Frame Rate Up-conversion Using HOE (Hierarchical Optical flow Estimation) Based Bidirectional Optical Flow Estimation

Mamoru Ogaki†, Tetsuya Matsumura††, Koji Nii†††, Masayuki Miyama†, Kousuke Imamura† and Yoshio Matsuda†,

College of Science and Engineering, Kanazawa University, Kakuma-machi, Kanazawa, Ishikawa, Japan†
Renesas Electronics Corporation, Kitaitami Works, 4-1 Mizuhara, Itami, Hyogo, Japan††
Renesas Electronics Corporation, Musashi Works, 5-20-1 Jyousui-honmachi, Kodaira, Tokyo, Japan†††

Summary
We propose a frame rate up-conversion algorithm using bidirectional optical flow estimation at each pixel based on a previously proposed Hierarchical Optical flow Estimation (HOE) algorithm. The HOE is extended from unidirectional flow estimation to bidirectional flow estimation in which a 3×3×2 multi-dimensional filter and a motion compensated image method are used for high accuracy of flow estimation. PSNRs of 30.29–45.99 dB were obtained for five test sequences with computer simulations. The PSNR improvements are 3.50–10.99 dB compared to a simple bidirectional block matching method and 0.26–1.88 dB compared to an existing method based on a bidirectional block matching method.

Key words:
frame rate up-conversion , optical flow, block matching, bidirectional flow estimation, motion compensation.

1. Introduction

In the field of consumer products in applications such as a digital television and personal computers, high-definition video contents on a large screen has been widely used because of the spread of digital broadcasting and the higher bandwidth of the internet. In parallel with this, high-quality video with a wide screen is penetrating into widely various industrial application fields such as automotive applications, surveillance cameras, and digital signage systems. Frame Rate Up-Conversion (FRUC) technology [1, 2], which generates intermediate frames from existing frames, has been developed as a means of realizing higher quality images in these application fields. Recently it has been applied in some classes of consumer products.

Intermediate frame can be generated doubling the previous frame or using the average image obtained from previous and subsequent frames. However, to generate smooth motion, a standard method of generating an intermediate frame is to do so by estimating the motion based on information from previous and subsequent frames, and by executing motion compensation against the current frame based on the estimated motion. For such motion estimation, block matching (BM) [3–6] is generally used. BM motion estimation is suitable for spatially averaged motions. Nevertheless, it has a fundamental weakness for spatially inhomogeneous motions such as rotation and scaling because it cannot predict those motions accurately. It degrades image quality of the intermediate frame. When a motion estimation error occurs, block noise of the intermediate frame often appears as a flickering of the screen, which causes a discomfort feeling. In addition, in the BM mechanism, the intermediate frame composed of blocks might cause noise artifact at the boundary of the blocks. A pixel level process is often required in the boundary area [2, 7, 8]. Therefore, to obtain images with smooth movement, it is desirable to estimate the motion basically in units of pixels, and to generate intermediate frames at the pixel level.

Optical flow technology has been proposed as a method to estimate motion at the pixel level [9–11], but some difficulties arise in motion estimation accuracy. Based on this basic concept, the Hierarchical Optical Flow Estimation (HOE) algorithm, which can realize high accuracy motion estimation, has been reported by the authors [12]. When this HOE method is applied to FRUC, a smoother intermediate frame is obtainable than when using estimation based on the BM method because the HOE method can estimate spatially heterogeneous motion with high accuracy. Moreover, the impact on the block boundary can be minimized.

As described in this paper, we propose a new FRUC algorithm using pixel units. It is based on an algorithm extended from the conventional unidirectional HOE method to bidirectional estimation. In Section 2, we propose the FRUC algorithm. This algorithm can improve accuracy using a motion-compensated image and bidirectional flow estimation instead of conventional unidirectional flow estimation. In Section 3, the results of the computer simulation are described. By computer

Manuscript received June 5, 2012
Manuscript revised June 20, 2012

simulation, we evaluate accuracy in the bi-directional flow estimation in section 2 and verify the effectiveness of using the motion compensation frame. Furthermore, this FRUC algorithm is executed for a typical standard sequence by simulation. We confirm the high efficiency of the proposed algorithm in comparison with other algorithms [5]. Conclusions are given in Section 4.

2. FRUC by extended HOE algorithm

HOE is an algorithm to estimate motions of pixels to be highly accurate. Fundamentally, it is unidirectional flow estimation. In estimation of motions of pixels on the frame at time *t*, three frames are used: the time *t* frame, its prior frame *t*-1 and its subsequent frame *t*+1. If the intermediate frame is generated between frame t and next frame *t*+1 based on the estimated motions on the frame *t* as in the FRUC, the generated pixel does not correspond to the integer pixel on the intermediate frame. Bidirectional motion estimation around the intermediate frame used in BM method can be applied to the HOE and can make pixels with estimated motion correspond to the integer pixels. In this case, motions to be estimated are ones on the intermediate frame to be generated from now on. Therefore, we can no three frames in motion estimation. Thus, the advantage of HOE of high-precision motion estimation using three frames will be lost.

In this section, we resolve these issues by extending the HOE and we apply it to the FRUC. In the following, we describe a bidirectional flow estimation algorithm on the basis of the HOE.

