Proceedings of the 6th IASTED International Conference
SIGNAL AND IMAGE PROCESSING
August 23-25, 2004, Honolulu, Hawaii, USA

Robust Motion Estimation for Video Sequences Based on Phase-Only Correlation

Loy Hui Chien and Takafumi Aoki
Graduate School of Information Sciences
Tohoku University
Aoba-yama 05, Sendai, 980-8579, Japan
email: loy@aoki.ecei.tohoku.ac.jp

ABSTRACT

In this paper, we present a robust Phase-Only Correlation (POC)-based motion estimation method for video sequences. Robust motion estimation is indispensable for many applications such as mesh-based video coding, stereo vision, and super-resolution imaging. In our proposed method, the motion vector of a point in a video frame is adaptively switched between motion vectors obtained by two motion estimation methods: (i) POC-based full search and (ii) POC-based hierarchical search. This approach can reliably detect both global and local motion in video sequences. We evaluate the robustness of our proposed method in mesh-based motion compensation. Experimental results show that our proposed method performs significantly better than conventional full search using Sum of Absolute Differences (SAD).

KEY WORDS

robust motion estimation, phase-only correlation, SAD, full search, hierarchical search

1 Introduction

Motion estimation is the process of determining the movement of the objects of a video sequence. The movement is usually expressed in terms of the motion vectors of selected points within the current frame with respect to another frame known as the reference frame. A motion vector represents the displacement of a point between the current frame and the reference frame.

Motion estimation is a fundamental task in numerous fields, such as image processing, image analysis, video coding, and computer vision. Robust high-accuracy motion estimation is essential for applications such as mesh-based motion compensation for video coding [1], stereo vision 3D measurement [2], and super-resolution imaging [3] (the reconstruction of a high-resolution image using multiple low-resolution images). Here, robustness refers to consistent pixel-level estimation of motion vectors with minimal false detection.

Among the various motion estimation methods, the block matching algorithm is most popular due to its simplicity. In block matching algorithms, an image block centered on a point in the current frame is compared with candidate blocks in the reference frame based on certain dissimilarity or similarity measures in order to find the best matching block within a predefined search area. An example of a dissimilarity measure is Sum of Absolute Differences (SAD). The motion vector of the point is given by the block displacement. Strategies for finding the best matching block are broadly classified into two types: full search methods and hierarchical search methods. The former is suitable for detecting local motion of individual objects, while the latter is suitable for detecting global motion of the scene.

Recently, a high-accuracy image matching technique using a Phase-Only Correlation (POC) function [4]–[6] has been developed. Using the POC function, we can estimate the translational displacement as well as the degree of similarity between two image blocks from the location and height of the correlation peak, respectively. It has been demonstrated that this matching technique can estimate the displacement between two images with 1/100-pixel accuracy when the image size is about 100×100 pixels [5].

This paper presents a robust POC-based motion estimation method for video sequences. Our proposed method combines the advantages of hierarchical search with those of full search. The motion vector of a point is adaptively switched between motion vectors obtained from POC-based hierarchical search and POC-based full search, so as to detect both global and local motion in video sequences. We evaluate the robustness of our proposed method using mesh-based motion compensation, where the quality of the motion compensated images is highly dependent on the robustness of the motion estimation method.

2 Phase-Only Correlation

Consider two $N_1 \times N_2$ images, $f(n_1, n_2)$ and $g(n_1, n_2)$, where we assume that the index ranges are $n_1 = -M_1, \cdots, M_1$ and $n_2 = -M_2, \cdots, M_2$ for mathematical simplicity, and hence $N_1 = 2M_1 + 1$ and $N_2 =$

