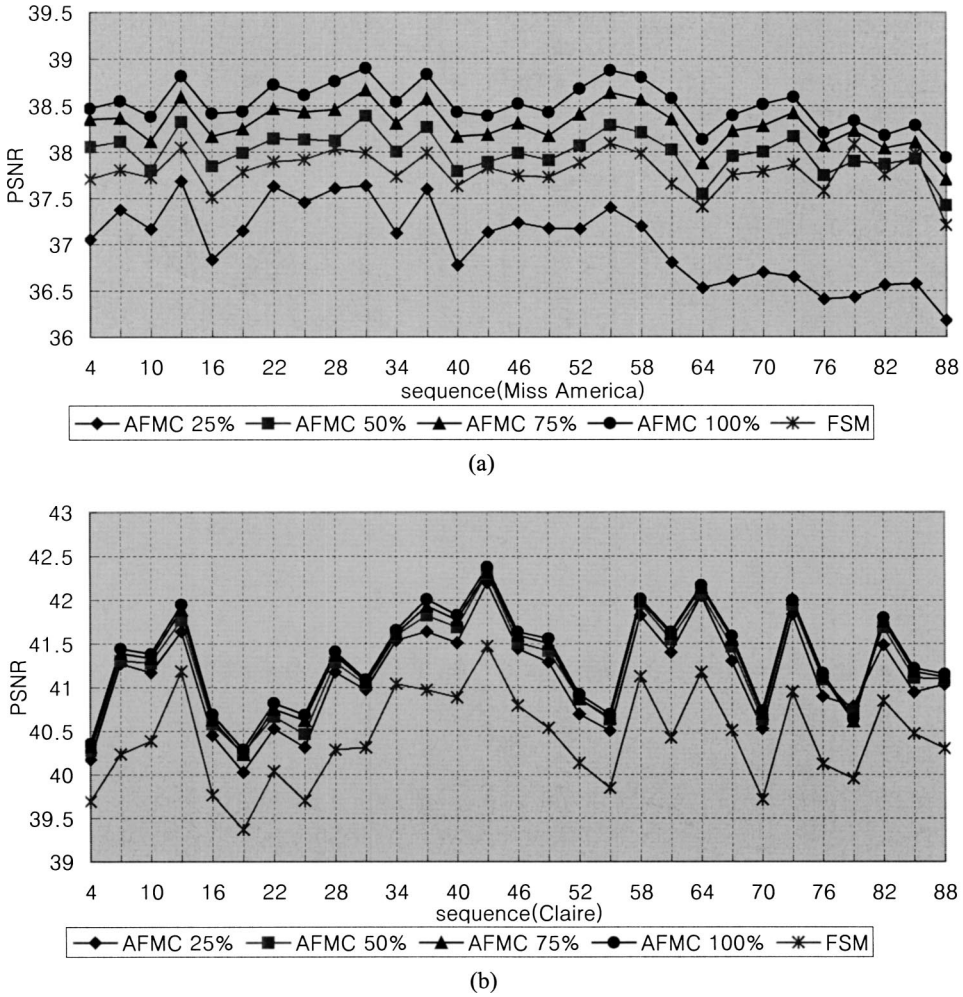

Fig. 14 Performances of the adaptive partial matching at different sending rates: (a) Miss America and (b) Claire.

Since the affine transformation is a linear interpolation of the motion in a triangular patch, it gives a better prediction than BMA.

4.3 *Performance of the Refinement Using Adaptive Partial Matching*

Now we compare performances after the refinement. An adaptive partial matching method can have different performances according to the sending rate. Figure 14 shows the performances at different sending rates. AFMC means a motion compensation using an affine transformation. It is sufficient for more than 50% of the control points for Miss America, and more than 25% of the control points for the Claire sequence. The case of sending rate 100% brings the maximum performance. However, we can get a sufficient performance for a 50% sending rate. We use a sending rate for around 50% of control points. It reduces computational complexity and number of coding bits.

4.4 *Amount of Coding Bits and Computational Complexity*

Until now, we have compared the performance of the predicted image. We need to compare the other important points such as computational complexity and number of sending bits.

Figure 15 shows a coding format to encode the data. From header_bit, we determine how large search ranges are used for motion estimation. It is encoded using two to four bits when we limit the search range up to ± 7. If a search range is three, we require only three bits, with an

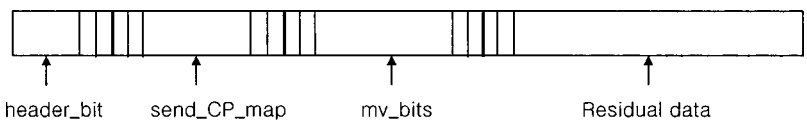

Fig. 15 Coding format for adaptive partial matching.

