Agent-based Hough Transform: A Way to the Improvement of the Execution Time in the Detection of the Dominant Straight Line in an Image

Brahim Nini
Larbi Ben M'hidi University
Constantine Road, PO. BOX 358
04000 Oum El Bouaghi, Algeria

Brahim Mehelain
Larbi Ben M'hidi University
Constantine Road, PO. BOX 358
04000 Oum El Bouaghi, Algeria

Bilel Flifel
Larbi Ben M'hidi University
Constantine Road, PO. BOX 358
04000 Oum El Bouaghi, Algeria

ABSTRACT
This paper focuses on the minimization of the time of the dominant straight line detection in an image using Hough Transform algorithm. The idea is a mixture between two domains, namely image processing and multi-agent systems. The importance of this work comes from the relying of image processing techniques on hardware accelerations. This paper demonstrates how the distribution of a purely sequential processing on a set of agents leads to an improvement from time execution point of view. The purpose is to reduce the execution time of Hough transform technique through the distribution of the algorithm on a set of reactive agents. This may allow the exploitation of a parallel or distributed environment. The main idea is based on the division of similar repeated processing with different parameters on several agents. It is a SIMD-like architecture according to Flynn classification. The obtained results are promising in the way that the execution time is at least divided by 4 comparatively to the use of the algorithm in its sequential form.

General Terms
Image processing, Multi-agent systems, Distributed computing.

Keywords
Hough transform, Agent, Straight line, Execution time.

1. INTRODUCTION
For many decades, image processing applications are based on sequential approaches to perform complex tasks, and consequently, are time-consuming. This has an impact on many other fields which are based on. Even nowadays, algorithms of image processing are getting more and more complex depending on the content to be analyzed. Despite recent advances in computer hardware, this complexity still remains an obstacle. One possible solution is that some of these operations can be split and done in parallel, or in a distributed fashion, depending on the nature of the algorithm. Hence, new approaches are used in tackling image analysis domain.

One of these approaches that are getting great importance in the research field is inherent in the multi-agent systems (MAS). Based on the artificial intelligence techniques, complex operations may be simplified in a cooperative manner.

Since the last decade, several multi-agent approaches have been proposed to improve image processing [1] through many points of view. According to [5], within the image processing domain, MAS are used according to two opposed software engineering strategies: 1) they can be used to exploit different macro level results provided by image processing algorithms, using negotiation among the agents, or 2) they can be used as artificial social systems, composed of entities which exploit micro level image processing results [8], adapted to the intrinsic image processing. The former uses several image processing algorithms at the same time. This is why these approaches do not deal in general with the inherent problems encountered in images processing. The agents use macro results provided by image processing algorithms and apply them as they are. Hence, one of their advantages is to analyze problems they are locally confronted with, and then use the algorithm which seems the more suitable to their local context [2]. In this work as for [8], we propose to deal with the core of image processing algorithms for an enhancement goal. In this context, many works propose to apply new views specific to MAS in image processing domain like the work in [3]. What is remarkable in most of them is that they focus mainly on image segmentation tasks.

Among the techniques that are interested in how to group isolated edge points into image structures, there is Hough Transform [7, 6]. It is used in the detection of regular curves such as lines, circles, ellipses, etc. Specifically, the straight-line detection case has been ingeniously exploited in many applications. It is based on the evaluation of two parameters: the angle varying into [0..360°] and the range depending on the position of each edge point.

The technique suffers, however, from poor time performance. This is due to two main reasons. The first one is its sequential structure where the same work is repeated 360 times for each edge point. The second one is the manipulated data structure that is a 2D array whose size is proportional to the image's size times 360. These are considered the Hough Transform technique's weaknesses.

Hough transform algorithm is used in general as a preprocessing task which requires an extra time to be added to a main task's processing time. The problem of the accumulator was extensively studied in [4], whereas many solutions where proposed for the dynamic part. In [10] for example, a description of an algorithm that uses Hough Transform for real-time pattern recognition is made. [9] Presents a boundary line-based method that optimizes

speed and accuracy by combining the Hough transform and contour algorithm in vehicles' license-plate recognition. Consequently, Hough Transform technique has seen many improvements in the literature, but always in the context of image processing scope.

