A NEW METHOD USING HOUGH TRANSFORM (HT) IN ROBOTIC VISION

Mahfoud HAMADA
[1] Electrotechnical Laboratory of Batna (LEB), Hadj Lakhdar University of BATNA
Chahid Boukhlouf Road, BATNA, 05000 Algeria, maayamahfoud@yahoo.fr

Abdelhalim BOUTARFA
[2] Advanced Electronics Laboratory (LEA), HL University of BATNA, Algeria
Chahid Boukhlouf Road, BATNA, 05000 Algeria, boutarfahal@yahoo.fr

Abstract: *Vision is an important sense for the navigation of mobile robots. Indeed this work presents a solution to an interesting and important problem, i.e. visual beacon detection for mobile robots. The proposed approach is based on a combination of a neural network pixel classifier and the Hough Transform to detect shapes in the incoming images. One of the objectives is to enable the robot (CESA) to move in an unspecified environment and acquire the necessary information for its vision. In view of the positive results obtained with a momentum of 0.001 and a coefficient of training equal to 0.015, we can conclude that our system is robust. Also, our algorithm allows a significant reduction of the computation time and can be therefore used in real-time applications. Moreover the proposed architecture can be easily fitted into Field Programmable Gate Array (FPGA) reconfigurable devices, since the present performance increase of this technology allows the implementation of complex applications while real-time constraints in Hough Transform for lines detection are respected for most of the video transmission standards. From this perspective, the present work constitutes an important step toward a better comprehension of the problem and proposes a solution that is robust under diverse conditions.*

Key words: *Hough Transform, Neural Network, Robot Vision, FPGA.*

1. Introduction

Vision systems for autonomous mobile robots must unify the requirements and demands of two very challenging disciplines: i) computer vision and image processing, and ii) robotics and embedded systems. While the state of the art in computer vision algorithms is quite advanced, many computer vision methods are intrinsically computationally expensive. Even an efficient implementation of such methods cannot fix this problem. Therefore, the resource demands of computer vision methods are in conflict with the requirements posed by robotics and embedded systems, which demand very short execution cycles for the control loops which read out and process sensor data, interpret and fuse them, and determine appropriate actions for the actuators. Particularly the real-time requirements of robotics seem to rule out most sophisticated computer vision methods, which is one of the reasons why some computer vision experts get discouraged to work on robot vision. As a consequence, robot vision software systems are often inflexible and hard to maintain, because they tend to contain hard-coded quick hacks, which for efficiency reasons try to exploit microoptimizations like performing multiple operations simultaneously, or because they are heavily model-based or incorporate application-specific heuristics. To address one aspect of this problem a solution is put forward.

In this paper, we propose a methodology (NN-HT) which is applicable in real-time and especially implemented in an FPGA chip. In that sense, the remainder of the paper is organized as follows: in section 2 we first review the previous work in robotic vision and in particular perception techniques in order to motivate the directions that we followed in the proposed methodology. In section 3, we proceed to the detailed description of the proposed process and show the hardware implementation (FPGA) of the novel algorithm and its validation. The obtained results on real images with robot CESA can be appreciated in section 4. Finally, section 5 summarizes our findings and gives a perspective of our research.

2. Previous work

The work of Meribout [1] has been carried out in relation to real-time HT hardware implementation while the L64250 [2] chip from LSI *Logic Corporation* uses a modified Hough Transform for real-time processing. M. Karabernou [3] has proposed the use of the CORDIC algorithm for fast Hough transform. This makes the HT problem suitable for parallel computing which would considerably reduce the processing time. Hough Transform has been successfully used in many applications. For instance, as a recent example, a Hough transform based on a line cluster detection method for rock mass discontinuity detection and analysis was shown by Deb [4]. A study on more formal statistical properties of Hough transform, such as the consistency of the estimator and the rate of convergence, was reported by Dattner [5]. Chunling Tu [6] proposed in his work, a hybrid method using a super resolution (SR) to improve Hough Transform (SRHT). This has allowed addressing the vote spreading and the peak splitting. In the current context, the interest to use the techniques of artificial intelligence solves many problems related to cognitive tasks such as learning, adaptation, generalization and optimization.

J. Tani [7] offers an interesting approach to robot navigation learning problem. In fact, the robot learns the model of the environment by recurrent neural network. Other solutions that use neural networks and fuzzy logic for motion planning and the representation of human knowledge applied to the navigation of robots are also found in [8].

To distinguish the main objects in image datasets, Bui Ngoc Nam [9] uses the graph-based segmentation and the normalized histogram (PODSH). There are some alternatives to find robust features that can be identified and characterized with a descriptor like the Surf [10]. For example Fontanelli [11] uses Sift for a wheeled vehicle visual servoing. Other detectors like Corner Detector in [12] have found corner features that are very common in semi-structured environments. In our previous work we used a neural network model for objects detection in the map of the robot's environment. This approach has allowed the study of the cluster analysis with range finder data [13]. In contrast to other works, we propose a new method that can be implemented by any FPGA device. Moreover, all previous work discussed so far concerns applications where robotic vehicles operate within natural environments. From this perspective, the present work constitutes an important step toward a better comprehension of the problem and proposes a solution that is robust under diverse conditions.