$2M_2+1$. Let $F(k_1,k_2)$ and $G(k_1,k_2)$ denote the 2D Discrete Fourier Transforms (2D DFTs) of the two images. $F(k_1,k_2)$ and $G(k_1,k_2)$ are given by

$$\begin{aligned} F(k_1,k_2) &= \sum_{n_1 n_2} f(n_1,n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} \\ &= A_F(k_1,k_2) e^{j\theta_F(k_1,k_2)}, \quad (1) \\ G(k_1,k_2) &= \sum_{n_1 n_2} g(n_1,n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} \\ &= A_G(k_1,k_2) e^{j\theta_G(k_1,k_2)}, \quad (2) \end{aligned}$$

where $k_1 = -M_1, \cdots, M_1$, $k_2 = -M_2, \cdots, M_2$, $W_{N_1} = e^{-j\frac{2\pi}{N_1}}$, $W_{N_2} = e^{-j\frac{2\pi}{N_2}}$, and the operator $\sum_{n_1 n_2}$ denotes $\sum_{n_1=-M_1}^{M_1} \sum_{n_2=-M_2}^{M_2}$. $A_F(k_1,k_2)$ and $A_G(k_1,k_2)$ are amplitude components, and $e^{j\theta_F(k_1,k_2)}$ and $e^{j\theta_G(k_1,k_2)}$ are phase components.

The cross-phase spectrum (or normalized cross spectrum) $\hat{R}(k_1,k_2)$ is defined as

$$\begin{aligned} \hat{R}(k_1,k_2) &= \frac{F(k_1,k_2)\overline{G(k_1,k_2)}}{|F(k_1,k_2)\overline{G(k_1,k_2)}|} \\ &= e^{j\theta(k_1,k_2)}, \quad (3) \end{aligned}$$

where $\overline{G(k_1,k_2)}$ denotes the complex conjugate of $G(k_1,k_2)$ and $\theta(k_1,k_2) = \theta_F(k_1,k_2) - \theta_G(k_1,k_2)$. The Phase-Only Correlation (POC) function $\hat{r}(n_1,n_2)$ is the 2D Inverse Discrete Fourier Transform (2D IDFT) of $\hat{R}(k_1,k_2)$ and is given by

$$\begin{aligned} &\hat{r}(n_1,n_2) \\ &= \frac{1}{N_1 N_2} \sum_{k_1 k_2} \hat{R}(k_1,k_2) W_{N_1}^{-k_1 n_1} W_{N_2}^{-k_2 n_2}, \quad (4) \end{aligned}$$

where $\sum_{k_1 k_2}$ denotes $\sum_{k_1=-M_1}^{M_1} \sum_{k_2=-M_2}^{M_2}$.

Now consider $f_c(x_1,x_2)$ as a 2D image defined in continuous space with real-number indices x_1 and x_2. Let δ_1 and δ_2 represent sub-pixel displacements of $f_c(x_1,x_2)$ to x_1 and x_2 directions, respectively. So, the displaced image can be represented as $f_c(x_1-\delta_1, x_2-\delta_2)$. Assume that $f(n_1,n_2)$ and $g(n_1,n_2)$ are spatially sampled images of $f_c(x_1,x_2)$ and $f_c(x_1-\delta_1,x_2-\delta_2)$, and are defined as

$$\begin{aligned} f(n_1,n_2) &= f_c(x_1,x_2)|_{x_1=n_1T_1, x_2=n_2T_2}, \\ g(n_1,n_2) &= f_c(x_1-\delta_1, x_2-\delta_2)|_{x_1=n_1T_1, x_2=n_2T_2}, \end{aligned}$$

where T_1 and T_2 are the spatial sampling intervals, and index ranges are given by $n_1 = -M_1, \cdots, M_1$ and $n_2 = -M_2, \cdots, M_2$. The POC function $\hat{r}(n_1,n_2)$ between $f(n_1,n_2)$ and $g(n_1,n_2)$ will be given by

$$\begin{aligned} &\hat{r}(n_1,n_2) \\ &\simeq \frac{\alpha}{N_1 N_2} \frac{\sin\{\pi(n_1+\delta_1)\}}{\sin\{\frac{\pi}{N_1}(n_1+\delta_1)\}} \frac{\sin\{\pi(n_2+\delta_2)\}}{\sin\{\frac{\pi}{N_2}(n_2+\delta_2)\}}, (5) \end{aligned}$$

where $\alpha \le 1$. The peak position of the POC function corresponds to the displacement between the two images, and the peak value α corresponds to the degree of correlation between the two images.

