system performance is nearly the same as for a system using an ideal mutlibit internal DAC. The added hardware in a switched-capacitor implementation for the proposed system involves only a few switches and capacitors, and some digital circuitry.

Acknowledgments: The authors are grateful to A. Ganesian for useful discussions.

18th October 1991

X. Xu and G. C. Temes (*Department of Electrical and Computer Engineering, Oregon State University, Corvallis, OR 97331-3211, USA*)

References

1 NAUS, P. J. A., and DIJKMANS, E. C., *et al.*: 'A CMOS stereo 16-bit D/A converter for digital audio', *IEEE J. Solid-State Circuits*, 1987, **SC-22**, pp. 390–394

2 KUP, B. M. J., DIJKMANS, E. C., NAUS, H., and SNEEP, J.: 'A bitstream D/A converter with 18b resolution'. ISSCC Dig. Tech. Paper, February 1991, pp. 70–71

3 MATSUYA, Y., UCHIMURA, K., IWATA, A., and KANEKO, T.: 'A 17-bit oversampling D/A conversion technology using multistage noise shaping', *IEEE J. Solid-State Circuits*, 1989, **SC-24**, pp. 969–975

4 CATALTEPE, T., KRAMER, A. R., LARSON, T. E., TEMES, G. C., and WALDEN, R. H.: 'Digitally corrected multi-bit ΔΣ data converters'. IEEE Proc. Int. Symp. on Circuits and Systems, May 1989, pp. 647–650

5 LESLIE, T. C., and SINGH, B.: 'An improved sigma-delta architecture'. IEEE Proc. Int. Symp. on Circuits and Systems, May 1989, pp. 372–375

6 PAULOS, J. J., BRAUNS, G. T., STEER, M. B., and ARDALAN, S. H.: 'Improved SNR using tri-level delta-sigma modulation'. IEEE Proc. Int. Symp. on Circuits and Systems, May 1987, pp. 463–466

DETERMINATION OF LINE LENGTH USING HOUGH TRANSFORM

M. W. Akhtar and M. Atiquzzaman

Indexing terms: Transforms, Signal processing

A new method is presented for determining the length of a line using the information contained in the Hough accumulator array. The length is calculated from the amount of spread of the votes in the accumulator array. As compared to other methods, the new method is highly efficient in terms of computing time.

Introduction: The Hough transform (HT) is widely used in machine vision and object recognition for detection of parameterisable geometric shapes. HT is a mapping from the image plane onto the parameter space. Consider the HT for straight lines using normal parametrisation. In this parametrisation, a straight line is represented by

$$\rho = x \cos \theta + y \sin \theta \qquad (1)$$

where, ρ is the length of the normal to the line segment from the image origin and θ is the angle of the normal with the positive x axis. The parameter θ is sampled while ρ is quantised. For each feature point, and for each sampled value of θ, the value of ρ is computed from eqn. 1 and quantised to the nearest value on the ρ axis. The corresponding cell in the accumulator array is incremented. Advantages of the HT are its robustness against noise and discontinuities in the pattern to be detected [1]. One of the major drawbacks of the HT is that it is highly compute-bound and requires large storage space. A second drawback of the HT for straight lines is that the conventional approach does not provide any information regarding the length and position of the line segment in the image plane. Several authqrs [2–4] have suggested methods of obtaining the position of the line segment. All the approaches are based on finding the endpoints by projecting the line segment on the image axis, and determining the length from the endpoints. All the methods assume that the Hough accumulator array is available. Hence, in determining the complexity of the algorithms (as shown below), the complexity of the initial construction of the accumulator array has not been taken into consideration.

