

Concurrent Hardware Alternatives for the Implementation of the Binary Hough Transform

Luciano da Fontoura Costa

RESUMO

Alternativas de hardware concorrente são consideradas para a implementação da transformada binária de Hough, uma técnica simples e efetiva para detecção de barras digitais. As alternativas consideradas incluem: processadores de alto desempenho para uso geral (transputers e TMS320C40), o processador de imagens paralelo MAPP2200, circuitos lógicos discretos e VLSI digital e de múltiplos-valores. Os méritos e limitações de cada uma destas opções para a implementação da transformada binária de Hough são discutidos.

ABSTRACT

Concurrent hardware alternatives for the implementation of the binary Hough transform, a simple and yet effective technique for digital bar detection, are presented and discussed. The alternatives considered in this paper include: general purpose parallel processors (transputers and the TMS320C40), the parallel image processor MAPP2200, discrete logic integrated circuits and VLSI (digital and multiple-valued circuits). Each of these options is discussed with respect to their relative merits and shortcomings as means for binary Hough transform implementation.

Grupo de Instrumentação e Informática,
Instituto de Física e Química de São Carlos,
Universidade de São Paulo,
Caixa Postal 369,
São Carlos, SP
13560-970 Brazil
phone: +55 (162) 71-9199
FAX: +55 (162) 71-3616
e-mail: luciano@uspfc.ifqsc.usp.br

1 Introduction

It is difficult to think of another image processing or computer vision technique that has become so popular as the Hough transform – HT [1, 2, 3]; it suffices to say that about 400 papers in the English language on the Hough transform had been published until 1991. The HT, introduced as a patent in 1962 [4], was later related to other techniques such as template matching [5], Radon transform [6], Fourier transform [7] and projections [8]. Its underlying principle consists in mapping the image feature elements, usually the edge-elements, into a discrete parameter (or Hough) space in such a way that peaks in the latter indicate *possible* instances of the sought pattern in the image. The current popularity of the HT can be explained by its following interesting properties: (a) it is simple and effective; (b) it searches *simultaneously* for all the instances of the sought pattern in the image; (c) it performs image segmentation and parametric fitting; (d) it allows certain levels of tolerance to distortion, occlusion or noise in the image; (e) it involves less computational complexity than many other pattern recognition alternatives such as template matching and (f) it is particularly suitable for implementation in parallel hardware and software [1, 2, 3, 9].

Although HTs can be generalized for detection of other curves and general patterns [2], it is the HTs for straight line detection that allow the best performance and simplicity. In fact, digital straight line segments as well as their generalization as digital bar segments [3, 9] are representation elements specially relevant to natural and computer vision for the following reasons: (i) any digitized curve can be represented or detected in terms of digital straight line segments and (ii) the mammalian visual cortex is in great part oriented to detection of bar segments [3, 10]. The series of possible analogies recently established between various of the HT aspects (e.g. modularity and parameter space organization) and the processing occurring in the mammalian visual cortex: [11, 12, 13] have provided further support to the HT as a specially significant pattern recognition technique.

Four issues have been identified as the principal barriers for the achievement of versatile and powerful computer vision systems: (a) need for real-time execution of a large amount of input data for which conventional Von-Neumann machines have often proved to be useless; (b) need for improved vision models and algorithms; (c) lack of knowledge about the natural vision systems which have been used as basis for many computer vision systems and (d) poor gray-level and spatial resolutions commonly adopted for image representation [14]. It should be emphasized that the odds of obtaining real-time versatile computer vision systems are improved whenever more than one of such barriers are simultaneously addressed. For instance, the development of improved algorithms should at the outset be oriented to parallel hardware or software implementations. Even issue (c), which relies primarily on biological experiments, should be attacked by computer vision researchers in the sense of achieving cross-fertilization between biological data and formal computational models [15].

The efforts reported in the present paper are part of a bigger ongoing project aimed at the development and implementation of a versatile framework for real-time intermediate-level vision [16] capable of describing a given input image in terms of its digital bar segments. Issues (a) to (d) above have been considered from the very beginning of the project development. This paper is aimed at the assessment of the advantages and shortcomings allowed by several representative concurrent hardware alternatives (pipeline, systolic or vectorial structures) for the implementation of the binary Hough transform – BHIT, which is the basis of the above mentioned framework. The BHIT is a variation of the standard HT for straight line detection which allows some interesting advantages such as no need for products and table look-ups and can be calculated without any rounding errors in integer arithmetic

(fixed-point).

The paper starts by specifying the problem and follows by presenting the BIIT and the series of parallel hardware options for its implementation, which includes: general and semi-general purpose processors (transputer and TMS320C40), the parallel image processor MAPP2200, discrete logic integrated circuits and VLSI (digital and multiple-valued circuits). The paper concludes with a discussion of the relative advantages and disadvantages of the considered alternatives.

