



HAL
open science

A Real Time Hough Transform Architecture Useable inside a WCE

Orlando Chuquimia, Andrea Pinna, Xavier Dray, Bertrand Granado

► **To cite this version:**

Orlando Chuquimia, Andrea Pinna, Xavier Dray, Bertrand Granado. A Real Time Hough Transform Architecture Useable inside a WCE. 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), Oct 2019, Nara, Japan. pp.1-4, 10.1109/BIOCAS.2019.8919052 . hal-02495530

HAL Id: hal-02495530

<https://hal.science/hal-02495530>

Submitted on 2 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Real Time Hough Transform Architecture Useable inside a WCE

CHUQUIMIA Orlando¹, PINNA Andrea¹, DRAY Xavier², GRANADO Bertrand¹

Abstract—To reduce the incidence of colorectal cancer (CRC), we propose a new paradigm of Wireless Capsule Endoscopy (WCE) that recognizes polyps in-situ. We embed an image processing chain in a System on Chip (SoC) that uses the Hough Transform as part of the processing to detect circles in High Definition (HD, 1920x1080) images. A circle here is a probable marker of colorectal cancer ; a polyp. Hough Transform is a widely used shape-based algorithm for object detection and localization. This technique can be generalized to circles. To embed the Hough Transform inside a WCE, considering real time execution and a limited area, several optimizations are necessary due to computational requirements. This paper presents an efficient, real-time architecture that can be integrated in an 8x8mm² embedded system of a HT algorithm for multi-circle detection. The architecture's area has been validated in a FPGA Xilinx Spartan 7 XC7S15-CPGA196 packaged in an area of 8x8mm². This architecture can run at 135.46MHz on a 1920x1080 pixels image.

I. INTRODUCTION

95% of CRC cases begin with the presence of a growth, known as a polyp, on the inner lining of the colon or the rectum. Of all the types of polyps, solely adenoma polyps are considered pre-cancerous [1].

To reduce the incidence of CRC, in [2] and [3], we have proposed a new WCE that automatically detects polyp in situ. We have proposed a specific processing chain embedded in a SoC to be integrated in the capsule.

In our chain, we use the Hough Transform (HT), a widely used technique for object localization since 1962 [4]. This technique can be generalized to detect circles in the images. In our processing, parts of the image that contain circles are called Regions Of Interest (ROI) and may contain a polyp. We select these ROI and use a learning algorithm to decide if it contains a polyp.

We focused on HT due to the measurements made on an ARM Cortex A9, using OpenCV library, running at 667 MHz. These measurements show an execution time of 998.260 ms for an image of size 1920x1080 (HD image). This execution time is 25 times higher than our goal of 40 ms. This result excludes an optimized software embedded implementation and obliges us to design a digital hardware implementation in FPGA or ASIC.

In [5], we first realized a state of the art to analyze the implementation of the HT in a FPGA, considering timing

and area constraints. Then, we chose to implement a specific version of HT introduced by Goneid et al. [6] in 1997.

The Goneid algorithm can successfully extract non-overlapping circles in binary images, even when the circle has an irregular shape due to the presence of random noise. This algorithm is easy to implement since each of the object's parameters is accumulated in its own one-dimensional space [7].

This algorithm was chosen because polyps found in colon examination images do not always have a regular circular shape [8], due to noise levels, and quality and resolution of the image. Therefore, the Goneid algorithm is suitable for our purpose.

In this article, we propose an optimized architecture to implement this version of HT to determine if it can be embedded in a WCE. Real-time execution is required, capable of processing 25 images per second. It is also necessary to consider the limited area of the WCE. A WCE has a diameter of 11.6 mm, so it is mandatory to integrate the processing inside a chip with a diagonal length of the same value. As we are working on the next generation of WCE, we are also considering HD images.

II. STATE OF THE ART

To realize the Goneid algorithm, three histograms are constructed, H_x , H_y and H_r , to compute the x-, y-coordinates and the radius of the circle candidates.

TABLE I
HT HARDWARE IMPLEMENTATIONS.

Ref.	FPGA	BRAMs/DSPs	Image size	Latency[ms] / Freq.[MHz]
[7]	Virtex-7	5562Kb / 398	400x400	5.338 / 181.812
[5]	Artix 7	2826Kb / 153	1080x1920	27.81 / 149.16

In [7], the authors propose a FPGA-based architecture to implement the Goneid algorithm for 400x400 images. The results of their implementation are shown in Table I. Multiple BRAMs were used to store the histograms H_x , H_y and H_r computed by 100 voting modules in parallel. As their implementation does not include HD images, we have extrapolated the latency for an HD image using a simple size factor. We obtained a latency of 69.18ms. Therefore, it was concluded that this architecture can not be executed in real time for HD images.

