designs [1, 7]. The resulting delay, energy and *EDP* per bit and the adopted transistor sizes (normalised to the minimum feature size L_{min}) are given in Table 1. The corresponding point in the energy-delay space is also highlighted in Fig. 4. Interestingly, from inspection of Table 1, the mixed D-FA/TG-FA circuit exhibits a 33% speed improvement with respect to the traditional D-FA. The speed improvement is achieved at the cost of a slightly lower energy increase, considering that *EDP* of the mixed D-FA/TG-FA is slightly lower than that of the D-FA. This means that the mixed D-FA/TG-FA has also a slightly better energy efficiency compared to the D-FA. Finally, the area occupied by the D-FA and TG-FA was found to be roughly the same (3.9×3.5 μm), thus the mixed D-FA/TG-FA and the traditional D-FA are essentially equivalent in terms of area.

Conclusions: The mixed D-FA/TG-FA approach to fast carry computation has been presented and evaluated. This approach is shown to be capable of a remarkably high-speed performance: for example, in a minimum *EDP* design, it is about 1.15 times the inverter delay with a fan-out of 4 (which in the adopted technology is 22.5 ps), which is very high compared to other topologies [3, 6]. Moreover, the proposed approach exhibits a greater than 30% speed advantage over the Domino topology with no degradation in the energy efficiency and area, thereby confirming that the approach is well suited and feasible for very high-speed applications.

15 March 2007
Electronics Letters online no: 20070752
doi: 10.1049/el:20070752

M. Alioto (*DII – University of Siena, Siena, Italy*)

E-mail: malioto@dii.unisi.it

G. Palumbo (*DIEES – University of Catania, Catania, Italy*)

References

1 Chandrakasan, A., Bowhill, W., and Fox, F.: 'Design of high-performance microprocessor circuits' (IEEE Press, 2001)
2 Callaway, T., and Swartzlander, E.: 'Low power arithmetic components', *in* Rabaey, J., and Pedram, M. (Eds.): 'Low power design methodologies' (Kluwer Academic Publishers, 1996)
3 Alioto, M., and Palumbo, G.: 'Analysis and comparison on full adder block in sub-micron technology', *IEEE Trans. Very Large Scale Integr. (VLSI) Systems*, 2002, **10**, (6), pp. 806–823
4 Chang, C.-H., Gu, J., and Zhang, M.: 'A review of 0.18 μm full adder performances for tree structured arithmetic circuits', *IEEE Trans. Very Large Scale Integr. (VLSI) Systems*, 2005, **13**, (6), pp. 686–695
5 Krambeck, R.H., Lee, C.M., and Law, H.-F.S.: 'High-speed compact circuits with CMOS', *IEEE J. Solid-State Circuits*, 1982, **17**, (3), pp. 614–619
6 Harris, D.: 'Skew-tolerant circuit design' (Morgan Kaufmann Publishers, 2001)
7 Gonzalez, R., Gordon, B.M., and Horowitz, M.: 'Supply and threshold voltage scaling for low power CMOS', *IEEE J. Solid-State Circuits*, 1997, **32**, (8), pp. 1210–1216

Efficient approach to extraction of texture browsing descriptor in MPEG-7

M.R. Hejazi and Y.-S. Ho

An efficient method for extracting the texture browsing descriptor (TBD) of MPEG-7 using the nonlinear modified discrete Radon transform is proposed. Results show that the proposed method is very fast and extracts the components of TBD efficiently.

Introduction: Texture is one of the most important features of an image and is widely used in image analysis. There are three descriptors for a texture in MPEG-7 [1]: the homogeneous texture descriptor (HTD), the edge histogram descriptor (EHD) and the texture browsing descriptor (TBD). In this Letter, we address the problem of the efficient construction of TBD in terms of direction, regularity and coarseness. These properties are highly related to the perceptual characterisation of a texture. Literature suggests that there are a number of approaches for extraction of these components. Two state-of-the-art approaches are the methods proposed by Manjunath *et al.* [2], and Lee and Chen [3]. The former has been recommended for MPEG-7. This Letter presents a rather simple, but efficient method for extraction of TBD using the nonlinear modified discrete Radon transform (NLMDRT). NLMDRT, which we proposed in a previous paper [4] (in [4], we called this transform the 'second proposed transform'), is a nonlinear modified version of the discrete Radon transform, where we apply the mean operator to sets with equal numbers of elements (i.e. pixel intensities) along a direction with parameter θ, instead of the summation on sets with different numbers of elements, as compared to the conventional Radon transform.

