pixels (bytes) over the total number of quantised coefficients using 256 levels (1 byte). Table 1 compares a single-structure NN (SSNN) [3] and ACA in terms of peak-signal-to-noise ratio (PSNR). The SSNN is trained and tested on the same image to avoid dependence of results on training data. Although this is impractical for NNs due to high training time requirements, it is useful for comparison purposes. Nevertheless, ACA gives higher PSNR for all four tested images and for three different *CR*. The parallel-structure NN (PSNN) [5] results in PSNR = 33.8 dB (*CR* = 8:1) for 'Peppers' when trained with 'Lena', and 33.0 dB when trained with 'Baboon'. The PSNR for ACA is 35.1 dB (*CR* = 8:1). Moreover, PSNN's dependence on the training data is apparent. The number of operations for PSNN (excluding training) is 63×10^6 for encoding 512×512 images (*CR* = 8:1). Although ACA's coding includes training, the number of operations is variable and smaller than that of PSNN: 45.5×10^6 for 'Lena' and 53.2×10^6 for 'Baboon' (image sizes: 512×512 and *CR* = 8:1).

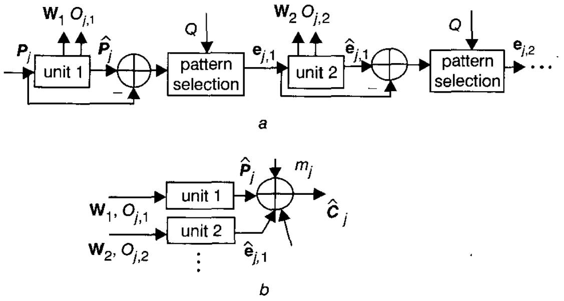

Fig. 2 *Detailed blocks*

a Weights/coefficients estimation block *b* Decoding block

Table 1: Comparison between single structure NN and proposed architecture in terms of PSNR (dB)

	Lena		Baboon		Peppers		Girl	
CR	SSNN	ACA	SSNN	ACA	SSNN	ACA	SSNN	ACA
16:1	29.2	31.8	21.6	21.8	28.8	31.7	28.9	30.3
8:1	31.9	35.8	22.9	23.4	31.9	35.1	31.8	34.1
4:1	35.5	38.9	25.1	25.9	34.9	37.4	35.0	38.2

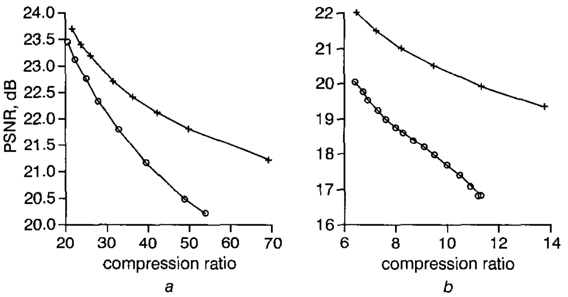

Fig. 3 *Comparison of proposed algorithm and JPEG for 'Baboon' and texture*

a For 'Baboon' *b* For texture
+ ACA -o- JPEG

Figs. 3*a* and *b* compare JPEG and ACA for the 'Baboon' and a textural image for different *CR*s. For these comparisons, ACA includes the lossless stage. The *CR* is defined as number of bytes in the original over number of bytes in the compressed image. The bytes in the compressed image consist of the arithmetic coded $O^q_{j,k}$ and dm_j, the quantisation step values Δ corresponding to each of the *K* units, and the *K* transform weights $\mathbf{W}^q_k$. It can be concluded that ACA provides significantly higher PSNR than JPEG especially for high *CR*s.

Conclusions: A novel adaptive cascade architecture (ACA) is proposed for image compression that produces higher PSNR than NN, and JPEG especially for large *CR*s. The superiority of ACA is more apparent for images 'rich' in information such as textures. For instance, Fig. 3*b* shows that the PSNR for ACA is around 2–3 dB higher than JPEG. The overall process of ACA exhibits similarities to JPEG. The major differences are the substitution of DCT with adaptive transforms, and the adaptive estimation of coefficient quantisation levels. Similar modifications can be applied to JPEG2000-like schemes with an expectation of additional improvement.