Table 1 Average performance of Miss America.

	FSM	AFMC 25%	AFMC 50%	AFMC 75%	AFMC 100%
PSNR(dB)	37.80	37.03	38.00	38.29	38.51
DCT/Q size(bytes)	1618	1689	1510	1498	1492
Refine_CP	396	155	269	357	357
Trans_CP	396	99	198	297	357
Overhead(bits)	0	400	400	400	400
Opt_SR	7 (fixed)	4.31	4.31	4.31	4.31
Total_Trans(bits)	16112	14704	14064	14760	15192
Total_Comp(c)		3018830	5179483	6847355	6847358
Total_Comp(r)	22809600	8941363α	17882726α	26824089α	32243097α

Table 2 Average performance of Claire.

	FSM	AFMC 25%	AFMC 50%	AFMC 75%	AFMC 100%
PSNR (dB)	40.42	41.12	41.24	41.30	41.34
DCTQ size(bytes)	1599	1488	1487	1496	1496
Refine_CP	396	167	316	357	357
Trans_CP	396	99	198	297	357
Overhead(bits)	0	400	400	400	400
Opt_SR	7 (fixed)	3.03	3.03	3.03	3.03
Total_Trans(bits)	4767	2680	3471	4272	4752
Total_Comp(c)		2232290	4133524	4656683	4656683
Total_Comp(r)	22809600	11176704α	22353408α	33530112α	40303872α

Table 3 Average performance for image warping methods (Miss America). Hexa7 is the hexagonal matching method (SR$=\pm 7$);[2] Gradient is the gradient constraint method;[2] Forward is the forward matching method;[3] and HGI is the hierarchical node interpolation method.[7]

	Hexa7	Gradient	Forward	HGI	APM
PSNR (dB)	38.39	37.07	38.37	37.80	38.51
DCT/Q size(bytes)	1491	1676	1505	1723	1492
Refine_CP	396	396	396	515	357
Trans_CP	396	396	396	515	357
Overhead(bits)	0	0	0	396	400
Opt_SR	7	-	7	7	4.31
Total_Trans(bits)	15096	16576	15208	17904	15192
Total_Comp(c)	22809600	792×792 Inv. Mat.	22910976	22995879	6847358
Total_Comp(r)	71651328α	71651328α	71752704α	97491968α	32243097α

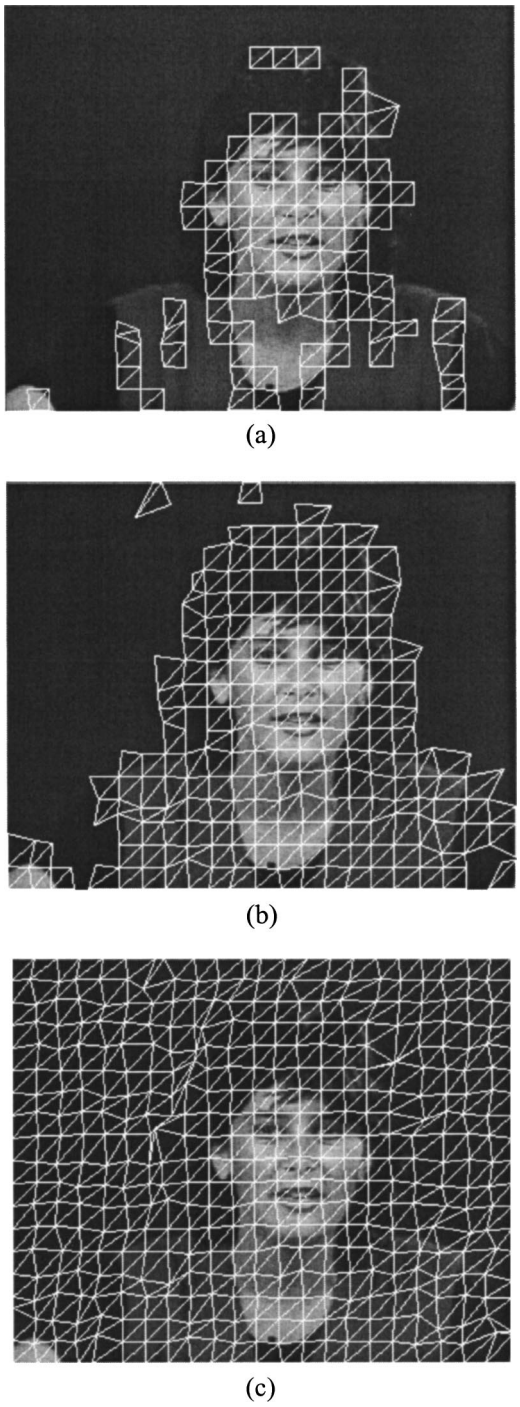

Fig. 16 Mesh topologies after adaptive partial matching (Miss America): APM with (a) 25%, (b) 50%, and (c) 100% of CP.