Structure of TBD: TBD consists of 12 bits: 6 bits for two dominant directions, 2 bits for regularity, and 4 bits for two coarseness components. A direction is quantised into six values, ranging from 0 to 150° with a step size of 30°. The regularity is graded on a scale of 1 to 4, ranging from irregular textures to textures with highly structured periodic patterns. A coarseness component is related to the image scale and associated with each direction. It is quantised into four values, with 0 indicating a fine grain texture, and 3 indicating a very coarse texture.

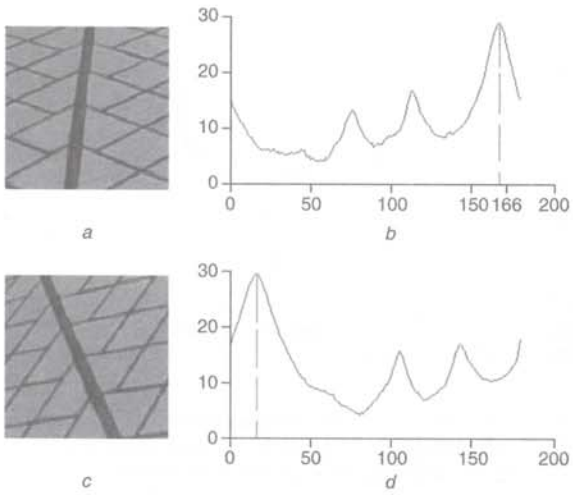

Fig. 1 *Variance arrays of image for two different orientations*
a Image *b* Variance array of image
c 30° rotated version of image *d* Variance array of rotated image

Direction estimation: Generally the texture of an image consists of anisotropic (directional) and/or isotropic (non-directional) textures. In [4], we showed that the performance of NLMDRT in the correct estimation of the dominant direction, that is the direction with more straight lines, was almost perfect for anisotropic textures and even quite high for textures with isotropic dominancy. To this end, NLMDRT is calculated for all directions with θ changing from 0 to 179°. We then compute the variance of the result for each direction and form the variance array S_R as

$$S_R(\theta) = Var_{\theta \in [0,180)}[R_{MM}(\lambda, \theta)] \quad (1)$$

where R_{MM} is the NLMDRT of the image as defined in [4], and the variance function is taken over all co-ordinates of a direction (i.e. λ). The variance array is expected to have a global maximum in the direction perpendicular to the dominant direction [4]. Thus, the first dominant direction D_1 is estimated by

$$D_1 = \text{argmax}_\theta (S_R) - 90^\circ \quad (2)$$

This fact can be verified easily for the multidirectional texture of Fig. 1. It can be seen that there is a peak for each main direction of the image in its variance array. We can also see that the variance array of the 30° rotated version of the image has the same peaks with a circular shift (approximately) equal to 30°. Using this scheme, we may even define a dominant direction for (almost) isotropic images, because pure isotropicity rarely occurs in real-world images. Since two dominant directions are used in the construction of TBD, we can easily estimate the second dominant direction (i.e. D_2) using (2), replacing the position of the second peak of S_R instead of its maximum.

Regularity estimation: The regularity is usually estimated by finding periodic patterns in the image [2]. In this work, we attempt to find this periodicity by comparing variance arrays of blocks of the image. For this purpose, we first divide the image into four non-overlapped blocks, and then calculate the variance array for each subimage using (1). Comparing a regular image with an irregular image, we expect the variance arrays of subimages of the regular image to be quite similar. We show this through an example. Fig. 2*a* is a near regular image, having a multidirectional textural structure, that is divided into four subimages. Smoothed variance arrays of these subimages are shown in Fig 2*b*. It can be seen that the variance arrays are almost similar.