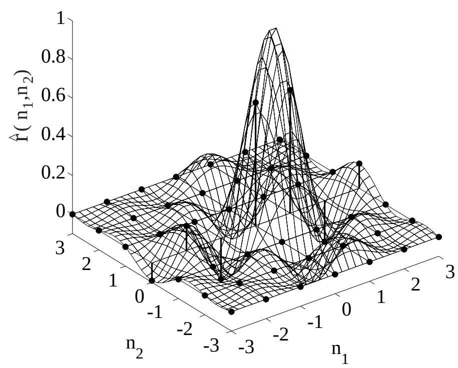


Figure 1. Function fitting for estimating the peak position.

For high accuracy sub-pixel image matching, the following techniques [5] are important: function fitting using equation (5) for high-accuracy estimation of sub-pixel displacement (δ_1,δ_2) and peak value α, application of Hanning window to reduce image boundary effects and low-pass filtering to reduce aliasing and noise effects. Figure 1 shows an example of function fitting using equation (5) to estimate the true position and height of the correlation peak.

In this paper, we use two versions of the POC function for image matching. We use a simplified version where only windowing technique and low-pass filtering is employed for fast computation. The displacement between two image blocks is estimated by detecting the position of the maximum value of the POC function $\hat{r}(n_1,n_2)$ in equation (4) with pixel accuracy. We also employ a high-accuracy version that uses function fitting, windowing and low-pass filtering technique for high-accuracy estimation of correlation peak α and sub-pixel displacement (δ_1,δ_2).

3 POC-Based Motion Estimation

In this section, our proposed POC-based motion estimation method is described. We describe two different search strategies for block matching: POC-based full search and POC-based hierarchical search. We also propose an adaptive search strategy that adaptively switches between the result of the above two algorithms for more robust motion estimation.

3.1 Full Search Method

In this section, we describe a POC-based full search motion estimation method (known thereafter as POC-FS) that requires less calculation than conventional full search while having almost no loss in optimality.

We consider two image blocks of size $W \times W$ with Hanning window function of the same size applied. Since the half-width of the Hanning window function is $\frac{W}{2}$, we may consider the maximum reliable displacement estimate between the two image blocks to be $\pm\frac{W}{4}$ both horizontally and vertically. Hence, instead of examining every candidate block, we only need to examine candidate blocks at $\frac{W}{4}$ pixel intervals in the search area.

Procedure for POC-FS

Input:

current image $I(n_1, n_2)$,
reference image $J(n_1, n_2)$,
point p in $I(n_1, n_2)$

Output:

corresponding point q of point p in $J(n_1, n_2)$,
motion vector v_p^{FS} of point p

Step 1: Extract an image block of size $W \times W$ centered on point p in the current frame $I(n_1, n_2)$.

Step 2: Calculate POC function between the image block in the current frame $I(n_1, n_2)$ and candidate blocks in the reference frame $J(n_1, n_2)$ every $\frac{W}{4}$ pixels in the search area. We use the simplified version of the POC function to obtain the rough correlation peak (defined as the maximum value of the POC function $\hat{r}(n_1, n_2)$ in Equation (4)) and the peak position which gives the displacement between the two blocks with pixel accuracy.

Step 3: Identify the top 3 matching blocks in Step 2, and displace the blocks by their displacement estimates. This brings the correlation peaks to the centers of the blocks. Re-calculate the POC function for the 3 new candidate blocks. The best matching block is the block with the highest correlation peak among the 3 blocks. Again, the simplified version of the POC function is used. The corresponding point q is centered on the best matching block.

Step 4: Find the motion vector $v_p^{FS} = q - p$.

In our experiments we set the window size as 32×32 and the search range as ±16 or ±32 both horizontally and vertically, depending on the expected range of motion.

3.2 Hierarchical Search Method

In the POC-based hierarchical search motion estimation method (known thereafter as POC-HS), some coarser versions of the original input images are created. This method is also known as the coarse-to-fine correspondence search technique [6]. The POC-based block matching starts at the coarsest image layer and the operation gradually moves to the finer layers. The motion vector detected at each layer is propagated to the next finer layer in order to guide the search at that layer. An overview of the technique is shown in Figure 2. Let p_o be the given point in the current image, and q_0 be the corresponding point in the reference image, and let p_l and q_l be the matching points at the l-th layer. The aim of the correspondence search is to find the corresponding point q_0 of point p_0 and in doing so, we obtain the motion vector of p_0 as $q_0 - p_0$.