2 The Binary Hough Transform

The BIIT is a variation of the standard HT for digital bar detection [9] which is based on McIlroy's modelling of digital straight lines [17]. Each point $P = (x, y)$ of a digital bar in the image defines a digital bar in the parameter space in such a way that any point in this latter digital bar defines yet another digital bar in the image space which contains P [9, 3]. The digital bar associated to each parameter space point Q corresponds to its 'receptive field', i.e. a high count will be obtained in Q whenever the image contains a line falling onto the 'receptive field'. It is straightforward to verify that parameter space points lying on a vertical line will define a family of parallel digital bars in the image space.

A possible formulation for the BIIT is given by expression (1), where j is a positive integer, M and C are respectively the bar slope and intercept and N_M is the amount of slope samples. For simplicity's sake, $N \times N$ images are assumed and the intercept resolution is fixed to be 1 (the general expression can be found in [3, 9]). In order to cover all the possible slopes, expression (1) has to be applied over every image feature element $P = (x, y)$ according to the four coordinate transformations given by (2) to (5). The accumulator array is incremented for each of the produced instances of C_v and i .

$$C_v(i) = y_v - i.(x_v/N_M) \quad (1)$$

$$\text{where } N_M = 2^j,$$

and

$$i = 0, 1, 2, \dots, N_M$$

$$(x, y) = (x_1, y_1) \quad (v=1) \quad (2)$$

$$(x, y) = (x_2, N - 1 - y_2) \quad (v=2) \quad (3)$$

$$(x, y) = (y_3, x_3) \quad (v=3) \quad (4)$$

$$(x, y) = (y_4, N - 1 - x_4) \quad (v=4) \quad (5)$$

The BIIT presents the following interesting features: (a) by processing the products through accumulated additions, the BIIT can be executed without products or table look-ups; (b) it allows optimal utilization of the memory required for the Hough space representation [18, 19, 3]; (c) it can be performed without rounding errors in fixed-point arithmetic and (d) it has been experimentally verified that the BIIT allows performance approximately equivalent to other standard HTs in respect to the amount of false and broken lines and to parameter estimation errors [3]. The BIIT also presents some shortcomings, common to most other HTs, such as the fact that some digital bars can be truncated or replicated during detection and that no information about the image connectivity is produced (sparse points can be detected as bars and the extremities of actual bars are not determined). Both these problems can nevertheless be easily circumvented by post-BIIT processings (respectively a merging and connectivity analysis stages) [9, 3]. The amount of false peaks resulting

in the Hough space can be reduced by adopting Gerig and Klein's strategy described in [20, 9].

3 Parallel Hardware for Implementation of the Binary Hough Transform

Being an important intermediate-level computer vision technique often required to run in real-time, many concurrent hardware structures have been considered for IIT implementation [2, 21, 3]. Although it has been experimentally verified that IITs are not particularly suitable for implementation in bidimensional processing arrays, they have proved to be generally suitable for implementation in dedicated architectures and linear SIMD and MIMD arrays [3]. A number of distinct concurrent hardware alternatives are considered as means of IIT implementation in the next subsections.

3.1 Concurrent Microprocessors

3.1.1 Transputer Networks

The transputer was introduced by INMOS as a self-contained microprocessor with support for multiprogramming which can be used as a building block of parallel processing systems based on the message-passing protocol. Transputers were primarily designed to execute the parallel language OCCAM [22] and, even though many other compilers have become commercially available, OCCAM is still the language which commonly allows the best performance. Although the transputer family currently includes many other members, we shall be restricted to the T400 (a more economic version of the T414) and the popular T800. The T400 is a 32-bit machine which includes two communication serial channels and 4Kbytes of internal memory; the T800 is basically a T400 which includes two additional channels, 2Kbytes of additional internal RAM and a 64-bit floating point unity.

In the specific case of Hough transform, the effectiveness of parallel implementations in transputer networks is favoured by the fact that the HT's input, ie the edge elements, can be processed in any order and consequently the image edge-elements can be distributed amongst a given number P of processing elements. Assuming uniform input distribution, the overall HT execution time becomes in great part determined by the intercommunications between distinct processing elements, which is by its turn determined by the adopted interconnection topology. Although the edge-elements could be divided between the processing elements by completely disregarding their position in the image, the option of partitioning the image in uniform segments, each to be assigned to a specific processing element, is henceforth adopted in order to allow the Hough space to be also conveniently partitioned (see [23]).

Transputer-based implementation of the BIIT have been successfully applied to several actual situations such as [24, 25, 26, 27]. Results of experimental performance assessments in transputer networks (either T400 or T800 since no products or real arithmetic are involved) are presented in Figure 1.

