Correspondence

Phase Information and Space Filling Curves in Noisy Motion Estimation

V. Bruni, D. De Canditiis, and D. Vitulano, *Member, IEEE*

***Abstract*—This correspondence presents a novel approach for translational motion estimation based on the phase of the Fourier transform. It exploits the equality between the averaging of a group of successive frames and the convolution of the reference one with an impulse train function. The use of suitable space filling curves allows to reduce the error in motion estimation making the proposed approach robust under noise. Experimental results show that the proposed approach outperforms available techniques in terms of objective (PSNR) and subjective quality with a lower computational effort.**

***Index Terms*—Complexity, motion estimation, fast Fourier transform (FFT), phase correlation, space filling curves.**

I. Introduction

MOTION estimation is crucial in several research areas such as video coding, image registration, etc. [1], [2]. The most famous approach is the block matching algorithm (BMA) [3] because of its accuracy and robustness to noise. However, it suffers from a high complexity so that a lot of research effort has been devoted to its speed-up (see, for instance, [4]–[7]). More recently, transform-based approaches have received an increasing interest. Most of them exploit the fast Fourier transform (FFT) shifting property [8]–[15] and work on blocks of the image, instead of single pixels as the more recent pel recursive [16], [17] and optical flow approaches [18], [19]. More precisely, if $\{b_k(x,y)\}_{k=0,1}$ are two aligned blocks, i.e., located at the same position in two successive frames (the analyzed group of frames, GOF), for a translational motion it holds

$$b_1(x,y) = b_0(x-d_x, y-d_y), \quad 0 \le x,y \le M-1 \tag{1}$$

where $M \times M$ is the block size, $b_0(x,y)$ is the reference block, x is the row index, y is the column index, and $\mathbf{d} \equiv (d_x, d_y)$ is the motion vector. Their normalized cross power spectrum is then

$$e^{j(\mathrm{Arg}(\hat{b}_0(\omega_x,\omega_y))-\mathrm{Arg}(\hat{b}_1(\omega_x,\omega_y)))} = \frac{\hat{b}_0(\omega_x,\omega_y)\hat{b}_1^*(\omega_x,\omega_y)}{|\hat{b}_0(\omega_x,\omega_y)\hat{b}_1^*(\omega_x,\omega_y)|} \tag{2}$$

where $\hat{\cdot}$, $\mathrm{Arg}(\cdot,\cdot)$ and $\cdot^*$, respectively, indicate the FFT, the phase, and the complex conjugation of $\cdot$. The inverse FFT of the left member of (2) combined with (1) allow to achieve the phase correlation surface that contains a peak $\delta(x-d_x, y-d_y)$ in correspondence to the shift $\mathbf{d}$ (just one translational motion in each block is considered). That is why this procedure is called phase correlation algorithm (PCA) [13].

The approaches oriented to reduce the computational effort of PCA can be split into two wide classes. The first one includes the discrete cosine transform-based methods: they are oriented to both avoid complex numbers and exploit the same domain used in video coding [14], [20]–[22]. Unfortunately, the resulting complexity is just slightly reduced. Approaches belonging to the second class estimate motion in the Fourier domain, avoiding the inversion of the transform. Among them, an efficient solution based on the sawtooth shape of the Fourier phase of two shifted blocks has been recently proposed by Balci and Foroosh [23], [24]. Using the phase separability [25], the number of phase cycles in each direction gives the motion

$$\begin{aligned} d_x &= \frac{M}{2\pi}\frac{\partial(\mathrm{Arg}(\hat{b}_0)-\mathrm{Arg}(\hat{b}_1))}{\partial\omega_x} \\ d_y &= \frac{M}{2\pi}\frac{\partial(\mathrm{Arg}(\hat{b}_0)-\mathrm{Arg}(\hat{b}_1))}{\partial\omega_y}. \end{aligned} \tag{3}$$

Its robustness to noise can be increased by a regularization, but with a higher computational effort.

This paper focuses on a novel approach for estimating motion directly in the FFT domain with a computational effort saving. It exploits a very simple but general result in the time domain: the average B of K aligned blocks in K successive frames is equivalent to the convolution $b_0 * \delta_{\mathrm{comb}}$ between the reference block b_0 (i.e., lying in the reference frame) and a suitable impulse train δ_{comb} (also known as *Dirac comb* or *sampling function*). This result allows us to estimate the displacement $\mathbf{d}$ by comparing the FFT phases of B and $b_0 * \delta_{\mathrm{comb}}$ in a nonzero frequency. This approach, called motion estimation based on phase information (MEPI), has the following advantages: *i)* motion is automatically estimated without search—as it happens using PCA (search of the best peak) and BMA (search of the best block); *ii)* a drastic reduction of the computational effort (with respect to PCA) is provided by avoiding the inverse FFT and by using two only values of the forward FFT; *iii)* GOF can contain an arbitrary number of frames further reducing its complexity. However, MEPI may fail for noisy sequences because it is based on two only phase values. A theoretical result that allows us to evaluate the precision of the estimated motion in case of noise is presented and then exploited by embedding two suitable space-filling curves in the MEPI's scheme [26]. These curves allow to increase the signal-to-noise ratio by following the main orientation of eventual edges in the considered blocks. Experimental results show that MEPI outperforms available techniques in terms of quality for not severe noise with a lower complexity.

