

Fig. 1c and/or Fig. 1d, we propose to equalise the gradients. Thus, $(a - d) = (b - \hat{x}) \Rightarrow \hat{x} = b + d - a$. This gradient equalisation could be justified due to two reasons: (i) it is highly unlikely for these gradients to have opposite signs between two consecutive scan-lines, (ii) it is highly likely that the pixel values change in an equivalent way between two scan-lines when moving to the right, from a diagonal edge. An interesting observation here is that the above predictor is well suited to cater for the needs of diagonal edges with width of a single pixel (Fig. 1d), unlike the predictors proposed in [2, 3]. In addition to the above, the computational cost of the proposed predictor is low since only one addition and one subtraction is necessary to calculate the predicted value, with no necessity for any divisions [3]. The predictor is also able to equalise gradients in a direction parallel to the detected diagonal edge, since $(a - d) = (b - \hat{x}) \Rightarrow (a - b) = (d - \hat{x})$.

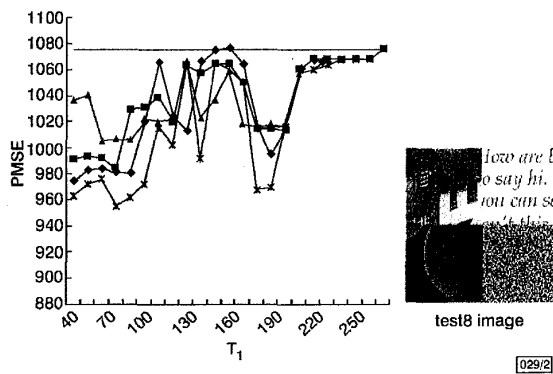


Fig. 2 Performance comparison for Test8 image

— JPEG-LS
 ▲ proposed $T_2 = 5$
 ■ proposed $T_2 = 10$
 ◆ proposed $T_2 = 15$
 * proposed $T_2 = 22$

Table 1: Performance comparison at $T_1 = 70$ [JPEG-LS: PMSE = 1075.535, CR = 1.915:1]

T_2	PMSE, proposed scheme at CR = 1.915:1	Improvement in PMSE	Number of detected diagonal pixels	Detected as diagonal pixels
		%		%
5	1006.125	6.45	850	1.30
10	984.029	8.51	1147	1.75
15	981.032	8.79	1488	2.27
22	954.793	11.23	1874	2.86

Experimental results and analysis: To check the validity of the proposed scheme, we performed experiments on a set of standard test image pairs. In Fig. 2 and Table 1, we provide a summarised performance comparison of the two methods, using the test image, Test8 (size 256×256) (Fig. 2). Test8 is a standard test image that consists of a wide selection of image properties and thus enables us to provide a detailed and fair comparison between the two methods. The performance of the proposed scheme is compared to that of JPEG-LS in terms of the mean squared error of the predicted image (PMSE), rather than the MSE of the reconstructed image, which is zero, due to the lossless image reconstruction. It is defined as

$$PMSE = \left[\sum_{i=1}^N (I_{original_i} - I_{predicted_i}) \right] / N$$

where N is the total number of pixels in the image, and $I_{original}$, $I_{predicted}$ are, respectively, the intensity values of original and predicted pixels at a given location. In our present work we claim that the proposed technique outperforms the JPEG-LS in terms of better prediction, leading to improvements of the lossless compressibility of the image. Within our current research context, we have not modified the statistical modelling (context modelling) part of the JPEG-LS code, which is the part instrumental in the selection of entropy coding length of prediction errors. Thus, smaller predictive errors obtained with the proposed method may

not always produce higher compression efficiency. As a result, equal compression ratios (1.915:1) are obtained for the two methods in our experiments.

To evaluate the performance of the proposed technique, we set the threshold $T_2 = 5, 10, 15$ and 22 . For each setting of T_2 we vary the threshold T_1 between 40 and 255 . Fig. 2 and Table 1 show the results obtained. The graphs in Fig. 2 show that the performance of the proposed algorithm outperforms the original JPEG-LS algorithm, within the entire range of the threshold T_1 , particularly performing best when $T_1 = 70$ for all tested values of T_2 . The graphs also show clear local minima around $T_1 = 190$. Closer investigation revealed that these minima are due to the right half of the test image having highly contrasting diagonal edges, which were the only ones detected and accurately predicted when $T_1 > 150$. For further analysis we summarise the performance of the proposed coder in Table 1. It is seen that the performance is best when $T_2 = 22$. At this setting an improvement of 11.23% is obtained for the PMSE and 1874 pixels (2.86% of total) are categorised as being on edges. When $T_2 < 22$, a lesser amount of pixels are categorised as being on diagonals, thus decreasing the performance of the proposed coder. However, when $T_2 > 22$ the accuracy of the definition of diagonals reduces, decreasing the relative performance of the coder.