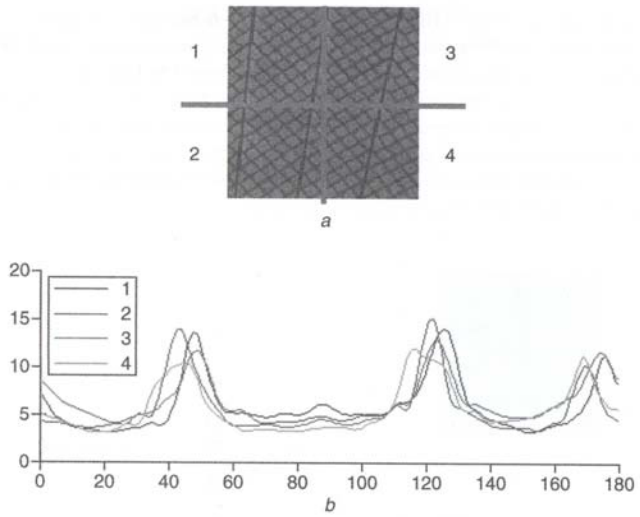

Fig. 2 *Regularity estimation*

a Image of size 128 × 128 and its four 64 × 64 subimages
b Variance arrays of subimages

In this Letter, the criterion for similarity measurement is based on the numbers and positions of peaks in the variance arrays. Let N_p^i be the number of peaks above the DC-bias line of the variance array of the ith subimage (i.e. mean(S_R^i)), and $P_{\text{pos}}^i(j), j = 1,\ldots,N_p^i$ be the positions of these peaks. We can then compare each pair of the variance arrays, i.e. the variance arrays of the kth and lth subimages, and calculate their similarity $SIM(k,l)$ based on the number and positions of peaks in P_{pos}^k that have proper matches in P_{pos}^l. We say that a peak q in P_{pos}^l is a proper match of a peak p in P_{pos}^k if q has the minimum position difference with p as compared to other peaks in P_{pos}^l, and if this difference is less than a threshold T_r. Here, we set T_r to 30°.

Let us assume that m is the number of peaks in P_{pos}^k that have proper matches in P_{pos}^l, and P_{mat}^k and P_{mat}^l are subsets of P_{pos}^k and P_{pos}^l whose elements are positions of these m peaks in P_{pos}^l and their proper matches in P_{pos}^l. We calculate $SIM(k,l)$ by

$$SIM(k,l) = [2m/(N_P^k + N_p^l)](1 - P_{diff}) \tag{3}$$

where

$$P_{diff} = |\overline{P_{mat}^k} - \overline{P_{mat}^l}|/T_r, \quad \text{if } P_{diff} > 1 \quad \text{then } P_{diff} = 1 \tag{4}$$

Calculating the similarity between all possible pairs of variance arrays, the regularity r_t is taken as the average of all these values, and has a value between 0 and 1. The extreme values 0 and 1 show irregularity and complete regularity, respectively. We need to quantise r_t on a scale of 1 to 4 before using it for TBD. This is easily done by applying the ceiling function to $4r_t$.

Coarseness estimation: We estimate the coarseness for a direction based on the principal frequency component of its corresponding peak on the variance array. This frequency component is reciprocally proportional to the peak width of the direction, where the peak width is defined as the distance between the intersection points of the two sides of the peak with the DC-bias line. Fig. 3 shows two images with different levels of coarseness and their corresponding variance arrays. Estimated peak widths for the dominant directions of two images have also been highlighted in Fig. 3. Comparing the two images, we see that the coarser image (Fig. 3*a*) has a bigger peak width. To this end, the coarseness C_i for the ith dominant direction of an image can be estimated by

$$C_i = \begin{cases} 3 + (\beta_i - 45)/(180 - 45), & \beta_i > 45° \\ 2 + (\beta_i - 15)/(45 - 15), & 15° < \beta_i \le 45° \\ 1 + (\beta_i - 7.5)/(15 - 7.5), & 7.5° < \beta_i \le 15° \\ \beta_i/7.5, & \beta_i \le 7.5° \end{cases} \tag{5}$$

where β_i is the peak width of the ith dominant direction. To apply this method for TBD, we calculate C_1 and C_2 using (5) and quantise them on a scale of 0 to 3, using the ceiling function as ceil(C_i) − 1.