This correspondence is organized as follows. The next section presents MEPI's theoretical model, the use of space filling curves in the noisy case (Section II-A), the proposed algorithm (Section II-B) and its complexity (Section II-C). Section III includes experimental results, discussions and comparisons with the most performing motion estimators.

II. Proposed Model

It can be proved that the average B of K aligned blocks subjected to a translational motion is equivalent to the convolution of the reference block b_0 with a suitable function δ_{comb}.

Proposition 1: If

$$B(x,y) = \frac{1}{K}\sum_{k=0}^{K-1} b_k(x,y), \quad 0 \le x,y \le M-1 \tag{4}$$

Manuscript received July 07, 2008; revised February 27, 2009. First published May 27, 2009; current version published June 12, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stanley J. Reeves.

The authors are with the Istituto per le Applicazioni del Calcolo "M.Picone"-C.N.R., 00161 Rome, Italy (e-mail: bruni@iac.rm.cnr.it; d.decanditiis@iac.rm.cnr.it; vitulano@iac.rm.cnr.it).

Digital Object Identifier 10.1109/TIP.2009.2019808

is the average of $K \in \mathbf{N}$ aligned blocks $\{b_k(x,y)\}_{k=0,1,\ldots,K-1}$ subjected to a translational motion whose shift is $(d_x, d_y) \in \mathbf{Z}^2$, b_0 is the reference block and δ_{comb} is an impulse train function

$$\delta_{\text{comb}}(x,y) = \frac{1}{K}\sum_{k=0}^{K-1}\delta(x-kd_x, y-kd_y) \tag{5}$$

then

$$B(x,y) = (b_0 * \delta_{\text{comb}})(x,y). \tag{6}$$

Proof: By putting (1) into (4), we get the equality: $B(x,y) = (1/K)\sum_{h=0}^{K-1} b_0(x-hd_x, y-hd_y)$. Each element in $B(x,y)$ is the sum of K suitably sampled elements of b_0. From (5), we get

$$\begin{aligned} B(x,y) &= \frac{1}{K}\sum_{h_1=0}^{(K-1)d_x}\sum_{h_2=0}^{(K-1)d_y}\left(b_0(x-h_1, y-h_2)\right. \\ &\quad \left.\times\sum_{k=0}^{K-1}\delta(h_1-kd_x, h_2-kd_y)\right) \\ &= \sum_{h_1=0}^{(K-1)d_x}\sum_{h_2=0}^{(K-1)d_y} b_0(x-h_1, y-h_2)\delta_{\text{comb}}(h_1,h_2) \\ &= (b_0 * \delta_{\text{comb}})(x,y). \end{aligned}$$

•

It turns out that the temporal averaging of corresponding blocks is equivalent to the spatial convolution of the reference block b_0 with an impulse train, whose sampling period depends on the motion vector $\mathbf{d} = (d_x, d_y)$. Using the convolution property, in the Fourier domain, we get

$$\hat{B}(\omega_x, \omega_y) = \hat{b}_0(\omega_x, \omega_y)\hat{\delta}_{\text{comb}}(\omega_x, \omega_y) \tag{7}$$

where $\hat{\delta}_{\text{comb}}(\omega_x, \omega_y) = \sum_{k=0}^{K-1} e^{-i(2\pi/2M-1)k(\omega_x d_x + \omega_y d_y)} = |\hat{\delta}_{\text{comb}}(\omega_x, \omega_y)| e^{-i(\pi/2M-1)(d_x\omega_x + d_y\omega_y)(K-1)}$. A comparison of the phases of both members of (7) at frequencies $(\omega_x, 0)$ and $(0, \omega_y)$ gives

$$\begin{aligned} d_x &= -(\text{Arg}(\hat{B}(\omega_x,0)) - \text{Arg}(\hat{b}_0(\omega_x,0))) \\ &\quad \times P(\omega_x, M), \quad \forall\,(\omega_x, 0) \end{aligned} \tag{8}$$

$$\begin{aligned} d_y &= -(\text{Arg}(\hat{B}(0,\omega_y)) - \text{Arg}(\hat{b}_0(0,\omega_y))) \\ &\quad \times P(\omega_y, M), \quad \forall\,(0, \omega_y) \end{aligned} \tag{9}$$

where $P(\omega_x, M) = ((2M-1)/(K-1)\pi\omega_x)$ and $P(\omega_y, M) = ((2M-1)/(K-1)\pi\omega_y)$. The choice of two independent frequencies allows a direct solution of the linear system giving the motion vector $\mathbf{d}$ without any search.

