stringent magnitude and delay characteristics, coefficient word-lengths >15 bits are rarely required.

Relative VLSI cost: The above designs were implemented in ES2 1 μm standard cell CMOS technology using the Preview Place and Route tools in Cadence DFWII, with a data wordlength $d = 18$ and $p = 4$ (2-port), $p = 5$ (3-port). Skew and deskew circuitry was included.

The maximum sample rates of the *lsb* and *msb* designs were found to be 60 MHz and 26.3 MHz, respectively. The core area of the *lsb* 3-port design was $2444 \times 2078 \mu m^2$. The *msb* 2-port design had a core area of $3430 \times 3716 \mu m^2$, 2.5 times the area of the *lsb* design. The VLSI requirements for the 2-port design are as follows: adders 44%, delays 8%, partial product generators 19%, and overflow circuitry 29%. The VLSI requirements for the 3-port design are as follows: adders 66%, delays 26% and partial product generators 8%.

Table 1: Relative VLSI cost

Design	VLSI area / $10^4 \mu m^2$				
	Coef length	Data length			
		8	12	16	24
msb 2-port	4	209	293	386	568
	8	243	347	463	687
	16	257	380	519	786
	24	260	394	556	865
lsb 3-port	5	87	132	168	250
	9	122	171	234	330
	13	157	218	279	420
	17	193	265	338	509
Ratio	4/5	2.40	2.22	2.30	2.27
	8/9	1.99	2.03	1.98	2.08
	12/13	1.64	1.74	1.86	1.87
	16/17	1.35	1.49	1.64	1.70

The main reasons for the *msb* first design having up to 2.5 times more area are therefore the more complex partial product generators and the overflow cells (optimised for area), which are required because of the redundant arithmetic. Table 1 lists estimated cell areas (excludes routing) for both designs for various values of d and p.

Conclusions: It has been shown that the use of *msb* first arithmetic for LTWDFs is unlikely to be worthwhile. The resulting architectures require up to 2.5 times more VLSI area than *lsb* first designs, while they have a lower sample-rate for most practical coefficient wordlengths.

12 February 1996
Electronics Letters Online No: 19960500

M.S. Anderson and S. Summerfield (*University of Warwick, Coventry, CV4 7AL, United Kingdom*)

References

1 ANDERSON, M.S., SUMMERFIELD, S., and LAWSON, S.S.: 'Realisation of lattice wave digital filters using three-port adaptors', *Electron. Lett.*, 1995, **31**, (8), pp. 628–629

2 HARRIS-DOWSETT, D., WICKS, A., and SUMMERFIELD, S.: 'A pipelining method for high speed VLSI wave digital filters'. 37th Midwest Symp. Circuits Syst., August 1994, **2**, pp. 1091–1094

3 LAWSON, S.S., and WICKS, T.: 'Design of efficient digital filters satisfying arbitrary loss and delay specifications', *IEE Proc. G*, 1992, **139**, (5), pp. 611–620

4 MCGOVERN, B.P., WOODS, R.F., and MCALLISTER, C.: 'Optimised multiply/accumulate architecture for very high throughput rate digital filters', *Electron. Lett.*, 1995, **31**, (14), pp. 1135–1136

5 MCQUILLAN, S.E., and MCCANNY, J.V.: 'A systematic methodology for the design of high performance recursive digital filters', *IEEE Trans. Comput.*, 1995, **44**, (8), pp. 971–982

6 SINGH, R.J., and MCCANNY, J.V.: 'High performance VLSI architectures for wave digital filtering', *J. VLSI Signal Process.*, 1992, **4**, pp. 269–278

7 PARHI, K.K., and MESSERSCHMITT, D.G.: 'Pipeline interleaving and parallelism in recursive digital filters - Part II: Pipelined incremental block filtering', *IEEE Trans. Acoust., Speech Signal Process.*, 1989, **37**, (7), pp. 1118–1134

Accelerated Hough transform using rectangular image decomposition

B. Gatos, S.J. Perantonis and N. Papamarkos

Indexing terms: Hough transforms, Image processing, Pattern recognition

A novel fast method for evaluating the Hough transform is proposed, which can be used to accelerate detection of prevalent linear formations in binary images. An image is decomposed using rectangular blocks and the contribution of each whole block to the Hough transform space is evaluated, rather than the contribution of each image point. The resulting acceleration in the calculation of the Hough transform field is demonstrated in two image processing experiments related to object axis identification and skew detection of digitised documents.