3. Proposed method

The problem that we are addressing is decomposed into two sub-problems, namely, computer vision method and FPGA implementation. In figure 1, we provide a schematic overview of the individual steps that are followed in the proposed process.

3.1 Problem and solutions

An effective vision system is very important for robots to sense the environments where they work and to detect the objects where they will operate. The system can greatly improve the robot's flexibility, adaptability and intelligence [14]. With the developments of optical engineering, embedded vision systems integrating camera and processing unit together, were developed in recent years.

However, it is necessary to investigated how to optimize sufficiently the limited computing capability in an embedded vision system. For this problem, a solution that is suitable is recommended: A real-time process based on the combination of the Hough Transform and neural networks is shown below in the block diagram of figure 1.

The objective of the flow chart is to permit to the CESA robot in figure 9 to understand better its perceived environment which will allow it to detect the beacons and to recognize their forms. A mapping by a Hopfield network is well adapted to compare with its database and to make the right decision.

3.2 Neural network implementation

3.2.1 The network architecture

After the calibration step of our aquisition system (sensors), we obtain gray level images. Then, we use the back-propagation algorithm of the gradient [15]. It has the merit of having two aspects: real time parallel processing and control of the process including the non-linearities.

Our neural network is a MLP (multilayer perceptron), often recommended in studies of pattern recognition, since it provides accurate results [16]. The network (9-6-3-1) in figure 2 gives better generalization for the entire image at the input of the network. To train the network, we use the generalized delta rule with momentum as learning paradigm.

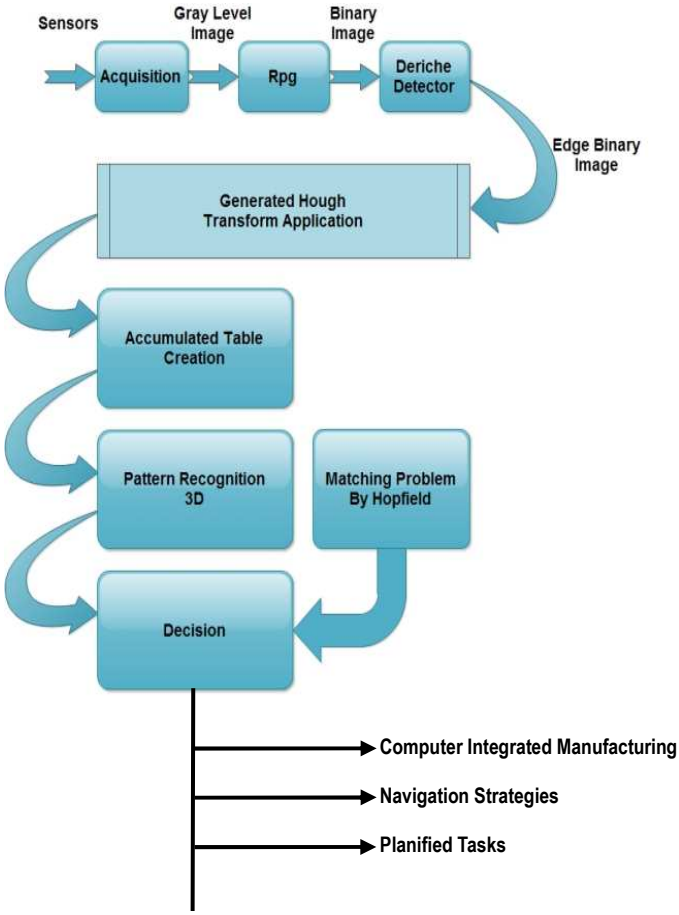


Fig. 1 Algorithm flowchart.

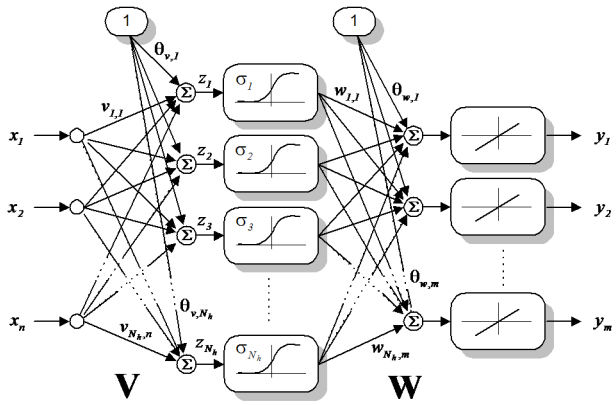
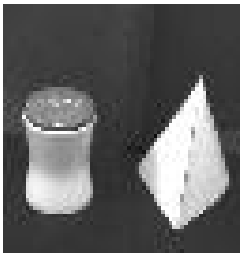


Fig. 2 Structure of an MLP neural network.

To reproduce our image or detect objects in a scene, it is necessary to develop a neural classifier. The latter consists of

a window of a multi-layer back propagation gradient: Multiple tests performed in the laboratory with windows of different sizes showed that the window (3*3) is the most appropriate. The figures below (3-a, 3-b and 3-c) illustrate the relevance of our choice [13].



(a)

(b) (c)

(a): Original input image;
(b): Results with a 3x3 window size;
(c): Results with a 5x5 window size.

Fig. 3 Test results with other window sizes.