In [5], we proposed a Real-Time HT Architecture (RT-HTA) to implement the Goneid algorithm that can be seen in Figure 1. It supports a maximum clock frequency of 149.16 MHz and can process 62 images per second. Thus, the real-time constraint has been reached. The results of our implementation are shown in Table I. The architecture computes the histograms H_x , H_y and H_r in parallel.

¹ LIP6, CNRS UMR 7606, Sorbonne Université, Paris, France.

² APHP, Hôpital Saint-Antoine, Sorbonne Université, Paris, France.

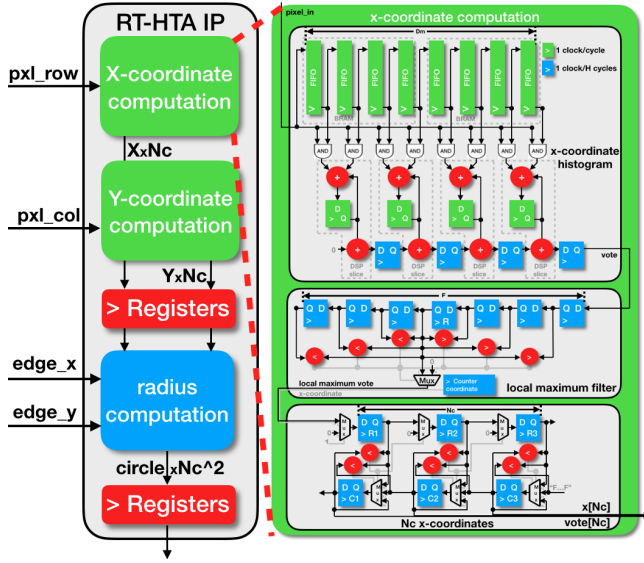


Fig. 1. Our digital architecture.

RT-HTA is composed of the five modules described below:

- **x- and y-coordinate computation:** these modules compute a histogram of the middle of the segments constituted by the points of contours. A histogram is built for the horizontal H_x and vertical H_y segments. The N_c higher local maximums of H_x and H_y become the N_c x-coordinates and N_c y-coordinates. N_c^2 center candidates are obtained from the combination of each x-coordinate and y-coordinate (see Figure 1);
- **radius computation:** this module builds, for each center candidate, a histogram H_r using Euclidean distance between the center candidate and each edge point. Once the histogram is built, this module assigns the most accumulated Euclidean distance as the radius. This module selects the center candidates as a real circle and the radius where the accumulation value is $> f * 4\sqrt{2}r$;
- **two registers module:** these modules register the N_c x-coordinates and the N_c y-coordinates of center candidates and the N_c^2 circles in parallel.

While we know that RT-HTA satisfies the real-time constraint, we need to determine if it can be implemented in a WCE. We choose to implement it in a FPGA Xilinx Spartan 7 XC7S15-CPGA196 packaged in $8 \times 8 \text{mm}^2$ surface area. This area implies a diagonal of 11.3 mm that is suitable for a WCE diameter of 11.6 mm. The results of the implementation are showed in Table II.

TABLE II
RT-HTA IMPLEMENTATION IN A XC7S15 FPGA.

Resource	Available	Utilization	Utilization %
LUT	8000	20637	258%
BRAM36K	10	78.5	785%
DSP	20	153	765%

From the results in Table II, we can observe that RT-HTA can not be integrated inside a $8 \times 8 \text{mm}^2$ FPGA without optimizations. We also notice that the BRAM and DSP resources

are the most used resources. After further analysis, it was determined that the x- and y-coordinate computation modules use the majority of the DSP and BRAM resources. Therefore, we have chosen to concentrate on the optimization of these two modules. To find how we can optimize them we have analyzed the centers candidates computation of Goneid algorithm. The analysis is described in the next section.

III. OPTIMIZATION OF CENTERS CANDIDATES COMPUTATION

The algorithm of Goneid is mainly composed of two parts:

- center candidates computation
- radius computation for each center candidate.

In [5], we have proposed the equation 1 to accelerate the x-coordinate computation. A similar equation was proposed for y-coordinate computation.

$$H_x\left(\frac{j+k}{2}\right) = \sum_{j=1}^{W-1} \sum_{k=j+1}^W \sum_{i=1}^H I_{(i,j)} * I_{(i,k)} \quad (1)$$

To obtain this equation, two changes were made to the original Goneid algorithm:

- All pixels in the image are considered, as opposed to solely the edge pixels as in the original Goneid algorithm.
- The summations order was changed, which can be seen in equation 1.