2.1 HOE-based bidirectional flow estimation

Suppose that intermediate frame *M* is generated by predicted motions from the successive two frames: *A* and *B*. As shown in Fig. 1(a), the integer pixel of frame *A* corresponds to the decimal pixel on the intermediate frame because this motion is generally decimal precision. Therefore, the problem arises such that the integer pixel on the intermediate frame cannot be determined uniquely in unidirectional motion estimation [7, 13, 14]. Although we might adopt a nearest integer pixel to an estimated decimal pixel, an integer pixel might appear in double or as a hole. It causes unnatural boundary noises on the generated frame.

In the proposed algorithm, as shown in Fig. 1(b), optical flows are estimated bidirectionally from frame *A* and frame *B* under constant velocity assumption. That is, frame *A* and frame *B* have movement of +*d* and –*d* to frame *M*, respectively.

When we denote the luminance value M(*x*, *y*, *t*) at pixel (*x*, *y*) in the intermediate frame to be generated, then the luminance values of the corresponding frame *A* and *B* are given as *A* (*x*-*u*, *yv*, *t*-1), *B* (*x*+*u*, *y*+*v*, *t*+1). Here, *d* = (*u*, *v*) represents the flow and *u* and *v* signify the *x* and *y* components of the flow, respectively. Assuming the conservation law of luminance, eq. (1) is obtained. ξ represents luminance change unrelated to motion caused by environments [15–17].

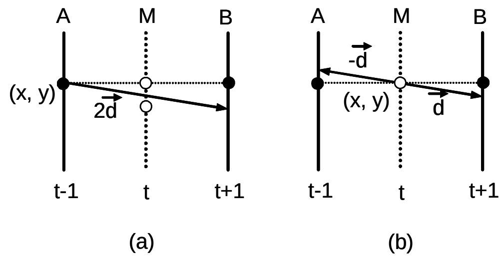

Fig. 1 (a) Unidirectional flow estimation and (b) proposed bidirectional flow estimation.

$$B(x+u, y+v, t+1) - A(x-u, y-v, t-1) = -\xi \tag{1}$$

Taylor expansion gives eq. (2) in first order approximation when *u* and *v* are small.

$$I_x u + I_y v + I_t + \xi = 0 \tag{2}$$

Here, I_x, I_y, I_z are given as shown in eq. (3).

$$\begin{aligned} I_x &= A_x(x, y, t-1) + B_x(x, y, t+1) \\ I_y &= A_y(x, y, t-1) + B_y(x, y, t+1) \\ I_t &= -A(x, y, t-1) + B(x, y, t+1) \end{aligned} \tag{3}$$

A_x, B_x and A_y, B_y are the spatial luminance gradients of A and B respect to *x* and *y*, respectively. *It* corresponds to the temporal luminance gradient. Flows *u* and *v* are determined as values minimizing the error function defined by eq. (4).

$$\begin{aligned} E(u,v) &= \iint dxdy\,(f^2 + \alpha^2 f_a{}^2 + \beta^2 f_b{}^2) \\ f^2 &= (I_x u + I_y v + I_t)^2 \\ f_a{}^2 &= u_x{}^2 + u_y{}^2 + v_x{}^2 + v_y{}^2 \\ f_b{}^2 &= \xi_x{}^2 + \xi_y{}^2 \end{aligned} \tag{4}$$

u_x, u_y, v_x, v_y, ξ_x, and ξ_y are spatial derivatives of *u*, *v*, and ξ for *x* and *y* and α and β are parameters. An integral is taken over the entire frame. *u*, *v*, and ξ which minimize eq. (4) are given as eq. (5).

$$(I_x u + I_y v + I_t + \xi) I_x = \alpha^2 \Delta u$$
$$(I_x u + I_y v + I_t + \xi) I_y = \alpha^2 \Delta v$$
$$I_x u + I_y v + I_t + \xi = \beta^2 \Delta v \tag{5}$$

In addition, the Laplacian Δu_{ij} at pixel (i, j) is approximated by eq. (6).

$$\begin{aligned}\Delta u_{i,j} &= \frac{1}{12}\left(u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i;1,j+1}\right) \\ &+ \frac{1}{6}\left(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j}\right) - u_{i,j} \\ &= \bar{u} - u_{i,j}.\end{aligned} \tag{6}$$

Applying the same approximation to v and ξ, the simultaneous linear equation is obtained as:

$$\begin{aligned} u &= \bar{u} - I_x \frac{I_x \bar{u} + I_y \bar{v} + I_t + \bar{\xi}}{\alpha^2 + I_x^{\,2} + I_y^{\,2} + \lambda^2} \\ v &= \bar{v} - I_y \frac{I_x \bar{u} + I_y \bar{v} + I_t + \bar{\xi}}{\alpha^2 + I_x^{\,2} + I_y^{\,2} + \lambda^2} \\ \xi &= \bar{\xi} - \lambda^2 \frac{I_x \bar{u} + I_y \bar{v} + I_t + \bar{\xi}}{\alpha^2 + I_x^{\,2} + I_y^{\,2} + \lambda^2} \\ \lambda &= \frac{\alpha}{\beta} \end{aligned} \tag{7}$$

Therein, u and v are calculated by iteration. In eq. (7), the subscripts i and j are omitted. Finally, the intermediate frame is generated from eq. (8) using u and v.

$$M(x,y,t) = \frac{1}{2}\{A(x-u, y-v, t-1) + A(x+u, y+v, t+1)\} \tag{8}$$

I_x in eq. (2) is a spatial luminance gradient at a pixel of the generated intermediate frame M (x, y). Results show that I_x is expressed simply as an average of the spatial luminance gradient at pixel (x, y) of frames A and B. With calculation of I_x in HOE, the accuracy improvement of the motion estimation can be accomplished using a 3×3×3 multi-dimensional gradient filter for the three frames: t-1, t, and t+1 [18].