The method described by Yamato *et al.* [2] is based on a heuristic approach. A line is determined by performing a least-squared error fit of the likely feature points. Projection of the line on the image axis gives the co-ordinates of the endpoints which are used to find the length of the line. The algorithm is iterative in nature, and uses least square curve fitting, thereby making it computation-intensive. The complexity of the algorithm has been found to be $O(n_f n_i + N)$, where n_f, n_i, and N^2 are the number of feature points, the number of iterations, and the size of the accumulator array, respectively. Sandler *et al.* [3] use the combinatorial Hough transform to determine ρ and θ of the line. All possible feature points are projected onto a one-dimensional histogram along either the x or y axis. A connectivity analysis of this one-dimensional histogram gives the end points of the line from which the length is determined. The algorithm has been found to be $O(n_f + N)$. Niblack and Tai [4] assume certain parameters of the line to be detected, and generate an accumulator array $\hat{H}$ using these parameters. The error between $\hat{H}$ and the accumulator array H generated by the actual feature points is minimised by varying the values of the assumed parameters. With many parameters to be varied, the procedure becomes highly compute-bound because of too many reaccumulations of the accumulator array. The complexity of the algorithm is $O(n_f n_i N + N^2)$.

The approaches available in the literature do not directly use the information in the accumulator array, and they are all highly computation-intensive. In this Letter, a new computationally-efficient method for finding the length of the line segment from the information contained in the accumulator array is proposed. The spread of the votes in the columns of the accumulator array is exploited to obtain the length. Its computational efficiency arises because it needs no reaccumulation of the parameter array, and does not require iterations.

Determination of length of line from accumulator array: A cell (i, j) in the accumulator array corresponds, in the image plane, to an infinite-length bar-shaped window at a distance ρ_i from the origin, of width $\Delta\rho$, and making an angle θ_j with the x axis, where $\theta_j = j\,\Delta\theta$, and $\rho_i = i\,\Delta\rho$. The number of votes in the cell is equal to the number of feature points lying in the window corresponding to that cell. The jth column of the accumulator array will be represented by C_j. Hence, the column p containing the peak is C_p. In general, it is not necessary that the actual angle of the line, denoted by θ_a, be exactly equal to θ_p. When this occurs, the line segment may cross several windows of width $\Delta\rho$ resulting in the spread of the votes in the ρ direction in C_p. The maximum spread (n_p^p) in C_p in the ρ direction was shown by Vanveen and Groen [5] to be given by

$$n_p^p = \left\lfloor \frac{l \sin\left(\frac{\Delta\theta}{2}\right)}{\Delta\rho} \right\rfloor + 2 \qquad (2)$$

The authors derived the equation for the purpose of suggesting an optimum value of $\Delta\rho$, assuming that the length of the line is known. In real-world applications, prior knowledge regarding the length of the line segment may not be available. The motivation behind the work reported in Reference 5 was to analyse the discretisation errors, and is completely different from the problem discussed in this Letter.

Using the concepts of probability, eqn. 2 can be approximated as [6]

$$n_p^p = \frac{l \sin\left(\frac{\Delta\theta}{2}\right)}{\Delta\rho} + 1 \qquad (3)$$

Because the angle θ_p of the window (corresponding to the peak cell) may not be exactly equal to θ_a, the maximum angle between the line segment and the window is $\Delta\theta/2$. Similarly, the maximum angle between the windows corresponding to the cells in any column C_j is $(\Delta\theta)/2 + d_{j,p}\,\Delta\theta$ [6], where $d_{j,p}$ is the number of columns between C_j and C_p, and is given by $d_{j,p} = |j - p|$. Hence, the spread of the votes in any column C_j is given by

$$n_p^j = \frac{l \sin\left(\frac{\Delta\theta}{2} + d_{j,p}\,\Delta\theta\right)}{\Delta\rho} + 1 \tag{4}$$

which can be rearranged to give

$$l = \frac{(n_p^j - 1)\,\Delta\rho}{\sin\left(\frac{\Delta\theta}{2} + d_{j,p}\,\Delta\theta\right)} \tag{5}$$

The length of the line can be obtained from eqn. 5 as follows. $\Delta\rho$ and $\Delta\theta$ are the resolutions used in constructing the accumulator array, and hence are known. The value of p is obtained by finding the column in the accumulator array with maximum votes. Any C_j, $j \neq p$ is chosen, and n_p^j is found by finding the first and the last nonzero elements in C_j. The ρ index of the first nonzero element in C_j, denoted by μ_f^j, is obtained by scanning C_j upwards starting from the bottom as shown in Fig. 1. The scanning is continued until the last nonzero element in C_j, whose ρ index is denoted by μ_l^j, is found. n_p^j is, therefore, given by $n_p^j = |\mu_l^j - \mu_f^j|$. Substituting values of n_p^j, $\Delta\rho$, $\Delta\theta$, and $d_{j,p}$ in eqn. 5 gives l.