Procedure for POC-HS

Input:

current image $I_0(n_1, n_2)$ $(= I(n_1, n_2))$,
reference image $J_0(n_1, n_2)$ $(= J(n_1, n_2))$,
point p_0 $(= p)$ in $I_0(n_1, n_2)$

Output:

corresponding point q_0 of point p_0 in $J_0(n_1, n_2)$,
motion vector $v_{p_0}^{HS}$ of point p_0

Step 1: For $l = 1, 2, \cdots, l_{max}$, create the l-th layer images $I_l(n_1, n_2)$ and $J_l(n_1, n_2)$, i.e., coarser versions of $I_0(n_1, n_2)$ and $J_0(n_1, n_2)$, recursively as follows:

$$I_l(n_1, n_2) = \frac{1}{4}\sum_{i_1=0}^{1}\sum_{i_2=0}^{1} I_{l-1}(2n_1 + i_1, 2n_2 + i_2),$$

$$J_l(n_1, n_2) = \frac{1}{4}\sum_{i_1=0}^{1}\sum_{i_2=0}^{1} J_{l-1}(2n_1 + i_1, 2n_2 + i_2).$$

In our experiments we set the value of l_{max} to 2 or 3 depending on the expected range of motion.

Step 2: For every layer $l = 1, 2, \cdots, l_{max}$, calculate the coordinate $p_l = (p_{l1}, p_{l2})$ corresponding to the original point p_0 recursively as follows:

$$p_l = \lfloor \tfrac{1}{2} p_{l-1} \rfloor = (\lfloor \tfrac{1}{2} p_{l-1\ 1} \rfloor, \lfloor \tfrac{1}{2} p_{l-1\ 2} \rfloor). \qquad (6)$$

Step 3: We assume that $q_{l_{max}} = p_{l_{max}}$ in the coarsest layer. Let $l = l_{max} - 1$.

Step 4: From the l-th layer images $I_l(n_1, n_2)$ and $J_l(n_1, n_2)$, extract two image blocks (of size $W \times W$) $f_l(n_1, n_2)$ and $g_l(n_1, n_2)$ with their centers on p_l and $2q_{l+1}$, respectively. For accurate matching, the size of image blocks should be reasonably large. In our experiments, we use 32×32 image blocks.

Step 5: Estimate the displacement between $f_l(n_1, n_2)$ and $g_l(n_1, n_2)$ with pixel accuracy using the simplified

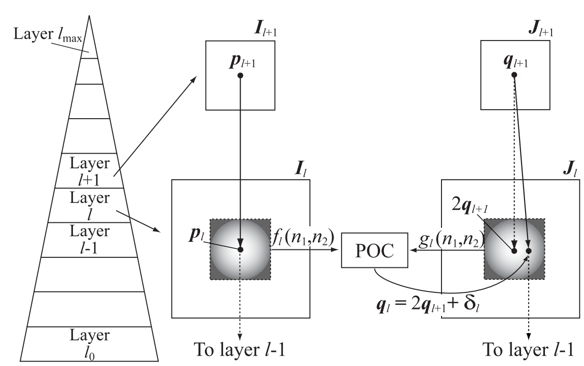


Figure 2. Block matching using a hierarchical coarse-to-fine approach.

version of the POC function, in which the displacement is determined to pixel-level accuracy. Let the estimated displacement vector be δ_l. The l-th layer correspondence q_l is determined as follows:

$$q_l = 2q_{l+1} + \delta_l. \tag{7}$$

Step 6: Decrement the counter by 1 as $l = l - 1$ and repeat from **Step 4** to **Step 6** while $l \geq 0$.

Step 7: Find the motion vector $v_{p_0}^{HS} = q_0 - p_0$.

3.3 Adaptive Search Method

To adapt to both global and local motion in video sequences, we propose a POC-based adaptive search motion estimation method (known thereafter as POC-HS/FS) that adaptively switches between POC-HS and POC-FS to detect correct motion.