In this algorithm, I_x should be calculated with high accuracy using another method because frame at t does not exist yet. I_x and I_y calculated with the usual two-point difference value do not yield high precision unlike the multi-dimensional gradient filter because I_x is simply an average of A_x and B_x. Using three frames such as A, prior A, and subsequent A for A_x calculation, four frames of t-3, t-1, t+1, and t+3, are required because of the B_x calculation. This is undesirable with regard to frame memory capacity and frame delay. In addition, it is not well consistent with motion compensated frame in the lower layer. In this algorithm, a 3×3×2 multi-dimensional gradient filter (a slight modification of a 3×3×2 multi-dimensional filter) is used. This point will be described later.

2.2 Improvement of flow accuracy using a motion compensated image

Determination of highly accurate flow will improve the quality of the intermediate frame. The concept of modification flow using motion compensated image is followed to improve the flow accuracy. This algorithm is based on a linear approximation, so it can only support a small movement. For a large movement, the motion compensated image and hierarchical image are applied [19, 20].

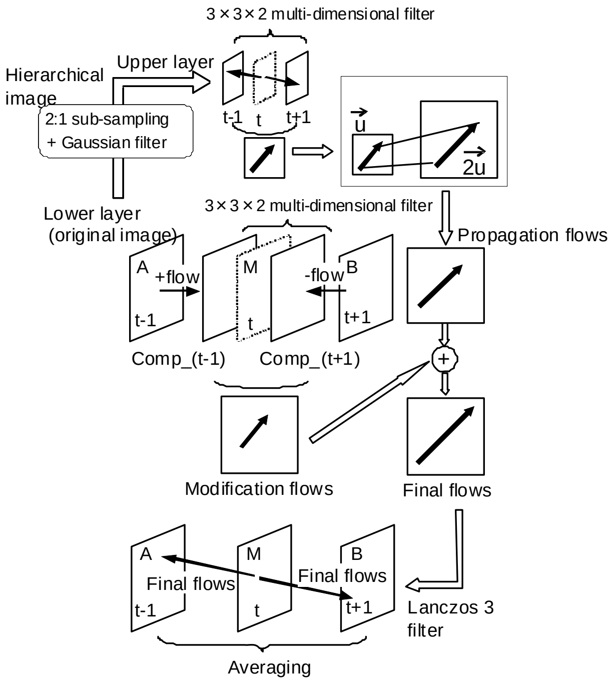

Fig. 2 Intermediate frame generation with bidirectional flow estimation using HOE algorithm

Figure 2 shows the flow of intermediate frame generation in the proposed bidirectional flow estimation based on the HOE. Firstly, applying the Gaussian filter and 2:1 sub-sampling, hierarchy images of t-1 and t+1 for frame A and B each are obtained. After creating the top layer, both the spatial luminance gradient and the temporal luminance gradient are calculated using the 3×3×2 multi-dimensional gradient filter. Flows are obtained by solving eq. (7) with iteration. Double of this flow is propagated to lower layer in the hierarchy. Using this propagation flow, motion compensated images, Comp_(*t*-1) and Comp_ (*t*+1), are generated from the frame of *t*-1 and *t*+1 [12]. Subsequently, the modification flow in this lower layer in the hierarchy is calculated again with application of the 3×3×2 multi-dimensional gradient filter. The sum of the modification flow obtained in this layer and the propagation flow from the upper layer is the flow of this layer. Repeating this process to lowest layer, a final flow is obtained. Finally, *A* (*x*-*u*, *y*-*v*, *t*-1) and *B* (*x*+*u*, *y*+*v*, *t*+1) are calculated using the final flow obtained and intermediate frame is generated with its average. The luminance value of the decimal pixel of *A* and *B* are generated from integer pixels with the Lanczos3 filter.

Introducing motion compensated image Comp_(*t*-1) and Comp_(*t*-1), the motion has been kept small at a lower layer in the hierarchy, which improves the estimation accuracy of flow together with hierarchical images. In creating the hierarchical image using 2:1 sub-sampling, a 5×5 Gaussian filter shown in Fig. 3 is applied at the same time.

$\frac{9}{400}$	$-\frac{3}{80}$	$-\frac{3}{25}$	$-\frac{3}{80}$	$\frac{9}{400}$
$-\frac{3}{80}$	$\frac{1}{16}$	$\frac{1}{5}$	$\frac{1}{16}$	$-\frac{3}{80}$
$-\frac{3}{25}$	$\frac{1}{5}$	$\frac{16}{25}$	$\frac{1}{5}$	$-\frac{3}{25}$
$-\frac{3}{80}$	$\frac{1}{16}$	$\frac{1}{5}$	$\frac{1}{16}$	$-\frac{3}{80}$
$\frac{9}{400}$	$-\frac{3}{80}$	$-\frac{3}{25}$	$-\frac{3}{80}$	$\frac{9}{400}$

Fig. 3 5x5 Gaussian filter.

