An effective mesh-based motion compensation technique for video coding

Chung Ming Kuo* and Mao Shung Hung
Department of Information Engineering, I-Shou University, Tahsu, 840, Kaohsiung, Taiwan
Received 31 January 2002; accepted 28 April 2003

Abstract

The block-matching algorithm is the most popular motion compensation technique in video coding. However, it cannot provide acceptable quality at very low bit rate. In this paper, a new mesh-based motion compensation method is proposed to attack the problem. First, a regular non-uniform mesh, which has regular structure with variable patch size, is presented. The patch size is varied according to motion activity of a video sequence. Next, a weighted interpolation block matching is developed to improve the estimate accuracy of displacements of grid points. It utilizes the motion correlation between a grid point and its associated patches. Finally, based on the new mesh and motion estimation scheme, an efficient motion compensation algorithm is developed. When compared to the conventional motion compensation techniques, the proposed method improves performance significantly with lower computational complexity and overhead information bits.

Keywords: BMA; Regular non-uniform mesh; Motion activity; Weight interpolation block matching

1. Introduction

Motion compensation plays an important role in video coding since it can reduce the temporal redundancy significantly (Li et al., 1994). The block-matching algorithm (BMA) is the most popular motion estimation/compensation technique and was adopted in the current video coding standards such as H.261/3 and MPEG-1/2

* Corresponding author. Fax: 886-7-6578944.
E-mail address: kuocm@isu.edu.tw (C.M. Kuo).
Since BMA employs a translational motion model, it suffers from blocking effects in the decoded images.

Recently, a mesh-based approach (MBA) was developed aiming at solving the above problem. The main idea behind the MBA is to model motion by a spatial transformation function that maps the coordinates of one image to form another image in a new coordinates system (Wolberg, 1990). In MBA, an image is partitioned into polygonal patches with a two-dimensional mesh. In general, the patches are triangles or quadrangles, leading to triangular or quadrilateral meshes, respectively. The vertices of each polygonal patch are known as grid (or control) points. The patches are non-overlapping in both reference and current frames. The spatial transformation function of the triangular patch can be described as an affine model, and that of the quadrilateral patch can be described as a bilinear or perspective model. An advantage of the mesh model over the conventional block model is its ability to represent more general types of motion such as rotation, zooming, and deformation. Thus it can avoid the blocky artifacts, especially in the presence of non-translational motions.

MBA has two major techniques for the implementation of spatial transformation. One is the backward type matching and the other is the forward type matching. The former generates a uniform mesh by partitioning the current frame into several equal-size rectangular blocks and finds the corresponding grid points in the previous reference frame. It is the most widely used techniques due to its simplicity. The latter generates a mesh in the previous frame, and then continues to track the grid points in the current frame. It can provide adaptive ways of motion compensation (Dudon et al., 1997; Tekalp et al., 1998; Wang and Lee, 1996a,b; Yaoping and Chengke, 1998), which are difficult with backward matching. The forward tracking can be used to continuously track control points of an object in successive frames, which is desirable in object-oriented coding. One obvious advantage of backward matching is that it can be possibly integrated in the current video coding standards. In this paper, we will focus on backward matching.

The major issues in mesh-based motion compensation are (a) mesh type selection and (b) motion estimation of a grid point. There are two types of meshes: regular and irregular. The regular mesh is easily to be generated and no extra information bits for representing mesh structure is needed. Therefore, it is widely used in backward matching. However, its fixed patch size cannot accommodate the non-stationary image characteristics and cope with object boundary exactly. Therefore, the performance based on this mesh is not good (may be lower than that of BMA (Wang and Ostermann, 1998)) due to the warping artifacts. The irregular mesh is generated according to the image content. The popular irregular meshes are Delaunay mesh (Dudon et al., 1997; Tekalp et al., 1998; Yaoping and Chengke, 1998), active mesh (Terzopoulos and Vasilescu, 1991; Vasilescu and Terzopoulos, 1992; Wang and Lee, 1994, 1996a,b), and quadtree mesh (Huang and Hsu, 1994). They achieve better performance and are suitable for forward matching. However, they are very complicated and computationally expensive. In this work, a regular mesh with variable patch size will be presented. The patch size is varied with the motion activity of image sequences; hence the mesh is dense in moving region and loose in the background region. It is simple but effective for motion compensation purpose.
The motion estimation of grid points is a more critical problem with the use of spatial transformation for MBA. Actually, a grid point is a part of its surrounding patches, thus the motion of the grid point is correlated with that of its attached patches. In this work, we will present a new scheme that exploits this correlation appropriately. The motion vector of a grid point is obtained by weighted averaging the motion vectors of the surrounding patches of the grid point. This paper presents a new MBA technique based on the proposed mesh and motion estimation scheme. Simulation results show that both subjective and objective qualities are improved significantly. The computational load is only slightly higher than that of the conventional method. Furthermore, it is possible to integrate the new method into the current video coding standards such as H.263.