Further experiments on a large set of test images revealed that the proposed coder performs best in images where a high percentage of diagonal edges are present. Within the Test8 image, the proposed algorithm performed best in the top left hand corner and bottom right hand corner sub-images. It was also observed that for a given image the selection of threshold T_1 should be based on the contrast (pixel value gradient) present across the diagonal edges ($T_1 = 60$, typical) and that the threshold T_2 should be based on the smoothness of pixel value variation expected to the right of a diagonal edge boundary ($T_2 = 8$, typical).

Conclusions: We have proposed an improvement to the JPEG-LS standard by introducing diagonal edge detection and a modification to its prediction scheme. It has been shown that the proposed scheme outperforms JPEG-LS by margins of up to 11.23% in terms of the mean squared error of the predicted image. The proposed method addresses some special issues that JPEG-LS fails to address and thus provides an attractive option for its performance enhancement.

© IEE 2001

31 July 2001

Electronics Letters Online No: 20010909
 DOI: 10.1049/el:20010909

E.A. Edirisinghe and S. Bedi (Department of Computer Science, Loughborough University, LE11 3TU, United Kingdom)

References

- [ISO/IEC97] ISO/IEC JTC 1/SC 29/WG1 FCD 14495 public draft, 'JPEG-LS: lossless and near-lossless coding of continuous tone still images', 16 July 1997
- WEINBERGER, M.J., SEROUSSI, G., and SAPIRO, G.: 'LOCO-I: a low complexity, context-based, lossless image compression algorithm'. Proc. IEEE Data Compression Conf., Snowbird, Utah, April 1996
- JIANG, J., GUO, B., and YANG, S.Y.: 'Revisit to JPEG-LS prediction scheme', *IEE Proc., Vis. Image Signal Process.*, 2000, **147**, (6)

Low power embedded extension algorithm for lifting-based discrete wavelet transform in JPEG2000

K.C.B. Tan and T. Arslan

A novel algorithmic technique for reducing power and area consumption of the discrete wavelet transform (DWT) on CMOS-based DSP architectures is presented. The technique reduces power by combining the data-extension procedure into the lifting-based DWT core. It is demonstrated that the new algorithm can be used to obtain more than 50% reduction in the amount of memory required, leading to significant reduction in area and power.

Introduction: The discrete wavelet transform (DWT) is increasingly being used for image coding. This is because it has features such as progressive image transmission by quality/resolution, and ease of compressed image manipulation. These features have also lead to significant interest in efficient algorithms for hardware realisation of the DWT. The conventional convolution-based DWT is computationally intensive and both area and power hungry. Some of these drawbacks were overcome by using the lifting based scheme for the DWT [1], which has been selected in the recently released JPEG2000 standard. Such developments aroused considerable interest in the implementation [2] of this algorithm. With the increase in demand for video telephony, embedded low power video becomes an essential feature. So far, only a few researchers have tackled the low power implementation of the DWT, through generic algorithmic techniques [3].

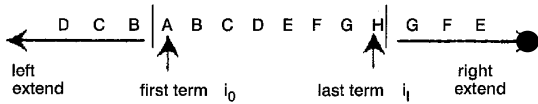


Fig. 1 Symmetrical data extension

In this Letter we present a novel algorithmic technique for reducing the power dissipation and memory requirements for data-extension prior to one-dimensional DWT. As DWT is implemented as a finite impulse response (FIR) filter, data-extension is important to ensure that the FIR filter, at the edge/boundary of an image, has sufficient data to produce a single output. Power reduction is achieved by combining and embedding data-extension into the main DWT algorithm in both the start and the end of the transformation process. This reduction is achieved as a result of the following: (i) reduction in the amount of storage memory since explicit extensions are not required, (ii) reduction in the number of read/write accesses since extension is only performed when the main data term is read or written, (iii) reduction in the number of arithmetic operations (see later). The data-extension used here is symmetrical (see Fig. 1) as specified by the JPEG2000 standard. The JPEG2000's 5/3 DWT (with lifting) is used to demonstrate this technique.