When switching between POC-HS and POC-FS, two factors come into consideration. The first factor is how well the image block matches with its best matching block. This is given by the correlation peak α of the POC function. The second factor is to what extent the motion vector at a point p correlates with the motion vectors of the set of its surrounding points S_p. We introduce a measure D, which represents how much the motion vector v_p differs from the motion vectors of the points in S_p. S_p may not necessarily refer to the adjacent pixels of p. We can freely define the surrounding points to have some distance from p.

$$D(v_p) = \sum_{s \in S_p} |v_p - v_s|. \tag{8}$$

POC-FS is effective for detecting local motions of individual objects, but the possibility of mismatching with a similar-looking block is high. On the other hand, POC-HS uses a larger matching area to increase the reliability, but may fail to detect the correct motion when the block overlaps with two or more objects with different motion. We use the following discriminant Z for each point p to decide which result to use.

$$Z = \frac{\alpha_p^{FS}}{\alpha_p^{HS}} \times \frac{D(v_p^{HS})}{D(v_p^{FS})}. \tag{9}$$

Here, α_p^{FS}, α_p^{HS} are the correlation peaks of the POC function for the best matching block found using POC-FS, POC-HS respectively. We use the function fitting technique described in Section 2 for high-accuracy estimation of the peak values. If $Z \geq 1.0$, then the v_p^{FS} is chosen, otherwise if $Z < 1.0$, then v_p^{HS} is chosen.

Procedure for POC-HS/FS
Input:
  current image $I(n_1, n_2)$,
  reference image $J(n_1, n_2)$,
  point p in $I(n_1, n_2)$
Output:
  motion vector v_p of point p
Step 1: Use POC-HS to find v_p^{HS} and α_p^{HS}.
Step 2: Use the following procedure to obtain v_p.
    If α_p^{HS} > threshold value k then
      $v_p = v_p^{HS}$
    else
      Use POC-FS to find v_p^{FS} and α_p^{FS}
      Calculate discriminant Z
      If $Z \geq 1.0$ then
        $v_p = v_p^{FS}$
      else
        $v_p = v_p^{HS}$
      end
    end

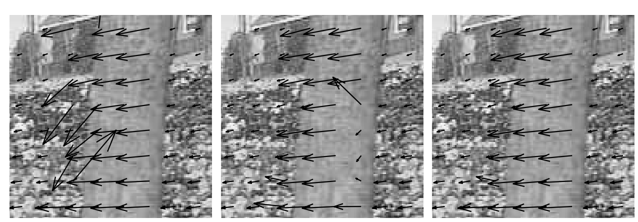
In our experiments we let $k = 0.5$.

An example of motion estimation using POC-FS, POC-HS and POC-HS/FS is shown in Figure 3 for the video sequence *Flower Garden*. POC-FS fails to detect the correct motion for some points close to the tree boundary because of occlusion. On the other hand, due to different motion of the tree and the background, POC-HS fails to detect the correct motion for some points within the tree. By adaptively switching between POC-FS and POC-HS, a more satisfactory result is obtained by POC-HS/FS.

4 Experiment and Evaluation

In our first experiment we investigate the robustness of POC-HS/FS for mesh-based motion compensation using a wide variety of MPEG-4 standard test video sequences that have mostly global motion, local motion or a good mix of both. We create the motion compensated image of the current frame and compare it with the current frame to calculate PSNR.

The motion compensation procedure for the current frame t is as follows. Nodes are positioned every 16 pixels to form a square mesh that partitions the image into square blocks. To avoid boundary occlusion effects, we exclude a frame boundary of 16 pixels



(a) (b) (c)

Figure 3. Motion estimation for *Flower Garden* using (a) POC-FS, (b) POC-HS, and (c) POC-HS/FS.

from the motion compensation process. Motion estimation is done for every node in the current frame t with respect to the reference frame $t-1$. The projective transformation of a point (n_1, n_2) in frame t to the point (n'_1, n'_2) in frame $t-1$ is defined as

$$\begin{pmatrix} n'_1 \\ n'_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_1\, h_2\, h_3 \\ h_4\, h_5\, h_6 \\ h_7\, h_8\; 1 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ 1 \end{pmatrix}, \qquad (10)$$

where $h_1 \sim h_8$ are the projective parameters of the projective transformation matrix H, shown in Figure 4. The coordinates of the four corner nodes of an image block in frame t and the coordinates of their corresponding nodes in frame $t-1$ are used to calculate H. Next, the inverse matrix H' is used to create the motion compensated image block of frame t, as shown in Figure 4. This procedure is repeated for every block to generate the motion compensated image of frame t.