This paper is organized as follows. Section 2 gives a review of the existing mesh-based motion compensation techniques. In Section 3, the proposed motion compensation method is introduced. The simulation results and discussions are given in Section 4. Finally, Section 5 summarizes the main results of this paper.

2. Review of conventional mesh-based motion compensation

2.1. Mesh type selection

Many mesh structures have been presented in the literatures. However, to our best knowledge, no clear classification has been made. In this work, we classify the mesh structures according to the grid-point assignment, patch size, and connectivity of grid points. If all horizontal (or vertical) mesh lines have the same number of grid points, and all the grid points in each line are collinear, then the mesh is regular; otherwise it is irregular. The regular mesh can be further classified into uniform patch and non-uniform patch according to patch size. If the path size of the entire mesh is constant, the mesh is uniform; otherwise, it is non-uniform. A grid point in a mesh is connected with other grid points. If the numbers of grid points connected to each grid point (except grid points at the boarder) in the mesh are the same, the mesh is conformal. Obviously, the regular mesh assures to be conformal. However, the irregular mesh can be conformal or non-conformal. Fig. 1 illustrates the various mesh types.

The regular uniform (RU) mesh is the simplest one. It is not desirable to send mesh geometry to the decoder. The RU mesh is suitable for both backward and forward matching. However, it cannot cope with object boundary exactly, and the fixed patch size is either too dense for background region or too loose for moving object. Therefore, the prediction gain based on this mesh may be lower than that of BMA (Wang and Ostermann, 1998) due to the warping artifacts.

There are many irregular meshes presented in the literature. The quadtree mesh and Delaunay mesh are non-conformal connected. The non-conformal connected property causes not only the complexity in maintaining mesh structure but also the discontinuity in image warping. The active mesh (Wang and Lee, 1994) is conformal connected, which iteratively adjusts a mesh from initially regular mesh by
minimizing mesh deformation, negative feature magnitude, the interpolation error, and the matching error. The generation procedure of the active mesh is very complicated. The active mesh is only suitable for forward matching.

In this paper, a regular mesh with variable patch size will be proposed. It is referred to as regular non-uniform (RN) mesh (Kuo et al., 1999). The patch size is varied with the motion activity of image sequences; hence the mesh is dense in moving region and loose in the background region. It is simple but effective for motion compensation purpose.

2.2. Motion estimation of grid points

Motion estimation of a grid point is another key factor for mesh-based motion compensation (Huang and Hsu, 1994; Nakaya and Harashima, 1991, 1994; Seferdis and Ghanari, 1993; Sullivan and Baker, 1991; Yaoping and Chengke, 1998). The estimate results will significantly affect the motion compensation performance; i.e., prediction gain and visual quality. In general, the motion estimation method can be divided into two categories (Wang and Ostermann, 1998). One of the approaches determines the displacement of each grid point by the block-matching algorithm (BMA) (Seferdis and Ghanari, 1993), which minimizes the block-matching error in a search area surrounding the grid. More specifically, BMA is applied to the block that centered at a grid point and the estimated block displacement is considered as the motion of the grid point. The block type matching considers the pixel similarity instead of the real prediction error (warping error). In addition, it estimates the displacement of each grid point independently. Therefore, this method is not sufficient to represent a concrete physical object that contains many interconnected grid points.