3.2.2 The momentum

The concept of momentum has been introduced to achieve a compromise between a low learning coefficient and an acceptable learning time. Momentum acts as a low pass-filter on the term of the weight change, since it reinforces the general trend and reduces the risk of oscillation.

The total number of weights is given by:

$$N = \sum_{i=1}^{n-1} L_i \cdot L_{i+1} \quad (1)$$

The network is trained by the delta rule with a momentum term which was introduced by Rumelhart [17]. At the $(t+1)^{th}$ iteration of the training, the network weight w_{ij} is updated with two components. The first component is proportional to the error signal e_j on the output $X_{ij,}$ for the neuron sending the activation signal. The second component is proportional to the amount of weight changes in the previous iteration (the momentum term).

$$\Delta wij\ (t+1) = wij(t) + C1.ej\ /Xij + C2.\Delta wij(t) \quad (2)$$

Where C_1 is the learning rate and C_2 the momentum term.

C_1 and C_2 are the main factors that affect learning and thus allow the tradeoff between system stability and the quality of the classification. Therefore, C_1 controls the speed of convergence in the training process and C_2 prevents the problem of oscillation in the vicinity of the solution.

Two types of transfer functions are usually used (sigmoid or hyperbolic). Ours must be continuous and monotonous, so our choice was the sigmoid function in our application, because it gives us the best results.

$$f(z) = 1/1.0 + e^{-z} \quad (3)$$

f(z) is between 0 and 1 when z varies between -∞ and + ∞. The derivative of f (z) is:

$$f'(z) = f(z).(1.0 - f(z)) \quad (4)$$

3.2.3 The back-propagation algorithm (Rpg)

The Rpg learning in the flowchart of figure 2 can be written in algorithmic form as follows:

We note that i, k, l, j are the layers of the neural network, respectively the input layer, the first hidden layer, the second hidden layer and the output layer for the multi-layer neural network classifier.

Procedure 1

Step 1:	*Calculate the output y_k for the first hidden layer*
Step 2:	*Calculate the output z_l for the second hidden layer*
Step 3:	*Calculate the output u_j of the output layer*
Step 4:	*Calculate the error of the output layer*
Step 5:	*Calculate the error of the second hidden layer using the output error*
Step 6:	*Calculate the error of the first hidden layer using the error of the second hidden layer*
Step 7:	*Calculate the weight variation for all the layers using the learning rate C1 and the momentum rate C2*
Step 8:	*Update the weights*
Step 9:	*if the algorithm converges (e3 inferior to a predefined threshold) stop the process, if not, return to step 1*

Note:

For our contribution we improve the rule of the modification of the weights in the algorithm of the back-propagation by adding the following equation:

$$\Delta W(t+1) = \Delta W(t) + 0.9\,W \quad (5)$$

Through experimentation, this major change enhances considerably the reliability of the algorithm.

3.3 Extraction of binary images by Deriche filter

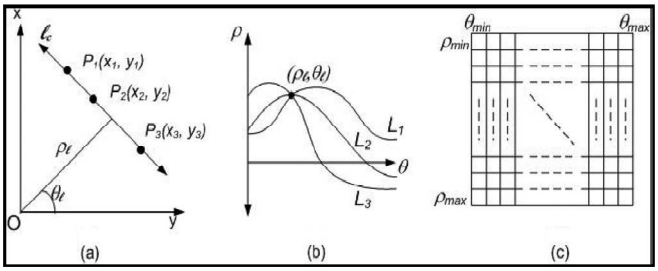
We proceed by applying a noise filtering operation to the 2D binary image that was computed in the previous step (Rpg). We obtain finally edge binary images.

All edge detectors are considered to be a triplet (algorithm, parameters and precondition). The precondition is represented by the context in which the pair (algorithm, parameters) runs correctly. It appears that the Deriche detector seems to be better suited for highlighting edges and noisy contours. This is an operator optimal edge detection based on Canny criteria implemented recursively. Moreover, it provides edges better localized, and thus produces a lower error of localization and omission. The edges of each binary image are extracted by the Deriche operator. There is a fairly significant noise reduction which reduced the recognition errors. The results confirm this.

3.4 Generated Hough Transform

The Hough Transform is an attractive technique because of its robustness towards noise. However, its main disadvantages are the large memory size required and the necessary complex and time consuming computations. The Hough Transform can be used to detect colinear points by studying the Hough space as shown in figure 4.

Points $P_1(x,y)$, $P_2(x,y)$ and $P_3(x,y)$, lying on the same line, share the same (ρ, θ) combination (ρ_1, θ_1), which implies that the sinusoidal curves of the colinear points intersect in the Hough space.



(a): Colinear points in image plane
(b): Intersecting sinusoids in Hough space
(c): Implementation as a 2D array of accumulators
Fig. 4 Straight lines Hough Transform.

It appears that the inputs are all pixels of the output of the line detection algorithm and the output is a class of an object. The neural network function is to classify a small region in the image as a part of object and his background. Once after line segments are detected by HT and matching lines pattern with the images are stored in the machine. The line segment extraction algorithm from the Hough space is performed by searching segments carried by each significant line segment in this space. A significant line segment is characterized by a peak in the Hough space, and for each detected peak, we remove from the Hough space the effect of its points. The used segment extraction algorithm outputs the polar coordinate (θ, ρ) of a line and the coordinates of its end points. The idea of the improved HT algorithm (GHT in figure 1) is to generate a number M of ρ values at the same time during a fraction of the θ interval in order to reduce both the effect of these errors on final results and the full HT computation time.