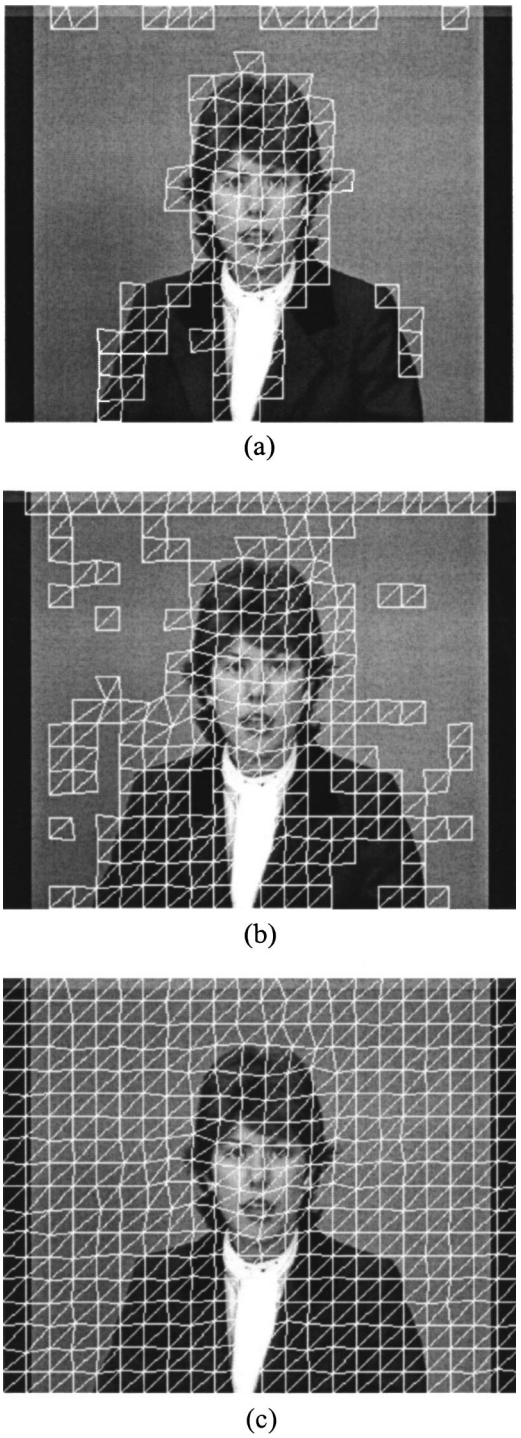

Fig. 17 Mesh topologies after adaptive partial matching (Claire): APM with (a) 25%, (b) 50%, and (c) 100% of CP.

additional one bit for the sign. Send_CP_map denotes blocks containing control points to be transmitted. mv_bits are used to encode each component of motion vectors. The residual image between the current and predicted images is encoded by DCT and quantization denoted as DCT/Q.

We arrange average results for FSM and AFMC in Tables 1 and 2. Around the sending rate 50% of CP, it shows some trade-off points. DCT/Q means the amount of bytes representing residual data coded by DCT and quantization. Refine_CP is a number of CPs required for the refinement using adaptive partial matching refinement, and Trans_CP means a number of CPs to be transmitted. Overhead includes header_bit (4 bits) and send_CP_map (396 bits). The proposed method uses optimal search ranges (Opt_SR). Finally, the results are accumulated as Total_Trans, which means the transmitted information (bits) and Total_Comp, which means computational complexity to compute the number of the multiplication. A letter in the brace *c* means the coarse motion estimation, and

r means the refinement. The factor α is a processing time that is required for the refinement using the affine transformation (AFMC).

As shown in Table 3, the proposed method (APM) is compared with conventional image warping methods. It can be an optimal solution in the sense of reconstructed image quality, number of coding bits, and computational complexity. Since each method requires a different operation, the measures of computation complexity are different: Hexa7, Forward, and hierarchical node interpolation (HGI) methods require the computation of MSE for blocks, while the gradient constraint method requires computation of inverse matrices and multiplication of matrices.