Fig. 1. Illustration of the various mesh types (a) regular-uniform, (b) regular-non-uniform, (c) irregular conformal connected, and (d) irregular non-conformal connected.
The other popular approach employs a recursive-updating manner to estimate the displacement of a grid point (Seferdis and Ghanari, 1993). More specifically, the displacement of a grid point is updated iteratively by minimizing the real prediction error of the patches attached to the grid. The prediction error is calculated by warping the corresponding patches in the reference frame to the current frame based on the estimate of motion of a grid point. Usually, the exhaustive search or gradient-based search is applied to locate the optimal displacement.

The second method achieves better performance than the first one, but it is very complicated and time consuming. The combination of the two approaches called the two-step method is more practical and has received great attention recently (Nakaya and Harashima, 1994; Nieweglowski and Haavisto, 1995; Nieweglowski et al., 1993). The first step, referred to as coarse estimation, employs BMA to obtain an initial guess of the displacement of a grid point. The second step, fine estimation, refines the initial guess by locally changing grid point position and by minimizing the resulting warping error. To avoid a significant increase of computation, the refinement is often performed at a very small search area. The two-step method may fail to converge since the coarse step that employs block matching often achieves a poor estimate, as mentioned before.

In order to improve the estimate accuracy of the coarse step, Nieweglowski et al. (1993) and Nieweglowski and Haavisto (1995) presented a modified BMA, in which a weighting mask is used to avoid the interference of the estimation. This algorithm still treats a grid point as a single pixel. Thus, the performance achieved is not good enough. In the following, we will present a new estimation scheme that considers a grid point as a part of the surrounding patches. It improves the estimate accuracy of the coarse step, thus yielding a better performance in motion compensation.

3. Proposed mesh-based motion compensation

3.1. RN mesh generation

As mentioned before, the conventional irregular meshes are too complicated; thus it is not attractive in real applications. On the contrary, RU mesh is not sufficient to characterize the motion activity. Generally, the motion is not stationary over the entire frame of an image sequence. Therefore, variable path size, instead of fixed one, should be selected such that better compensation can be achieved. In this work, we design a regular mesh in which the patch size is varied with motion activity. Generally, the variance of the difference of two consecutive frames is a good criterion to measure motion activity. We employ the local variance of the difference frame to adjust the patch size. The patch size is varied from 8 to 32 in either horizontal or vertical direction.

We first calculate the variance of the frame difference by

\[ v_{FD} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (f_{FD}(i,j) - \bar{f}_{FD})^2, \]

(1)
where $f_{FD}$ denotes the intensity difference between the current and previous frame. The frame size is $M \times N$, and $f_{FD}$ is the mean of the difference frame. The mesh can be generated starting from horizontal direction or vertical direction. The following statement is for the horizontal direction. An input frame is partitioned into horizontal strips, each with height of 32 pixels, and then the local variance of a horizontal strip is calculated by

$$v_{SP}^{32}(k) = \frac{1}{M \times 32} \sum_{i=1}^{M} \sum_{j=1}^{32} \left( f_{FD}(i, j + 32 \times (k - 1)) - \bar{f}_{SP} \right)^2,$$

where $k = 1$ to $N/32$ and $\bar{f}_{SP}$ is the horizontal strip mean. Finally, we compare $v_{SP}^{32}(k)$ with the threshold $v_{FD}$. If $v_{SP}^{32}(k) > v_{FD}$, the strip contains moving objects, thus it is further partitioned into two strips with equal size. The partition continues until the strip variance is below the threshold, or it reaches the minimum size of 8. After the partition in horizontal direction is finished, the partition in the vertical direction is then performed in a similar manner to complete the mesh generation.

The mesh structure must be transmitted as an overhead for both backward and forward matching. It can be efficiently encoded with a variable length code. Let “1” denote further partition, and “0” end partition, we can readily obtain the mesh structure code, which is a small overhead. For example, the coarsest partition (patch dimension is 32 for both horizontal and vertical directions) for a CIF format of 352 x 288 yields 11 vertical strips and 9 horizontal strips. Thus, $11 + 9 = 20$ bits are required for the representation of the coarsest partition, which is the minimum overhead. Each horizontal (or vertical) strip can be further partitioned at most two times in a binary split manner. Therefore, at most three bits are needed to represent each horizontal (or vertical) partition. As a result, the maximum overhead is $3 \times (11 + 9) = 60$ bits. There are five types of possible partitions (structures) for each strip, and the corresponding structure codes are given in Fig. 2b. If the five types have the same probability of occurrences, then the average code length for each structure is $\frac{1}{5}(1 + 4 \times 3) = 2.6$ bits. Thus, the average overhead for representing a mesh is $(11 + 9) 2.6 = 52$ bits.