© IEE 2003 *6 July 2003*
Electronics Letters Online No: 20030811
DOI: 10.1049/el:20030811

D. Charalampidis (*Electrical Engineering Department, University of New Orleans, 2000 Lakeshore Drive, New Orleans, LA 70148, USA*)

References

1 COTTRELL, G.W., MUNRO, P., and ZIPSER, D.: 'Image compression by backpropagation: an example of extensional programming' *in* SHARKEY, N.E. (Ed.): 'Models of cognition: a review of cognition science' (Norwood, NJ, 1989)

2 CARRATO, S., and MARSI, S.: 'Parallel structure based on neural networks for image compression', *Electron. Lett.*, 1992, **28**, (12), pp. 1152–1153

3 BENBENISTI, Y., *et al.*: 'Fixed bit-rate image compression using a parallel-structure multilayer neural network', *IEEE Trans. Neural Netw.*, 1999, **10**, (5), pp. 1166–1172

4 WALLACE, G.K.: 'The JPEG still picture compression standard', *IEEE Trans. Consum. Electron.*, 1992, **38**, (1)

5 CHRISTOPOULOS, C., SKODRAS, A., and EBRAHIMI, T.: 'The JPEG2000 still image compression coding system: an overview', *IEEE Trans. Consum. Electron.*, 2000, **46**, (4)

Half-pixel filter of MC-DCT compressed video

G. Cao, J. Li and Y. Zhang

A novel half-pixel filter is proposed to extract one block with half-pixel precision motion vector in the DCT domain. The proposed filter reduces the computational complexity by integrating the interpolation and translation into a single step, while improves the video quality compared with the existing half-pixel filter.

Introduction: How to extract one block not aligned 8×8 blocks from the motion compensated discrete cosine transform (MC-DCT) format bit streams is one of the key steps to implement the DCT-domain transcoder (DDT) [1]. The problem of DCT-domain inverse motion compensation (IMC) is illustrated in Fig. 1. Let $MV(x, y)$ denote the motion vector (MV) carried by the current block *B*. In general, the reference block $\hat{B}$ determined by $MV(x, y)$ may not be aligned to the original 8×8 blocks in the reference frame, and may intersect with four neighbouring blocks B_i, $i = 1, 2, 3, 4$. B_{13} is the corresponding subblock of $\hat{B}$ intersected with B_1. The related rows and columns in B_{13} are denoted by *m* and *n*, and determined by $MV(x, y)$.

Following Chang [2], $\hat{B}$ can be expressed as appropriate window and shift operations. However, Chang's method is cumbersome to extract one block with half-pixel in horizontal and vertical directions. Four blocks need to be extracted, then averaged. Assuncao [3] proposed a half-pixel filter to interpolate the DCT coefficients. The results are then extracted using Chang's method. The computational complexity is reduced compared with Chang's method. However, Assuncao's method not only separates interpolation from translation, but also introduces some distortion in those blocks located on the macroblock's right and bottom boundaries.

In this Letter we propose a novel half-pixel filter to extract one block not aligned 8×8 blocks with half-pixel precision motion vector in the DCT domain. The proposed filter integrates the filtering and translation into a single step, and significantly reduces the image distortion and drift errors. Compared with Assuncao's method, our method improves 0.37 dB in video quality.

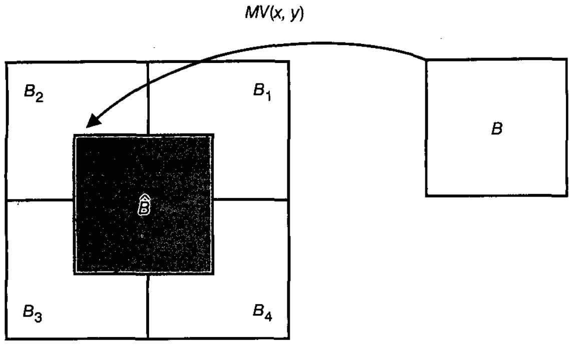

Fig. 1 *Inverse motion compensation*
$\hat{B}$: reference block
B: current block
B_i, $i = 1, 2, 3, 4$: four neighbouring blocks
$MV(x, y)$: MV of B

Novel half-pixel filter of MC-DCT: If $MV(x, y)$ only has integer-pixel precision, $\hat{B}$ can be expressed as $\hat{B}_{I,m,n}$, in which I denotes integer-pixel precision, through a superposition of appropriate window and shift operations determined by m and n as Chang's translation

$$\hat{B}_{I,m,n} = \sum_{i=1}^{4} H_{i1} B_i H_{i2} \qquad (1)$$

where H_{ij}, $i = 1, 2, 3, 4, j = 1, 2$ are appropriate sparse 8×8 matrices of zeros and ones. If $MV(x, y)$ has a half-pixel precision in both vertical and horizontal directions, the IMC interpolates a pixel value from four neighbouring pixels, i.e.