This paper is a presentation of an approach that distributes the Hough Transform algorithm on a set of reactive autonomous agents in order to exploit it in a distributed or parallel environment. It consists in mixing the domain of SMA with image processing since this approach has proved many advantages [5]. The used idea is to divide the angle varying in [0..360°] into smaller intervals and make a set of agents responsible of the processing relative to each interval. From image processing point of view, each agent is responsible of detecting the dominant line in a specific interval of inclination in an image. The dominant straight line in the image becomes simply a decision to be done relatively to the partial results.

In order to present these results, the paper is divided into four sections. The next one reviews the Hough transform for straight lines. The following section details how the idea of mapping a set of agents on the technique is done. It presents the obtained results and discusses them in relation to the sequential form of the technique. The final section concludes this work and presents its future extension.

2. HOUGH TRANSFORM TECHNIQUE FOR LINE DETECTION

Hough transform (HT) is a technique which can be used to isolate features of a particular shape within an image. It is used for the detection of regular curves such as lines, circles, ellipses, etc. In general, its purpose is not inherent in itself. It is always used as a preprocessing technique in order to prepare an image for further processing. Because of its importance, it is largely used even though its calculation and memory costs are very high.

In spite of the HT technique's usefulness in image processing domain is out of the scope of this paper, it is possible to summarize its important features into two points:

- Tolerant to gaps in feature boundary descriptions.
- Unaffected by image noise.

Given an image of size N x M, HT algorithm checks each pixel $p(x, y)$ of detected edges. It assumes that primitive edge detection has already been performed on the image. It takes each one and converts it into a new space. In fact, it transforms each significant edge pixel belonging to (x, y) space into a straight line in this new space expressed in the form $y = mx + c$. Clearly, many lines go through a single point (x, y). Consequently, each line will have a slope (m) and an intercept (c) such that the equation holds true. The available space needs then to hold this set of lines in an array (called the accumulator array). Since m and c may vary from $-\infty$ to $+\infty$, it is safer to use angles, transforming to polar coordinates (r,θ) (see Figure 1).

The principle of HT algorithm consists in varying θ from 0° to 360° for each edge pixel value (x, y), and then calculating $r = x.\cos\theta + y.\sin\theta$. Given an accumulator array of size ($N+M$, 360), the elements that lie in the box (b x b) with center (r,θ) (being the position in the accumulator array) are incremented, i.e. for each edge's pixel, extract the 360 lines passing by it whereas each line is governed by the equation ($x.\cos\theta + y.\sin\theta = r$). Clearly, if the box is (1 x 1), only one element of the array is incremented; if the box is (3 x 3), nine elements are incremented. This gives a thick line in the new space so that intersections are not missed. The increment of the value of the same position many times indicates that the same line is passing by several pixels, i.e. several lines having the same values of m and c. This is why HT algorithm finally looks for the highest values in the accumulator arrays (r, θ) and thus identify the pair (r, q) that are most likely to indicate the line in (x, y) space.

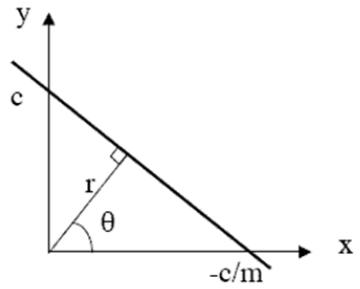

Figure 1. One line belonging to (r,θ) space of HT passing by a pixel

3. AGENT-BASED HOUGH TRANSFORM

One of the main problems encountered in image processing is the execution time which is generally high because of the hardness of the extraction of the relevant data in a given context. This problem is also one of the HT algorithm which shows poor time performance. To solve such problem and give it a solution, we propose to distribute the HT algorithm on a group of reactive agents. In a parallel hardware environment, this would lead to a clear enhancement.

Within this work, we are interested in the dominant line in an image, i.e. the most important one having the maximum number of pixels. The dominant line is then extracted using the radius and angle sets as being related to the maximum value in the accumulator. The purpose is to apply an idea in which it is possible to share the HT algorithm on a set of agents where each one is responsible of one part.