The bit rate for motion information is dependent on the number of grid points. The size of a strip with the coarsest resolution is 32 pixels. If further partition is needed for the strip, there are four different types of partitions, as shown in Fig. 2c. Assume the five strip types have the same probability of occurrences, then the average number of partition is $\frac{1}{5}(1 + 2 + 3 + 3 + 4) = 2.6$. So the average number of grid points is $(11 \times 2.6)(9 \times 2.6) = 669.24$. Each grid point required 8 bits (4 bits for $x$- and 4 bits for $y$-components) to represent motion vector. Total number of bits for grid motion information is about 5354 bits. The mesh structure code overhead compared with the bit rates of grid motion information is only $52/5354 = 0.97\%$. Therefore, the overhead of mesh structure is very small. The generated meshes for Claire sequence CIF and QCIF format are shown in Figs. 2a and b, respectively. Since the QCIF format is not divisible by 32, we extend the frame boundary using the similar scheme in H.263 unrestricted mode. It is obvious that the mesh density is higher in moving area than that in still area. The corresponding mesh structure codes are listed Fig. 2c.
The idea behind the RN mesh is similar, in some sense, to the advanced prediction (AP) mode in H.263. In AP mode, a macroblock can be subdivided into four sub-blocks, each with one motion vector. It can be viewed as a motion compensation technique with variable block size. Since the small block is used in the higher motion activity region or the discontinuous boundary, the motion compensation prediction
gain is increased significantly. However, the overall complexity and overhead information bits are also increased. In the new mesh structure, the patch size is decreased with the increase of motion activity. Therefore, the mesh is dense in the moving area and boundary region and loose in the background area. This will result in good accuracy of motion compensation. Fig. 3 depicts the RU and RN meshes, which have approximate the same number of patches. It indicates that the proposed RN mesh can characterize the region of higher activity and moving boundary better than conventional RU mesh.

In summary, the new mesh has advantageous properties as follows: (a) The structure of the mesh is conformable connected; thus it can reduce the discontinuity of image warping. The estimation procedure of displacement of a grid point is easier and more computationally efficient. (b) The density of the mesh is increased gradually instead of suddenly, thus resulting in more visual pleasing quality in compensation. (c) The generation procedure of the mesh is very easy and efficient. (d) The overhead of mesh structure code is very small.

3.2. Motion estimation of grid point

3.2.1. Weighted interpolation block matching

In general, each grid point (except that at border) is attached with four patches, thus it is a part of these patches. Therefore, the motion of a grid point is highly correlated with that of its surrounding patches. In this work, the patch is actually a block with variable size. Therefore, we first apply the conventional BMA to estimate the displacement of each block (patch), then obtain the motion of a grid point by interpolating the displacements of the surrounding four blocks (patches) as

\[
\begin{align*}
\hat{d}_{x_{gr}}^e &= w_{ul}^e d_{ul}^e + w_{ur}^e d_{ur}^e + w_{ll}^e d_{ll}^e + w_{lr}^e d_{lr}^e, \\
\hat{d}_{y_{gr}}^e &= w_{ul}^e d_{ul}^e + w_{ur}^e d_{ur}^e + w_{ll}^e d_{ll}^e + w_{lr}^e d_{lr}^e,
\end{align*}
\]

Fig. 3. Comparison of the regular uniform mesh and regular non-uniform mesh (a) regular uniform mesh and (b) regular non-uniform mesh.
where \([d_{nx}, d_{ny}]\) is the displacement vector of the grid point \(n\). \([d_{ux}, d_{uy}], [d_{ur}, d_{ur}], [d_{ll}, d_{lr}], [d_{ll}, d_{lr}]\) are the displacement vectors of the upper left, upper right, lower left, and lower right blocks, respectively. Each displacement vector contains two components, representing horizontal displacement and vertical displacement, respectively. The \(w\) is the corresponding weighting factor.