A. Noisy Case

In the presence of noise, (6) becomes

$$C(x,y) = (b_0 * \delta_{\text{comb}})(x,y) + \frac{1}{K}\sum_{k=0}^{K-1}\epsilon_k(x,y) \tag{10}$$

where $C(x,y) = (1/K)\sum_{k=0}^{K-1} c_k(x,y)$ and $c_k(x,y) = b_k(x,y) + \epsilon_k(x,y)$ are the original (clean) blocks $b_k(x,y)$ with an additive i.i.d. Gaussian noise $\epsilon_k(x,y) \sim N(0,\sigma^2)\ \forall\, k$. Equation (10) is invariant to any linear preprocessing on the noisy frames c_k. On the other hand, nonlinear denoising operators are usually expensive and less performing than singular value decomposition (SVD) based approaches [25]. Therefore, our goal is not to remove noise, but to derive an upperbound for the error on motion estimation when (8) and (9) are evaluated on noisy blocks c_k.

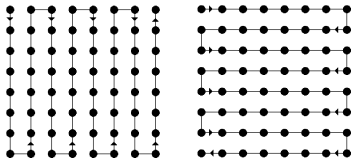

Fig. 1. Two adopted space-filling curves following the vertical (**left**) and the horizontal (**right**) direction.

Proposition 2: The upperbound of the error Δ_x for d_x in (8) with noisy blocks c_k is

$$\begin{aligned} \Delta_x^2 \le \sigma^2\left(\frac{1}{K}\frac{1}{|\hat{C}(\omega_x,0)|^2} + \frac{1}{|\hat{c}_0(\omega_x,0)|^2} + \frac{2}{K}\right) \\ \times P^2(\omega_x, M), \quad \forall\,(\omega_x, 0). \end{aligned} \tag{11}$$

The upperbound of Δ_y for d_y in (9) has the same form, with $(\omega_x, 0)$ replaced by $(0, \omega_y)$.

Proof: Equation (8) evaluated on noisy frames c_k can be rewritten as

$$d_x = P(\omega_x, M)\left(\text{atan}\left(\frac{m_4}{m_3}\right) - \text{atan}\left(\frac{m_2}{m_1}\right)\right)\Big|_{m_i=\bar{m}_i\ i=1,2,3,4}$$

where $\bar{m}_1 = \text{Re}(\hat{C}(\omega_x,0)) = \text{Re}(\hat{B}(\omega_x,0)) + (1/K)\sum_{k=0}^{K-1}\text{Re}(\hat{\epsilon}_k(\omega_x,0)) = m_1 + \eta_1$, $\bar{m}_2 = \text{Im}(\hat{C}(\omega_x,0)) = m_2 + \eta_2$, $\bar{m}_3 = \text{Re}(\hat{c}_0(\omega_x,0)) = m_3 + \eta_3$, $\bar{m}_4 = \text{Im}(\hat{c}_0(\omega_x,0)) = m_4 + \eta_4$. Hence, d_x depends on the variables m_1, m_2, m_3, m_4 that are affected by the errors $\eta_1, \eta_2, \eta_3, \eta_4$. Exploiting the property of an i.i.d. normal random array with zero mean and finite variance in the Fourier domain, as well as the orthogonality property of the cosine and sine bases, the errors $\eta_j,\ j = 1,\ldots,4$ are still normal distributed with zero mean and covariance matrix with nonzero entries: $\text{Cov}(\eta_1,\eta_3) = \text{Cov}(\eta_2,\eta_4) = \sigma^2/K$ and $\text{diag}(\text{Cov}) = \{\sigma^2/K, \sigma^2/K, 1, 1\}$. Therefore, the error Δ_x^2 for d_x can be estimated by applying the error propagation law for derived measures (see chap. V of [27]): $\Delta_x^2 = \sum_{i=1}^4\sum_{j=1}^4(\partial d_x/\partial m_i)(\partial d_x/\partial m_j)|_{\substack{m_i=\bar{m}_i \\ m_j=\bar{m}_j}}\text{Cov}(\eta_i,\eta_j)$. After a simple algebra, we get

$$\begin{aligned} \Delta_x^2 = \sigma^2 P^2(\omega_x, M)\left(\frac{1}{K}\frac{1}{|\hat{C}(\omega_x,0)|^2} + \frac{1}{|\hat{c}_0(\omega_x,0)|^2}\right. \\ \left. - \frac{2}{K}\frac{\text{Re}\left(\hat{C}(\omega_x,0)\hat{c}_0^*(\omega_x,0)\right)}{|\hat{C}(\omega_x,0)|^2|\hat{c}_0(\omega_x,0)|^2}\right). \end{aligned}$$

Considering $|\text{Re}(z)|/|z|^2 \le 1$ for any complex number z, we get (11). A similar result can be provided for Δ_y^2. •