Using the same notations as above, the ρ value generated in a sub-interval m of the θ-axis (the θ-axis is divided into M sub-intervals) is given by the following equation:

$$\begin{cases} \rho_\theta = x\cos\theta + y\sin\theta \\ \theta = n\varepsilon + m\pi/M \\ 0 \le n\varepsilon < \pi/M \\ 0 \le m < M \\ 1 \le M < K \\ \varepsilon = \pi/K \end{cases} \quad (6)$$

This algorithm is applied on binary edge images obtained in our case after using the Deriche edge detection algorithm described in [18]. It uses the two procedures (2 and 3) which summarize the voting process and the segment extraction algorithms.

Procedure 2: the voting process Algorithm

Begin
For each binary edge image point Do,
For each sub interval [mK/M , (m+1) K/M] of the θ-axis Do
Calculation of the ρ's initial values $\rho_{(mK/M)}$ *and* $\rho_{(mK/M)+(K/2)}$
For $0 < n < (K/M - 1)$ *Do,*
Calculation of the $\rho_{(n+1)+(mK/M)}$ *and the* $\rho_{(n+1)+(mK/M)+(K/2)}$ *values*
For each calculated ρi *we increasing by one the cell (iε, ρ i) from the Hough space*
EndDo.
EndDo.
EndDo.
End.

So, these parameters (K and M) are very important and must be chosen so that the accumulated errors are kept in an acceptable range. The used segment extraction algorithm outputs the polar coordinate (θ, ρ) of a line and the coordinates of its end points.

Procedure 3: the line segment extraction Algorithm

Begin
The Max peak $(n_{Max}\varepsilon, \rho_{Max})$ *value extraction from the Hough space.*
While this value is superior of an experimentally threshold ThH Do
For each binary edge image point forming this peak Do
For $0 < n < (K/M - 1)$ *in each sub interval*
[mK/M, (m +1)K/M] of the θ -axis Do
Decreasing by one the cell ($n\varepsilon$, ρ_n*) from the Hough space*
EndDo.
EndDo.
For the whole binary edge image points forming this peak Do
looking for all adjacent points whose lengths are superior to the same threshold ThH.
EndDo.
End While.
End.

3.5 Matching problem by Hopfield model

Given the state of the art, we opted for the Hopfield model that is most compatible with our application. We show that the matching problem can be formulated as an optimization task where you have to satisfy the constraints of correspondence. An energy function representing these constraints on the solution is minimized. Then the connection matrix is deduced to change the network to its stable state. Figure 5 shows the structure of two-dimensional Hopfield which can be likened to a matrix.

Lyapunov function for a binary two-dimensional network of Hopfield [19] is given by:

$$E = -\frac{1}{2}\sum_{u,l}\sum_{v,m} T_{ulvm}\, V_{ul}\, V_{vm} - \sum_{u,l} I_{ul}\, V_{ul} \quad (7)$$

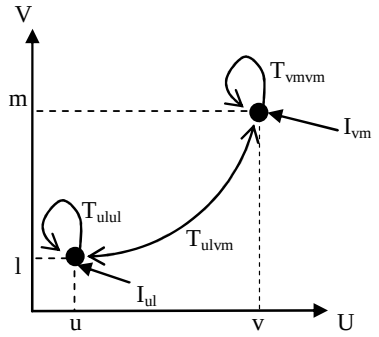


Fig. 5 Structure of the Hopfield network.

V_{ul} et V_{vm} represent the binary states (output) neurons (u,l) and (v,m), which can be set to 1 (active) or 0 (inactive).

T_{ulvm} is the connection weight between the two neurons.
This connection is symmetric ($T_{ulvm} = T_{vmul}$).
It is shown that for a stable network, It would require to each neuron has no connection on itself, ie $T_{ulul} = 0$ and I_{ul} is the initial entry for each neuron. In this network, we used one hand the constraints of intensity and disparity, and secondly the law of oneness and the scheduling.

3.6 Hardware implementation of the HT Algorithm

3.6.1 The proposed architecture

Three main architectures have been proposed for the HT algorithm:

- The ρ generation.
- The voting process.
- The control bloc.

The general expression of the HT algorithm is written as follows:

$$\begin{cases} 0 \le m < \frac{M}{2}, 1 < M \le K, \ \varepsilon = \frac{\pi}{K} \ and \ 0 \le n < \frac{K}{M} \\ \rho_{n+1+\frac{m.K}{M}} = \rho_{n+\frac{m.K}{M}} + \varepsilon.\rho_{n+\frac{m.K}{M}+\frac{K}{2}} \\ \rho_{n+1+\frac{m.K}{M}+\frac{K}{2}} = \rho_{n+\frac{mK}{M}+\frac{K}{2}} - \varepsilon\rho_{n+\frac{m.K}{M}} \\ \rho_{\frac{m.K}{M}} = x.cos\left(\frac{m.K}{M}.\varepsilon\right) + y.sin\left(\frac{m.K}{M}.\varepsilon\right) \\ \rho_{\frac{m.K}{M}} = y.cos\left(\frac{m.K}{M}.\varepsilon\right) - x.sin\left(\frac{m.K}{M}.\varepsilon\right) \end{cases} \quad (8)$$

ε is the resolution in the θ-axis.
M is the number of the ρ values generated at the same time.
K is the number of the divisions of the the θ-axis.
ρ_i is the ρ value obtained from the discrete angular coordinate i along the the θ-axis.
(x, y) are the image point coordinates.
$\left(\rho_{\frac{mK}{M}}\right)$ and $\left(\rho_{\frac{mK}{M}+\frac{K}{2}}\right)$ are the seed values of the ρ generated at the beginning of each sub-interval of the θ-axis.