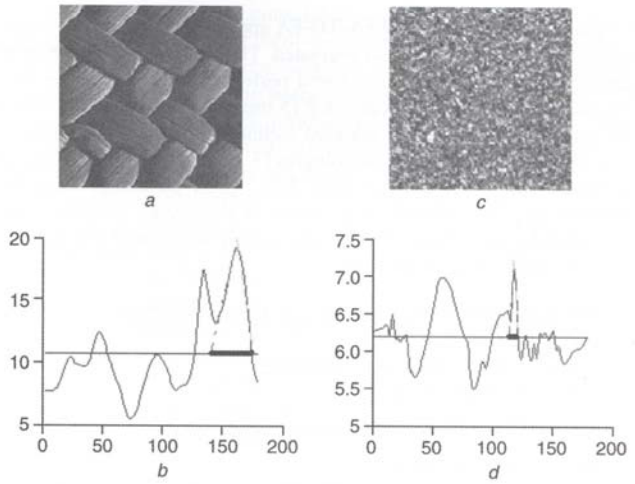

Fig. 3 *Coarseness estimation*

a Semicoarse anisotropic texture
b Variance array of image in *a* and estimated peak width of its dominant direction
c Fine isotropic texture
d Variance array of image in *c* and estimated peak width of its dominant direction

Results: To show the efficiency of the proposed scheme, we compared its performance with the recommended method for TBD [2], as well as the method proposed in [3]. For this purpose, we generated 448 images from 112 texture images of the Brodatz album, similar to the second experiment in [3]. We then classified these images into directional, high regular, and low regular. As shown in Table 1, while the classification rate of the proposed method is as high as the other approaches, its average computational time for producing TBD is significantly less than for the other methods.

Table 1: Performance comparison of different methods

	Proposed method	MPEG-7 TBD	Method in [3]
Correct classification rate	96.2%	96.3%	95.5%
Average computational time	0.7 s	8.9 s	1.2 s

Acknowledgments: This work was supported in part by IITA, in part by MIC through RBRC, and in part by MOE through the BK21 project.

22 January 2007
Electronics Letters online no: 20070208
doi: 10.1049/el:20070208

M.R. Hejazi and Y.-S. Ho (*Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju 500-712, South Korea*)

E-mail: m_hejazi@gist.ac.kr

References

1 MPEG-7 Visual Experimentation Model (XM) Version 10.0, ISO/IEC/JTC1/SC29/WG11, Doc. N4063, March 2001

2 Manjunath, B.S., Wu, P., Newsam, S., and Shin, H.D.: 'A texture descriptor for browsing and similarity retrieval', *Signal Process., Image Commun.*, 2000, **16**, (1), pp. 33–43
3 Lee, K., and Chen, L.: 'An efficient computation method for the texture browsing descriptor of MPEG-7', *Image Vis. Comput.*, 2005, **23**, pp. 479–489
4 Hejazi, M.R., Shevlyakov, G., and Ho, Y.S.: 'Modified discrete Radon transforms and their application to rotation-invariant image analysis'. 8th IEEE MMSP, October 2006, pp. 429–434

Non-predictive multi-scaled based MLVQ video coding for robust transmission over mobile channels

M.F.M. Salleh and J. Soraghan

A new non-predictive video codec for mobile applications is presented. The scheme omits the prediction step in the temporal axis and increases robustness of its transmission in the mobile channel. The same subbands from each frame in the group of frames (GOP) are joined together to exploit their spatial-temporal redundancies. The significant vectors of the joined subbands within the GOP are quantised using a novel multi-stage lattice vector quantisation. This process reduces quantisation errors and enhances the reconstructed frame quality. Experimental results are shown to be significantly better than H.263 and comparable to the current H.264 standards in an erroneous hilly terrain mobile environment using the TETRA channel simulator for some test video sequences.

Introduction: The need for efficient video compression for multimedia application is due to the high volume of video data and the limited bandwidth of the mobile channel [1]. Video compression schemes generally aim to exploit both the spatial and the temporal redundancy of the video sequence. There has been extensive research effort in this field for the last 40 years, as reviewed in [2]. Video coding standards such as H.263 [3] and H.264 [4] utilise motion compensation and motion estimation to reduce temporal redundancy. Besides, there have been works for the non-standard video coding that omit this process, as presented in [5, 6]. In [5] a new three-dimensional subband coding for video is reported to achieve good balance between coding efficiency and error resilience. Akbari and Soraghan [6] have developed a wavelet based video coding scheme that aims for robust video transmission in a noisy mobile environment. In this work, the grouping of frames has eliminated the need for prediction technique. The work in [6] uses adaptive vector quantisation in coding the high frequency subbands. The significant coefficients are identified and quantised using the LBG algorithm which involves high computational demand to generate the codebook. The computational demand can be significantly reduced using lattice vector quantisation [5, 7].