Introduction: The Hough transform [1] has emerged in recent decades as a powerful method for many image processing and pattern recognition applications. A problem which has attracted much attention in the literature is the development of fast variants of the original Hough transform. To this end different techniques have been proposed, including parallel implementation [2], hierarchical schemes [3] and application dependent fast algorithms [4]. In this Letter, we propose a novel method for evaluating the Hough transform for binary images, based on the decomposition of the images into rectangular blocks of foreground pixels. Fast evaluation of the Hough transform field is achieved by analytically calculating the contribution to cells in the Hough accumulator array of a whole rectangular block rather than of each individual pixel. The method is particularly effective in applications where prevalent linear features cannot be captured correctly using edge detection, so that the whole image should preferably be used. Significant acceleration (up to 10 times) is achieved in a number of applications using the proposed block Hough transform (BHT), as compared with the usual point Hough transform (PHT) implementation.

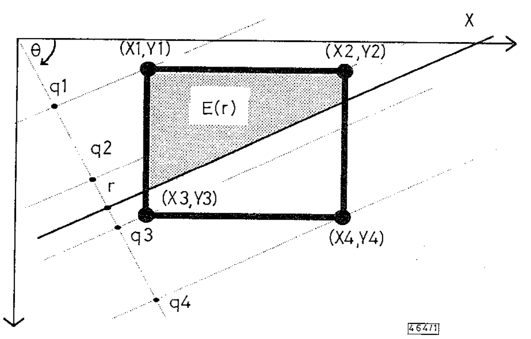


Fig. 1 *Evaluation of rectangular block contribution to Hough transform space*

Algorithm: Given a binary image, we are interested in the detection of straight lines parametrised by $r = x\cos\theta + y\sin\theta$ (see Fig. 1). To this end we firstly form a decomposition of the image into rectangular blocks of foreground pixels, as follows: starting from a foreground pixel (i, j) of the image, we identify the rectangle T_1 that can be extracted if we move point (i, j) to the right until we find a background pixel (i_1, j) and then we move segment $(i, j) - (i_1-1, j)$ parallel to itself and downwards until we find a segment $(i, j_1) - (i_1-1, j_1)$ which has at least one background pixel. We then calculate the area of the rectangle T_2 that can be extracted if we move point (i, j) downwards until we find a background pixel (i, j_2) and move segment $(i, j) - (i, j_2-1)$ parallel to itself and to the right until we find segment $(i_2, j) - (i_2, j_2-1)$ which has at least one background pixel. Of the two rectangles T_1 and T_2 we chose the one which has the larger area as part of the image decomposition. We remove all points of this rectangle from the original image and repeat this process for the modified image iteratively, until no foreground pixels remain.

Next we consider each of the rectangular blocks R of pixels and evaluate the contribution of its points to each cell in the accumulator array of the Hough transform space. Let the co-ordinates of pixels in a rectangle R be defined by $x \in (k_1 - 1/2, k_2 + 1/2)$, $y \in (l_1 - 1/2, l_2 + 1/2)$. Clearly, $(k_2 - k_1 + 1)(l_2 - l_1 + 1)$ pixels lie in the interior of R. Consider a cell in the accumulator array which corresponds to all straight lines determined by the parameters θ and $-1/2 + r_1 < r < 1/2 + r_1$, with r_1 integer. Clearly, the number of points P in R contributing to this cell can be approximated by the area occupied by the intersection of the rectangle R and of the strip in the xy plane restricted between the straight lines $(r_1 - 1/2, \theta)$ and $(r_1 + 1/2, \theta)$ (small discrepancies are due to the discrete image grid). The calculation of P can be carried out analytically as follows: for a given θ we first consider the area $E(r_0)$ of the region defined by the intersection of R and the semi-plane $r < r_0$ whose boundary is the line (r_0, θ) (see Fig. 1). Let X_i, Y_i, $i = 1,..., 4$ be the co-ordinates of the four corners Q_i, $i = 1, ..., 4$ of R in the order in which they are encountered as the fanuly of parallel straight lines (r_0, θ) sweeps the plane moving towards increasing r_0. Moreover, let $q_i = X_i\cos\theta + Y_i\sin\theta$. Straightforward geometrical calculations lead to the following formulas:

$$E(r_0) = \begin{cases} 0 & \text{if } r_0 < q_1 \\ \frac{(r_0-q_1)^2}{2|\cos\theta\sin\theta|} & \text{if } q_1 \le r_0 < q_2 \\ \frac{(q_2-q_1)^2}{2|\cos\theta\sin\theta|} + (r_0-q_2)L & \text{if } q_2 \le r_0 < q_3 \\ \frac{(q_2-q_1)^2+(q_4-q_3)^2-(q_4-r_0)^2}{2|\cos\theta\sin\theta|} + (q_3-q_2)L & \text{if } q_3 \le r_0 < q_4 \\ \frac{(q_2-q_1)^2+(q_4-q_3)^2}{2|\cos\theta\sin\theta|} + (q_3-q_2)L & \text{if } r_0 \ge q_4 \end{cases} \quad (1)$$

for $\theta \neq k\pi/2$, k integer, and

$$E(r_0) = \begin{cases} 0 & \text{if } r_0 < q_1 \\ (r_0 - q_1)L & \text{if } q_1 \le r_0 < q_3 \\ (q_3 - q_1)L & \text{if } r_0 \ge q_3 \end{cases} \quad (2)$$

for $\theta = k\pi/2$, k integer.

L depends on whether Q_2 has the same x or y co-ordinate with Q_1

$$L = \begin{cases} \frac{l_2-l_1+1}{\cos\theta} & \text{if } X_2 = X_1 \\ \frac{k_2-k_1+1}{\sin\theta} & \text{if } Y_2 = Y_1 \end{cases} \quad (3)$$

Finally, P can be readily computed as

$$P = E(r_1 + 1/2) - E(r_1 - 1/2) \quad (4)$$

As a result of the above considerations, our algorithm proceeds as follows:

(1) The image is decomposed into rectangular blocks.
(2) For all blocks and every angle θ
   (i) the quantities q_i are computed and sorted
   (ii) $E(1/2 + r_1)$ is computed using eqns. 1 – 3 for all integers r_1 between $[q_1]$ and $[q_4] + 1$, where $[q]$ denotes the integer part of q.
   (iii) P is computed for all integers r_1 between $[q_1]$ and $[q_4]$ using eqn. 4.

Experimental results: To demonstrate the efficiency of BHT two image processing applications are considered, namely the problem of identifying elongated objects in the field of view of a camera and the problem of estimating the skew of an image document after digitisation. In all experiments the Hough field produced using BHT is very similar to the one produced using PHT and the time consumed is dramatically shorter when using BHT (error caused by the discrete image grids amounts to <1% per cell value). For both PHT and BHT, we use look-up tables for all function calculations involving cos θ and sin θ in order to speed up all processes.

Three chess pieces are placed within the field of a CCD camera and a 512 × 512 pixel image of the field of view is produced (Fig. 2). Object axes correspond to the prevalent image lines, so the problem of object axis estimation reduces to the problem of estimating peaks of the Hough field. Considering only edge pixels is time saving but is not applicable to the task at hand, because part of the required axes lies in the interior of individual objects. The time consumed to construct the Hough field and locate the three object axes is dramatically shorter when using BHT instead of PHT (3 and 37s, respectively, on a PC 486DX/33MHz).

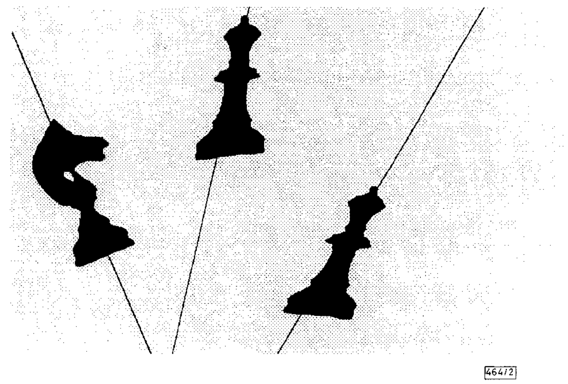


Fig. 2 *Object axis determination using Hough transform*

Time needed is dramatically decreased when using BHT instead of conventional PHT

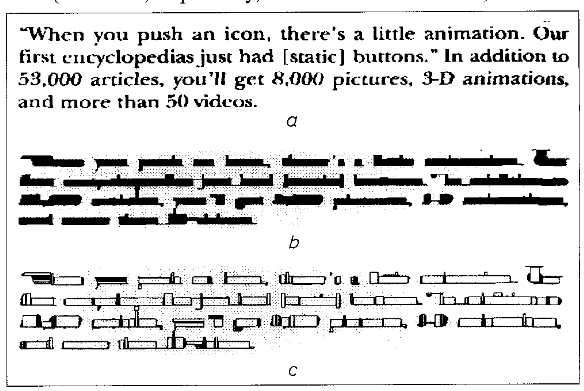


Fig. 3 *Images of writing*

a Part of a document image
b Final image after RLSA application
c Block decomposition of the final image