3.1 Distribution of the HT Algorithm

The distribution of the algorithm is based on the interval of angles varying from 0° to 360°. The basis of the HT algorithm is that it should evaluate r for all the 360 possible lines and this for every pixel edge. Hence, agents interfere at the level of the angle θ. To do it, two layers of agents are used. The first one is composed of many agents having the same function with different parameters as it will be seen later on. However, just one agent composes the second layer. Its function is to create the other agents and synthesize their results. Based on this approach, the group of agents is created and each one is assigned a given interval in the range of [0°..360°] to work with. The used principle is to divide the 360 degrees on the fixed number of

agents and make each one responsible of the detection of the dominant line relatively to its interval according to the HT algorithm. It is then called local line.

According to this idea, each agent has its own partial accumulator on which it works. As a consequence, the corresponding accumulator should be (*N x M, int*) where *int* = *endV – beginV* + 1 is the size of the interval [*beginV..endV*] each agent is assigned. In another words, each agent is responsible of the detection of the dominant line in a given orientation which may be called local dominant line. To do it, the full HT algorithm is applied on the entire image but in its specific interval.

A central agent has the function of creating the other agents and detecting the global dominant line. For this, Each agent communicates its parameters as being the angle, the range and the corresponding value in its local accumulator of the local dominant line to the central agent. The latter, after gathering all the required pieces of information from all agents, selects the highest value to identify the most dominant line in the image.

3.2 Implementation

The realization of the solution uses Jade platform based on the Java language. A dual core processor was used for the obtained results shown below. The application begins by creating the first agent whose function is to create the other agents and to detect the global dominant line. It is named ``AgentG", meaning Agent Guide. The number of created agents depends upon the expected number of the test. Each created one is given the name of its specific interval (see Figure 2).

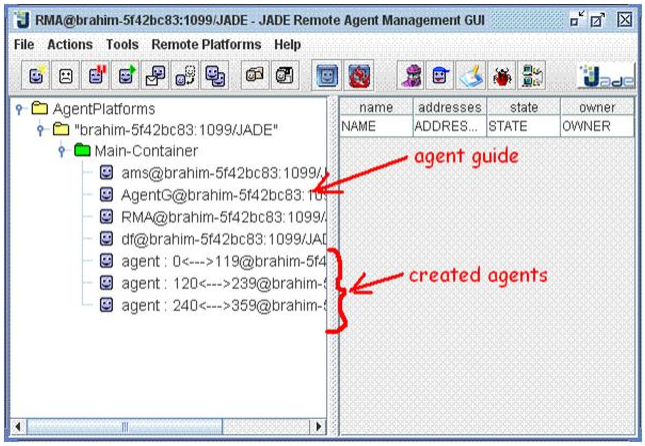

Figure 2. Example of agents' names. In this case three agents are created.

After the decision of AgentG of the dominant global line, the application redraw it in another color using the following equation: $y = (-ax+c)/b$, Where $a = \cos(angle)$, $b = \sin(angle)$, and $c = range$ (see Figure 3).

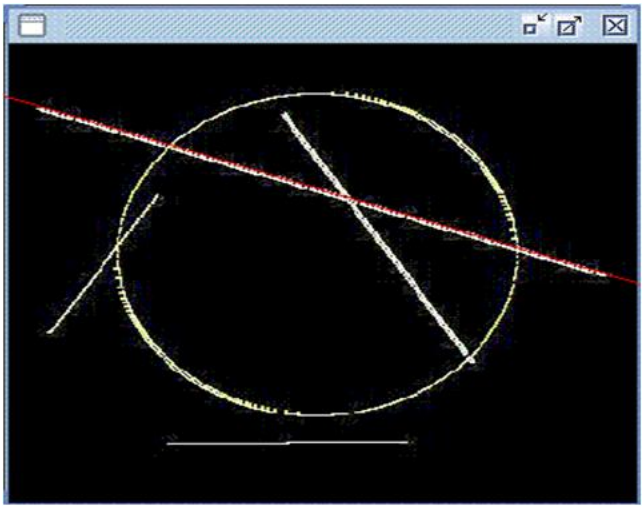

Figure 3. One used image with the dominant line detected.

