MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

Real-time Hough Transform Based Circular Shape Extraction

Mahmoud MERIBOUT, Takeshi OGURA and Mamoru NAKANISHI
NTT LSI Laboratories
email: meribout@aecl.ntt.jp

Abstract

In this paper, two circular shape extraction methods based on CAM (Content Addressable Memory) concept are proposed. The first method based on Hough Transform uses only the gradient amplitude information, while the second uses in addition preselected gradient direction to estimate the center coordinates of the circle and then apply the HT circle extraction to extract it's radius. The advantages of both our propositions consist of their performance to keep low both the hardware amount, and the computation time for extraction of unknown, noisy circular shapes.

1 Introduction

In spite of recent developments in VLSI technology, image processing applications have been slow to emerge because the real-time processing requirement is very costly. This is especially true for high and intermediate image processing levels, such as feature extraction.

One of the most efficient ways to extract features is to use the Hough transform(HT). However, this requires a large computation time and capacity storage, both of which increase with the complexity of the shape to be extracted (3D and 5D parameter space for respectively circle and ellipse extraction). Therefore, many parallel HT implementations have been reported in the literature [1][2][3]. Unfortunately, these architectures generally requires a large amount hardware, moreover, they are unsuitable for real-time HT-based algorithms, especially for complex curve extraction. This is mainly due to the bottleneck associated with memory access.

This paper shows how better performance, using inexpensive hardware, can be obtained, by means of an enhanced memory access techniques called Content Addressable Memory (CAM). Results of an evaluation of circular shape extraction using the CAM-based HT algorithms show the effectiveness of the CAM technique.

Successful real-time CAM-based HT line extraction has already been proposed and evaluated successfully by part of authors [4]. Moreover, circular shapes extraction is also important for realizing practical vision applications.

Specifically, this paper describes and evaluates two parallel circular shape extraction algorithms : circular shape extraction CAM-based HT that uses only gradient magnitude ; and CAM-based HT, using gradient direction information. As a result, real time requirements with low hardware amount are achieved since both voting and 3D peak extraction, which mainly compose the proposed HT algorithms, are efficiently executed within the CAM.

2 Hough Transform Algorithm on CAM-based System

Both our proposed algorithms stress our CAM technology implementation in a Highly Parallel Image Processing architecture (HiPIC) [5].

2.1 CAM-Based System Hardware Architecture

The hardware architecture of the CAM-based system (Fig 1) is based on the HiPIC concept (Highly-parallel Integrated Circuits and Systems). This is a modular architecture comprising :

- a preprocessing module for low level image processing,
- an FPGA block for control,
- a RISC/DSP processor for system control and serial data processing,
- frame/data memory.

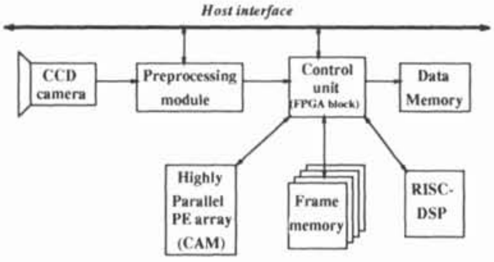


Figure 1. HiPIC: Highly-parallel Integrated Circuits and System.

Beside it's flexibility for supporting different image processing applications, such as line or segment extraction and 3D reconstruction; this architecture is robust due to the fact that the CAM is a SIMD-like architecture in which a high number of PEs perform simple operations such as inc, add, data shift, and data comp in parallel at high speed. These features make our proposed hardware implementation for circular HT more efficient than other architectures.

2.2 HT algorithm implementation

Each CAM word corresponds to a quantization point in Hough space and acts as a processor element (PE), a decision circuit, and accumulator (Fig 2). The value computed by the PE is stored in the decision circuit, which decides whether voting to the accumulator must be done or not after search operations.