The error Δ_x depends on the inverse signal-to-noise ratio (SNR): $(\sigma^2/|\hat{C}(\omega_x,0)|^2)$ and $(\sigma^2/|\hat{c}_0(\omega_x,0)|^2)$. Hence, the selection of the two lowest frequency values could be the best choice, since the spectrum of many natural images has a decreasing decay [26], [28]. A way for reducing the error in (11) may be to increase the size of the GOF. However, the hypothesis of translational motion may become false. A simple but effective tool can be then adopted: the space-filling curves [26]. A space-filling curve passes through every point of a 2-D region of the image. Its aim is to increase the clean signal correlation, while leaving the noise unchanged. Each block can be then modeled through a curve whose preferential direction is parallel to the main orientation of the edges inside it. Bearing in mind the motion separability [25], the most suitable curves are the ones shown in Fig. 1, that account for horizontal and vertical edges. Curves like the Hilbert (sometimes called Peano) one [26] are not suitable for our purposes because of their anisotropy and the lack of both a closed form and a linear dependence on the horizontal and vertical motion. MEPI can be then extended as

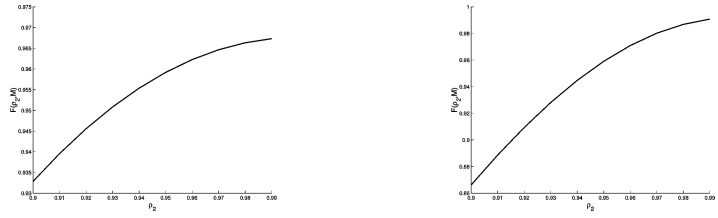

Fig. 2. $F(\rho_2, M)$, as in (13), versus $\rho_2 \in [.9, .99]$ with $M = 8$ (**left**) and $M = 16$ (**right**).

follows. The errors Δ_x and Δ_y are compared with a threshold τ, that depends on the required precision for the motion vector. If they are less than τ, the estimated motion vector (d_x, d_y) is adopted. If one of the two errors, say Δ_y, over-exceeds τ, then d_y is estimated as follows. The block is reorganized using the horizontal scan for further correlating the clean information. Then, d_y is estimated using (9) where $P(1, M) = (2M^2 - 1)/(K - 1)\pi$ while $\hat{C}(2\pi/M^2)$ and $\hat{c}_0(2\pi/M^2)$ are the second coefficient of the 1-D FFT of the resulting noisy 1-D signals. Since the motion for this signal is $Md_x + d_y$, d_y can be derived by simply subtracting the known quantity Md_x from it. Finally, if both Δ_x and Δ_y over-exceed τ, motion vector cannot be estimated and then it is supposed to be equal to $(0, 0)$. The next proposition proves the existence of cases where the reorganization of the block via one of the two proposed space filling curves improves the accuracy of motion estimation. To this aim, each block is modeled as a first-order discrete stationary random field with a separable covariance matrix [28, p.36].

Proposition 3: Let c be a generic block of size $M \times M$ with separable and stationary covariance matrix $r(u, v) = \sigma^2 \rho_1^{|u|} \rho_2^{|v|}, \forall\, u, v$, where $\rho_1 = r(1, 0)/\sigma^2$ and $\rho_2 = r(0, 1)/\sigma^2$ are the one-step correlations in the u and v directions. If $\rho_1 < \rho_2$, let $\hat{c}(0, 2\pi/M)$ and $\hat{c}(2\pi/M^2)$ respectively be the second Fourier coefficient of the block and its 1-D horizontal raster scan. Hence

$$|\hat{c}(0, 2\pi/M)|^2 < |\hat{c}(2\pi/M^2)|^2 \tag{12}$$

is equivalent to

$$\rho_1 < F(\rho_2, M) \quad \text{with}$$
$$F(\rho_2, M) = \frac{\rho_2(\cos(2\pi/M^2) - \cos(2\pi/M))}{1 + \rho_2^2 - \rho_2(\cos(2\pi/M^2) - \cos(2\pi/M))}. \tag{13}$$

A similar result holds if $\rho_2 < \rho_1$ and the vertical raster scan is considered.

Proof: In case of an horizontal raster scan, the correlation matrix for the resulting 1-D signal is $r(u) = \sigma^2 \rho_2^{|u|}, \forall u$. It is the correlation matrix of a stationary first-order Markov sequence. From the Wiener–Khinchin Theorem it holds: $|\hat{c}(\omega)|^2 = \hat{r}(\omega)\ \forall\ \omega$, where $\hat{r}(\omega)$ is the spectral density function [28]. Then, (12) is equivalent to: $\hat{r}(0, 2\pi/M) < \hat{r}(2\pi/M^2)$, i.e.,

$$\sigma^2 \frac{\left(1 - \rho_1^2\right)\left(1 - \rho_2^2\right)}{\left(1 + \rho_1^2 - 2\rho_1 \cos(0)\right)\left(1 + \rho_2^2 - 2\rho_2 cos(2\pi/M)\right)} < \sigma^2 \frac{1 - \rho_2^2}{1 + \rho_2^2 - 2\rho_2 \cos(2\pi/M^2)}.$$

Since $(1 + \rho_2^2 - 2\rho_2 \cos(\bullet)) = |1 - \rho_2 e^{-i\bullet}|^2 > 0$, a simple algebra gives (13). •

Equation (13) is graphically proved in Fig. 2. It shows the behavior of $F(\rho_2, M)$ for $\rho_2 > .9$, that is the usual range in real images (page 37 of [28]). Therefore, when (13) holds, space-filling curves are convenient since they increase SNR that regulates the error Δ_y.