3.3 Results of the Proposed Technique

To show the obtained results, some experiments have been made on different images. To emphasize the effect of the edges' pixels number, the application was executed using three types of images, namely Im1 (Figure 4), Im2 (Figure 5), and Im3 (Figure 6), which are different from complexity point of view. The complexity to be understood here is based on the number of detected pixels during edge detection process.

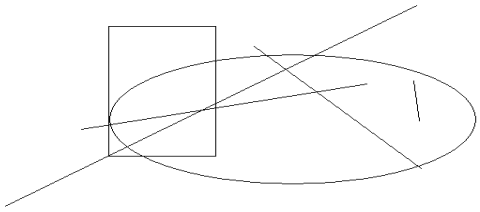

Figure 4. The image with 2370 pixels.

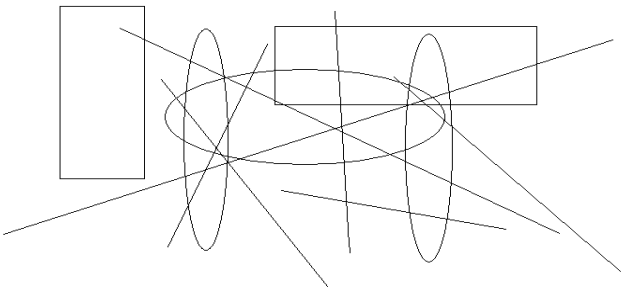

Figure 5. The image with 5378 pixels.

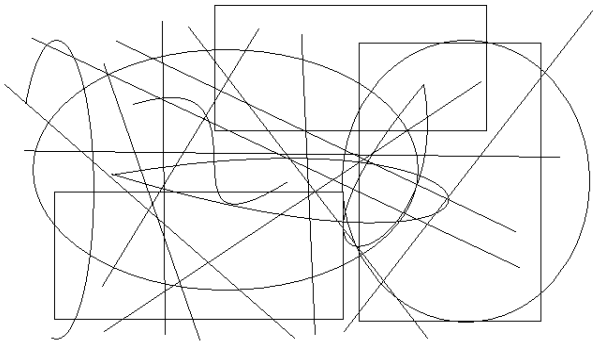

Figure 6. The image with 10070 pixels.

According to each case, a different number of agents was involved, varying from 1 to 360. The used principle, as already explained, is to divide the 360 degrees on the chosen number of agents and make each one responsible for the detection of the local dominant line relatively to its related interval according to HT algorithm. The made experiments are based on the study of the effect of such number of agents on the execution time in relation to images' complexity. We get a sample of results grouped in Table 1 and represented graphically in Figure 7. Each mentioned value in the table is the mean value of ten values obtained after complete executions. Note that the number 0 agent means the HT algorithm running in its sequential form. Also, the number of detected edges' pixels is put between parentheses.

Table 1. Mean time of HT algorithm execution applied on different images, having different number of edges' pixels, with different number of agents.

Agents	Im1 (2370)	Im2 (5378)	Im3 (10070)
0	544.25	755.65	1342.50
1	530.22	777.86	1400.82
2	511.36	685.65	1077.77
4	237.58	517.67	1097.82
10	106.84	172.39	306.32
20	113.91	312.38	402.54
50	96.36	163.81	185.65
100	97.42	96.67	147.52
200	114.50	95.36	98.90
360	74.07	76.16	74.48

3.4 Discussion

(Figure 7) shows three curves. Each one represents the variation of the execution time of the distributed algorithm which depends on the number of agents used in the analysis of three different images having specific number of edges' pixels.

A particular situation is represented by the number of agents as being equal to 0. It is the case where no agent is used; i.e. we use the algorithm of Hough Transform in its sequential form. What is remarkable in this situation is that the execution time increases when just one agent is used. Even though it is almost similar to have one agent or to run the algorithm in its sequential form, this situation may be explained by the time devoted to the communication between Agent Guide and the created one.