3.1.2 Digital Signal Processors – The TMS320C40

The TMS320C40 is currently the most recent member of the Texas Instruments TMS320 digital signal processor family and is amongst the highest performance 32-bit microprocessors available in the market, operating at 275 MOPS and transferring data at a rate of

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

Table 1: Execution times and speed-ups for the BHT in 1 and 4 transputers (for 128 samples of the slope parameter); N_{ife} = amount of processed image feature elements.

N_{ife}	1xT800	4xT800	speed-up
64x64	3.47 s	909 ms	3.82
32x64	1.75 s	459 ms	3.81
16x64	895 ms	254 ms	3.52
8x64	467 ms	137 ms	3.40

320 Mbytes/s with a 40ns cycle time [28, 29]. Its principal features include six parallel bidirectional ports, each provided with a respective DMA channel for interprocess communications; global and local buses (100 Mbytes/s each) for data and program fetches; a 40-bit floating-point/integer multiplier ALU, a 32-bit barrel shifter; program cache (512 bytes) and Harvard architecture. Single precision (32 bits) and extended precision (40-bits) floating-point products as well as 32-bit integer multiplications can all be executed in a single cycle. Standard TMS320C40 packages present 325 pins.

The BHT can be effectively implemented in the TMS320C40 by making use of the barrel shifters and DSP-typical add-accumulate and loop control mechanisms. The BHT regularity and underlying data flow ensures that the instruction pipelining in the TMS320C40 will perform effectively. Message-passing-network implementations analogous to those described for the transputer can also be applied with the benefit that a small overhead will be implied due to the faster inter-processors communication allowed by the parallel channels, although at the cost of compounding the dynamic reconfiguration of the topology. Despite the fact that the TMS320C40, as well as most DSPs, can perform floating-point products at the same speed as integer additions, the BHT remains an interesting alternative compared to other standard, multiplication-based HTs because of its adequacy for fully accurate calculations of the intercept parameters.

The TMS320C40 is currently being considered as the DSP-core for a high-performance subsystem for image processing and computer vision, to be integrated into a larger system intended for general applications in visual inspection and computer vision. This project is being jointly formulated by the IFQSC-USP, Dept. Electrical Eng.-USP and ICMSC-USP.

3.1.3 Parallel Image Processors – The MAPP2200

Being the first device ever to incorporate a CCD sensor-array and a parallel image processor into the same chip (a 68-lead LCC package with a glass window), the IVP's¹ – 'smart optical sensor' MAPP2200 can be mounted in a camera in the same way as a standard CCD chip but with the bonus of delivering versatility and processing power [30]. The MAPP2200 includes a 256x256 CCD array, 114 digital 256-bit registers, a 256-bit analog register and three pixel processing unities (point, neighborhood and global logical unities) – operation occurs in a SIMD fashion over all the 256 bi-operands of the accumulator or digital registers). The MAPP2200 can be straightforwardly interfaced to a 16- or 32-bit supervisory microprocessor host through a 16-bit bidirectional bus; the instructions to be executed in the MAPP2200 must be scanned by the host over the MAPP2200. Most instructions are executed in 250 ns.

¹Integrated Vision Products, Sweden.

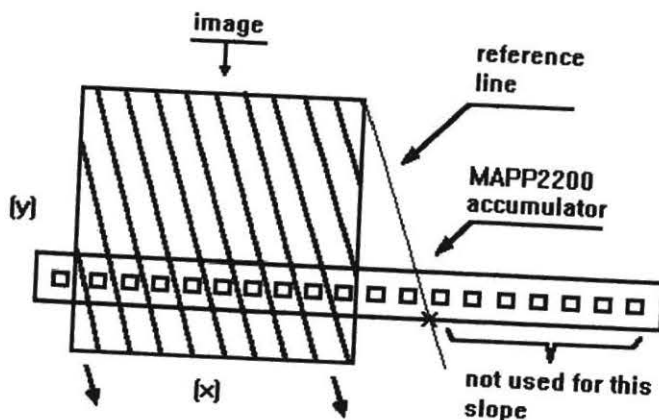


Figure 1: The strategy for implementing the BITT in the MAPP2200.

Since 256-bit registers are the standard data element in the MAPP2200, implementations of BHT algorithms which handles the image edge elements individually are not likely to yield good performance. The alternative currently adopted in our project consists in exploring the partitioning of the image space implied by the BITT formulation: we have already seen in section 2 how a family of digital bars is defined in the image respective to vertically aligned parameter space points (see also [8]). Thus, by taking the lowest digital line as reference, it is possible to process the whole image for each slope by counting into the MAPP2200 accumulator all the edge elements which falls within each of the digital bars. The reference digital bar is generated by the host through the BITT and, since the intercept resolution is fixed to be one, each of the accumulator bits can be associated to a digital bar in the image (such a strategy implies that the image segments size should be limited to half of the accumulator length). Such a processing strategy, which relies strongly on the register shift instructions of the MAPP2200, is illustrated in Figure 1. The image has to be fully scanned for each considered bar slope. Processing of slopes with absolute values larger than 1 can be achieved simply by exchanging the x - and y -coordinates of all the image feature elements and re-applying the discussed strategy.