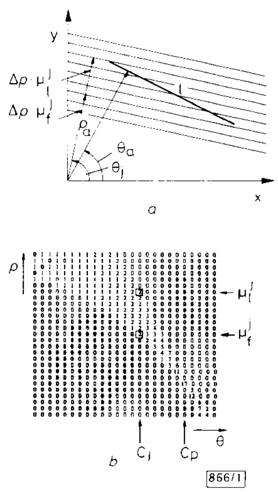

Fig. 1 *Bar shaped windows corresponding to cells in accumulator array*

a Windows in image plane
b Cells near peak in accumulator array

The above procedure is carried out for a number of values of j, and the different lengths l obtained are averaged to obtain an average length l_{ave} as

$$l_{ave} = \frac{1}{(j_2 - j_1)} \sum_{j=j_1}^{j_2} \frac{(n_p^j - 1)\,\Delta\rho}{\sin\left(\frac{\Delta\theta}{2} + d_{j,p}\,\Delta\theta\right)} \tag{6}$$

The number of columns for which the averaging is performed depends on the accuracy desired. Table 1 shows simulation results for several representative lines, for averaging over 4, 6, 8, and 10 columns.

In general, digitised points corresponding to a line segment are not colinear as shown in Fig. 2. This figure shows the breaking up of the line segment into smaller horizontal line segments [6] for a line having $\theta_a > 45°$. (x_1, y_1) and (x_2, y_2) are the actual endpoints and (x_3, y_3) and (x_4, y_4) are the points obtained due to digitisation. (x_1, y_1) and (x_2, y_2) should vote into the cells (μ_f^j, j) and (μ_l^j, j), respectively. For $\theta_a > 45°$ (as in Fig. 2), if C_j is chosen such that $j - p$, then there is a possibility that votes from (x_3, y_3) and (x_4, y_4) will be accumulated in (μ_f^j, j) and (μ_l^j, j), respectively, and not from (x_1, y_1) and (x_2, y_2). This will result in a small spread s_1. On the other hand, if $j < p$ is selected, votes from (x_1, y_1) and (x_2, y_2) are going to be accumulated in (μ_f^j, j) and (μ_l^j, j), respectively, giving rise to a spread s_2 which is the true spread arising from the complete length of the line. This is because s_1 assumes that (x_3, y_3) and (x_4, y_4) are the end points of the line.

Table 1 SIMULATION RESULTS FOR COLUMNS TAKEN ON BOTH SIDES OF PEAK

θ_a	θ_p	Actual length of line	l_{ave} for $(j_2 - j_1) =$ 4	6	8	10
deg	deg					
23	23·25	61·847	65·42	60·83	59·98	60·21
28	28·31	68·884	60·25	62·31	63·96	63·39
30	30·33	70·327	69·41	68·52	68·41	68·23
42	42·47	79·310	65·42	68·21	69·83	69·97
56	55·62	84·315	75·75	77·56	79·71	79·75
59	58·65	81·633	75·75	77·56	76·84	77·46
65	64·71	77·389	75·75	75·10	74·99	75·04

$\Delta\rho = 0{\cdot}91$, $\Delta\theta = 1{\cdot}0$

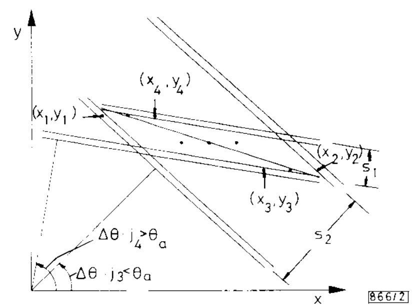

Fig. 2 *Erroneous detection of end points due to choice of C_j, $j > p$ for $\theta_a > 45°$*