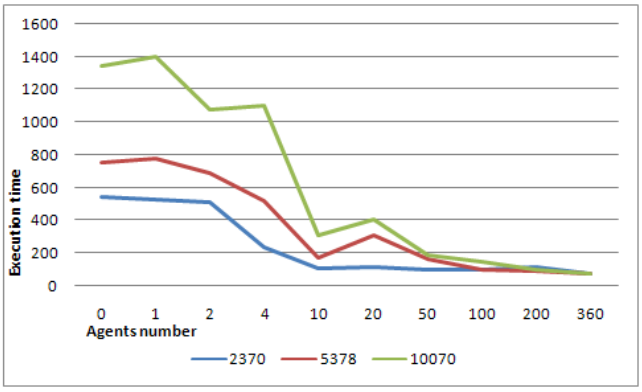

Figure 7. Representation in curves of the results enumerated in Table 1.

What is observable, as being the most important result of this work, is that the execution time decreases when the number of agents increases. This remarkable behavior of the algorithm has the following features:

- The amount of time shown in (Table 1) is not important in itself. It may change under different execution environments. The relative values obtained correspondingly to different images and numbers of agents are more significant.
- The execution time of the sequential form is remarkably reduced and divided, in average situations, at least by 4. In the case of complex images, this may reach the ratio of 1/20.
- It is observable that when the number of agents increases, the execution time becomes independent from the images' complexity whereas it is directly dependent in the case of the sequential form of HT algorithm.
- What is also remarkable is that the use of agent solution does not make an observable effect on the execution time of the sequential algorithm if a small number of agents are used.
- The execution time tends to be stable when the number of agents goes beyond 50 whatever images' complexity is.

4. CONCLUSION

In this paper, one case of the borrowing from multi-agent system to image processing is shown as being able to bring some improvement. The mixture is based on the split of the HT algorithm into pieces that are processed separately by agents. Each agent extracts the straight line in the image corresponding to a given inclination. The dominant straight line existing in the used image is extracted through gathering all partial results.

The implemented application uses Jade platform with Java language. It allows comparing the amount of time required to

reach the objective when varying the number of agents and edge points. The results show that the number of 50 agents and up is optimum whatever the number of detected edges' pixels is. The execution time is reduced considerably and can reach the ratio of 1/20 relatively to the sequential form of HT algorithm in the case of complex images.

The extension of this work will deal with its generalization regarding the Hough Transform principles. In fact, the latter is a technique that is used for the detection of many geometrical shapes provided that is has an analytical expression. It would be important to generalize the solution so that it becomes based on the multi-agent systems regarding the obtained results for straight lines.

5. REFERENCES

[1] C. Spinu, C. Garbay, and J.M. Chassery. A multi-agent approach to edge detection as a distributed optimization problem. In Proceedings of the 13th International Conference on Pattern Recognition, volume 2, Vienna, Austria, 25-29 Aug 1996.

[2] Ernst G. P. Bovenkamp, Jouke Dijkstra, Johan G. Bosch, and Johan H. C. Reiber. Multi-agent segmentation of ivus images. Pattern Recognition, 37(4):647663, 2004.

[3] E. Duchesnay, J.-J. Montois, and Y. Jacquelet. Cooperative agents society organized as an irregular pyramid: A mammography segmentation application. In Pattern Recognition Letters, volume 24, pages 24–35, 2003.

[4] J. Illingworth and J. Kittler. A survey of the hough transform. In Computer Vision, Graphic and Image Processing, volume 44, pages 87–116, 1988.

[5] J. Mahdjoub, Z. Guessoum, F. Michel, and M. Herbin. A multi-agent approach for the edge detection in image processing. In EUMAS'06. Fourth European Workshop on Multi-Agent Systems, Lisbon, Portugal, December 14-15 2006.

[6] P. V. C. Hough. A method and means for recognizing complex patterns. US Patent, 3,069,654, Dec 1962.

[7] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. ACM, pages 11–15, 1972. 5.

[8] Smaine Mazouzi, Mohamed Batouche, and Zahia Guessoum. A self-adaptative multiagent system for segmentation and reconstruction of 3d scenes. In AISTA 2004 in cooperation with the IEEE Computer Society Proceedings, Luxembourg, November 2004.

[9] Tran Duc Duan, Duong Anh Duc, and Tron Le Hong Du. Combining hough transform and contour algorithm for detecting vehicles' license-plates. In International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, October 20-22 2004.

[10] Xin Lin and Kazunori Otobe. Hough transform algorithm for real-time pattern recognition using an artificial retina camera. OPTICS EXPRESS, 8(9), April, 23 2001.