Because the sizes of the surrounding four patches are not the same for the non-uniform mesh structure generated in this work, the weight values associated are different in general. The determination of the weights is based on the following criteria:

1. The horizontal displacement and vertical displacement are independent.
2. The region with higher motion activity should be assigned larger weighting factors. This means that a small surrounding block should contribute more than a large one.
3. The weighting factors for horizontal and vertical directions should be determined independently because the width and height of a patch are obtained independently.

According to the above criteria, we define the weighting factor for the upper left block as follows. The definitions for other attached blocks are similar and thus neglected here.

\[
\begin{align*}
    w_{xul} &= \frac{1}{4} \sum_{k \in J_n} \frac{x_k}{x_p}, \\
    w_{yul} &= \frac{1}{4} \sum_{k \in J_n} \frac{y_k}{y_p},
\end{align*}
\]

where \(J_n = \{ul, ur, ll, lr\}\), which denotes the four blocks attached to the grid point \(n\). The \(x_p\) and \(y_p\) are the width and height of the block \(p\). If the four blocks are with equal size, all the weighting factors are the same and equal to 1/4. In such case, the displacement of the grid point is simply the average of the displacements of four patches.

In the proposed mesh, the neighboring blocks in the same horizontal (vertical) strip have the same height (width). Therefore, \(w_{ul} = w_{ur}, w_{ll} = w_{lr}\), which simplifies the calculation of weighting factors.

Compared to the standard BMA (SBMA) and modified BMA (MBMA) (Nieweglowski and Haavisto, 1995; Nieweglowski et al., 1993), the proposed weighted interpolation block matching (WIBM) can improve the estimation accuracy significantly, especially on the moving object boundary. We use a simple example to demonstrate the improvement. Figs. 4a and b are two successive artificial images with five numbered objects. There are different types of motions contained in the five objects such as zooming for object 1, clockwise rotation for object 2, pure translation for object 3, counterclockwise rotation for object 4, and deformation for object 5. We applied backward matching using MBMA and WIBM, and the resulting mesh maps are shown in Figs. 4d and e, respectively. For fair comparison, the regular uniform (RU) mesh is used for both schemes although our system employs regular non-uniform mesh.

On the neighborhood of object boundary, some patches cover a small part of object, say patches a and b in Fig. 4c. Since object 3 has pure translation, the corresponding grid points of patches a and b should move in same direction.
Fig. 4. The comparison of MBMA and WIBM for artificial images (a) previous frame, (b) current frame, (c) mesh map on the current frame, (d) mesh map on the previous frame by MBMA, (e) mesh map on the previous frame by WIBM, (f) predicted image of (d) with MBMA, (g) predicted image of (e) with WIBM, (h) difference between (f) and (b), and (i) difference between (g) and (b).
The SBMA or MBMA estimates the displacements of each grid point independently by forming block centered at the grid point. The forming block may not contain any object information. Thus it often results in incorrect estimate of the displacement, and causes warping artifact, as illustrated in Fig. 4d. However, in the proposed WIBM, its surrounding patches determine the displacement of a grid point, so the object information will be included in. The warping artifact is thus significantly reduced. Figs. 5a and b are the enlarged versions of the mesh maps. It is obvious that that the new method has achieved reasonable estimation results in the shade regions shown in Fig. 5b. On the contrary, the mesh map obtained by MBMA is over-deformed in the same region.

The other advantage is that the WIBM can cope with covered/uncovered background better than SBMA and MBMA. Figs. 6a and b indicate that the Claire turns head to the left. Thus, near to left side of her face the uncovered background appears. When the MBMA is applied to estimate the displacement of grid points in the left-side face of the current frame, the grid points will match to background in
the previous frame and incorrect match is yielded since the MBMA utilizes pixel similarity. However, the proposed WIBM will achieve correct estimate since it utilizes patch contents. Fig. 6 also compares the motion tracking ability of MBMA and WIBM. For clear demonstration, here we intentionally use two adjacent frames separated with significant time difference; i.e., frame 90 and frame 96. The predicted frame is thus distorted seriously. However, the proposed method achieves better-predicted quality obviously, as seen in Figs. 6f and g.