Figure 6 shows the *m*th module that realizes the ρ generation architecture defined by expression (8). The *m* value indicates which sub-intervals the ρ generation module is working on. *M/2* modules in parallel are necessary to generate the Hough distributions.

3.6.2 FPGA implementation

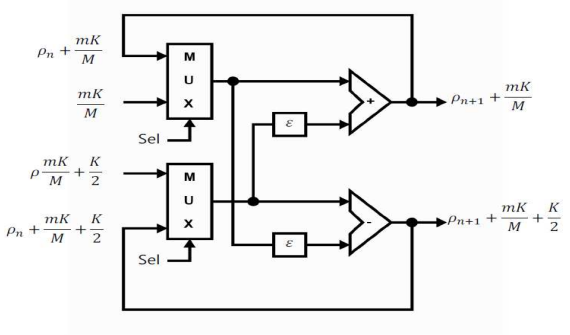


Fig. 6 The ρ generation architecture of the *m*TH module.

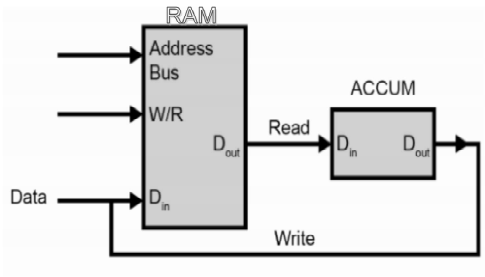


Fig. 7 The voting process architecture bloc.

The presented architecture has been implemented using an 8-bits Virtex II Xilinx Xc 250-5fg456C FPGA for a particular case of M=4. The design entry used the VHDL hardware description language based on ISE Fondation Conception EDA tools. Expression (8) is rewritten as follows:

$$\begin{cases} \rho = \rho + \varepsilon\rho_{n+(\frac{K}{2})} \\ \rho_{n+1+(\frac{K}{2})} = \rho_{n+(\frac{K}{2})} - \varepsilon\rho_n \\ \rho_{n+1+(\frac{K}{2})} = \rho_{n+(\frac{K}{4})} + \varepsilon\rho_{n+(\frac{3K}{4})} \\ \rho_{n+1+(\frac{3K}{4})} = \rho_{n+(\frac{3K}{4})} - \varepsilon\rho_{n+(\frac{K}{4})} \\ 0 \le n < \frac{K}{4} \\ With: \ \rho_0 = x \ ; \ \rho_{\frac{K}{2}} = y \ ; \ \rho_{\frac{K}{4}} = \frac{\sqrt{2}}{2}(x+y) \ ; \\ \rho_{3K/4} = \frac{\sqrt{2}}{2}(y-x) \end{cases} \quad (9)$$

Since the value ε must be selected as $1/2^p$ and K/M value must be an integer, we have chosen K= 48 which implies that p=4. Two modules from figure 6 and four blocs from figure 7 are necessary to generate and store all Hough distributions.

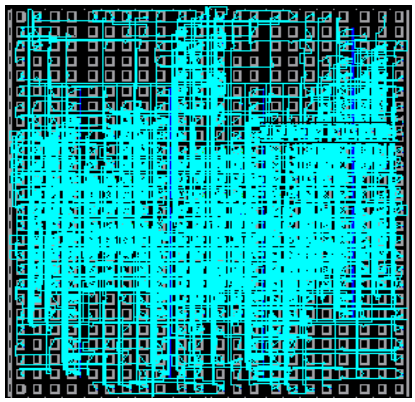


Fig. 8 HT parameters extraction circuit.

The used memory will be addressed over 14 bits and the stocked data will be coded over 9 bits for a binary edge image of size 256x256. The resulting architecture can be clocked at 606.081MHz and can generate in parallel four ρ each 11.852 ns.

Table 1: The occupation of space FPGA

SLICE (CLB_S) OCCUPIED	**618/1536**	OCCUPATION OF **37%**
IOB_S OCCUPIED	**138/200**	OCCUPATION OF **69%**
RAM	**19/24**	OCCUPATION OF **81%**

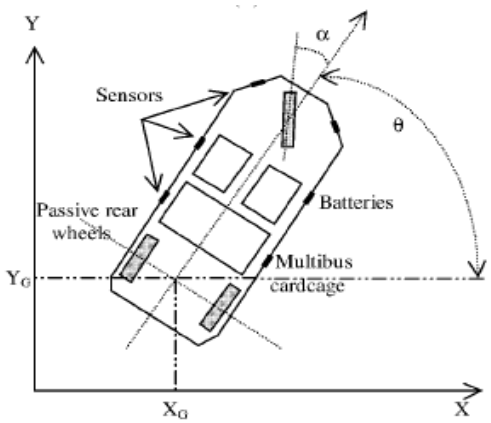
Figure 8 shows the Virtex II Xilinx Xc 250-5fg456C FPGA package. It utilizes: 37% or 618 out of 1536 Slices (CLBs), 69% or 138 out of 200 IOBs and 81% or 19 out of 24 RAM.