For x-coordinate computation, we have implemented the summations i and k of equation 1 in parallel. FIFO memories were used to shift the columns. Adders and registers were used to accumulate the votes between one column and the other two consecutive columns.

We used BRAM to implement the FIFO memories, DSP to implement the adders and Flip Flops to implement the registers.

The advantage of our architecture is that one accumulation value for a coordinate x of histogram H_x is obtained each time a column is read.

After an analysis of equation 1, we notice that reordering summations can accelerate the computation but with a high resource cost due to the shift of the columns. An alternate method to obtain an accumulation value, requiring minimal resources, must be found.

In summations j and k of equation 1, the column j is multiplied with the rest $W - j$ columns. However, as we only need to detect circles with a maximum diameter of Dm , we only need to multiple the column j with the next Dm columns. We rewrite the equation 1 as the equation 2.

$$H_x\left(\frac{j+k}{2}\right) = \sum_{j=1}^{W-1} \sum_{k=j+1}^{j+Dm} \sum_{i=1}^H I_{(i,j)} * I_{(i,k)} \quad (2)$$

We can reorganize the summations of equation 2 to obtain the equation 3.

$$H_x\left(\frac{j+k}{2}\right) = \sum_{i=1}^H \sum_{j=1}^{W-1} I_{(i,j)} * \left(\sum_{k=j+1}^{j+Dm} I_{(i,k)} \right) \quad (3)$$

The variable change $n = \frac{j+k}{2}$, produces equation 4.

$$H_x(n) = \sum_{i=1}^H \sum_{j=1}^{W-1} I_{(i,j)} * \left(\sum_{n=j+1}^{j+\frac{D_m}{2}} (I_{(i,2n-j-1)} + I_{(i,2n-j)}) \right) \quad (4)$$

We can reorganize the summations of equation 4 to obtain the equation 5.

$$H_x(n) = \sum_{i=1}^H \sum_{j=1}^{W-1} \sum_{n=j+1}^{j+\frac{D_m}{2}} I_{(i,j)} * I_{(i,2n-j-1)} + I_{(i,j)} * I_{(i,2n-j)} \quad (5)$$

As $I_{(i,j)}$ is a pixel with a binary value, we can replace the multiplication with a logic *AND*.

To optimize the area, we can approximate the addition with a logic *OR*. Although a vote is lost for each consecutive contour pixel, the shape of the histogram does not change, and we obtain the same local maximums. The equation can be rewritten as the equation 6.

$$H_x(n) = \sum_{i=1}^H \sum_{j=1}^{W-1} \sum_{n=j+1}^{j+\frac{D_m}{2}} (I_{(i,j)} \text{ AND } I_{(i,2n-j-1)}) \text{ OR } (I_{(i,j)} \text{ AND } I_{(i,2n-j)}) \quad (6)$$

We can now use equation 6 that computes the same histogram H_x as equation 1. The advantage of the equation 6 is that we obtain an accumulation value of histogram H_x for each pixel read. Furthermore, it only uses simple logic operators *OR* and *AND*.

The latency and the area of architecture are impacted because we use simple *OR* and *AND* operators instead of addition operators and we shift simple binary pixels instead of entire columns using FIFO memories.

Figure 2 shows the testing results of the modified algorithm. The algorithm identified a circle (seen in green in c) from the flat polyp (seen in a).

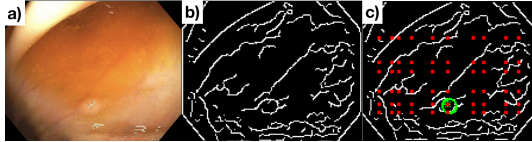


Fig. 2. a) Original image, b) edge image, c) circles detected.

IV. X-COORDINATE ARCHITECTURE

We realized a new version of the x-coordinate computation module that implements the modified version seen in equation 6. It is a 3-stages pipelined architecture that is shown in Figure 3. It computes N_c x-coordinates for a maximum circle diameter D_m , $D_m > 1$.

In the first stage, for an image of size $W * H$, the image needs to be read pixel by pixel in line, and each pixel is shifted using a simple 1-bit Flip-Flop. With this architecture, we can compute one input pixel with D_m and other pixels of the same line in parallel. The votes are shifted using 1 bit Flip-Flops and are accumulated in one memory. To accumulate the votes, we propose the architecture illustrated in Figure 3. This architecture is implemented in a single BRAM that enables a simultaneous read and write in one cycle and avoids collisions.