To further demonstrate the quality of the proposed motion estimation scheme, we used the SBMA, MBMA, and the WIBM to estimate the motion vectors of three standard test sequences. Again, the regular uniform mesh is employed for all schemes. The bilinear warping is used to obtain the motion compensated prediction.

Fig. 5. The enlarged versions of Figs. 4d and e.
Fig. 6. Comparison of the tracking ability (a) frame 96, (b) frame 90, (c) mesh map of frame 96, (d) mesh map of frame 90 using MBMA method, (e) mesh map of frame 90 using WIBM, (f) the predicted frame using mesh in (d), and (g) the predicted frame 96 using mesh in (e).
frame. The prediction gains in terms of peak-signal-to-noise ratio (PSNR) are listed in Table 1. It indicates that the proposed method obviously achieves higher prediction gain.

The WIBM obtains a rough estimate of grid points displacements. After WIBM, a small range \((-1 \sim +1)\) of MBMA is then applied to find the initial displacements of all grid points.

3.2.2. Refinement of motion vector

The above scheme estimates the motion of a grid point with block-matching error, instead of real prediction error. Thus it is still suboptimal and needed to be further refined based on real prediction error. In this work, the coarse estimate with the above scheme is used as an initial guess, and then a refinement procedure similar to HMA in (Nakaya and Harashima, 1994), or motion vector refinement in (Nieweglowski and Haavisto, 1995) is performed around a small searching window \((3 \times 3)\) centered at the initial values. The \(3 \times 3\) window results in eight test motion vectors. For each test vector the four surrounding patches are warped and the prediction errors are calculated. The smallest prediction error yields the final motion vector. The process repeats for all grid points of an image, and then begins the second iteration until convergence is achieved.

The refinement obviously increases the computational complexity of the motion estimation. The purpose of motion compensation is mainly to compensate moving areas. Thus, we can apply the refinement only for the grid points in the portions of high-density mesh, for example, the portion with patches sizes \(8 \times 8, 8 \times 16,\) or \(16 \times 8\). In general, this can save the computational complexity about 50\% or higher, depending on the moving activity of the sequences.

The proposed motion estimation of grid points contains three parts: (a) The WIBM obtains a coarse estimate of grid point’s displacements, (b) a small range \((-1 \sim +1)\) search using MBMA is then applied to find the initial displacements of all grid points, and (c) refinement based on the criterion of real prediction error is performed. The detail computational complexity analysis will be discussed in the following section.

4. Simulation results

The simulations are performed on the test sequences “Claire,” “Salesman,” and “Trevor,” which are CIF format with \(352 \times 288\) luminance pixels, \(2 \times 176 \times 144\)

<table>
<thead>
<tr>
<th>Sequence</th>
<th>SBMA</th>
<th>MBMA</th>
<th>WIBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claire</td>
<td>37.96</td>
<td>38.03</td>
<td>38.06</td>
</tr>
<tr>
<td>Salesman</td>
<td>36.39</td>
<td>36.38</td>
<td>36.39</td>
</tr>
<tr>
<td>Trevor</td>
<td>32.71</td>
<td>32.63</td>
<td>32.89</td>
</tr>
</tbody>
</table>

Table 1
Average PSNR (dB) of predicted image using various search schemes
chrominance pixels, and the frame rate of 30 Hz. In our simulations, we only consider the luminance sequences at frame rate of 10 Hz. For comparison, the mesh-based MBMA, the conventional SBMA with full search, and AP mode are also performed using the same test sequences. In SBMA, the block size is $16 \times 16$, and search ranges are $\pm 7$ and $\pm 15$, respectively. To further compare the performances of the various techniques, in SBMA the search with half-pixel accuracy is also performed for the search range of $\pm 7$. For AP mode, the subdivision is determined by the result of full search SBMA. If the mean absolute difference (MAD) of macroblock $B_i(m,n)$ in the entire search area is greater than a threshold $T$, then the $B_i(m,n)$ is divided into four subblocks. The best estimate of the motion vector $B_i(m,n)$ is used as the initial guess of the motion vector of the subblock, and a search area $-p/2$ to $p/2$ is refined for each subblock. In our simulation, the threshold $T$ is 4 for subdivision. For MBMA, a uniform mesh with fixed size of $16 \times 16$ is used, which yields 437 (i.e., $23 \times 19$) grid points in an image frame. The maximum displacement of a grid point in either $x$ or $y$ direction is also limited in the range of $\pm 7$, and a $21 \times 21$ weighting mask is chosen (Nieweglowski and Haavisto, 1995).