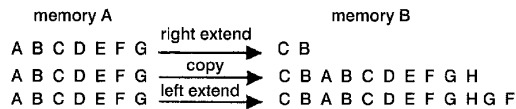


Fig. 2 Conventional straightforward data-extension

Algorithm: Conventional straightforward data-extension involves extending the data to the right, copying all the original data and lastly performing a left extension on the data (see Fig. 2). The number of data terms to extend will depend on whether the data set starts or ends with an even or odd number indexed term. If the data starts or ends with an even number indexed term, then two data terms are extended to the left and the right of the data set (see Figs. 3 and 4). If the data starts or ends with an odd number indexed term, then a single data extension is added to both the left and the right of the data set (see Figs. 3 and 4). Eqns. 1 and 2 describe the implementation of the 5/3 lifted forward DWT. Eqn. 1 is used for the computation of the odd coefficients, whereas eqn. 2 is used for the computation of the even coefficients. The computations of the odd coefficients have to be performed first, followed by the computations of the even coefficient.

$$Y(2n + 1) = X(2n + 1) - \left[\frac{X(2n) + X(2n + 2)}{2} \right] \quad (1)$$

$$Y(2n) = X(2n) + \left[\frac{Y(2n - 1) + Y(2n + 1) + 2}{4} \right] \quad (2)$$

Figs. 3 and 4 show the relationship between the output $Y(n)$ with its extended input $X(n)$. Our embedded extension algorithm splits the homogenous conventional computation process into three parts. The first part we term the Start Process, the second part the Normal Process (which is the same as the conventional computation), and the third part the End Process. The data-exten-

sions are only embedded during the Start Process and the End Process. The Start Process embeds the left extension and the End Process the right extension. There are two equations for computing the coefficients in the Start Process. The first, eqn. 3, is the embedded extension equation for the computation of the first odd coefficient when the input data starts with an odd number indexed term. The second, eqn. 4, is used for the computation of the first even coefficient when the input data starts with an even number indexed term. Note that i_0 in $Y(i_0)$ indexes the first term of the data.

$$Y(i_0) = X(i_0) - X(i_0 + 1) \quad (3)$$

$$Y(i_0) = X(i_0) + \left[\frac{Y(i_0 + 1) + 1}{2} \right] \quad (4)$$

As $X(i_0 - 1)$ is equal to $X(i_0 + 1)$ (see Fig. 3), eqn. 1 will reduce to eqn. 3. Also, using the fact that $Y(i_0 - 1)$ is equal to $Y(i_0 + 1)$ (see Fig. 4), eqn. 2 reduces to eqn. 4. Similarly in the End Process, eqns. 1 and 2 reduce to eqns. 5 and 6, respectively. Eqns. 5 and 6 are the embedded extension equation for the computation of the last odd coefficient, when the input data ends with an odd number indexed term (see Fig. 4), and the last even coefficient when the input data ends with an even number indexed term (see Fig. 3), respectively. The i_l in $Y(i_l)$ indexes the last term of the data.

$$Y(i_l) = X(i_l) - X(i_l - 1) \quad (5)$$

$$Y(i_l) = X(i_l) + \left[\frac{Y(i_l - 1) + 1}{2} \right] \quad (6)$$

As can be deduced from the new set of equations (eqns. 3 – 6), using this technique, computation steps are reduced as the entire computation of the odd coefficient is omitted when there is either an even number indexed start or end terms. In the same effect, a read access and two arithmetic operations are omitted in computing the first/last odd coefficient when there is either an odd number indexed start or end term. This technique can be extended to be used in the inverse DWT with the 5/3 filter and/or in both the forward and inverse DWT with a 9/7 filter.