2.3 Calculation of luminance gradient

Both the spatial luminance gradient and temporal luminance gradient of frames *A* and *B* are necessary in the flow calculation. For simple approximation of difference at successive two points, the accurate flows are unobtainable because the process is too sensitive to subtle luminance changes and noises. The calculation of the spatial luminance gradient and temporal luminance gradient using a three-tap multi-dimensional gradient filter was proposed conventionally [12, 18]. As described in Section 2.1, the multi-dimensional gradient filter is inapplicable as it is. Therefore, 3×3×2 dimensional gradient filter (modified the multi-dimensional gradient filter) is used. Figure 4 shows pixels for use in this 3×3×2 multi-dimensional gradient filter. Table 1 shows its coefficients.

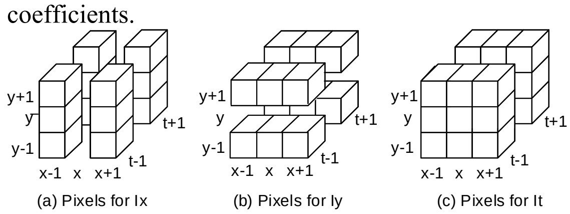

Fig. 4 Pixels for spatial and temporal gradient calculations.

	$k_0/p_0/d_0$	$k_1/p_1/d_1$
Spatial LPF coefficients (k)	0	0.500
Temporal LPF coefficients (p)	0.552	0.224
Gradient coefficients (d)	0	0.455

Table 1 LPF coefficients and gradient coefficients.

In the calculation of the spatial luminance gradient, coefficients of Low Pass Filter (LPF) of the conventional multi-dimensional gradient is applied in the spatial direction. The average is taken in the temporal direction. In the same manner, temporal luminance gradient is calculated. The Low Pass Filter (LPF) coefficients of a conventional multi-dimensional gradient filter are applied in the spatial direction. Gradient coefficients as shown in Table 1 are applied to the calculation of temporal gradient.

A conventional multi-dimensional gradient filter, even in the temporal direction, applies the spatial direction LPF coefficients shown in Table 1. In this filter, it is understood that the k1 coefficient is equal to 0.5 by sharing 0.276 each from P0 coefficient of 0.552 to frame for both side. As the bidirectional flow estimation is based on an assumption of constant speed, the image of the frame *t*-1, and *t*+1 are not so different. Consequently, almost identical precision is expected with conventional multi-dimensional gradient filters. It is consistent that I_x is expressed as average of A_x and B_x, as shown in eq. (3). In the calculation of temporal luminance gradient, there is no problem because the frame at t is not included in the conventional 3×3×3 multi-dimensional gradient filter.

These values are calculated concretely as follows. First, we explain the spatial luminance gradient. I^{Lt} (*x*, *y*, *t*) is generated from frames *A* and *B* with eq. (9) using the coefficients of the temporal direction LPF filter as shown in Table 1.

$$I^{Lt}(x,y,t) = k_1\{A(x,y,t-1) + B(x,y,t+1)\} \tag{9}$$

By eq. (10) I^{Lty} (*x*, *y*, *t*) is obtained with application of the spatial LPF in the *y* direction against I^{Lt} (*x*, *y*, *t*). The final spatial luminance gradient for the x direction Ix (x, y, t) is expressed from I^{Lty} (x, y, t) using gradient coefficients d_1 by eq. (11).

$$I^{Lty}(x,y,t) = p_0 I^{Lt}(x,y,t) + p_1\left\{I^{Lt}(x,y-1,t) + I^{Lt}(x,y+1,t)\right\} \tag{10}$$

$$Ix = d_1\left\{I^{Lty}(x+1,y,t) - I^{Lty}(x-1,y,t)\right\} \tag{11}$$

In the same manner, Iy (x, y, t) is also calculated from I^{Lty} (x, y, t) as eqs. (12) (13).

$$I^{Ltx}(x,y,t) = p_0 I^{Lt}(x,y,t) + p_1\left\{I^{Lt}(x-1,y,t) + I^{Lt}(x+1,y,t)\right\} \tag{12}$$

$$Iy = d_1\left\{I^{Ltx}(x,y+1,t) - I^{Ltx}(x,y-1,t)\right\} \tag{13}$$

Second, we explain the calculation of temporal luminance gradient. As shown in eqs. (14) and (15), after applying the spatial LPF to A and B for the x and y directions, the luminance gradient for t is calculated as the eq. (16) using the gradient coefficient d_1. I^{Lyx} (x, y, t +1) of eq. (16) is expressed for frame B.

$$I^{Ly}(x,y,t) = p_0 A(x,y,t-1) + p_1\left\{A(x,y-1,t-1) + A(x,y+1,t-1)\right\} \tag{14}$$

$$I^{Lyx}(x,y,t) = p_0 I^{Lx}(x,y,t-1) + p_1\left\{I^{Lx}(x-1,y,t-1) + I^{Lx}(x+1,y,t-1)\right\} \tag{15}$$

$$It(x,y,t) = d_1\left\{I^{Lyx}(x,y,t+1) - I^{Lyx}(x,y,t-1)\right\} \tag{16}$$

3. Simulation results

3.1 Accuracy of bidirectional flow estimation

The accuracy of the proposed bidirectional flow estimation was verified by computer simulation. It is compared to the HOE algorithm, which realizes the highest level accuracy currently [12, 15].