To calculate H accurately, sub-pixel accuracy of motion vectors is required. We use the high-accuracy version of the POC function described in Section 2, where function fitting is employed to calculate the sub-pixel displacement after the motion vectors have been estimated to pixel-level accuracy using POC-HS/FS.

In our experiment, we also compare our proposed method with SAD-based full search motion estimation method (known thereafter as SAD-FS). The block size for SAD-FS is set at 16×16, corresponding to the half-width of the 32×32 Hanning window used for POC-based image matching. For sub-pixel estimation of motion vector, we use bilinear interpolation to estimate to 0.125-pixel accuracy.

Video sequences often contain uniform, textureless and featureless regions that cause mismatching. An example of this is the sky in the video sequences *Shinjuku* and *Flower Garden*. To prevent such mismatching, we identify nodes in areas that have low standard deviation of luminance and set their motion vectors to zero.

The average PSNR of the motion compensated images is given in Table 1. We see that the average PSNR is higher for POC-HS/FS than for SAD-FS. In general, POC-HS/FS produced more robust estimation of motion vectors than SAD-FS, hence resulting in more accurate motion compensation. For video sequences that contained similar-looking textures such as *Shinjuku*, *Church* and *Mobile Calendar*, POC-HS/FS was significantly more robust, outperforming SAD-FS by 2.5∼10 dB. *Kiel Harbour*, which contains a gradual zoom-in, had only a slightly higher PSNR for POC-HS/FS. In Table 1, we also observe that the average number of block matches for pixel-level matching is significantly lower for POC-HS/FS.

Figures 5 and 6 plot the PSNR per frame for *Mobile Calendar* and *Shinjuku* respectively. In both cases the PSNR for POC-HS/FS is consistently higher than for SAD-FS. Figures 7 and 8 give examples of motion compensated images. We observe that the motion compensated images for POC-HS/FS are much closer to the original images.

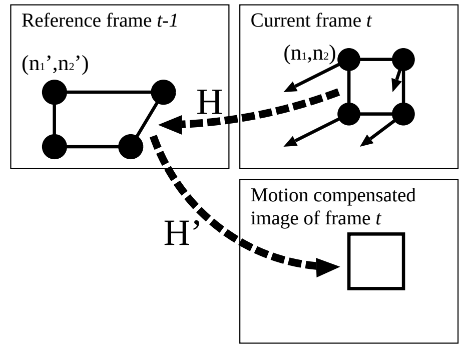


Figure 4. Motion compensation procedure for a block.

In our second experiment we investigate the effectiveness of the adaptive switching mechanism of POC-HS/FS. We employ 50 frames of the video sequence *Flower Garden*, which contains fast horizontal camera motion, along with an even faster-moving foreground tree object (an example is shown in Section 3.3). We call this sequence *Flower Garden*∗. We perform the motion compensation procedure for each current frame t with respect to reference frames $t-b$, where $b = 1$,2,3 and 4, to simulate various degrees of global and local motion.

The average PSNR for POC-HS, POC-FS, and POC-HS/FS, along with the percentage of vectors that were adaptively switched from POC-HS to POC-FS are shown in Table 2. We observe that the percentage of switched vectors increases with larger apparent disparity between the global motion of the background and the local motion of the tree object. For each value of b, the average PSNR for POC-HS/FS is highest, thus demonstrating the effectiveness of the switching mechanism.

5 Conclusion

This paper presents a robust POC-based motion estimation method for video sequences. In the proposed method, the motion vector is adaptively chosen from the result of POC-based hierarchical search and POC-based full search, so as to accommodate both global and local motion. The proposed method is evaluated using mesh-based motion compensation, where robust motion estimation is essential. We have demonstrated that the proposed method is generally more robust than the conventional SAD-based full search method.