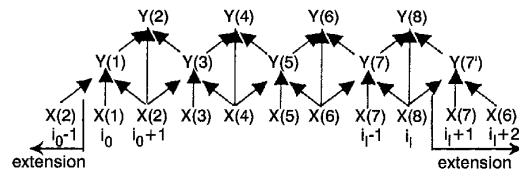


Fig. 3 Odd start term and even end term data

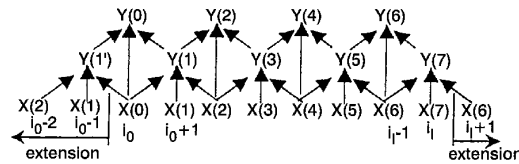


Fig. 4 Even start term and odd end term data

Analysis: Three techniques were evaluated in a one-dimensional access with a 5/3 DWT filter. These are: the conventional straightforward extension, extension by scheduled access and our new embedded extension algorithm. The extension by scheduled access is a scheme where the data (read) access controller is scheduled to read the original data in a manner that the data are extended systematically, so that the DWT processor can calculate all the coefficients in a homogenous process. Analysis of eqns. 3 – 6 clearly points to both memory and power savings as could be seen from the last column of Table 1. Note that 'N' in Table 1 represents the number of data in a row or column and 'Number of operations' refers to read/write access operations. Three cases of extension are considered in the calculation of computational and memory savings. In the first case (Case A), we consider the extension with input data that starts with an odd number indexed term and ends with an odd number indexed term. The second case (Case B) considers input data that either starts with an even number indexed term and ends with an odd number indexed term or starts with an odd number indexed term and ends with an even

number indexed term. The third case (Case C) considers extensions with input data that starts with an even number indexed term and ends with an even number indexed term. The output of our embedded algorithm is verified and compared to that of the conventional straightforward extension using MATLAB. In the conventional straightforward extension, each data term will incur a two-operation overhead with a read access and write access. In extension by scheduled access, each data in the extension only incur one operation. Our new embedded extension algorithm does not incur any extension overhead. Furthermore, our algorithm saves up to an additional eight arithmetic operations (for data that starts and ends with an even number indexed term) per row/column, as several computations are eliminated as shown above. Although the savings with our embedded extension algorithm in one-dimensional process are only a few operations more than those of extension by scheduled access, savings will be considerable when it comes to two-dimensional processing of a whole frame. As Table 1 shows, significant savings in memory and number of operations are achieved. Such major savings lead to significant savings in both area and power. The algorithm has a slight overhead in terms of a small increase in the area of the control logic. This is mainly due to incorporating Start/End states into the controller. Although there is an increase in control area, the overall reduction in power consumption will not be affected, since the increase is insignificant compared to the savings achieved due to the reduction in memory size and the number of arithmetic operations. Furthermore, the additional states can be effectively manipulated by appropriate power-down mechanisms.

Table 1: Memory and power (as number of operation) saving per row or column of processing

Technique	Case	Memory used	Memory reduction	Number of operations (for extension access only)	Reduction in number of operations
Conventional (straightforward) extension	A	$N + 2$	0	$2N + 4$	0
	B	$N + 3$	0	$2N + 6$	0
	C	$N + 4$	0	$2N + 8$	0
Extension by scheduled accessing	A	-	$N + 2$	2	$2N + 2$
	B	-	$N + 3$	3	$2N + 3$
	C	-	$N + 4$	4	$2N + 4$
Embedded extension algorithm (with 5/3 DWT)	A	-	$N + 2$	-2	$2N + 6$
	B	-	$N + 3$	-5	$2N + 11$
	C	-	$N + 4$	-8	$2N + 16$

Conclusions: An algorithm for the low power data extension of the DWT on CMOS-based DSPs has been presented. The algorithm reduces power by combining the data-extension into the main lifted-based DWT such that the computation steps are reduced and hence resulting in a reduction in the switched capacitance sector of the dynamic power. We have shown that the algorithm can be used to obtain more than 50% memory and power reduction.

Acknowledgments: This work is supported by the Engineering and Physical Sciences Research Council under Grant No. GR/N08322.