B. Algorithm

For a GOF composed of K frames f_k of size $\bar{M} \times \bar{M}$, split each f_k into nonoverlapping $M \times M$ blocks $c_{j,k}$. $c_{j,k}$ is the jth block in the kth frame f_k with $j = 1, \ldots, \lfloor(\bar{M} \times \bar{M}/M \times M)\rfloor$. For each j:

1) Compute the average C_j of the K corresponding blocks in K successive frames.
2) Evaluate the phase of $\hat{C}_j$ at the frequency values $(\omega_x, \omega_y) = (0, 1)$ and $(\omega_x, \omega_y) = (1, 0)$.
3) Compute the phase of $\hat{c}_{j,0}$ at the same frequency values.
   If $\sigma^2 = 0$, then
   - Compute the motion $\mathbf{d} = (d_x, d_y)$ using (8) and (9).

   else
   - Evaluate Δ_x and Δ_y using (11). Let $3|\Delta_x|$ be the maximum error for d_x, according to the 3σ law [27] (similarly for Δ_y), and let τ be the largest admissible error value, then:
     a) if $3|\Delta_x| < \tau$ and $3|\Delta_y| < \tau$, then estimate d_x and d_y using (8) and (9);
     b) if $3|\Delta_x| < \tau$ and $3|\Delta_y| > \tau$, then estimate d_x using (8). Compute the second coefficient of the FFT of the 1-D signal achieved through the curve in Fig. 1(right). Its motion is $(d_y + Md_x)$. Then subtract Md_x (known) from it to get d_y;
     c) if $3|\Delta_x| > \tau$ and $3|\Delta_y| < \tau$, then apply operations in 3.b), but inverting the roles of x and y and using the curve in Fig. 1(left);
     d) if $3|\Delta_x| > \tau$ and $3|\Delta_y| > \tau$, then set $d_x = 0$ and $d_y = 0$.
4) Assign the motion vector $\mathbf{d} = (kd_x, kd_y)$ to each block $c_{j,k}, k = 1, 2, \ldots, K - 1$.

C. Complexity

We will denote addition, multiplication, division, and comparison, respectively, with $\mathbf{a}$, $\mathbf{m}$, $\mathbf{d}$, and $\mathbf{c}$. For the component x, setting $K = 2$ and omitting arguments, (8) can be rewritten as: $d_x = -P(\mathrm{atan}(\mathrm{Im}(\hat{B}\hat{b}_0^*)/\mathrm{Re}(\hat{B}\hat{b}_0^*))$. Hence, atan argument computation, in terms of $\hat{b}_0$ and $\hat{b}_1$, requires 6 $\mathbf{m}$s, 4 $\mathbf{a}$s, and 1 $\mathbf{d}$. For $\omega_x = 1$, $-P$ requires 1 $\mathbf{m}$, 1 $\mathbf{a}$, and 1 $\mathbf{d}$. Atan computation requires $\log_2(M/2) + 1\mathbf{c}$s: one for the sign, the remaining $\log_2(M/2)$ to determine its absolute value through the search in a binary tree (built once for all). In fact, the (precomputed) values, corresponding to all possible integer shifts are $M/2$, which is the largest integer shift that guarantees a comparison among aligned blocks. With regard to $\hat{b}_0(\omega_x, 0)$, $M(M - 1)$ $\mathbf{a}$s are required for the sum along columns (or rows) plus the 1-D FFT of the resulting signal. The latter exploits the symmetry of cos function in the unitary circle: Re requires $M - 3$ $\mathbf{a}$s and $M/4 - 1$ $\mathbf{m}$s, while $M/2 - 1$ $\mathbf{a}$s and $M/4 - 1\mathbf{m}$s are necessary for Im by exploiting the complementary role of cos and sin. Finally, 2 $\mathbf{d}$s are required for the normalization of FFT. The same amount of operations is required for $\hat{b}_1(\omega_x, 0)$ (or for $\sum_{k=1}^{K-1} \hat{b}_k$ in case of $K > 2$). Considering both x and y direction, the total amount of operations per pixel is 4.3281 for $M = 16$, while its complexity is $O((K + 2)M^2)$.