In this Letter, a non-predictive video code is presented for robust transmission in mobile channels based on the work presented in [6]. Instead of using the adaptive vector quantisation (AVQ), the novel multi-stage lattice vector quantisation [8] is used to process a group of video sequences. Since LVQ reduces computation in codebook generation, this allows the new video coding scheme to have multi-stage processes (multi-stage lattice VQ) which reduce quantisation errors and enhance the reconstructed frame quality.

The results obtained for the new video coding scheme in an erroneous mobile hilly terrain (HT) environment using the TETRA channel simulator [9] are significantly better than the H.263 codec and comparable to the H.264 codec.

Video encoder architecture: The wavelet-based video codec encoder architecture is shown in Fig. 1. The video codec takes a video sequence and passes it to a frame buffer. The buffer dispatches several frames at a time to the DWT blocks, thus effectively grouping the video sequence into a group of frames (GOF). The DWT block performs 2-D discrete wavelet transforms on the GOF. In this system, we use three levels of wavelet decomposition, which is sufficient for a QCIF sequence.

The coefficients of the high frequency subbands from each frame are first subdivided into a predefined unit block size of $N \times N$, which ultimately defines the size of the vector dimension. Each of the high frequency subbands (e.g. HH_1) is selected from all frames and joined together to form in single vector list, there will be nine separate vector lists for a group of four frames. These vector lists are then passed to a vector selection process. The significant vectors are identified by comparing the vector energy to a certain threshold. The vectors with energy values greater than a threshold value are preserved. The significant or preserved vectors are than passed to the multi-stage lattice VQ (MLVQ) for lossy compression. The MLVQ process produces two outputs, i.e. the scale list, and index sequence which is then coded using the variable length coding. Details of the MLVQ technique are readily available in [8]. The location information of the significant vectors is defined as the MAP sequence in binary ones and zeros. The MAP sequence is one if the vector is significant and zero otherwise. The MAP sequence is then represented in a quadtree structure via the quadtree coding.

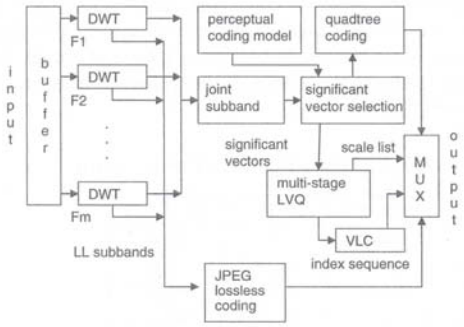

Fig. 1 *MLVQ video encoder architecture*

The advantage of joining the high frequency subbands with a GOP is that if one of the joint subbands' data stream is corrupted then the GOP still can be reconstructed using the other joint subbands and the low frequency subbands.

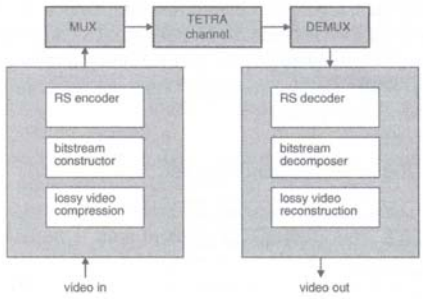

Fig. 2 *Block diagram of MLVQ video codec transmission*

Video transmission: The block diagram of the transmission is shown in Fig. 2. The lossy video compression is the same process as the non-predictive MLVQ encoder discussed in the preceding Section which encodes the video sequence with some compression gain. The compressed bitstream is then classified according to a predefined syntax structure in the bitstream constructor process. In this stage the bitstream is enciphered into two parts, i.e. the header and texture. In the RS encoder process the forward error correction (FEC) codes are added to the bitstream using the Reed Solomon codes. In the next stage, the coded header and texture are combined together in an alternating structure before they are passed through the TETRA channel simulator. The received data are first demultiplexed to the header and texture bitstreams. Then the RS decoder process eliminates the added bit redundancy in the coded bitstreams. Then the bitstream decomposer process deciphers the received bitstreams to meaningful data for video decoding. The final stage is the lossy video reconstruction process, where the compressed video sequence is reconstructed.

Error resilient: We developed the equal and unequal error protection (EEP/UEP) schemes of the MLVQ video data for transmission in a