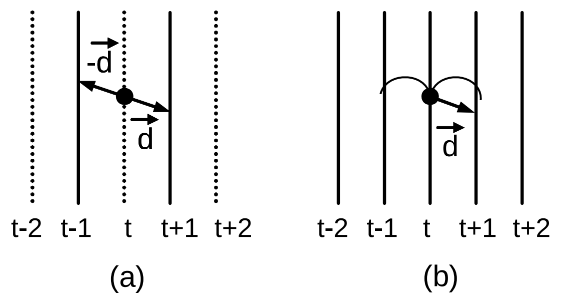

Fig. 5 (a) Bidirectional flow estimation in the proposed algorithm and (b) unidirectional flow estimation in the original HOE algorithm.

In the proposed method, when the motion of the frame t is estimated, a 3×3×2 multi-dimensional gradient filter is applied with respect to the frame of previous t-1 and next t+1, as shown in Fig. 5(a). To prepare the magnitude of flow estimated, we used three frames at t-2, t, and t+2 in the HOE algorithm, as shown in Fig. 5(b).

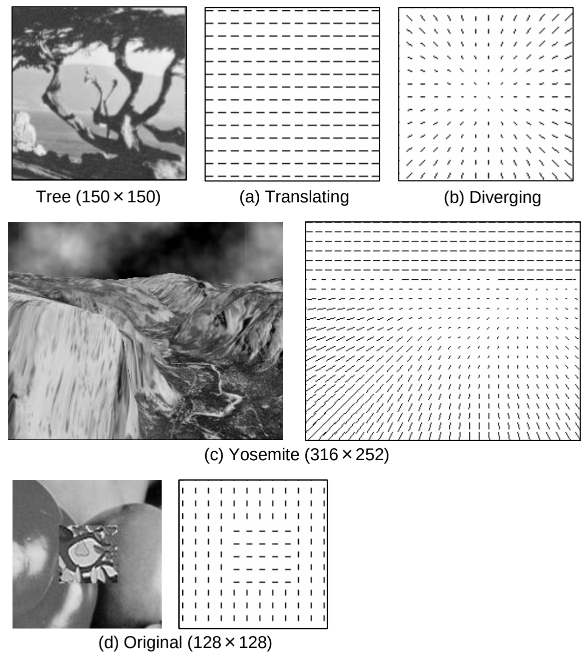

Fig. 6 Test sequences and its correct flows.

Four test sequences with correct flow were selected: Translating Tree (Trans), Diverging Tree (Div), Yosemite (Yos), and originally created sequence (Org). Figure 6 shows the test sequences and their correct flow. Trans is a sequence in which the entire image moves in 1.73–2.26 pixels/frame in the right horizontal direction. Div zooms around the neighborhood of center with movements of 1.29 and 1.86 pixels/frame on the left and right, respectively. Yos is a cloud sequence for which the entire image moves to right direction about 2 pixels/frame, whereas the lower-left portion is running about 4 pixels/frame. In the original sequence (Org), the central rectangle object moves to right with 1 pixel/frame in the horizontal direction, while the background is moving upper in the 1 pixel/frame along the vertical direction. As an index of accuracy, the Mean Angular Error (MAE) and the Mean velocity Magnitude Error (MME) defined by eqs. (17) and (18) were used [21, 22].

$$MAE = \frac{1}{MN}\sum_{i,j} \cos^{-1}\left(\frac{u_c v_c + u_e v_e + 1}{\sqrt{(u_c^2 + v_c^2 + 1)(u_e^2 + v_e^2 + 1)}}\right) \tag{17}$$

$$MME = \frac{1}{MN}\sum_{i,j}\left|\sqrt{u_c^2 + v_c^2 + 1} - \sqrt{u_e^2 + v_e^2 + 1}\right|. \tag{18}$$

Here, (u_c, v_c) is a correct flow and (u_e, v_e) represents the obtained flow. Iterations are terminated when it reaches the preset 150 times or when the amount of update of the flow reaches less than 0.0001. Here, parameters are $\alpha = 10$ and $\beta = 5$.

Figures 7 and 8 show the MAE and MME in case of $L = 2$, 3, 4, respectively. The proposed method estimates the motion equivalent to 2d because two frames are used, while HOE estimates the motion d using three successive frames. In spite of this, almost equal accuracy is obtained from both the proposed method and HOE. Trans and Yos sequences include motion of about 3.0–4.0 pixels/frame. In this simulation, the proposed method must detect the motion of about 8 pixels/frame because the motion between two frames is twice. Results show that the flow accuracy with L=2 has degraded in both sequences. HOE is confirmed to detect the motion of about 3.5–4.0 pixels/frame in two layers [15]. When executing FRUC actually, intermediate frames are generated between two frames that have a motion of 4 pixels/frame. Therefore, it is in a sufficient range for detection in two layers. The estimation accuracy of bidirectional flow estimation does not mean special degradation compared to the HOE.

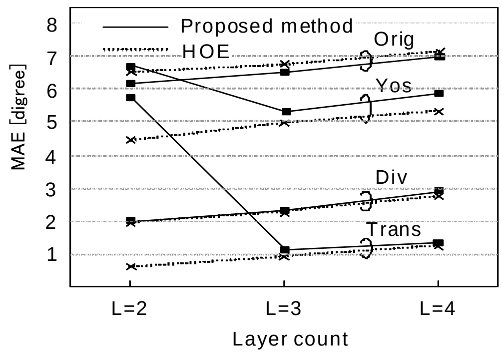

Fig. 7 MAE (Mean Angle Error) of estimated flows in the proposed method and the original HOE algorithm.