In OCR applications image skew correction is an important preliminary step before proceeding to character segmentation and recognition [5]. The Hough transform has been applied with success to skew estimation [6], but its application introduces significant time delay in the recognition process. We have corrected the skew of several documents using combinations of different preprocessing schemes and Hough transform implementations. We have applied two preprocessing methods to the original document image, namely edge detection and runlength smoothing algorithm (RLSA), as shown in Fig. 3 [7]. Both PHT and BHT were then applied to the resulting images in order to determine the skew. For all combinations excellent results were obtained concerning the accuracy of skew angle determination (maximum error = 0.1°). Results concerning the time consumed by the different methods are shown in Table 1. For all combinations BHT is faster than the corresponding PHT implementation. The quickest result is obtained using BHT in combination with RLSA.

Table 1: Using RLSA preprocessing and BHT is the quickest combination for finding the skew angle of a digitised document

Preprocessing tool		Time, s	
RLSA (smoothing)	Edge detection	PHT	BHT
no	no	111	50
no	yes	63	—
yes	no	206	17
yes	yes	35	—

Conclusions: A novel fast method was presented for applying the Hough transform to binary images. The method is based on the decomposition of the images into rectangular blocks of foreground pixels. Its efficiency was demonstrated in two image processing applications, where significant acceleration was achieved compared to the classical point Hough transform implementation.

19 February 1996
Electronics Letters Online No: 19960510

B. Gatos and S.J. Perantonis (*Institute of Informatics and Telecommunications, National Research Center 'Demokritos', GR-153 10, Aghia Paraskevi, Greece*)

N. Papamarkos (*Electric Circuits Analysis Laboratory, Department of Electrical and Computer Engineering, Demokritus University of Thrace, 67100 Xanthi, Greece*)

B. Gatos: also with Electric Circuits Analysis Laboratory, Department of Electrical and Computer Engineering, Demokritus University of Thrace, 67100 Xanthi, Greece

References

1 DUDA, R.D., and HART, P.E.: 'Use of the Hough transform to detect lines and curves in pictures', *Commun. ACM,* 1972, **15**, pp. 11–15
2 KANNAN, C.S., and CHUANG, H.Y.H.: 'Fast Hough transform on a mesh connected processor array', *Inf. Proc. Lett.,* 1990, **33**, (5), pp. 243–248
3 GUIL, N., VILLALBA, J., and ZAPATA, E.L.: 'A fast Hough transform for segment detection', *IEEE Trans. Image Process.,* 1995, **4**, (11), pp. 1541–1548
4 KE-QING, L., and QI, T.: 'A fast Hough transform for inspecting accurate needle-type meter gauges'. Proc. IAPR Workshop on Computer Vision, 1988, pp. 195–198
5 LE, D.S., THOMA, G.R., and WECHSLER, H.: 'Automated page orientation and skew angle detection for binary document images', *Pattern Recognit.,* 1994, **27**, (10), pp. 1325–1344
6 HINDS, S.C., FISCHER, J.L., and D'AMATO, D.P.: 'A document skew detection method using run-length encoding and the Hough transform'. Proc. 10th Int. Conf. Pattern Recognition, 1990, **1**, pp. 464–468
7 WONG, K.Y., CASEY, R.G., and WAHL, F.M.: 'Document analysis system', *IBM J. Res. Devel.,* 1982, **26**, (6), pp. 647–656

Improved rate 1/4 convolutional codes

G. Wade, M. Tomlinson and P. Van Eetvelt

Indexing terms: Convolutional codes, Error correction codes, Error detection codes

The performance of parallel architectures for generating rate 1/4 high constraint length convolutional codes is examined via information weight distributions. It is found that these codes offer a significant decoding advantage, while having a comparable bit error rate (BER) performance to classical codes.