The reverse situation arises in the case of $\theta_a < 45°$. For $\theta_a = 45°$, there is no such problem because no two digitised points are either on the same horizontal of vertical line, and no ambiguity regarding the end points can occur. Therefore, the accuracy of the length of the line segment computed using eqn. 6 can be further enhanced by applying the following rule to the selection of the C_js for which l will be determined:

Use C_j, $j > p$ if $-45° < \theta_a < 45°$, $135° < \theta_a < 225°$. Use C_j, $j < p$ if $45° < \theta_a < 135°$, $225° < \theta_a < 315°$. Use any C_j for $\theta_a = 45°$, 135°, 225°, and 315°.

θ_a is approximated by θ_p. Use of the above rule while selecting columns in the accumulator array considerably reduces the error in the average length of the line as can be seen by comparing Table 1 with Table 2.

Table 2 SIMULATION RESULTS FOR COLUMNS TAKEN ON ONE SIDE OF PEAK

θ_a	θ_p	Actual length of line	l_{ave} for $(j_2 - j_1) =$ 4	6	8	10
deg	deg					
23	23·25	61·847	61·17	61·51	61·85	61·78
28	28·31	68·884	68·39	68·88	69·55	69·79
30	30·33	70·327	71·65	71·14	70·68	70·69
42	42·47	79·310	79·24	79·67	79·25	79·43
56	55·62	84·315	85·11	84·40	84·45	84·31
59	58·65	81·633	81·59	81·24	81·51	81·24
65	64·71	77·389	79·87	78·35	77·74	77·48

$\Delta\rho = 0{\cdot}91$, $\Delta\theta = 1{\cdot}0$

Concluding remarks: The standard Hough transform for straight line detection does not provide any information regarding the length of the line. Several methods have been proposed in the literature for determination of line length. All the methods are based on projections, and are highly computation-intensive as is evident from the complexities as stated earlier. A very simple and computationally efficient method for determination of line length from the spreading of the votes in the accumulator array has been proposed. The complexity of the proposed approach has been found to be $O(1)$. This has been achieved at the expense of a small degradation in the accuracy as compared to other methods. Results obtained using this method have also been presented.

23rd September 1991

M. W. Akhtar and M. Atiquzzaman (*Department of Electrical Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia*)

References

1 ILLINGWORTH, J., and KITTLER, J.: 'A survey of the Hough transform', *Comput. Vis. Graph. Image Proc.*, 1988, (44), pp. 87–116

2 YAMATO, J., ISHII, I., and MAKINO, H.: 'Highly accurate segment detection using Hough transform', *Systems and Computers in Japan*, 1990, **21**, (1), pp. 68–77

3 SANDLER, M. B., DE FONTURA, C. L., and BEN-TZVI, D.: 'Performance improvements to the Hough transform', *IEE Conf. Publication*, 19th–22nd March 1990, pp. 98–103

4 NIBLACK, W., and TRUONG, T.: 'Finding line segments by surface fitting to the Hough transform'. IAPR Int. Workshop on Machine Vision Applications, 18th–30th November 1990, Tokyo, pp. 1–4

5 VANVEEN, T. M., and GROEN, F. C. A.: 'Discretisation errors in the Hough transform', *Pattern Recognition*, 1981, **14**, (1–6), pp. 137–145

6 AKHTAR, M. W.: 'Towards an optimal accumulator size for the Hough transform'. MSc Thesis, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, August 1991

MATCHED FILTER FOR Gbit/s APPLICATIONS

B. Enning

Indexing terms: Filters, Signal processing

A matched filter circuit for a rectangular pulse system is presented. It is composed of two transversal filter sections. Owing to its simplicity it is attractive for Gbit/s signal processing. Frequency response and time domain measurements of the realised filter are given.