Since 8-bit additions between two data (i.e. two complete 256-byte image rows) is performed bit-serially in the MAPP2200, a total of about 25 cycles are required by this operation. We are currently developing a simulator for the MAPP2200 in Pascal which will be used to assess the performance of the above described strategy for the BHT implementation in this processor.

3.2 Dedicated Parallel Implementations

The following subsection presents the systolic structures that are the basis for the dedicated alternatives to be addressed in the remaining subsections.

3.2.1 Systolic Architectures for the Binary Hough Transform

The simplicity and the spatial and temporal regularity of the BHT processing suits particularly well to pipelined/parallel (ie systolic) execution. Three main systolic designs have been proposed for BHT implementation which are respectively based on the following three underlying processing strategies: (a) progressive calculation of the products through accumulate additions; (b) binary decomposition of the operands and (c) combination of these two strategies [31, 32, 33, 3]. Such systolic structures are illustrated in Figure 2 for $N_M = 4$, where / and . respectively indicate right and left binary shifts and the triangles represent delay-elements (or latches). The additions with $N - 1$ and U are required in order to avoid negative values during the fixed-point calculation.

Each of these systolic structures presents its relative advantages and disadvantages: the first structure implies the greatest number of delay-elements, and consequently the longest execution time but does not require any signal line-crossing (which compounds VLSI designs); the second alternative demands fewer delay-elements at the expense of a number of signal line-crossings; an interesting balance between these parameters is achieved in the third design, which has the further advantage of being inherently modular (the basic module is a systolic structure of the second type, which is cascaded according to the underlying principle of the first structure). Features which are common to these three systolic designs are presented in the following:

Speed – Assuming the conservative hypothesis that each of the basic operations in the BHT bit-level systolic architectures can be performed in about 10ns, a whole 1732×1732 image can be processed in just 30ms;

Simplicity – Circuits are required only for integer addition and latching, with no need for tables or products (the binary shifts can be straightforwardly implemented by wiring);

Regularity – Each of the three topologies presents substantial inherent data flow and control regularity, which favours dedicated implementations;

Efficiency of hardware resources utilization – It can be easily verified that any of the systolic designs presented in this paper allow full efficiency for hardware resources utilization [34], which commonly implies faster and more economic operation;

Input – The systolic structures require their input, i.e. the image feature elements, to be fed in a bit-stream fashion, which is easily achieved by using simple systolic edge-detectors (see [35]), which can be directly connected to the bit-stream output of standard cameras.

It should be observed that additional circuitry needed for implementing the accumulator array updating is also required, which can be straightforwardly obtained by using counters and memory modules and will not be considered in the present paper due to space limitation (for more detail see [35, 3]).

Any of the three basic systolic structures presented above assume that the data is processed in words ('word-parallel'). Bit-serial and bit-level versions of these structures are also feasible and can be straightforwardly obtained. Bit-serial designs are particularly suitable for economic, non real-time implementations; bit-level structures, which are obtained by performing each addition in bit-pipelining (fine-grain concurrency), consist in effective ways of achieving high execution throughputs [33, 3]. Such systolic BHT structures have proved to be amongst the highest performance and most effective dedicated concurrent hardware alternatives for Hough transform execution [3].

The BHT systolic architectures were formally validated with respect to logical operation and synchronization by exploiting the transputer multiprogramming capabilities: each processing element (or part of it) was assigned to an OCCAM process and the physical