© IEE 2001

17 July 2001

Electronics Letters Online No: 20010915
DOI: 10.1049/el:20010915

K.C.B. Tan and T. Arslan (Department of Electrical Engineering, The University of Edinburgh, The King's Buildings, Mayfield Road, Edinburgh EH9 3JL, Scotland, United Kingdom)

E-mail: benny.tan@ee.ed.ac.uk

References

- SWELDENS, W.: 'The lifting scheme: a new philosophy in biorthogonal wavelet constructions', *Proc. SPIE*, 1995, **2569**, pp. 68-79
- ANDRA, K., CHAKRABARTI, C., and ACHARYA, T.: 'A VLSI architecture for lifting-based wavelet transform'. IEEE Workshop on Signal Processing Systems, SiPs 2000, Oct. 2000, pp. 70-79
- MASUPE, S., and ARSLAN, T.: 'Low power DCT implementation approach for CMOS-based DSP processors', *Electron. Lett.*, 1998, **34**, (25), pp. 2392-2394

Polar co-ordinates shape adaptive discrete transform-based watermarking scheme for arbitrarily-shaped object

Yun-Ho Ko and Seong-Dae Kim

A new watermarking scheme for arbitrarily-shaped image is proposed. Using co-ordinates conversion to log-polar and polar co-ordinates shape adaptive discrete transform (PSADT), the proposed watermarking method can be applied to any arbitrarily-shaped image such as MPEG-4 VOP and is much more robust to geometrical attacks than conventional ones.

Introduction: The new standard, MPEG-4, offers an object-based representation of image, where individual video objects (VOs) are coded into separate bit streams. Compared to the conventional frame-based compression techniques, the object-based representation enables MPEG-4 to cover a broad range of emerging multimedia applications. Digital image watermarking techniques have also been proposed for copyright protection of digital image. However, conventional watermarking schemes are most vulnerable to geometrical attacks such as rotation and scaling since they may completely change the alignment of the watermark with respect to pixels [1, 2].

In this Letter, we propose a new watermarking scheme for an arbitrarily-shaped object, which is designed to be unaffected by any geometrical attack. The proposed watermarking scheme operates in the frequency domain based on an approach similar to the method of Barni *et al.* [2]. To make our scheme robust to geometrical attacks, we convert Cartesian co-ordinates to log-polar co-ordinates and then apply a new shape adaptive transform to handle arbitrarily-shaped objects.

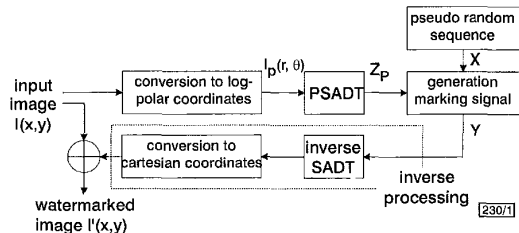


Fig. 1 Proposed watermarking scheme

Watermark embedding: The proposed watermarking scheme is shown in Fig. 1. We first convert co-ordinates of the original image $I(x, y)$ to log-polar co-ordinates $I_p(r, \theta)$ of $N_r \times N_\theta$ elements with the origin of log-polar co-ordinates corresponding to the centre of gravity (x_{cen}, y_{cen}) of the arbitrarily-shaped mask (\mathcal{R}) . Let Δ_r and Δ_θ be the discrete sampling step size in the radial and angular direction, respectively. Consider a point (x, y) in the original image $I(x, y)$, the mapping is defined as

$$\begin{aligned} x - x_{cen} &= \exp(r\Delta_r) \cos(\theta\Delta_\theta) \\ y - y_{cen} &= \exp(r\Delta_r) \sin(\theta\Delta_\theta) \end{aligned} \quad (1)$$

where

$$\begin{aligned} r &= \{0, 1, \dots, N_r - 1\} & \theta &= \{0, 1, \dots, N_\theta - 1\} \\ \Delta_r &= \frac{\ln\left(\max_{x,y \in \mathcal{R}} \sqrt{(x-x_{cen})^2 + (y-y_{cen})^2}\right)}{N_r - 1} & \Delta_\theta &= \frac{2\pi}{N_\theta} \end{aligned}$$

Using this co-ordinates conversion, shift and scaling of the original image have no effect on the log-polar co-ordinates image $I_p(r, \theta)$. Also, rotation in the Cartesian co-ordinates will be converted to circular shift in the polar co-ordinates.

In the polar co-ordinates shape adaptive discrete transform (PSADT) stage, only the active pels within the polar co-ordinates image are transformed in a similar way to SA-DCT in MPEG-4 [3]. This will result in a number of coefficients (Z_p) which is identical to the number of active pels, so that the watermark signal which will be inserted in the next stages does not appear outside of the mask (\mathcal{R}) . Fig. 2 illustrates the successive steps involved for performing a PSADT. To perform a vertical transform, the columns are shifted and aligned to the upper border of the block