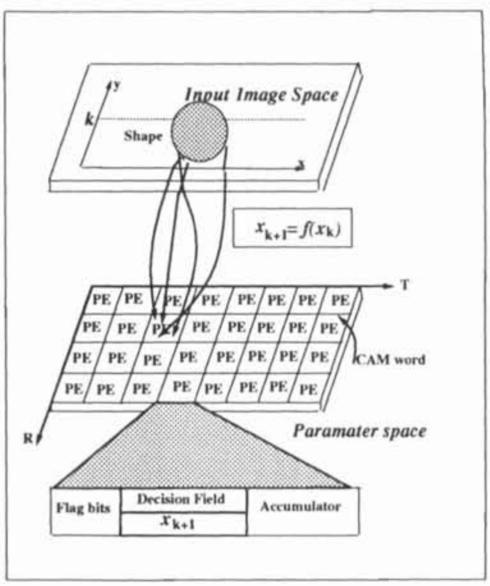


Figure 2. Hough Transform: Implementation on the CAM

The flow of these algorithms can be described as following :

- *For each line $k+1$ in the image space, the decision field is updated for each CAM word according to the following recursive equation :*

$$x_{k+1} = f(x_k), \tag{1}$$

  where f depends on the shape to be extracted, and x_k is a function of the edge pixel in line line k.

- *This updating is followed by CAM-cell voting using coordinates of the edge pixels in line $k+1$.*
- *After the whole image is scanned, both thresholding and maximal peak extraction are achieved simultaneously for each accumulator array.*

3 Proposed Algorithms

3.1 Circle Detection CAM-Based HT

The cartesian equation of a circle shows that 3 parameters (a, b, c) are required:

$$(x-a)^2 + (y-b)^2 = c^2 \tag{2}$$

Where (x, y) are the image coordinates, (a, b) the coordinates of the center of the circle, and c it's radius.

In order to avoid multiplication operations, the equation computed by each PE, for an image $N \times N$ is modified from eq.(2) to the linear and recursive equation (3):

$$x_a(y) = (x-a)^2 + y^2 = x_0 + 2b \times y. \tag{3}$$

$$x_0 = c^2 - b^2,$$

$$\text{where}: (a, b, c) \in N \times N \times N$$

The original 3D parameter space is subdivided into $(\frac{N}{4}) \times (2D)$ sub-parameter spaces (b, c). For each sub-parameter space, the following steps are performed :

- *The terms b and x_0 in (eq.3) are loaded in the decision field of each CAM word whose address is formed by the concatenation of parameters b and c,*
- *The whole image is scanned, and for each edge point belonging to line y, parallel search of $X_{a_{(4\times i)+k}}(y)$ (where $k \in [0, 3]$, and $i \in [0, \frac{N}{4}]$) is performed. At this step, and because of the quantization errors, the searching mask is not fixed but shifted to the left by $|x-a|$ bits in the decision field.*

- *At the end of the line, the voting is carried out in parallel for each word. This is followed by updating the decision field corresponding to the next scan line.*

- *After the whole image is scanned, all CAM words whose corresponding accumulator is above a threshold $th = f(c)$ (where f is a function depending on the radius c) are found.*

- *Global 3D peak extraction of three successive a values (a_{i+1}, a_{i+2} and a_{i+3} cell arrays in Fig 3) is performed by the CAM. This process consists of maximal extraction followed by resetting the accumulators of the surrounding 26 neighboring CAM words.*

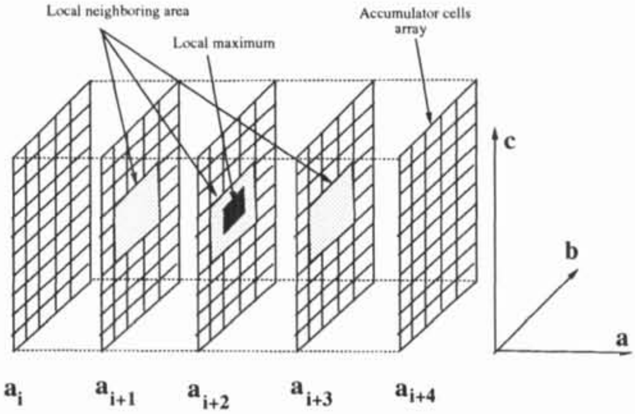


Figure 3. Global 3D peak extraction,

Since a circular shape intercepts an horizontal line at a maximum of two pixels, two voting levels are achieved. The first voting level is performed for each edge pixel and uses a 2 bits wide buffer. At the end of the scan line, the amount of this buffer is added to the accumulator field during the second level voting step. This has the merit of limiting the number of long sequence bit serial operations required to increment the accumulator field.