$$\hat{B} = \frac{1}{4} \sum_{u=0}^{1} \sum_{v=0}^{1} \hat{B}_{I,m+u,n+v} \qquad (2)$$

The above interpolation can be integrated with translation as a single step. Let $E^s = \{1, 2, \ldots, s\}$, where s is an integer number. We first define two kinds of sparse $s \times s$ matrices of zeros $R_s = (R_s(i, j))_{i, j \in E^7}$ and $L_s = (L_s(i, j))_{i, j \in E^7}$, such that

$$R_s(i,j) = \begin{cases} 0.5, & \text{if } j = i, j \in E^7 \\ 0.5, & \text{if } j = i + 1, \;\; i,j \in E^7 \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

$$L_s(i,j) = \begin{cases} 0.5, & \text{if } j = i, j \in E^7 \\ 0.5, & \text{if } j = i - 1, \;\; i,j \in E^7 \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

A set of novel half-pixel filters in the pixel domain is designed as given by

$$\hat{B} = \sum_{i=1}^{4} P_{i1,m} B_i P_{i2,n} \qquad (5)$$

where $P_{i1,m}$ and $P_{i2,n}$, $i = 1, 2, 3, 4$ perform interpolation, windowed and shifted operations, and can be defined as follows:

$$P_{11,m} = P_{21,m} = \begin{bmatrix} 0 & R_m \\ 0 & 0 \end{bmatrix} \qquad (6)$$

$$P_{31,m} = P_{41,m} = \begin{bmatrix} 0 & 0 \\ L_{9-m} & 0 \end{bmatrix} \qquad (7)$$

$$P_{12,n} = P_{42,n} = \begin{bmatrix} 0 & R_n \\ 0 & 0 \end{bmatrix} \qquad (8)$$

$$P_{22,n} = P_{32,n} = \begin{bmatrix} 0 & 0 \\ L_{9-n} & 0 \end{bmatrix} \qquad (9)$$

where 0 denotes a zero matrix with appropriate dimension. Because of the distributive property of matrix multiplication with respect to the DCT [2], the coefficients of $\hat{B}$ in the DCT domain can be obtained using the DCT matrices $DCT\,(P_{i1,m})$ and $DCT\,(P_{i2,n})$ as follows:

$$DCT(\hat{B}) = \sum_{i=1}^{4} DCT(P_{i1,m}) DCT(B_i) DCT(P_{i2,n}) \qquad (10)$$

where $DCT(\cdot)$ represents the DCT operation. $DCT(P_{i1,m})$ and $DCT(P_{i2,n})$ are the new half-pixel filters. Because L_s is the transpose of R_s, i.e. $L_s = R_s^T$, only eight $DCT(P_{11,m})$, $1 \leq m \leq 8$ matrices should be pre-computed and stored for MV with half-pixel precision. After the filtering, no further translation operation is needed.

Obviously, a 9×9 block is needed to interpolate an 8×8 block, and the half-pixel filters should also consider the adjacent macroblock's block. However, Assuncao proposed a half-pixel filter according to the block's position within the macroblock (MB) (top left: C_1, top right: C_2, bottom left: C_3, bottom right: C_4) as given by

$$C_i^h = \begin{cases} C_i P_{11,8} + C_{i+1} P_{11,1}, & i = 1, 3 \\ C_i F_3^h, & i = 2, 4 \end{cases} \qquad (11)$$

$$C_i^v = \begin{cases} P_{11,8}^T C_i + P_{11,1}^T C_{i+1}, & i = 1, 2 \\ F_3^v C_i, & i = 3, 4 \end{cases} \qquad (12)$$

where the definitions of F_3^h and F_3^v are the same as that of $P_{11,8}$ and $P_{11,8}^T$, respectively, except that the right-most element at the bottom equals 1 rather than 0.5, which results in some distortion in those blocks located on the right and bottom boundaries of the MBs. Furthermore, Chang's translation is needed after the interpolation for the Assuncao's method.

Table 1: Matrix operations

Method	Multiplication	Addition
Chang	96	60
Assuncao	36	16
Novel	24	12

The matrix operations to calculate four MC-DCT blocks in a luminance MB with half-pixel precision MV are shown in Table 1. The proposed novel filter just needs 24 multiplications and 12 additions, the same as Chang's method to extract a luminance MB with integer pixel MV. A significant reduction in computational complexity can be achieved.