3.2 Effectiveness of motion compensated image

In the proposed method, the accuracy of the flow estimation is improved by introducing motion compensated image in addition to hierarchical structure in order to suppress motions small in each layer. We confirmed an effectiveness of a motion compensated image using four sequences stated in the previous section. The flows were simulated in the proposed algorithm with and without the motion compensation. The results of the MAE and MME are shown in Figs. 9 and 10, respectively (L = 3). In the case of no motion compensated image, the propagation flow is used as the initial value of iteration in the lower layer.

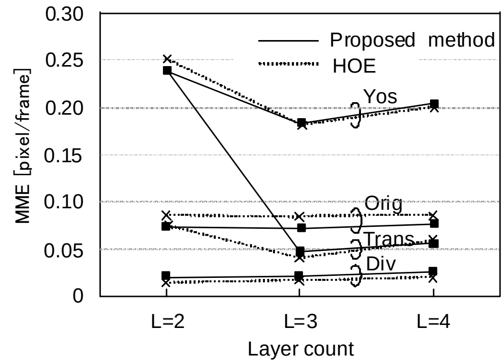

Fig. 8 MME (Mean velocity Magnitude Error) of estimated flows in the proposed method and the original HOE algorithm.

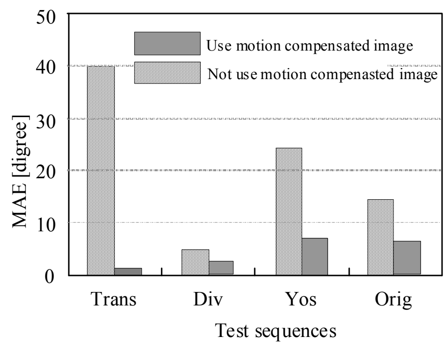

Fig. 9 Improvements due to the motion compensation images: MAE.

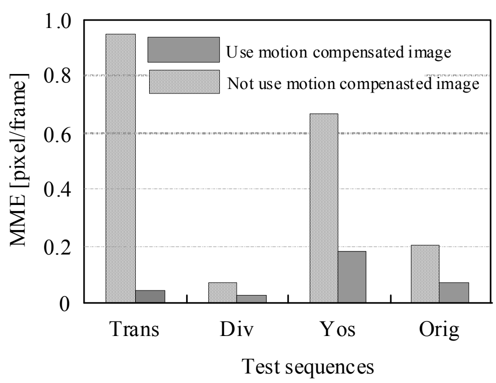

Fig. 10 Improvements due to the motion compensation images: MME.

As shown in Figs. 9 and 10, it is readily apparent that applying the motion compensation image has engendered accuracy improvement. In case of not using a motion compensated image, the accuracy of the Trans was markedly degraded. This is the same reason that the flow of Trans with large movement deteriorated significantly in L=2 as described in Section 3.1. Similarly, in Yos with the same degree of movement, the accuracy in the case of no motion compensated image degrades greater than that in the case of using one. The introduction of motion compensated image together with the hierarchical method can be confirmed to realize profound improvement of flow accuracy in estimating large motion.

3.3 FRUC with the proposed algorithm

FRUC simulation was executed using the proposed algorithm. Five sequences were used for comparison with other existing algorithms: *foreman*, *table tennis*, *flower*, *mobile*, and *Akiyo*. All of these are CIF 30 fps. *Flower* is a 250 frames sequence in all. The other four sequences are 300 frames sequences. In these sequences, an odd frame is generated from the even frame. Table 2 shows computer simulation results.

	Frame average	Bidirec-tional BM	MOFRUC ref. [5]	Proposed method
Foreman	28.40	25.45	32.62	32.88
Table	27.99	28.97	32.17	32.47
Flower	19.19	24.50	30.40	32.01
Mobile	23.88	19.30	28.41	30.29
Akiyo	46.24	37.18	46.78	45.99

Table 2 Comparison with other FRUC algorithms: PSNR.

PNSR in Table 2 presents the average of the entire generated even frame (*flower*, 125 frames; the other four, 150 frames). For comparison, the PSNR in other three method: the simple frame average, bidirectional BM method and in an earlier report [5] are also listed in Table 2. In the BM method, the 8×8 block size and the bidirectional full search with search range of ±8 were used. This search range is equivalent to the hierarchical structure with four layers in the proposed method and both algorithms can detect the same degree of motion. Reference [5] gives the highest level PSNR compare to other existing methods to the best of our knowledge, although there are little differences among sequences.

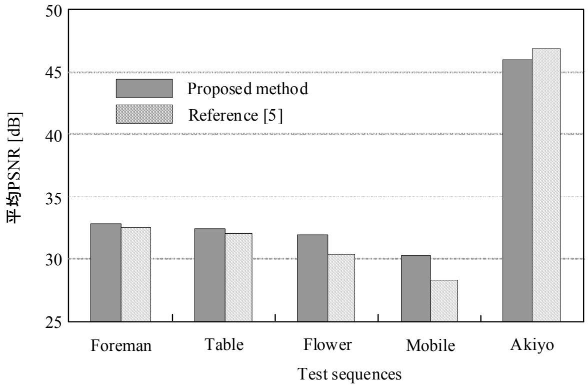

Fig. 11 Comparison with other FRUC algorithms: PSNR.