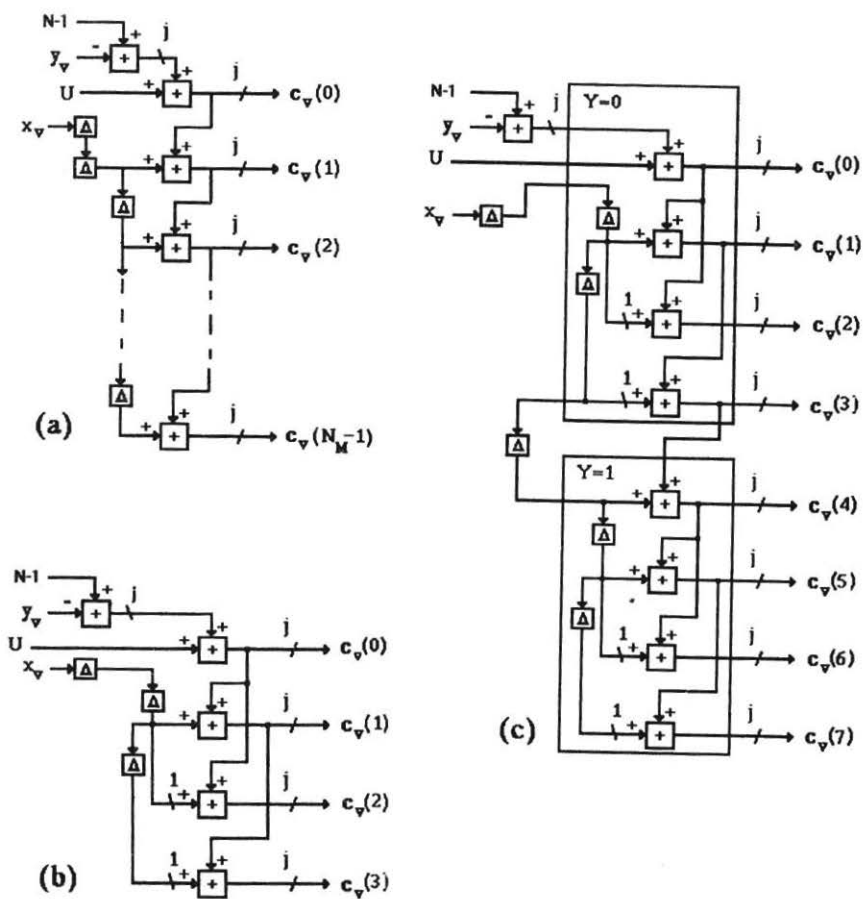


Figure 2: The three basic BHT systolic structures respectively based on progressive addition (a), binary decomposition (b) and combination of these strategies (c); $l' = 2^{j-1} - 1$.

	Word-level	bit-serial
number of columns	2	1
rows per column	11	9
stages per row	94	121
array area	3.44 mm ²	1.61mm ²
clock period	10ns	30ns

interconnections between these elements were implemented as channels. Though the employed version of the OCCAM-2 did not allow variable PAR control, it was possible to verify the correctness of the proposed designs [3].

3.2.2 Discrete Digital Circuits

Though this certainly corresponds to the most cumbersome alternative, it is also the less expensive and most versatile dedicated implementation strategy, at least in the light of our currently available technological capabilities (no commercially available EPLD suitable for BHT systolic implementation has so far been identified by the author). Having decided by the TTL technology, we tried to design a basic module which exploited as well as possible the available TTL devices. It turned out that one of the most suitable options was to use four-bit adders in such a way that the additions are serially executed in chunks of 4-bits. Such a strategy can thus be classified as intermediate between bit-level and word-parallel BHT systolic structures. The current design is based on a dual output (i.e. implement two of the outputs $C_r(i)$ in expression 1) module, composed of six low- and medium-scale integration TTLs. Such modules can be cascaded in such a way as to obtain the aimed slope resolution. The execution rate of such a structure, primarily intended to be interfaced to an IBM-PC compatible machine, is expected to be about 50MHz for 'LS' TTL devices.

3.2.3 Digital VLSI Implementations

Digital VLSI integrated circuits dedicated to word-parallel and bit-serial calculations versions of the above discussed second (and consequently a module for the third structure) architectural option have been designed by using the SOLO 1200, a software package for designing of custom integrated circuits, and fully simulated [36, 37]. The basic BHT operator, i.e. the adder, has been designed using the SOLO 1200 'draft' schematic entry. Given that the available total IC area and amount of input/output signals (pins) were respectively limited to 27.4 mm² and 84 pins, the word-parallel version had to be organized into bit plane modules (three bits per module). The principal obtained features for the word-parallel and bit-serial designs (for eight outputs) are presented in table 3.2.3. As expected, the word-parallel architectures allowed a smaller period of clock at the expense of a larger IC area. Also, it should be observed that the obtained used area falls short of the total available area, which means that larger bit-serial structures can be easily implemented into a single chip: larger word-parallel modules are limited by the amount of available pins.

3.2.4 Multiple-Valued Logic VLSI Implementations

It is quite well known amongst the computer science community that digital circuits and computers were introduced and have been thriving as means of obtaining improved computation reliability compared to that allowed by analogic technology. What is commonly

Table 2: Principal features of the considered concurrent hardware alternatives (1 – processor only; 2 – approximated, for 20MHz devices in batches of 1000; 3 – for 16 outputs; 4 – 40MHz devices; G/S = general/semi-general; D = dedicated; T_b = approximated execution time for the addition of two binary words (for single processing element).