As a result of the above proposition, the computation time is short, and proportional to $O(\frac{N^4}{A})$, where A is the number of CAM words. However, both the hardware amount and the processing time grow with the image size. This is mainly due to the delay caused during $a - space$ scanning.

One alternative for further reducing the computation time is to subsample the input image, and consequently the parameter space, by a factor K as a preliminary mechanism to focus attention on candidate circle parameters. Exact circle parameters are then extracted using the above described method. In this case, the computation time is proportional to $O(\frac{N^4}{A \times K^3})$. Section 4 shows the efficiency of this method.

A second alternative consists to use the gradient direction of each edge pixel to estimate the circle center coordinates (a, b). This is described next.

3.2 Circle Detection CAM-Based HT using Gradient Direction

In this method, the computation time of the above method is reduced by taking advantage of useful image information: the gradient direction. Differentiating eq.(2) with respect to x, we get:

$$dx \neq 0 \Rightarrow x(y) = a - (y - b) \times \frac{dy}{dx} \quad (4)$$

$$else \qquad y = b \quad (5)$$

Therefore, if θ denotes the gradient angle at (x, y), then the value of $\frac{dy}{dx}$ is either $tg(\theta + \frac{\pi}{2})$ or $tg(\theta - \frac{\pi}{2})$.

The flow of the algorithm is as follows:

- *For each edge pixel belonging to line y, equs.(5) and (6) are computed using a look-up table for multiplication operation. Voting task is then performed,*

- *After the whole image is scanned, thresholding and maximal search are carried out to extract a set A of center circle coordinates,*

- *Using eq.(3), a second voting is performed, by evaluating for each boundary point a set of parameters (a, b, c) that possibly pass through A.*

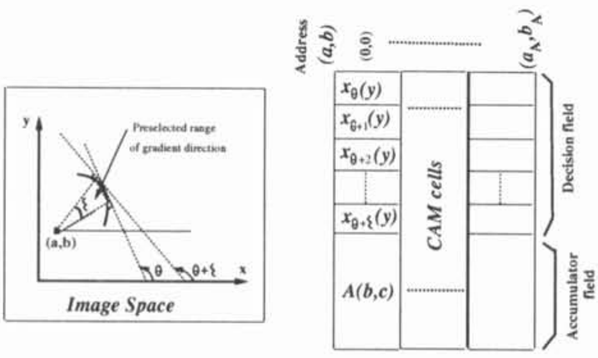
It's obvious that $eq.4$ is very costly. This is why only a limited range of gradient direction $s[\theta, \theta + \varepsilon]$ is selected in step 1 of the algorithm (Fig. 4(a)).

For each scan line $y + 1$, the decision field is updated according to the equation :

$$x(y+1) = x(y) - \frac{dy}{dx} \quad (6)$$

$$x(0) = a + \frac{dy}{dx} \times b$$

Figure 4(b) shows the corresponding CAM word configuration.



-a- -b-

Figure 4. Circle detection based HT using gradient girection

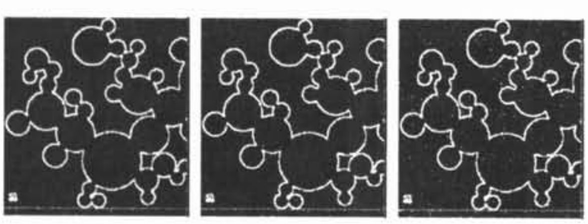
Compared to the first algorithm, the complexity is reduced from $O(\frac{N^4}{A \times K^3})$ to $O(\frac{N^3 \times B}{A \times K^3})$, where B is the number of selected gradient direction intervals.

4 Simulation Study

In this section, both the accuracy and hardware/processing time performances of our algorithms are discussed. These simulation results are evaluated using a CAM-based simulator coded in C-language.

4.1 Circle Extraction Performance

To evaluate their accuracy, three input images with different $\frac{S}{N}$ ratio were generated (Fig.5).