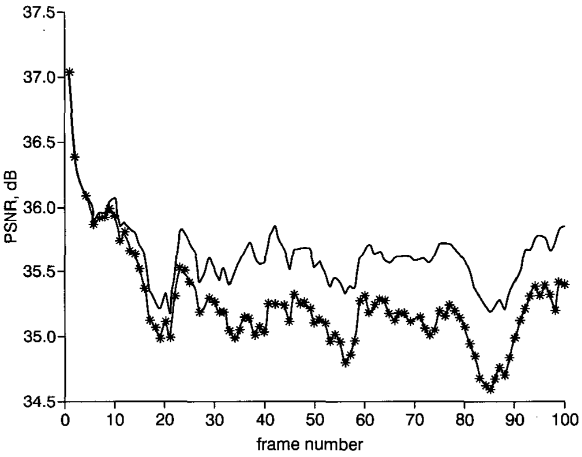

Fig. 2 *PSNR using different methods in DDT*
—— novel
—✱—Assuncao

Simulation results: Since the drift errors introduced in DDT have to be evaluated, we have encoded a high quality bit stream using the Foreman sequence with a fixed quantiser step of size 6. This bit stream was then transcoded using two different half-pixel filters in DDT with a fixed quantiser step of size 10. A percentage of 47.35 has half-pixel precision MV for Luminance blocks. As Fig. 2 shows, compared with Assuncao's method, our method improves 0.37 dB in video quality. We also implemented Chang's method to extract the block with half-pixel precision MV. The proposed filters achieved the same quality as Chang's method.

Conclusion: We have proposed a novel half-pixel filter to extract one block not aligned 8 × 8 blocks with half-pixel precision motion vector in the DCT domain which integrates interpolation and translation into a single step. The proposed filter significantly reduced the image distortion and drift errors compared with the Assuncao's half-pixel filters, and achieved the same video quality as Chang's method.

Acknowledgment: This research was supported by the Hi-tech Research and Development Program of China under grant 2001AA114060.

© IEE 2003 *17 June 2003*
Electronics Letters Online No: 20030817
DOI: 10.1049/el:20030817

Gang Cao, Jintao Li and Yongdong Zhang (*Digital Technology Laboratory, Institute of Computing Technology, Chinese Academy of Sciences, P.O. Box 2704, Beijing 100080, People's Republic of China*)

E-mail: caog@ict.ac.cn

References

1 ASSUNCAO, P., and GHANBARI, J.: 'Post-processing of MPEG2 coded video for transmission at lower bit-rates'. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP'96, Atlanta, GA, USA, May 1996, Vol. 4, pp. 1999–2002

2 CHANG, S.-F., and MESSERSCHMITT, D.G.: 'Manipulation and compositing of MC-DCT compressed video', *IEEE J. Sel. Areas Commun.*, 1995, **13**, pp. 1–11

3 ASSUNCAO, P., and GHANBARI, J.: 'A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams', *IEEE Trans. Circuits Syst. Video Technol.*, 1998, **8**, pp. 953–967

Translation, rotation and scale stabilisation of image sequences

S. Ertürk

Translation, rotation and scale stabilisation of image sequences is presented with phase correlation based estimation of motion parameters. Phase correlation and log-polar magnitude spectra representation is utilised to obtain translation, rotation and scale parameters. Stabilised frame positions are obtained through Kalman filtering of absolute parameters, and image frames are processed accordingly so as to result in a sequence that displays smooth global movements only.

Introduction: Image sequence stabilisation (ISS) aims to remove undesired translation, rotation or scale variations so as to produce a compensated image sequence that displays intentional global camera movements only. Undesired camera motions causing disturbing fluctuations in the image are commonly encountered in image sequences acquired through hand-held cameras due to operator instability and in cases where the camera is located on a moving framework such as a vehicle or even a mobile phone.