modelo	T_b	type	cost
T400/T800	50ns	G/S	US\$42/US\$220 ^{1,2}
TMS320C40	25ns	G/S	US\$360 ^{1,4}
MAPP2200	20ns	G/S	medium
discrete	20ns	D	US\$50 ³
digital VLSI	10ns	D	high
MVL-VLSI	2.5ns	D	high

less known is that analogic, and particularly analogic-digital hybrid circuits are nowadays making a come-back as interesting alternatives for VLSI implementations. One of the most representative of such tendencies, and the one considered in this paper, is multiple-valued logic – MVL – VLSI, which presents potential for reducing the interconnection complexity and the amount of elementary devices thus yielding more compact and faster VLSI integrated circuits. Although alternative strategies are possible, this paper is restricted to LSI multiple-valued CMOS current-mode circuits based on the radix-4 signed-digit – SD – numeric representation system [38, 39, 40]. One of the most remarkable inherent features allowed by such circuits is that additions can be straightforwardly performed through simple wiring according to Kirchhoff's current law [39, 40].

A possible current-mode circuit for addition of two radix-4 SD values A and B giving C is presented in Figure 3(a) [39, 40], where the triangle represents an n-channel current mirror; BDCI and TD respectively stand for bi-directional current input and threshold detector and BTD is a TD provided with the buffering circuitry required for synchronization (via clock) of the pipelined operation. A remarkable property of such a full adder, which requires 26 transistors, is that its execution time is independent of the word length [39, 40]. Since a single full-adder is obviously not enough for most applications, it becomes necessary to cascade them in order to process longer words, as illustrated in Figure 3(b), where each 'IQ' means an 'inverted quantizer' which also contains 26 transistors. For the typical situation considered in [40], a non-pipelined SDFA with the inverted quantizer execute in about 11ns. The pipelined SDFA presented in Figure 3 allows a reduction of about 50% of the execution time, consequently resulting in a quite effective basic operator for implementation of the BHT systolic structures. It is easy to verify that suitable BHT MVL systolic structures can be obtained from their digital equivalents (Figure 2) except for the fact that 1-bit, instead of 1-bit, outputs are produced by each adder. Such MVL BHT structures thus not only allow a reduction of the basic clock, but also a substantial performance improvement due to the production of parallel outputs corresponding to 2 bits (or four values).

4 Comparing the Alternatives

Table 2 presents the principal features of the concurrent hardware alternatives for BHT implementation discussed in the present paper.

As could be expected, the dedicated designs resulted faster at the expense of less ver-

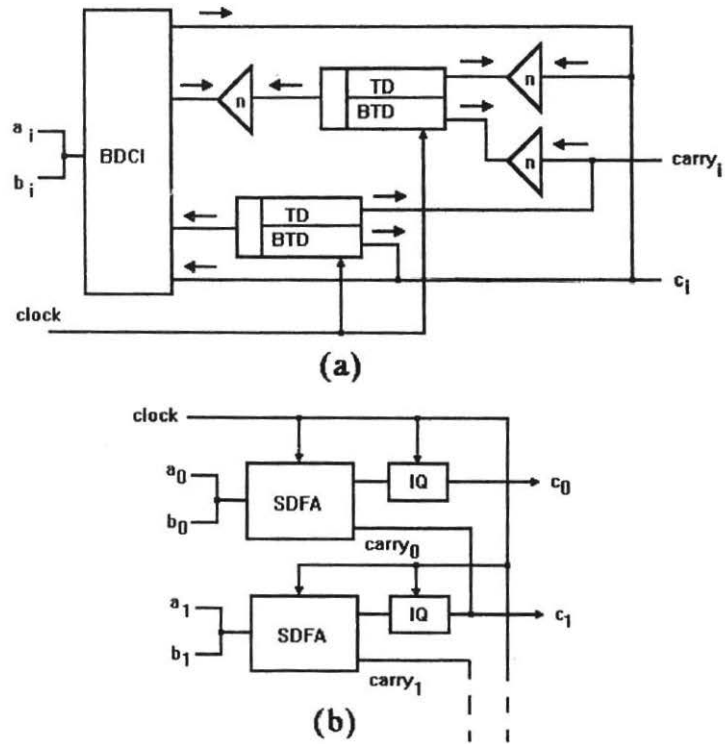


Figure 3: The radix-4 signed-digit full adder - SDFA (a) and its cascade combination in order to allow the addition of longer words (b).

satility. On the other hand, the slower concurrent microprocessors can also be used to implement other operations (such as edge-detection, thinning, merging and connectedness analysis) required by the overall intermediate-level vision framework. In fact, the choice of one of the above alternatives will in great part depend on the application requirements and the aimed overall cost. As already observed, the option involving discrete digital circuits (TTLs) allows the lowest cost and yet the best flexibility and one of the fastest execution rates. The principal shortcoming of such an approach is the required physical size, which can be attenuated by using small-size TTL packages. The transputer and TMS320C40 are interesting alternatives for building versatile systems and for post-HT stages in the intended intermediate-level vision framework. The MAPP2200 is an innovative device whose main feature is the compaction of the circuitry: it would be quite interesting to have an MAPP2200-based camera able to perform most of the image pre-processing and the BHT speedily. The MVL-VLSI is one of the most exciting alternatives for the BHT implementation which presents potential for good integration densities and excellent execution rates. We are currently collaborating with the Dept. of Electronic Engineering, Tohoku University (see acknowledgements), towards a MVL-VLSI BHT design. The principal obstacles to be overcome are the formal assessment of pipelined execution of the MVL full adder and the production aspects.