Fig. 3. Considered test sequences and images. From left to right: Mobile, Coastguard, Flower, Tennis, and Pentagon.

TABLE I
RESULTS OF MEPI, PDM, PCA, FS-BMA, AND 2-BT ON MOBILE, COASTGUARD AND FLOWER SEQUENCES

		Mobile		*Coastguard*		*Flower*	
Method	**No. of op. per Pixel** (16×16)	**PSNR** (16×16)	**PSNR** (8×8)	**PSNR** (16×16)	**PSNR** (8×8)	**PSNR** (16×16)	**PSNR** (8×8)
MEPI	4	26.72 db	28.52 db	33.85 db	35.20 db	24.63 db	25.69 db
PDM	44	25.64 db	26.98 db	32.90 db	33.20 db	24.23 db	25.43 db
PCA	45	24.78 db	25.44 db	32.26 db	32.53 db	24.07 db	25.37 db
FS-BMA	589	22.99 db	23.88 db	30.47 db	31.58 db	23.79 db	25.22 db
2BT	10	22.72 db	22.99 db	29.93 db	30.50 db	23.43 db	24.55 db

TABLE II
PENTAGON IMAGE. SUBPIXEL COMPARISON AMONG MEPI AND PDM [24]

True shifts	(0.167, -0.5)	(0.67, 0.25)	(-0.33, -0.167)	(0.33, 0.33)
PDM	(0.161, -0.502)	(0.68, 0.244)	(-0.34, -0.161)	(0.333, 0.328)
MEPI	(0.163, -0.502)	(0.672, 0.272)	(-0.333, -0.177)	(0.334, 0.338)

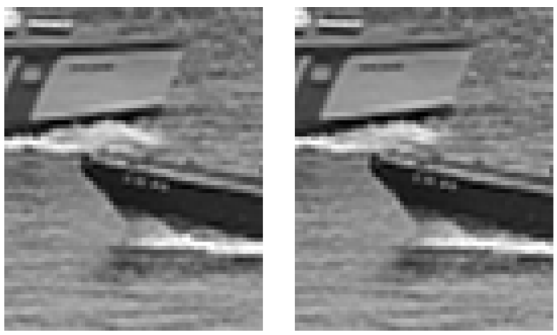

Fig. 4. Detail of Frame no. 31 of Coastguard sequence: original (**left**) and motion compensated via MEPI (**right**).

In case of noise, the error bound in (11) requires 6 $\mathbf{a}$s, 3 $\mathbf{d}$s, and 8 $\mathbf{m}$s. The same is required for Δ_y^2 plus the two $\mathbf{c}$s in step 3) of the algorithm. The computation of atan is still obtained by searching among all its possible (precomputed) values, corresponding to all possible shifts, $[-M/2+Md_x, M/2+Md_x]$ in step 3.b) [or $[-M/2+Md_y, M/2+Md_y]$ in step 3.c)]; the number of $\mathbf{c}$s is still $\log_2(M/2)+1$. The two (for $\hat{c}_0$ and $\hat{c}_1$) 1-D FFTs on M^2 samples organized along one of the 2 space filling curves again need $2(2M^2-4)$ operations. The amount of operations per pixels is 4.4687 in case of step 3.a) of the algorithm (as well as in 3.d, except for atan computation) and 8.4414 in case of step 3.b) or 3.c). The complexity is $O((K+6)M^2)$.

III. EXPERIMENTAL RESULTS AND CONCLUSIONS

MEPI has been tested on three well-known sequences (see Fig. 3): Mobile (140 frames of size 240×352), Coastguard (300 frames of size 352×288), and Flower (60 frames of size 352×240). Table I shows its peak signal-to-noise ratio (PSNR) values, averaged on the whole motion compensated sequence, for $\sigma = 0$, using GOFs of size two ($K = 2$) and blocks of size 8×8 and 16×16. MEPI has been compared with the classical phase correlation algorithm (PCA) [2], Balci and Foorosh approach in [23], denoted with phase difference method (PDM), the classical full search—block matching algorithm (FS-BMA) [3] and the recent two-bit transform (2-BT) [30] based on bit-planes. We note that MEPI performs better than the other methods, improving both the more sophisticated PDM and the classical PCA. In particular, the latter may give false peaks in case of multiple motion or a not precise equivalence between two blocks, i.e., when the actual block is not the precise shifted copy of the reference one. On the contrary, MEPI is more robust thanks to the selection of the lowest frequency in step 2) of the Algorithm and the use of the sum of subsequent frames, that is able to suppress or reduce eventual differences between two aligned blocks. In order to show that this choice does not produce artifacts, in Fig. 4 there is a detail of frame no. 31 of Coastguard that contains the aforementioned effects along with its motion compensated version. This region is composed of 80×64 pixels with topmost and leftmost corner located at pixel (97, 81). For this test, a GOF of size two with 16×16 blocks has been used. It can be noted that no evident distortion appears. However, the most attractive aspect of MEPI stems from its complexity, as shown in Section II-C. The second column of Table I compares the (rounded) number of operations (where the same weight has been assigned to $\mathbf{a}$, $\mathbf{m}$, $\mathbf{d}$ and $\mathbf{c}$) required by MEPI, PDM [23], 2BT [30], FS-BMA [3], and PCA [2]. It can be noted that there is an evident saving of the computational effort with respect to the most competitive approach 2-BT that requires 9.9025 operations per pixel and also with respect to a possible speed-up of PCA via the Discrete Cosine Transform [21], that requires 20 operations per pixel. In a nonoptimized matlab code, the CPU time for Mobile is 47.2 s (Pentium 4, 3.06 GHz, ram 512 Mb). Another test sequence has been selected: Tennis, shown in Fig. 3. In particular, the first 25 frames have been considered, where there is just pure translational motion, using 16×16 blocks. The results, in terms of PSNR, of the three Fourier-based approaches MEPI, PCA and PDM respectively are 33.98, 33.61, and 33.41 db. They are very similar because the common hypothesis of pure global translation is completely fulfilled. However, MEPI requires about 1/10 of the operations that are necessary for PCA and PDM. The same observation holds for Flower sequence (in Table I).