-a- -b- -c-

Figure 5. Three noisy input images

Table 1 shows the hit ratios obtained using the two proposed CAM-based HT algorithms. For this test, various buffer sizes (BF=1 bit or BF=2 bits) of the first level voting and subsampling sizes K were used (see sec.3.1) for the first algorithm. In the second, three range of gradient direction were used: $\theta \in [0, \varepsilon]$, $\theta \in [\frac{\pi}{2}, \frac{\pi}{2} + \varepsilon]$, and $\theta \in [\frac{\pi}{4}, \frac{\pi}{4} + \varepsilon]$ ($\varepsilon \in \{1^0, 2^0, 3^0\}$).

		Input image					
		S/N=100%		S/N=80%		S/N=60%	
		Algorithm					
K	BF	1	2	1	2	1	2
1	2	100%	95%	97%	81%	79%	72%
1	1	100%	89%	83%	63%	64%	49%
2	2	100%	-	79%	-	51%	-
2	1	100%	-	68%	-	21%	-
4	2	100%	-	47%	-	-	-
4	1	100%	-	21%	-	-	-

Table 1. Hit ratio.

4.2 Processing time

The computation time for different image sizes was evaluated using a finite word length simulation(Fig.7). The following hypothesis were considered :

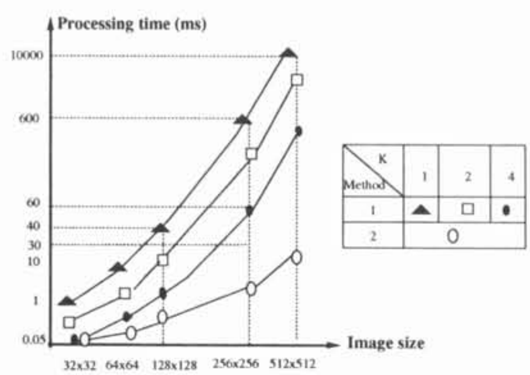


Figure 7. Processing time performance.

- One 40-Mhz VCAM chip of ($4kwords \times 72bits$) [5],
- Buffer size: $BF = 1$,
- Three ranges of gradient direction.

4.3 Discussion

As a result of using the CAM concept for HT, circular shape extraction is performed efficiently in an attractive computation time. In the future, better hit ratio performance can be obtained using weighted voting[4]. The first algorithm uses only the gradient magnitude, while the second adds the gradient direction information. This makes the later method more robust in accuracy results, hardware requirements, and extensibility for more complex curve detection. It's drawback is that some prior informations about the image are required to adequately select the range of gradient direction, otherwise, higher hardware amount will be required.

5 Conclusion and Future Work

In this study, an efficient implementation of CAM-based HT has been proposed. To confirm it's performance against previous propositions, two methods for circular shape extraction have been implemented and evaluated. Since these algorithms require a high number of simultaneous write and search operations, we showed that, by taking advantage of CAM features, both the hardware amount and execution time are reduced. This makes our architecture the most efficient for such applications. In the future, the implementation of these algorithms on our architecture will be achieved. Furthermore, and using the same architecture, more complex curves extraction is envisaged using the Generalized Hough Transform.

References

[1] A. Rosenfeld, J. Ornelas and Y. Hung, *"Hough Transform Algorithms for Mesh-connected SIMD parallel Processors"*, Computer Vision Graphics Imaging Process, Vol 41, pp. 293-305, 1988.

[2] C. Guera and S. Hambrusch, *"Parallel Algorithms for Line Detection on a Mesh"*, Journal of Parallel Distributed Computing, pp. 1-19, 1989.

[3] H.Y. Chuang and CC.Li, *"A Systolic Array for Straight Line Detection by Modified Hough Transform"*, In Proc. IEEE Workshop on Computer Architecture for Pattern Analysis and Image Database Management, pp. 30-304, 1985.

[4] M. Nakanishi, T. Ogura, *" A Real Time CAM-based Hough Transform Algorithm and Its Performance Evaluation"*, to appear in Proc ICPR, 1996.

[5] T. Ogura, M. Nakanishi, T. Baba, Y. Nakabayashi, R. Kasai, *"A 336-kbit Content Addressable Memory for Highly Parallel Image Processing"*, Proc. CICC, May, 1996.