Unlike Djekoune [20] who ran out of circuit area and therefore was unable to generate all ρ values for their corresponding θ values because they made use of LUT to store sine or cosine function required by the HT, we were not confronted to this problem in our test setup.

The circuit area is sized according to the M value, the greater the area. The free circuit area can be used to include one or more functions such as straight-line extraction function. This will allow us to avoid additional circuitry to be placed on different chips that would slow down the execution time because of a slower off-chip communication.

4. Experimental results

The CESA vehicle in the figure 9 is an experimental vehicle designed to operate autonomously within a structured office or factory environment. It has been designed with several goals in mind. First and foremost, CESA is tested with which to experiment with such things as robot programming language and sensor integration/data fusion techniques. Second, CESA is designed to be low cost, and depend on only one camera and two active telemetric sensors. CESA has a tricycle configuration with a single front wheel, which serves both for steering and driving the vehicle, and two passive load-bearing rear wheels.



(a)

(b)

(c)

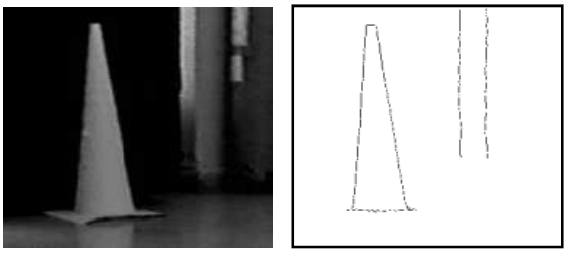
Fig. 9 Robot CESA.
(a): Structure of the vehicle and its reference center coordinates
(b): The CESA autonomous vehicle
(c): Robot CESA with embedded system

The onboard control system uses several sensors embedded telemetry assets *a*nd a vision system to provide the position and heading information needed to guide the vehicle along specified paths between stations. The external, or off-board, part of the system consists of a global path planner which stores the structured environment data and produces the path plans from one station to the next. It also includes a radio communication link between this supervisory controller and the vehicle. The CESA vision system is composed of a camera iVC 500 CCD and a microcontroller-based data acquisition card. The 80C51 microcontroller controls the whole operation of the card.

The acquisition is performed in real time in 5ms. A binary edge-recognition algorithm was implemented on a field programmable gate array (FPGA) on this card. The framework is successfully applied in different robotic scenarios, such as biologically-motivated neural network learning, neural object classification in an office environment and reliable high speed image processing in the RoboCup middle-size robot league [21].

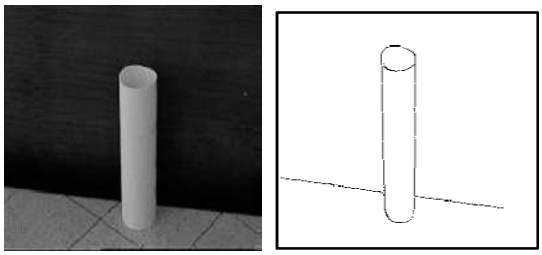
For this purpose, we use the laboratory environment as a field of action for our experimental mobile platform By comparison with our work a similar robotic mapping method has been developed in this method, scan matching is combined with a heuristic for closed loop detection and a global relaxation method, results in a 6D concurrent localization and mapping on natural surfaces [22].

In all experiments, various types of beacons (polyhedral, cylindrical and rectangular) are used. The proposed algorithm was applied to the images numbered 10-a, 11-a, 12-a and 13-a. The results are shown respectively in figures 10-b, 11-b, 12-b, and 13-b.



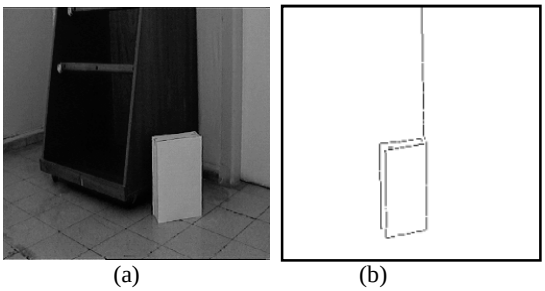
(a) (b)

Fig. 10 Algorithm results of polyhedral beacon.
(a): Original input image
(b): Recognition results



(a) (b)

Fig. 11 Algorithm results of cylindrical beacon.
(a): Original input image
(b): Recognition results



(a) (b)

Fig. 12 Algorithm results of rectangular beacon.
(a): Original input image
(b): Recognition results

(a)

(b)

Fig. 13 Algorithm results of differents objects.
(a): Original input image
(b): Recognition results

Increasing the number of hidden layers implies training difficulties and decreases the performance. Thus, we conclude that the four-layer network (two hidden layers) is sufficient for the segmentation problem. Also, we can say that increasing the input window size increases the computing time and the output images that resulting are noisier. The rate parameters for the back-propagation learning affect not only the learning rate but also the classification performance. The appropriate choice for our problem is a very small learning rate (C_1=0.001) combined with a medium momentum rate (C_2=0.015).