Even if MEPI has not been specifically designed for subpixel estimation, in principle it may be adopted for this purpose too. MEPI has been tested on Pentagon image in Fig. 3, using the method in [11] for producing two shifted images. Table II contains a comparison with PDM. MEPI's performance is promising and very similar to PDM one, except for some cases where it may give worse results—see d_y in the second test. It can be explained by considering that PDM is based on a very powerful regularization approach that exploits all available frequencies, while MEPI uses just one phase value that might result not accurate for precise noninteger shift estimation.

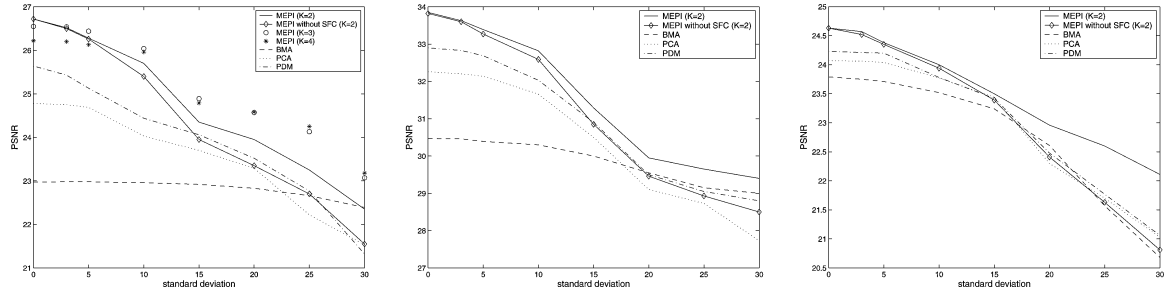

Fig. 5. Mobile (**left**), Coastguard (**middle**) and Flower (**right**) sequences: MEPI with $K = 2$ (with and without space filling curves), PCA, PDM, and BMA results. Mobile figure also includes MEPI with $K = 3, 4$.

In presence of noise, MEPI has been tested on Mobile, Coastguard and Flower sequences using 16×16 blocks, GOFs of size two, with and without the use of space filling curves, as shown in Fig. 5. PSNR is the average on 30 independent trials. The threshold τ in step 3) of the algorithm has been set to .5 in order to achieve a precision of 1 pixel for (integer) motion estimation. MEPI outperforms alternative approaches for $\sigma < 30$ with space filling curves and $\sigma < 15$ without, confirming the advantage in using space filling curves in case of noise. MEPI has also been compared in Fig. 5 using GOFs of size 3 and 4 on Mobile sequence highlighting a further advantage of this approach. MEPI's experiments and theoretical model assume i.i.d. Gaussian noise, since it embeds both the (additive) thermal photodetector noise and the (multiplicative) film grain noise [31]. Hence, in real applications, the robust median estimator in [29] can be applied if the noise variance σ^2 is unknown. For more complicated noise (additive signal dependent) a simple algorithm like MEPI is not effective, since more operations or, at least, more frequencies should be accounted for. However, MEPI's primary objective was to limit the influence of noise in motion estimation without increasing the computational effort, as it is required, for instance, in ancient film restoration. In fact, MEPI's computational effort is still competitive under noise, as shown in Section II-C. Nonetheless, MEPI's performance can be further improved in step 3.d) of the algorithm, by embedding more sophisticated schemes like the one in [24]. Future research will be oriented to both make MEPI more robust for subpixel estimation and extend it to more general motion and noise models.

REFERENCES

[1] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images: A review," *Proc. IEEE*, vol. 76, no. 8, pp. 917–935, Aug. 1988.

[2] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering*. New York: CRC, 2000.