Introduction: It is known that the optimum receiver for rectangular signals in a system without bandwidth restrictions, corrupted by white Gaussian noise, is a matched filter which converts the rectangular input signal to a signal of triangular shape at its output [1]. This corresponds to a frequency response of the filter

$$G = \frac{\sin(\pi T f)}{\pi T f} \qquad (1)$$

where T is the duration of a single signal element.

Although a paper was published recently [2] where a receiver with a matched filter for a 565 Mbit/s optical PSK system was presented, matched filter detection is not common in broadband transmission systems. This may partly be related to the moderate expected improvement in receiver sensitivity of less than 1 dB when compared to conventional receiver filters [1], and partly be related to the expected high effort when realising a matched filter for broadband applications.

The circuit configuration presented in Fig. 1 shows that the required filter characteristic can be approached very easily, hence allowing an improvement in receiver sensitivity.

Principle of operation: Fig. 1*a* shows the block diagram of the circuit. It consists of two cascaded transversal filter circuits with delays $\tau_1 = T/2$ and $\tau_2 = T/4$.

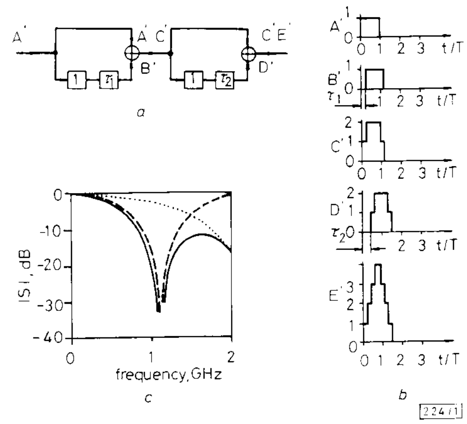

Fig. 1 *Matched filter configuration*

a Block diagram
b Principle time scheme
c Frequency response (maximum value 0 dB for convenience), calculated
– – – – response of first stage
. . . . response of second stage
——— overall response

The time response is depicted in Fig. 1*b* indicating a staircase signal shape of the output signal with a triangular envelope.

The frequency response of the filter

$$S = 4 \,.\, \cos(\pi/2 \,.\, fT) \,.\, \cos(\pi/4 \,.\, fT) \qquad (2)$$

is depicted in Fig. 1*c*. Comparing S from eqn. 2 with $4 \,.\, G$ from eqn. 1 it turns out that the approach of the desired function $\sin(x)/x$ is excellent for $0 \leq f \leq 1/T$. Here the deviation is less than 1 dB, and for $1/T < f \leq 1{\cdot}7/T$ it is still less than 3 dB.

Realised circuit: One stage of the realised transversal filter circuit is depicted in Fig. 2. It consists of an input stage which mainly achieves a 50 Ω input match by the resistive voltage divider 33 Ω, 18 Ω and the output stage with an open ended transmission line of delay $\tau/2$ at the collector. This output stage creates the desired frequency response $V_{out}/V_{in} \propto \cos(\pi\tau f)$ [3]. The time response of a 1·12 Gbit/s NRZ signal is shown in Fig. 3: 3*a* shows the eye diagram of the input signal of the filter and 3*b* the output signal of the complete two stage circuit. Clearly the required triangular shape of the output signal is identified.

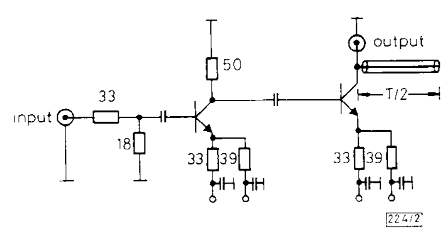

Fig. 2 *Realised circuit of one transversal filter stage*

Base bias has been omitted

Conclusion: For a signal with rectangular shape, matched filter detection can improve the receiver sensitivity by approximately 0·8 dB when compared with a conventional filter [1]. This Letter shows that appropriate circuit schemes exist which allow the implementation of a matched filter in Gbit/s systems. The presented configuration of two cascaded simple