The mobile robot has been made familiar with the environment during an initial training phase. The artificial neural learning system is made to detect only the beacons from other items such as the blurs shown in all figures 10, 11, 12 and 13. Although the mobile robot is moving in the scene, the visual information (input data to the NN) is computed using a stochastic parametric activation function of the neuron. The experiments have been carried out in a noisy environment. With our algorithm, the remaining noise in the background of the scene is removed from the filter stage. The correspondence between the object of interest (beacon) and the filtered image is found by matching the two-dimensional projections of the beacon by considering all possible sets of the available beacons in the internal environment of the mobile robot. The data base includes artificial landmarks. While joining the technical multi-resolution and the data fusion, our system remains robust at the price of an extremely appreciable computing time per execution cycle.

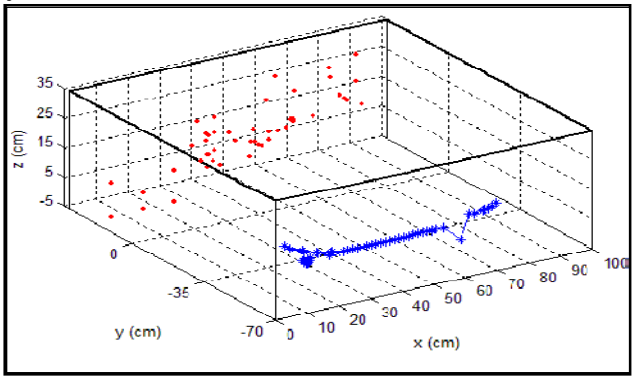


Fig. 14 Three-dimensional plot of the estimate state.

The procedures including the image feature detection and tracking method, feature initialization, system startup procedure are integrated. In this experiment, the robot moves from the left- to right-side of the field, and the estimate state and image features are depicted as a 3D map shown in the figure 14. In the plot, the dots indicate the landmarks obtained from the initialized image features and the asterisks represent the state of the camera equipped on the robot. Therefore, our mobile robot performs the self-localization and mapping procedures simultaneously. The time requirements to process a frame, after the learning stage, is about 5 milliseconds during the operation of the mobile robot, which seems better than other published results obtained under the same working conditions.

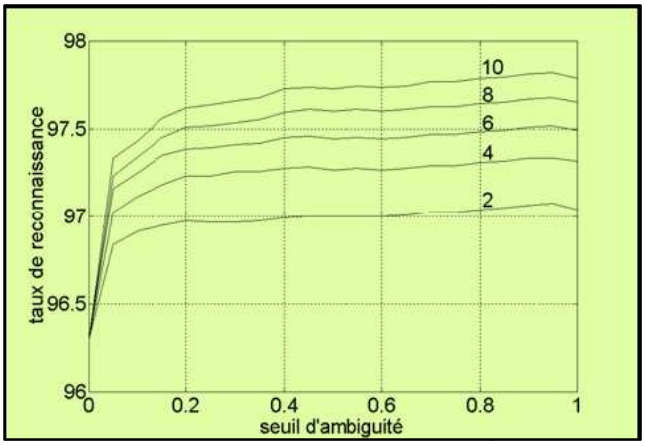


Figure 15 Recognition rate in % of hybrid classifier based on thresholds of confusion and ambiguity.

The hybrid classifier was evaluated on the basis of tests. Figure 15 indicates the evolution of the recognition rate depending on the level of ambiguity and confusion on the threshold of the objects used in the database. The recognition rate also increases with the level of ambiguity.

The results show the effectiveness of our method for recognition beacons on all forms. The recognition rate is estimated at 97.8%.

Our experiments have been carried out on an automated guided vehicle (AGV) shown in figure 9c, which is equipped with a camera. The obtained results on the original images are satisfactory since the objects are clearly recognized.Also, we obtained a valuable insight into the role of network topology, rate parameters, training sample set and initial weights.

The FPGA device has been implemented in the visual navigation system of our CESA mobile robot. After edge restoration, the device's results may be used either as the basic data to determine the depth of an object in three dimensions (stereoscopic vision) by geometric reasoning, or as input data to other algorithms such as the matching algorithms developed in [23-27] and also to endow the mobile robot with the intelligence required to perceive its environment. Two semi-local constraints on combinations of neighbourhood correspondences are derived (one geometric, the other photometric). They allow testing the consistency of correspondences and hence to reject falsely matched neighbourhoods.

The table 2 below shows the numerical results obtained during treatment.

Table 2: Experimental results

TYPES OF BEACONS	RECOGNITION RATE
POLYHEDRAL BEACON	97.7 %
CYLINDRICAL BEACON	97.9 %
RECTANGULAR BEACON	97.8 %
OTHER OBJECTS	97.8 %

According of the proved experimental results, one can conclude that our method in comparison with previous works on subject [28-31] is also very effective and faster to be employed in practical applications. We show clearly the excellence of our system based on table 3.