In our work, the adaptively variable partial matching method has a very preferable trade-off, and several extensions are possible for applications. When we use small channel capacity, if real-time operation is required, coarse motion estimation and compensation is a sufficient solution.

4.5 *Results of Mesh Topology*

In Figs. 16 and 17, we can see mesh topologies after the adaptive partial matching using n % of control points (CP). Adaptive partial matching can effectively track moving objects.

5 Conclusions

We show that wide search ranges do not always improve reconstructed image quality due to the large deformation of meshes. By considering the point, we determine variable search ranges adaptively to estimate a motion vector at each node point. To reduce computational complexity, we also introduce an adaptive partial matching method instead of applying the hexagonal matching method on the whole image. By simulating the proposed scheme with DCT and quantization, we show that the proposed scheme is a suitable method in low bit-rate coding. The high performance is due to the fact that the affine transformation maintains an acceptable level of prediction error while we reduce the number of control points to n % of the maximum number of control points, such as 25, 50, 75, and 100%. As a result, we develop a more efficient motion estimation method than block-matching methods and other image warping methods in the view of computational complexity, comparable image quality, and fewer number of coding bits.

References

1. J. Nieweglowski, T. G. Campbell, and P. Haavisto, ''A novel video coding scheme based on temporal prediction using digital image warping,'' *IEEE Trans. Consum. Electron.* **39**(7), 141–150 (1993).
2. Y. Nakaya and H. Harashima, ''Motion compensation based on spatial transformations,'' *IEEE Trans. Circuits Syst. Video Technol.* **4**(5), 339–356 (1994).
3. Y. Wang and O. Lee, ''Use of two-dimensional deformable mesh structures for video coding, Part I: The synthesis problem: Mesh-based function approximation and mapping,'' *IEEE Trans. Circuits Syst. Video Technol.* **6**(11), 636–646 (1996).
4. O. Lee and Y. Wang, ''Non-uniform image sampling and interpolation over deformed meshes and its hierarchical extension,'' *Proc. SPIE* **2501**, 389–400 (1995).
5. Y. Wang and O. Lee, ''Active mesh—A feature seeking and tracking image sequence representation scheme,'' *IEEE Trans. Image Process.* **3**(8), 610–624 (1994).
6. Y. Altunbasak and A. M. Tekalp, ''Closed-form solutions for polygon-based node motion estimation,'' *Proc. SPIE* **2727**, 356–364 (1996).
7. C. L. Huang and C. Y. Hsu, ''A new motion compensation method,'' *IEEE Trans. Circuits Syst. Video Technol.* **4**(1), 42–52 (1994).
8. G. J. Sullivan and R. L. Baker, ''Motion compensation for video compression using control grid interpolation,'' in *1991 International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 2713–2716, IEEE, Piscataway, New Jersey (1991).
9. B. Girod, ''The efficiency of motion-compensating prediction for hybrid coding of video sequences,'' *IEEE J. Sel. Areas Commun.* **SAC-5**(7), 1140–1154 (1987).
10. G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, New York, NY (1990).
11. R. Crane, *A Simplified Approach to Image Processing*, Prentice Hall, Englewood Cliffs, NJ (1997).

Dong-Keun Lim received the BS degree in electronics engineering from Chonbuk National University, Jeonju, Korea, in 1994, and MS and PhD degrees in information and communications engineering from Kwangju Institute of Science and Technology (K-JIST), Kwangju, Korea, in 1997 and 2003, respectively. He joined Hyundai Electronics (now Hynics Semiconductor), Ichon, Korea, from 1993 to 1995, where he designed applicaiton specific integrated circuit (ASIC) very large scale integration (VLSI) chips for image and video processing. Since 2003, he has been with Samsung Electronics, Suwon, Korea, where he is involved in development of the advanced digital high-definition television (AD-HDTV) system. His research interests include digital image and video coding, digital video and audio broadcasting, content-based video representation and processing, and their VLSI design and system implementations with software and hardware interfaces.

Yo-Sung Ho received the BS and MS degrees in electronic engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively, and the PhD degree in electrical and computer engineering from the University of California, Santa Barbara, in 1990. He joined the Electronics and Telecommunications Research Institute (ETRI), Daejon, Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, New York, where he was involved in development of the advanced digital high-definition television (AD-HDTV) system. In 1993, he rejoined the technical staff of ETRI and was involved in development of the Korean direct broadcast satellite (DBS) digital television and high-definition television systems. Since 1995, he has been with Kwangju Institute of Science and Technology (K-JIST), where he is currently a professor in the information and communications department. His research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video and audio broadcasting, and content-based signal representation and processing.