In the proposed mesh, the number of grid points is varied according to the characteristics of an image sequence. In order to fairly compare with MBMA under the same situation, we limit number of grid points to be below 85% of the RU mesh with patch size $16 \times 16$. For a CIF format, there are 437 (i.e., $19 \times 23$) grid points. Thus, if the number of grid points is over 370, we increase the threshold value by 5 and then merge the partitions until the number is below 370. We compare the experimental results of different methods in terms of average PSNR, visual quality, computational complexity, and the entropy of overhead for representing motion vectors and mesh geometry.

4.1. Performance evaluation

The performance of the proposed method is evaluated in terms of average peak-signal-to-noise ratio (PSNR) and visual quality. The PSNR (dB) is defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}},$$

where MSE is the mean square error between the original image frame and predicted frame. The average PSNRs of the predicted sequences for the test sequences using the above four methods are given in Table 2. The WIBM with RU mesh and RN mesh are also carried out, respectively. We try to investigate the influence of motion estimation and mesh structure on performance. It can be seen that the PSNR of WIBM with RN mesh is higher than SBMA from 1.5 to 2.35 dB, higher than SBMA with half pixel accuracy from 0.43 to 1 dB. The proposed method is also higher than AP mode from 0.49 to 1.19 dB, and higher than MBMA from 0.68 to 1.23 dB. Obviously, the proposed method can improve the prediction gain significantly. In the same situation, the WIBM with RU mesh is also better than MBMA method. It indicates that the motion estimation with weighted interpolation outperforms the modified block matching.
Figs. 7 and 8 show the PSNR vs. frame number for the test sequences Claire and Salesman. For Salesman, the PSNR improvement of our method is significant, yet for Claire it is relatively small. This is due to the fact that the motion characteristic of Salesman is more complicated than Claire. Our method gives more improvement for complex motions since it employs variable patch size and exploits motion correlation appropriately. Fig. 9 shows the PSNR comparison of new method, SBMA and AP mode. Again the new method achieves better performance.

Fig. 10 shows the predicted images of Claire sequence with SBMA, MBMA, AP mode, and the proposed WIBM-RN. It is found that proposed method achieves the best visual quality.

4.2. The computational complexity

In this section, the computational complexity for SBMA, SBMA with half pixel accuracy, AP mode, MBMA, and the proposed method will be compared. The major computations in these techniques include block matching (Bm), weighting (wei), bi-linear transformation (Bt), and bilinear interpolation (Bi). The required operations per pixel for these computations are as follows:

- Bm: one addition and one subtraction, denoted as 1(+) and 1(−).
- wei: one multiplication, denoted as 1(×).
- Bt: 4(+) and 4(×).
- Bi: 4(+) and 4(×).

The computations incurred in the four methods are summarized in Table 3. For example, for each grid point, WIBM-RN requires computations including block matching, weighting block matching, and refinement. The computational complexity listed in the brackets will be described in the following. The motion estimation parameters are: block size $N = 16$, search area $P = \pm 7$ and weighting mask $M = 21$.

Then the computation complexity can be estimated as

1) $16^2 \times (2 \times 7 + 1)^2 (\text{Bm}) = 57600 \times 2 (+/−) = 115.2k (+/−)$.
2) $16^2 \times ((2 \times 7 + 1)^2 + 8) (\text{Bm}) + 8 \times 16^2 (\text{Bi}) = 59648 \times 2 (+/−) + 256 \times (4 (+) + 4 (×)) = 127.4k (+) + 8.1k (×)$.
3) $4 \times 8^2 \times (2 \times 3 + 1)^2 (\text{Bm}) = 12544 \times 2 (+/−) = 25.1k (+/−)$.
4) $(2 \times 7 + 1)^2 \times 21^2 (\text{Bm} + \text{wei}) = 99225 \times (2 (+/−) + 1(×)) = 198.5k (+/−) + 99.2k (×)$.