5 Concluding Remarks

This paper has presented recent results concerning the consideration of several representative concurrent hardware alternatives for the implementation of the binary Hough transform, an effective technique for digital bar detection. Three concurrent microprocessors (transputers, TMS320C40 and MAPP2200) and three dedicated concurrent structures (discrete digital circuits, digital VLSI and MVL-VLSI) have been presented and its advantages and shortcomings for BHT implementation identified. It was verified that suitable BHT implementation strategies can be obtained for all the addressed concurrent hardware alternatives; the dedicated options resulting particularly interesting as means of achieving high execution rates at affordable costs.

The results obtained in this paper can be extrapolated to predict the suitability of the considered hardware alternatives for the implementation of other image processing and computer vision techniques which present similar data flow and control structure. In fact, the HT, in addition to its popularity as an effective pattern recognition technique, has also been established as a benchmarking algorithm.

6 Acknowledgements

The developments reported in this paper have only been possible through the collaboration and support from several people and institutions. The 'Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP' financed part of the work reported here through processes 87/1741-8 and 82/3848-2; special thanks go to Dr. T. Higuchi and Dr. T. Aoki (Tohoku University, Japan) for having kindly collaborated in the MVL BHT design and for his suggestions about MVL-pipelining (Figure 3), Prof. T. Forchheimer (IVP, Sweden) who supplied documentation on the MAPP2200; Mr. Panagiotis Tzionas (Greece), who designed and simulated the BHT digital VLSI integrated circuit during his M.Sc. programme at King's College London and to Mr. Marcelo Tozin and Mr. Marcos F. Inoue who are currently

working respectively on the MAPP2200 and discrete circuit implementations of the binary Hough transform.

References

- [1] H. Maitre. Un panorama de la transformation de Hough. *Traitement du Signal*, 2(4):305–317, 1985.
- [2] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.
- [3] L. da F. Costa. *Effective Detection of Line Segments with Hough Transform*. PhD thesis, King's College, University of London, London, UK, May 1992.
- [4] P. V. C. Hough. Method and means for recognizing complex patterns. *United States Patent Office*, Dec. 1962. Patent 3,069654.
- [5] G. C. Stockman and A. K. Agrawala. Equivalence of Hough curve detection to template matching. *Communications of the ACM*, 20(11):820–822, Nov. 1977.
- [6] S. R. Deans. Hough transform from the Radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(2):185–188, Mar. 1981.
- [7] G. Hall, T. J. Terrell, and L. M. Murphy. Implementation of the Radon transform using a dynamically switched transputer network. In D. J. Pritchard and C. J. Scott, editors, *Transputer Applications 90*, pages 156–163. IOS Press, 1990.
- [8] J. L. C. Sanz and I. Dinstein. Projection-based geometrical feature extraction for computer vision: algorithms in pipeline architectures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):160–168, Jan. 1987.
- [9] L. da F. Costa and M. B. Sandler. Effective detection of straight-line segments with Hough transform. *Computer Vision, Graphics, and Image Processing*, 1992. Accepted for publication.
- [10] P. H. Lindsay and D. A. Norman. *An Introduction to Psychology*. Harcourt Brace Jovanovich, 1977.
- [11] D. H. Ballard, G. E. Hinton, and T. J. Sejnowski. Parallel visual computation. *Nature*, 306:21–26, Nov. 1983.
- [12] G. Blasdel. Orientation selectivity, preference, and continuity in monkey striate cortex. *The Journal of Neuroscience*, 12(8):3139–3161, Aug. 1992.
- [13] Luciano da F. Costa. On visual cortex and hough transforms. *Psychology*, 1993. Submitted.
- [14] T. O. Binford. Survey of model-based image analysis systems. *Artificial Intelligence*, 17:205–244, 1981.
- [15] D. Marr. *Vision*. W. H. Freeman, 1982.