With this architecture, we obtain the same histogram as the previous version of RT-HTA.

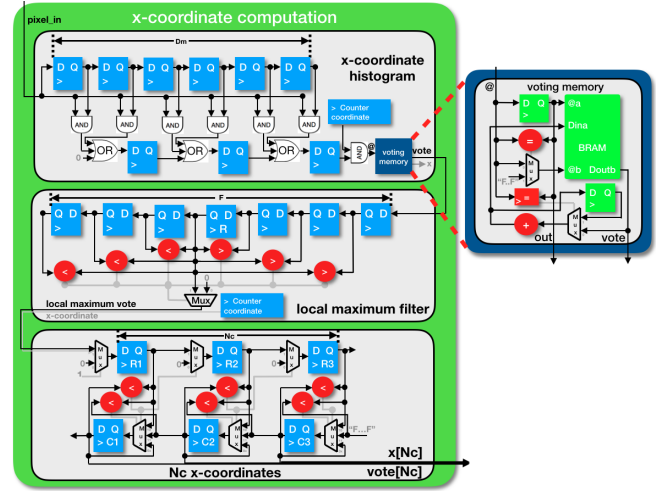


Fig. 3. x-coordinate computation architecture.

In the second and third stages, we use the same architecture as the previous version shown in Figure 1. In the second stage, the vote values are filtered in order to find the local maximum. A sliding window of size F , that corresponds to the minimum Euclidian-distance between two center candidates, is used to compare a vote value with the $\frac{F}{2}$ previous values and with the $\frac{F}{2}$ posterior values. The output is the vote corresponding to the largest histogram value for a x-coordinate. If there are two circles with centers separated by an Euclidean-distance less than F pixels in the image, only the circle with more edge points will be detected.

Finally, in the third stage, all the local maximum histogram values are stored in registers R_i and C_i . Hence, the larger values will be gradually transferred to the right side through the registers C_i . This process is executed until all the lines in the image are read. Finally, the largest local maximum histogram values are stored in C_i and their respective coordinates correspond to the N_c x-coordinates of the circles candidates.

Using a similar architecture, N_c y-coordinates are computed. Each x- and y- coordinates are combined to obtain N_c^2 center candidates.

In the new HT-RTA for a HD image, the x- and y-coordinates range from 1 to 1920. The address is then coded in 11bits and only 2 BRAM of 36Kbits are necessary.

A comparison of the necessary resources to implement this architecture and to implement the previous architecture for HD images is available in Table III. From these results, we notice that the new architecture uses 83% less BRAM and 100% less DSP in comparison to the original version of RT-HTA.

TABLE III
RESOURCES NECESSARIES FOR X- AND Y-COORDINATES COMPUTATION.

Resources	RT-HTA original	RT-HTA optimized	Reduction %
BRAM36Kb	12	2	83%
DSP	108	0	100%

V. AREA FPGA VALIDATION

To implement our new RT-HTA, we need $\frac{20(W+H)}{36K} + \frac{N_c^2}{2}$ BRAMs of 36Ks and $4N_c + N_c^2$ DSP slices in order to detect N_c^2 circles with a maximum circle diameter of D_m and separated at least F pixels in an image of $W \times H$ size. For a HD image it is necessary to use 10 BRAM of 36Kbits and 32 DSP slices.

The number of circles to detect, N_c^2 , is limited by the number of BRAM and DSP48 available. The maximum circle diameter D_m is limited by the number of Flip Flops and Look-Up Tables (LUT) available.

To determine if our HT IP could be contained in a WCE with more or less 11mm of diameter, we have implemented it on a FPGA Xilinx Spartan 7 XC7S15-CPGA196 packaged in $8 \times 8 \text{mm}^2$ area with a diagonal of 11.3mm. The results of implementation are showed in Table IV.

TABLE IV

NEW RT-HTA IMPLEMENTATION IN A XC7S15 FPGA.

Resource	Available	Utilization now	Utilization before	Reduction
LUT	8000	7735(97%)	20637(258%)	63%
BRAM36K	10	10(100%)	78.5(785%)	87%
DSP	20	20(100%)	153(765%)	87%

Our new RT-HTA present significant reduction of resources and it can be contained in a FPGA Xilinx Spartan 7 packaged in $8 \times 8 \text{mm}^2$ integrable in the next generation of WCE.

VI. EXECUTION TIME SOC VALIDATION

To validate the execution time of our architecture, we prototyped it on a SoC system, the Digilent ZedBoard XC7Z020.