Table 2
Average PSNR (dB) with backward matching

<table>
<thead>
<tr>
<th>Sequence</th>
<th>SBMA (7)</th>
<th>SBMA (7) (half pixel)</th>
<th>SBMA (15)</th>
<th>SBMA (AP)</th>
<th>MBMA</th>
<th>WIBM-RU</th>
<th>WIBM-RN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claire</td>
<td>37.47</td>
<td>38.84</td>
<td>37.49</td>
<td>38.76</td>
<td>39.29</td>
<td>39.34</td>
<td>39.95</td>
</tr>
<tr>
<td>Salesman</td>
<td>36.91</td>
<td>37.72</td>
<td>37.02</td>
<td>38.19</td>
<td>37.77</td>
<td>37.88</td>
<td>38.68</td>
</tr>
<tr>
<td>Trevor</td>
<td>33.36</td>
<td>34.46</td>
<td>33.48</td>
<td>34.10</td>
<td>34.05</td>
<td>34.57</td>
<td>35.03</td>
</tr>
</tbody>
</table>
Fig. 7. The PSNR comparison for Claire sequence at 10 Hz.

\[(7 + 1)^2 \times 21^2 (Bm + wei) = 21609 \times (2 (+/-) + 1 (\times)) = 43.2k (+/-) + 28.2k (\times).\]
\[2 \times 9 \times 4 \times 16^2 (Bt + Bi + Bm) = 18432 \times ((4 (+) + 4 (\times)) + (4 (+) + 4 (\times)) + 2 (+/-)) = 184.3k (+/-) + 147k (\times).\]
The above results are listed in Table 4. It is obvious that the required computation of the proposed method is less than that of MBMA.

In MBA, the number of motion vectors that needed to be transmitted is equal to the number of grid points of a mesh. In the proposed WIBM, the
overhead information needed to be sent includes mesh structure code and motion vector code. The average number of grid points and overhead bits for the test sequence Claire are compared in Table 5. The average number of grid
points of WIBM is less than that of MBMA and SBMA; thus it needs the least overhead information bits, although the transmission of extra mesh structure code is necessary.

Fig. 10. Comparison of predicted images (a) original Frame 87, (b) original frame 90, (c) predicted image by SBMA, (d) predicted image by MBMA, (e) predicted image by AP mode, and (f) predicted image by proposed method.
5. Conclusion

In this paper, we have developed a new mesh-based motion compensation method for video coding. A regular non-uniform (RN) mesh that provides simple and regular structure with variable patch size is presented. The patch size is varied according to motion activity of a video sequence. Because the mesh generated is denser in moving area than in still area, the motion compensation pay more attention on moving area such that better performance is achieved. We also developed a weight interpolation block-matching scheme for the estimate of displacements of grid points.
This scheme exploits efficiently the motion correlation between neighboring patches, thus improving the estimate accuracy. Simulation results show that the proposed algorithm improves the performance significantly with lower computational complexity and overhead information bits.

References


Chung-Ming Kuo received the B.S. degree from the Chinese Naval Academy, Kaohsiung, Taiwan, in 1982, the M.S. and Ph.D. degrees all in electrical engineering from Chung Cheng Institute of Technology, Taiwan, in 1988 and 1994, respectively. From 1988 to 1991, he was an instructor in the Department of Electrical Engineering at Chinese Naval Academy. Since January 1995, he has been an associate professor at the same institution. In August 2000, he joined the I-Shou University, where he is currently an associate professor with the Department of Information Engineering. His research interests include object-based video compression and retrieving, image processing, multimedia signal processing, and optimal estimation.

Mao-Shung Hung received the B.S. and M.S. degree all in information engineering from the I-Shou University in 1997 and 1999, respectively. In 1999, he joined the MicroLinnks Technology Corporation, where he is S/W engineer. His research interests are in image/video compression, image processing, and multimedia communications.