Translational jitter is the most commonly encountered as well as the most disturbing case, rotational fluctuations are usually second in importance, while scale fluctuations are encountered rather infrequently. To maintain intentional motion effects and avoid picture loss in sequences containing significant camera movements such as pan, the jitter element of the assessed global motion should be resolved, instead of removing detected global movements entirely. Desired motions can commonly be resolved on the basis of smoothness, as intentional motions are commonly smooth while undesired jitter is rather random in nature and of higher frequency. Translational jitter stabilisation through Kalman filtering and fuzzy adaptive Kalman filtering of absolute frame positions has been demonstrated in [1] and [2], respectively. In this Letter Kalman filtering based stabilisation is extended to take rotation and scale variations into account. Phase correlation based motion estimation with log-polar magnitude spectra representation is utilised to resolve translation, rotation and scale parameters.

Translation, rotation and scale stabilisation: Utilisation of Fourier transform properties for translation, rotation and scale estimation has been presented in [3]. If an image $f_2(x, y)$ is a spatially shifted version of the image $f_1(x, y)$ with a contrast difference accumulated by α, then

$$f_2(x, y) = \alpha \cdot f_1(x - x_0, y - y_0) \tag{1}$$

The cross-power spectrum is defined as

$$\frac{F_1(u, v) \cdot F_2^*(u, v)}{|F_1(u, v) \cdot F_2^*(u, v)|} = e^{j2\pi(ux_0+vy_0)} \tag{2}$$

where F_1 and F_2 are the corresponding Fourier transforms. The phase correlation surface is obtained as the inverse discrete Fourier transform of the cross-power spectrum:

$$P(x, y) = F^{-1}(e^{j2\pi(ux_0+vy_0)}) = \delta(x + x_0, y + y_0) \tag{3}$$

and the phase correlation surface will approximately be zero everywhere except at the displacement corresponding to the shift between the images. If $f_2(x, y)$ is a translated and rotated replica of $f_1(x, y)$, the translation will affect only the phase component of the Fourier spectra, while the magnitude spectra of the images will be rotated versions of each other, with a rotation angle equal to the image rotation. Hence if represented in polar coordinates, θ_0 denoting the rotation between the two images, the magnitude spectra are related by

$$M_2(\rho, \theta) = M_1(\rho, \theta - \theta_0) \tag{4}$$

Therefore, when using polar representation of the magnitude spectra the rotation amount will result in a corresponding shift, which can be obtained using phase correlation. If $f_2(x, y)$ is a scaled version of $f_1(x, y)$ with scale factor a, the Fourier transforms will be related by

$$F_2(u, v) = \frac{1}{a^2} F_1\left(\frac{u}{a}, \frac{v}{a}\right) \tag{5}$$

By converting the axes to logarithmic scale, scaling is converted to a translational movement in the form of

$$F_2(\log u, \log v) = \frac{1}{a^2} F_1(\log u - \log a, \log v - \log a) \tag{6}$$

so as to enable estimation of the scale factor through phase correlation. By combining logarithmic scaling with polar representation, i.e. employing log-polar representation, it is possible to obtain rotation and scale parameters (θ_0 and a) in a single step:

$$M_2(\log \rho, \theta) = M_1(\log \rho - \log a, \theta - \theta_0) \tag{7}$$

Interframe translation, rotation and scale parameters are estimated through phase correlation making use of the log-polar representation of the magnitude spectra, and evaluated motion parameters are accumulated to obtain absolute translation against frame number, absolute rotation against frame number and absolute scale factor against frame number signals. Kalman filtering is utilised to smooth the absolute motion signals, removing high frequency jitter, while retaining low frequency movements. The differences between the Kalman filtered outputs and the original signals are used as correction parameters for stabilisation of the image sequence. Image frames are shifted, rotated and scaled by the corresponding correction parameters so as to obtain a stabilised sequence. Normally more intensive stabilisation is desired for the rotational component; hence the process noise variance of the Kalman filter handling rotation is kept at a comparatively lower value. It is possible to utilise fuzzy adaptive tuning of the process noise variances of the Kalman filters as described in [2], in order to adapt the filter characteristics to changing motion dynamics.

Experimental results: Fig. 1 shows two sample frames of an image sequence containing deliberately introduced intensive translational and rotational jitter. Interframe motion parameters are evaluated using phase correlation and log-polar representation of the magnitude spectra of the images. A 256 × 256 log-polar representation is utilised, resulting in a rotation angle resolution of $360°/256 \simeq 1.4°$. It is possible to achieve finer resolutions, for instance a log-polar representation of size 512 × 512 will give an angle resolution of 0.7°, at the expense of increased computational load. Note that the Fourier transform properties based rotation estimation approach evaluates the best rotation with respect to the centre of the image frames, while in practice the rotation origin is usually not exactly located at