In the four sequences aside from *Akiyo*, the proposed method improves PSNRs of 3.5–11.82 dB compared to the simple frame average method and BM method. It was also confirmed that higher PSNRs of 0.26–1.88 dB were obtained compared to Ref. [5]. Figure 11 shows the PSNR of the proposed method and the Ref. [5]. The proposed method is particularly effective in images including fine structures or complex motions such as *flower* and *mobile* because the FRUC is executed at the pixel level. In the case of *Akiyo*, all methods give very high PSNR. There is no apparent difference in subjective evaluation because most images in the *Akiyo* sequence are still images.

Figure 12 shows the 290th frame and the 48th frame in mobile sequence, which has the smallest and largest difference of PSNR between the BM method and the proposed method, respectively. As shown in Fig. 12(b), some objects are dropped in the upper center and in the lower left portion (ball). Seeing carefully, block noises appear in many parts. On the other hand, as shown in Fig. 12(c), unnatural noises do not appear in the frame in the proposed method. However, the black ball at the left bottom is missed because the movement of the object is too large to capture. As setting the search range ± 8 in the BM method, this ball is also missed in the BM method. Compared with Fig. 12(e) and Fig. 12(f), which have a largest PSNR differences, almost all of characters on the calendar are lost completely in the BM method, while the proposed method generates them completely.

Figures 13(a–c) shows expanded images of the areas encountered by white line at the left bottom in Figs. 12(d–f), respectively. Many block boundary noises such as the surface of the ball and the leaves of the tree appear in the BM method as shown in Fig. 13(b). However, in the proposed method shown in Fig. 13(c), objects are reproduced almost completely, even in portions with the fine structure and object boundaries.

Figure 14 shows the 164th frame of flower sequence with the smallest PSNR difference between the BM and the proposed method. Three images are shown in Fig. 14: (a) the original image, (b) the image generated by the BM method (b), and (c) the image generated using the proposed method. Although the PSNR difference between image (b) and (c) is small, detailed portions such as birds and trees on the street are lost in the BM method, while these objects are completely generated in the proposed method as in Fig. 14(c). The proposed method is extremely effective.

(a) Original image:290th frame (b) BM method（PSNR:24.58 dB） (c) Proposed method（PSNR:27.69 dB）

(d) Original image:48th frame (e) BM method（PSNR:18.34 dB） (f) Proposed method（PSNR:33.87 dB）

Fig. 12 Mobile: (a) Original image:290th frame, (b) BM method, (c) proposed method, (d) original image 48th frame, (e) BM method, and (f) proposed method.

(a) Original image:48th frame (b) BM method（PSNR:18.34 dB） (c) Proposed method（PSNR:33.87 dB）

Fig. 13 Mobile: (a) Original image: 48th frame (b) BM method, and (c) proposed method.

(a) Original image:164th frame (b) BM method（PSNR:24.30 dB） (c) Proposed method（PSNR:26.83 dB）

Fig. 14 Mobile: (a) Original image: 164th frame (b) BM method, and (c) proposed method

4. Conclusion

The FRUC algorithm in the pixel unit using optical flows was proposed. The optical flow estimation is based on the extended Hierarchical Optical flow Estimation (HOE) algorithm, in which the unidirectional flow estimation with successive three frames is extended to bidirectional one with successive two frames. In the proposed algorithm, the same accuracy was achieved to the original HOE in spite of using two frames. Owing to the bidirectional flow estimation, doubling or missing of the pixel does not appear on the generated frame in the FRUC. By computer simulation, the PSNR of 30.29–45.99 dB are obtained in five sequences: *foreman*, *table tennis*, *flower*, *mobile*, and *Akiyo*. The PSNR improvements are 3.50–10.99 dB compared to a simple bi-directional block matching method and 0.26–1.88 dB compared to an existing "Multiple Objective Frame Rate Up-conversion" method based on a bidirectional block matching method.

References

[1] B. W. Jeon, G.-I. Lee, S.-H. Lee, and R.-H. Park, "Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure," IEEE Trans. Consum. Electron., vol.49, no.3, pp. 499–508, Aug., 2003.

[2] B. Choi, "New Frame Rate Up-Conversion using Bi-directional Motion Estimation," IEEE Trans. Consum. Electron., vol.46, no.3, pp. 603–609, Aug., 2000.

[3] T. Ha, S. Lee, and J. Kim, "Motion compensated frame interpolation by new block-based motion estimation algorithm," IEEE Trans. Consum. Electron., vol.50, no.2, pp. 752–759, May 2004.

[4] S. Fujiwara, and A. Taguchi, "Motion compensated frame Rate Up-conversion Based on Block Matching Algorithm with multi size blocks," Proceeding of International Symposium on Intelligent Signal Processing and Communication Systems, pp. 353–356, Dec., 2005.

[5] Chong, O. C. Au, W.-S. Chau, and T.-W. Chan, "Multiple Objective Frame Rate Up Conversion," Proc. IEEE International Conference on Multimedia and Expo, pp. 253–256, Jul., 2005.

[6] ISO/IEC13818-2, "Information Technology Generic Coding of Moving Pictures and Associated Audio Information: Video," *ISO/IEC 13818-2,* Nov., 1995.

[7] S.-H. Lee, O. Kwon and R.-H. Park, "Weighted adaptive motion-compensated frame rate up-conversion," IEEE Transactions on Consumer Electronics, vol.49, no.3, pp. 485–492, Aug. 2003.

[8] K. Hilman, H. W. Park, and Y. Kim, "Using motion compensated frame-rate conversion for the correlation of 3:2 pulldown artifacts in video sequence," IEEE Trans. Circuits and Systems for Video Tech., vol.10, no.6, pp. 869–877, Sep., 2000.