-
- [16] L. da F. Costa. Towards a versatile framework for intermediate-level computer vision. In *Proc. IAPR International Workshop on Machine Vision Applications 1992*, pages 261 – 264, Tokyo, Japan, Dec. 1992.
- [17] M. D. McIlroy. A note on discrete representation of lines. *AT & T Technical Journal*, 64(2):481–490, Feb. 1984.
- [18] L. da F. Costa and M. B. Sandler. Performance improvements and performance evaluation of the binary Hough transform. In *Proc. V European Signal Processing Conference*, Barcelona, Spain, Sep. 1990.
- [19] L. da F. Costa and M. B. Sandler. Detecting straight line segments in $O(n^2)$. In *International Workshop on Visual Form*, pages 165 – 174, Capri, Italy, May 1991.
- [20] G. Gerig and F. Klein. Fast contour identification through efficient Hough transform and simplified interpretation strategy. In *8th International Joint Conference on Pattern Recognition*, pages 498–500, Paris, France, 1986.
- [21] M. G. Albanesi. Architectures for the Hough transform: A survey. Technical Report RIDIS-39-90, Dipartimento di Informatica e Sistemistica, Università degli Studi di Pavia, Pavia, Italy, May 1990.
- [22] INMOS. *Transputer Reference Manual*. Prentice-Hall, 1988.
- [23] D. Ben-Tzvi, A. Naqvi, and M. Sandler. Synchronous multiprocessor implementation of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 52:437–446, 1990.
- [24] L. da F. Costa, X. Leng, M. B. Sandler, and P. Smart. A system for semi-automated analysis of clay samples. *Review of Scientific Instruments*, 62:2163–2166, Sep. 1991.
- [25] L. da F. Costa, D. R. Andrews, and M. B. Sandler. Quality control of ultra-sound transducers with the binary Hough transform. In *19th International Symposium on Acoustical Imaging*, Bochum, Germany, 1991.
- [26] L. da F. Costa, M. B. Sandler, and S. Velastin. Applying image analysis techniques to optimize the production of cork stoppers. In *Proc. SIBGRAPI'92*, pages 13–16, LINDOIA, Brazil, Nov. 1992.
- [27] L. da F. Costa and M. B. Sandler. Application of the binary Hough transform to image compression. In *6th International Conference on Digital Processing of Signals in Communications*, pages 56–60, Loughborough, UK, Sep. 1991.
- [28] R. Simar, P. Koeppen, J. Leach, S. Marshall, D. Francis, G. Mekras, J. Rosenstrach, and S. Anderson. Floating-point processors join forces in parallel processing architectures. *IEEE Micro*, pages 60–69, Aug. 1992.
- [29] Texas Instruments Inc. *TMS320C4X User's guide*, 1991.
- [30] A. Astrom and R. Forchheimer. Mapp2200 smart vision sensor. programmability and adaptivity. In *MVA '92 – IAPR Workshop on Machine Vision Applications*, pages 17–20, Tokyo, Japan, Dec. 1992.

-
- [31] L. da F. Costa and M. B. Sandler. A binary Hough transform and its efficient implementation in a systolic array architecture. *Pattern Recognition Letters*, 10(5):329–334, Nov. 1989.
 - [32] L. da F. Costa and M. B. Sandler. The binary Hough transform and its implementation. In *Proc. 1990 SPIE/SPSE Symposium on Electronic Imaging Science and Technology – Curves and Surfaces in Computer Vision and Graphics*, pages 183–193, Sta. Clara, USA, Feb. 1990. Paper No. 1251-21.
 - [33] L. da F. Costa and M. B. Sandler. Multiple-output multipliers for computer vision and digital signal processing. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 2633–2636, Singapore, 1991.
 - [34] L. da F. Costa and J. F. W. Slaets. On the efficiency of parallel pipelined architectures. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 39(9):2086–2089, 1991.
 - [35] L. da F. Costa and M. B. Sandler. A complete and efficient real time system for line segment detection based on the binary Hough transform. In *Proc. Euromicro '90 Workshop on Real Time*, pages 205–213, Horsholm, Denmark, Jun. 1990.
 - [36] Tzionas Panagiotis. VLSI implementation of the binary Hough transform. Master's thesis, Department of Electronic and Electrical Engineering, King's College London, University of London, London, UK, Sep. 1989.
 - [37] L. da F. Costa, P. Tzionas, and M. B. Sandler. On the VLSI implementation of the binary Hough transform. *IEE Colloquium Digest*, (1990/95):1–4, May 1990.
 - [38] A. Avizienis. Signed-digit number representations for fast parallel arithmetic. *IRE Transactions on Electronic Computers*, pages 389–400, Sep. 1961.
 - [39] M. Kameyama, S. Kawahito, and T. Higuchi. A multiplier chip with multiple-valued bidirectional current-mode logic circuits. *IEEE Computer*, pages 837–845, Apr. 1988.
 - [40] S. Kawahito, M. Kameyama, T. Higuchi, and H. Yamada. A 32x32-bit multiplier using multiple-valued mos current-mode circuits. *IEEE Journal of Solid-State Circuits*, 23(1):124–132, Feb. 1988.