[3] J. M. Jou, P.-Y. Chen, and J.-M. Sun, "The gray prediction search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 843–848, Sep. 1999.

[4] H.-W. Cheng and L.-R. Dung, "EFBLA: A two phase matching algorithm for fast motion estimation," *Electron. Lett.*, vol. 40, no. 11, pp. 660–661, May 2004.

[5] C. J. Kuo, C. H. Yeh, and S. F. Odeh, "Polynomial search algorithm for motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 5, pp. 813–818, Aug. 2000.

[6] K. N. Nam, J.-S. Kim, R.-H. Park, and Y. S. Shim, "A fast hierarchical motion vector estimation algorithm using mean pyramid," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 4, pp. 344–351, Aug. 1995.

[7] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

[8] L. H. Chien and T. Aoki, "Robust motion estimation for video sequences based on phase-only correlation," in *Proc. IASTED Int. Conf. Signal and Image Proc.*, Aug. 2004, pp. 441–446.

[9] Y. M. Chou and H. M. Hang, "A new motion estimation method using frequency components," *J, Vis. Commun. Image Represent.*, vol. 8, no. 1, pp. 83–96, Mar. 1997.

[10] S. Erturk, "Digital image stabilization with sub-image phase correlation based global motion estimation," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1320–1325, Nov. 2003.

[11] H. Foroosh, J. Zeroubia, and M. Berthold, "Extension of phase correlation to sub-pixel registration," *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 188–200, Mar. 2002.

[12] L. Hill and T. Vlachos, "Motion measurement using shape adaptive phase correlation," *Electron. Lett.*, vol. 37, no. 25, pp. 1512–1513, Dec. 2001.

[13] C. D. Kughlin and D. C. Hines, "The phase correlation image alignment method," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Sep. 1975, pp. 163–165.

[14] M. Biswas and T. Nguyen, "A novel motion estimation algorithm using phase plane correlation for frame rate conversion," in *Proc. IEEE Conf. Signals, Syst., Comput.*, Nov. 2006, vol. 1, pp. 492–496.

[15] A. Molino, F. Vacca, and G. Masera, "Design and implementation of phase correlation based motion estimator," in *Proc. IEEE Int. SOC Conf.*, Sep. 2005, pp. 291–294.

[16] A. Netravali and J. Robbins, "Motion-compensated television coding: Part I," *Bell Syst. Tech. J.*, vol. 58, no. 3, pp. 631–670, 1979.

[17] J. D. Robbins and A. N. Netravali, "Recursive motion compensation: A review," in *Image Sequence Processing and Dynamic Scene Analysis*, T. S. Huang, Ed. Berlin, Germany: Springer-Verlag, 1983, pp. 73–103.

[18] G. Le Besnerais and F. Champagnat, "Dense optical flow by iterative local window registration," in *Proc. IEEE Int. Conf. Image Processing*, 2005, pp. 137–140.

[19] M. Ye, R. M. Haralick, and L. G. Shapiro, "Estimating piecewise—Smooth optical flow with global matching and graduated optimization," *IEEE Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1625–1630, Dec. 2003.

[20] M. Li, M. Biswas, S. Kumar, and T. Nguyen, "DCT- based phase correlation motion estimation," in *Proc. IEEE Int. Conf. Image Processing*, 2004, pp. 445–448.

[21] U.-V. Koc and K. J. R. Liu, "DCT-based motion estimation," *IEEE Trans. Image Process.*, vol. 7, no. 7, Jul. 1998.

[22] M. Song, A. Cai, and J.-A. Sun, "Motion estimation in DCT domain," in *Proc. ICCT*, 1996, vol. 2, pp. 670–674.

[23] M. Balci and H. Foroosh, "Inferring motion from the rank constraint of the phase matrix," in *Proc. ICASSP*, Mar. 2005, vol. 2, pp. 925–928.

[24] M. Balci and H. Foroosh, "Subpixel estimation of shifts directly in the fourier domain," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1965–1972, Jul. 2006.

[25] W. S. Hoge, "Subspace identification extension to the phase correlation method," *IEEE Trans. Med. Imag.*, vol. 22, no. 2, pp. 277–280, Feb. 2003.

[26] D. Salomon, *Data Compression: The Complete Reference*. New York: Springer, 2004.

[27] J. K. Taylor and C. Cihon, *Statistical Techniques for Data Analysis*. New York: CRC, 2004.

[28] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[29] D. L. Donoho, "Denoising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, Mar. 1995.

[30] A. Erturk and S. Erturk, "Two-bit transform for binary block motion estimation," *IEEE Trans. Circuits Syst Video Technol.*, vol. 15, no. 7, pp. 938–946, Jul. 2005.

[31] K. J. Boo and N. K. Bose, "A motion-compensated spatial-temporal filter for image sequences with signal-dependent noise," *IEEE Trans. Circuits Syst Video Technol.*, vol. 8, no. 3, pp. 287–298, Jun. 1998.