[9] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," Artificial Intelligence, vol.17 pp.185–203, 1981

[10] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," Proceedings of Imaging Understanding Workshop, pp. 121–130, 1991.

[11] S. H. Wang and S. U. Lee "A Hierarchical Optical Flow Estimation Algorithm Based on the Interlevel Motion Smoothness Constraint," Pattern Recognition, vol.26, no.6, pp. 939–952, 1993.

[12] N. Minegishi, J. Miyakoshi, Y. Kuroda, T. Katagiri, Y. Fukuyama, R. Yamamoto, M. Miyama, K. Imamura, H. Hashimoto, and M. Yoshimoto, "VLSI architecture study of a real-time scalable optical flow processor for video segmentation," IEICE Trans. Electron., vol.E89-C, no.3, pp. 230–242, Mar., 2006.

[13] C. K. Wong and O. C. Au, "Fast motion compensated temporal interpolation for video," Proc. SPIE: Visual Communications and Image Processing, pp. 1108–1118, May, 1995.

[14] C.-W Tang and O. C. Au, "Unidirectional motion compensated temporal interpolation," Proc. of IEEE Int. Symp. on Circuits and Systems, vol.2, pp. 1444–1447, Jun., 1997.

[15] M. A. Gennert and S. Negahdaripour, "Relaxing the brightness constancy assumption in computing optical flow," A. I. Memo, no.975, M.I.T., Jun., 1987.

[16] J. M. Odobez and P. Bouthemy, "Robust multi-resolution estimation of parametric motion models applied to complex scenes," Publication interne IRISA, no.788, 1994.

[17] Y. Muratsubaki, A. Kurokaw, M. Miyama, K. Imamura, and Y. Matsuda, "Fast computation of optical flow under brightness change for hardware implementation," IEICE Trans. Inf. & Syst., Vol. J93-D, No.9, pp. 1766-1777, Sep., 2010 (in Japanese).

[18] E. P. Simoncelli, "Design of multi-dimensional derivative filters," First IEEE Int'l. Conf. Image Processing, vol. I, pp. 790–793, Nov., 1994.

[19] P. J. Burt and E. H. Anderson, "The Laplacian pyramid as a compact image code," IEEE Trans. Commun., vol. COM-31, pp. 532–540, Apr., 1983.

[20] J. Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm," Intel Corp., Microprocessor Research Labs., OpenCV Documents, 1999.

[21] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunk: Combining local and global optical flow methods," Int'l. Jour. Computer Vision, vol.61, no.3, pp. 211–231, 2005.

[22] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow Techniques," Int'l. Jour. Computer Vision, vol.15, no.1, pp. 43–77, 1994.

Mamoru Ogaki was born in Ishikawa, Japan, in 1988. He received his B.S Degree from Kanazawa University, Ishikawa, Japan, in 2011. His research interest is VLSI designs for image processing.

Tetsuya Matsumura received the B.E. and Ph.D. degrees from Kyushu Institute of Technology, Fukuoka, Japan in 1984 and 2001, respectively. He joined Mitsubishi Electric Corporation in 1984. He was transferred to Renesas Technology Corporation in 2003. He currently works on software based multimedia platform for Car Information System in the Software Platform Division, Renesas Electronics Corporation, Itami, Hyogo, Japan.

Koji Nii received the B.E. and M.E. degrees from Tokushima University, in 1988 and 1990, respectively, and the Ph.D. degree in informatics and electronics engineering from Kobe University, Hyogo, Japan, in 2008. He joined Mitsubishi Electric Corporation in 1990. He was transferred to Renesas Technology Corporation in 2003. He currently works on R&D of low-power design techniques in the Technology Development Unit, Renesas Electronics Corporation, Kodaira, Tokyo, Japan. Dr. Nii holds 76 US patents, and published 22 papers and 66 talks at conferences. He is a Technical Program Committee of the IEEE CICC and a member of the IEEE SSCS and EDS. He is also a Visiting Associate Professor of Graduate School of Natural Science and Technology, Kanazawa University.

Masayuki Miyama was born on March 26, 1966. He received a B.S. degree in Computer Science from the University of Tsukuba in 1988. He joined PFU Ltd. in 1988. He received an M.S. degree in Computer Science from the Japan Advanced Institute of Science and Technology in 1995. He joined Innotech Co. in 1996. He received a Ph.D. degree in electrical engineering and computer science from Kanazawa University in 2004. He is an assistant professor at Kanazawa University Graduate School of Science and Technology. His present research focus is VLSI designs for real-time image processing.

Kousuke Imamura received the B.S., M.S. and Dr. Eng. degrees in Electrical Engineering and Computer Science in 1995, 1997 and 2000, respectively, all from Nagasaki University. He is currently an Associate Professor in the Institute of Science and Engineering at Kanazawa University. His research interests are high efficiency image coding and image processing.

Yoshio Matsuda was born in Ehime, Japan, on October 26, 1954. He received the B.S. degree in physics and the M.S. and Ph.D. degree in applied physics from Osaka University in 1977, 1979, and 1983, respectively. He joined the LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan, in 1985. He was engaged in development of DRAM, advance CMOS logic, and high frequency devices and circuits of compound semiconductors. Since 2005, he has been a professor of Graduate School of Natural Science and Technology at Kanazawa University, Japan. His research is in the fields of integrated circuits design where his interests have includes multimedia system, low power SoCs, and image compression processors.