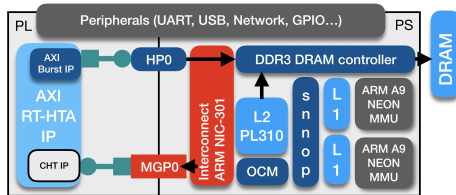


Fig. 4. Diagram of our AXI RT-HTA IP core integrated on a Zynq.

In Figure 4, we illustrate the integration of the new RT-HTA IP in this SoC.

We have realized a pipeline of three operations; first, write an image into the DRAM memory and read computed circles, second, center computation and third, radius computation.

Computation time of our new HT-RTA to detect N_c^2 circles with a maximum circle diameter of D_m separated at least by F pixels for an image of $W \times H$ size takes $2(W * H + W + 2 + \frac{D_m}{2} + \frac{F}{2} + N_c)$ cycles of latency and $W * H + W + 2 + \frac{D_m}{2} + \frac{F}{2} + N_c$ cycles of interval between two images. As we see, the latency and the interval are directly linked to the image size.

To validate the functionality of our new RT-HTA, we have realized an implementation to detect $N_c^2 = 16$ circles with a maximum circle diameter of $D_m = 108$ and a sliding window $F = 40$ in a HD image. Details of the implementation can be seen in Table V.

As we can see in Table V our HT architecture can work with a maximum frequency of 140.10 MHz alone and 135.46 MHz

TABLE V
AXI CHT IP CORE IMPLEMENTATION RESULTS.

Architecture	CHT IP	AXI CHT IP	System
Freq. maximum[MHz]	140.10	135.46	135.46
Test Freq.[MHz]	125		
Latency[ms] / Interval[ms]	33.24 / 16.62		
Throughput[Mpixel/s] / fps[frame/s]	125 / 60		

in the global system. That is due to the distance between the furthest DSP slice and the AXI AMBA interconnection. We have validated our new RT-HTA with a frequency of 125 MHz to process a HD image each 16.62ms. Our digital architecture can process an image of 1920×1080 pixels in less than 40 ms, as is expected, and provides a 60 fps throughput.

VII. CONCLUSIONS

In this paper, we have proposed an efficient real-time architecture to implement the HT that can be integrated in an $8 * 8 \text{mm}^2$ FPGA and help localize polyps in a WCE. We have conceived an efficient architecture to significantly reduce the use of memory and DSP.

Our architecture supports a maximum clock frequency of 140.10 MHz to detect up to 16 circles with a maximum circle diameter of 108 pixels on a HD image.

This architecture was functionally validated in a SoC system, a Xilinx Zynq-7000 XC7Z020, where it uses 10 BRAMs of 36Kb, 32 DSP slices, 8355 LUTs and 8651 Flip Flops. It produces a latency of 33.24 ms and an interval between two images of 16.62 ms for a HD image. Finally, this architecture can process 60 images per second.

REFERENCES

- [1] J. Ferlay, I. Soerjomataram, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. M. Parkin, D. Forman, and F. Bray, "Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012," *International journal of cancer*, vol. 136, no. 5, pp. E359–E386, 2015.
- [2] C. Orlando, P. Andrea, D. Xavier, and B. Granado, "Polyps recognition using fuzzy trees," in *Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on*, pp. 9–12, IEEE, 2017.
- [3] J. Silva, A. Histace, O. Romain, X. Dray, and B. Granado, "Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer," *International Journal of Computer Assisted Radiology and Surgery*, vol. 9, no. 2, pp. 283–293, 2014.
- [4] P. V. Hough, "Method and means for recognizing complex patterns," Dec. 18 1962. US Patent 3,069,654.
- [5] C. Orlando, M. Andrea, D. Xavier, and B. Granado, "FPGA-Based Real Time Embedded Hough Transform Architecture for Circles Detection," in *2018 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pp. 31–36, Oct. 2018.
- [6] A. Goneid, S. El-Gindi, and A. Sewisy, "A method for the hough transform detection of circles and ellipses using a 1-dimensional array," in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 4, pp. 3154–3157, IEEE, 1997.
- [7] X. Zhou, Y. Ito, and K. Nakano, "An efficient implementation of the one-dimensional hough transform algorithm for circle detection on the fpga," in *Computing and Networking (CANDAR), 2014 Second International Symposium on*, pp. 447–452, IEEE, 2014.
- [8] O. Romain, A. Histace, J. Silva, J. Ayoub, B. Granado, A. Pinna, X. Dray, and P. Marteau, "Towards a multimodal wireless video capsule for detection of colonic polyps as prevention of colorectal cancer," in *Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on*, pp. 1–6, IEEE, 2013.

¹Thanks to Hana Raffoul for her corrections supported in this paper.