Further comparison of POC-based methods with SAD-based methods, as well as the application of the proposed method to stereo image matching, moving object segmentation, etc. are left for future study.

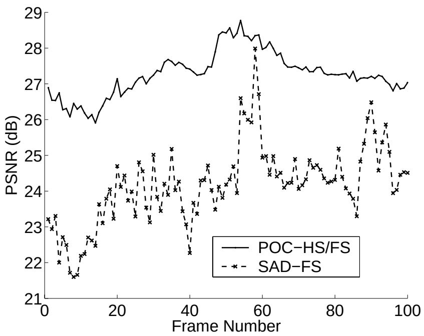


Figure 5. PSNR of motion compensated images for *Mobile Calendar*.

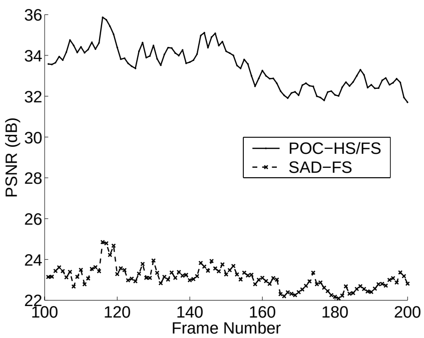


Figure 6. PSNR of motion compensated images for *Shinjuku*.

Acknowledgements

The authors wish to thank Mr. Satoshi Kondo and the team at AV Core Technology Development Center/Image Processing Group, Matsushita Electric Industrial Co., Ltd., for their kind support and advice.

References

[1] G. J. Sullivan and R. L. Baker,"Motion compensation for video compression using control grid interpolation," *Proc. IEEE ICASSP*, Vol. 4, pp. 2713–2716, May 1991.

[2] O. D. Faugeras, *Three-Dimensional Computer Vision*, MIT Press, 1993.

[3] C. P. Sung, K. P. Min, and G. K. Moon,"Super-resolution image reconstruction: A technical overview," *IEEE Signal Processing Magazine*, Vol. 20, No. 3, pp. 21–36, 2003.

[4] C. D. Kuglin and D. C. Hines,"The phase correlation image alignment method," *Proc. Int. Conf. on Cybernetics and Society*, pp. 163–165, 1975.

[5] K. Takita, T. Aoki, Y. Sasaki, T. Higuchi, and K. Kobayashi,"High-accuracy subpixel image registration based on phase-only correlation," *IEICE Trans. Fundamentals*, Vol. E86-A, No. 8, pp. 1925–1934, August 2003.

[6] K. Takita, M. A. Muquit, T. Aoki, and T. Higuchi,"A sub-pixel correspondence search technique for computer vision applications," *IEICE Trans. Fundamentals*, August 2004,(to be published).

(a) (b) (c)

Figure 7. Motion compensation images for *Mobile Calendar*: (a) Original image, (b) POC-HS/FS, and (c) SAD-FS.

(a) (b) (c)

Figure 8. Motion compensation images for *Shinjuku*: (a) Original image, (b) POC-HS/FS, and (c) SAD-FS.

Table 1. Average PSNR of motion compensated images for POC-HS/FS, SAD-FS (unit: dB). (The average number of block matches for pixel-level matching is shown in parentheses.)

Sequence	POC-HS/FS	SAD-FS
Shinjuku	33.04 (1.2×10^3)	23.02 (2.5×10^5)
Church	34.43 (8.2×10^2)	31.93 (4.0×10^5)
Kiel Harbour	26.55 (1.4×10^3)	26.21 (4.0×10^5)
Flower Garden	31.68 (7.8×10^2)	30.00 (3.3×10^5)
Foreman	35.04 (1.5×10^3)	33.44 (4.6×10^5)
Mobile Calendar	27.28 (1.0×10^3)	24.17 (4.8×10^5)

Table 2. Average PSNR of motion compensated images for POC-HS, POC-FS, and POC-HS/FS (unit: dB) for *Flower Garden*$*$, and average percentage of switched vectors.

b	POC-HS	POC-FS	POC-HS/FS	%
1	28.55	27.12	28.57	0.8
2	25.33	24.03	25.48	2.4
3	23.43	22.50	23.88	5.1
4	21.70	21.16	22.44	8.2