Introduction: Recursive systematic convolutional codes approaching the Shannon limit have been discussed recently in the literature [1]. These half rate systematic codes are generated from two encoders working in parallel. In this Letter we examine the weight distribution of linear non-systematic codes generated from two half rate encoders working in parallel, corresponding to a rate 1/4 code in unpunctured form. Both recursive and non-recursive parallel encoder architectures are examined.

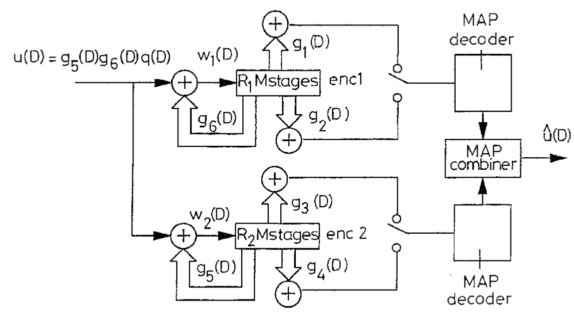
Since each code is linear we measure its distance properties by calculating the distances of remerging paths to the all-zeros code sequence. In general there will be many remerging paths of distance d from the all-zeros sequence. Denoting the total number of information paths associated with paths of distance d by B_d, this process is usually described by a polynomial [2]

$$f(D) = \sum_{d=d_{free}}^{\infty} B_d D^d \qquad (1)$$

The weight distribution B_d can then be used to compute an upper bound on the bit error rate (BER) as [2],

$$BER < \frac{1}{2} \sum_{d=d_{free}}^{\infty} B_d \operatorname{erfc}\left(\sqrt{\frac{E_b}{4N_0}d}\right) \qquad (2)$$

Usually only the first few terms are required to evaluate the upper bound.



367/1

Fig. 1 *Rate 1/4 recursive, parallel encoder-decoder system*

Recursive parallel R=1/4 codes: The system in Fig. 1 applies feedback to two non-systematic rate one half encoders working in parallel. Since feedback cannot affect the d_{free} of the corresponding rate one half code, a sensible choice of polynomials is to let $g_1(D)$ and $g_2(D)$ correspond to the optimal, classical $R = 1/2$ non-systematic code. A computer search then optimises $g_3(D)$, $g_4(D)$, $g_5(D)$ and $g_6(D)$ in terms of d_{free} and weight distribution. In general, polynomials $u(D)$ and $g_6(D)$ ($g_5(D)$) are relatively prime and so $w_1(D)$ ($w_2(D)$) will be a semi-infinite sequence. An efficient computer search is then to apply a sequence of the form $u(D) = g_5(D)g_6(D)q(D)$, as indicated, where $q(D)$ is of realistically small order. Finite sequences $w_1(D)$ and $w_2(D)$ will then eventually clear registers R_1 and R_2 and the required code distance d is then the Hamming weight of the remerging path.

Some optimal code polynomials for the system in Fig. 1 are given in Table 1 (the corresponding weight distributions have been verified up to register length 7 using the transfer function approach). In general there are other polynomial sets for a fixed M which yield the same d_{free}, although their weight distributions yield an inferior BER.

Table 1: Weight distribution of optimal recursive parallel codes of rate 1/4, constraint length $K = 2M + 1$

Register length, M	Feed forward polynomials (octal)		Feed back polynomial (octal)	Minimum weight	Total information weight				
	g_1 / g_3	g_2 / g_4	g_6 / g_5	d_{free}	B_{dfree}	$B_{dfree+1}$	$B_{dfree+2}$	$B_{dfree+3}$	$B_{dfree+4}$
5	53	75	53	29	36	40	24	42	80
	65	77	57						
6	171	133	165	33	32	48	56	60	82
	175	117	123						
7	247	371	325	38	143	0	230	0	293
	301	251	263						
8	561	753	541	42	122	0	300	0	420
	771	537	405						
9	1167	1545	1401	46	186	0	202	0	562
	1751	1553	1561						
10	2335	3661	3001	50	194	0	398	0	736
	3233	3537	3231						

Decoding: The system in Fig. 1 can be modelled as a single register encoder of constraint length $K = 2M + 1$. Thus, $M = 6$ yields a code with an effective constraint length of 13 and $d_{free} = 33$. Rather than use a 2^{12} state trellis decoder, the architecture in Fig. 1 permits the use of two relatively simple MAP decoders each based on 6-stage registers, as indicated. Each decoder computes the *a posteriori* probabilities of an information bit u_j being 1 or 0, and the results are simply combined to obtain an estimate of the log-likelihood ratio for u_j.