TABLE 3: COMPARISON OF SEARCH METHODS

SEARCH METHODS	ADVANTAGES	DISADVANTAGES
PROPOSED METHOD WITH HYBRID COMBINATION	deterministic method very effective technical analysis results = 97.8 %	need wide work space time < 500ms
F. BONIN-FONT [28]	deterministic method very effective technical analysis results > 93.0%	work space time consuming > 500ms
JAEDO KIM [29]	deterministic method very effective LPR results = 93.0%	time consuming > 500ms
KUMAR PARASURAMAN [30]	deterministic method very effective financial ratios LPR results = 98.0%	time consuming > 500ms
M. ANTÓN_RODRÍGUEZ [31]	technical analysis very effective results = 97.30%	time consuming > 500 ms

Limitations of our method:

In order to optimize this methodology from an economical point of view, additional variables must be considered in the development and validation of the methodology. A similar work with NeuroHough transform of RANSAC would be interesting for performance comparison.

5. Conclusion and future work

Robotic vision is a key and difficult problem in the navigation of mobile robots, because their background is complicated and changeable. A combination of a neural network pixel classifier and the Hough Transform to detect shapes in the incoming images is proposed in this paper. The difficulties of applying advanced computer vision to autonomous mobile robots in dynamic environments are discussed. We have proposed a hybrid method to achieve parallel computation for higher performance. It consists of combining the techniques based on a Hough transform for straight line detection associated to neural networks for object detection and recognition. The HT is used because it is a well-known method for the detection of parametric curves in binary images, and it was recognized as an important means of searching for beacons and features in binary images. The presented system has the major advantage that, along with increasing the number of the outputs of the network, the processing time remains the same as for a neural network with a single output. We observe other benefits using our hybrid approach compared with classical calculus methods that are as follows:

1. They do not require the complex derivative evaluations and they are easy to understand.
2. They do not get caught in local minima as easily as the classical methods.

We believe that our work has enabled us to provide a solution to the problems of pattern recognition in mobile robotics. A direction for the future is to improve our work by introducing the further support in the following specification:

1. Integrating heterogeneous sensor modules, such as ultrasonic range data and an active stereo-vision system or other types of sensors.
2. Merging accurately topological maps, or hybrid maps created by different mobile robots.
3. Using better encoders with higher resolutions can improve the performance of the algorithms.

Finally, the method RANSAC (Random Sample Consensus) is a method of probabilistic voting which seems to be interesting for reducing the computation time. However, only its inclusion in a hybrid process could enlighten us as to the performance advantage of this method. We reviewed the robust estimation methods used in the robotic vision. It appears that the use of voting techniques such as Hough Transform or the consensus method of Ransac represents a good compromise between robustneess and algorithmic efficiency despite a reduced speed of convergence.

ACKNOWLEDGMENT

The authors would like to express their great appreciation to the anonymous reviewers of this paper for their constructive reviews and suggestions for improving this paper. They would also like to thank image team of CReSTIC of Troyes in France for their help.

References

[1] M. Meribout, M. Nakanishi. *A real-time image segmentation on massively parallel architecture*. Real-Time Imaging Vol. 5, 1999, p. 279–291.

[2] LSI Logic, L64250. Histogram/Hough Transform Processor, 1989.

[3] M. Karabernou, F. Terranti. *Real-time FPGA implementation of Hough Transform using gradient and CORDIC algorith.,* Image and Vision Computing, No.23, 2005, p. 1009–1017.

[4] D. Deb, S. Hariharan, U. Rao, C.H. Ryu. *Automatic detection and analysis of discontinuity geometry of rock mass from digital images.* Comput. Geosci, Vol. 34, No.2, 2008, p. 115–126.

[5] Dattner, I. *Statistical properties of the Hough transform estimator in the presence of measurement errors.* J. Multivariate Anal, Vol. 100, No.1, 2009, p. 112–125.

[6] C. Tu, B. Jacobus van Wyk, K. Djouani, Y. Hamam and S. Du. *A Super Resolution Algorithm to Improve the Hough Transform.* ICIAR'2011, 8th International Conference on Image Analysis and Recognition, Vol. 6753, Burnaby, Canada, June 22-24, 2011, p. 80-89.

[7] J. Tani. *Model-based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective.* IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 26, No.3, 1996, p. 421-436.

[8] D. Janglová. *Neural Networks in Mobile Robot Motion.* International Journal of Advanced Robotic Systems, Vol. 1, No.1, 2004, p. 15-22.

[9] B. Ngoc Nam, Pham The Bao. *Principal objects detection using graph-based segmentation and normalized histogram.* International Journal of Computer Science Issues, Vol. 9, No.1, Issue 1, January 2012, p. 47-49.

[10] H. Bay, T. Tuytelaars, L. Van Gool. SURF: speeded up robust features. In: Proceedings of the Ninth European Conference on Computer Vision, 2006.

[11] D. Fontanelli, A. Danesi, A. Bicchi. *Visual servoing on image maps. In: Springer Tracts in Advanced Robotics.* Experimental Robotics, Vol. 39, 2008, Springer, New York.

[12] I.Parra, D. Fernández, JE. Naranjo, R. García, MA. Sotelo, M. Gavilán. *3d visual odometry for road vehicles.* Journal of Intelligent Robotics and Systems, Vol. 51, No.1, 2008, p. 113–134.

[13] A. Boutarfa. *An approach to beacons detection for a mobile robot using a neural network.* Proceedings of the 18th International Conference Modelling and Simulation, Montreal, Quebec, Canada, May 30, 2007, p. 118-124.

[14] F. Bonin-Font, A. Ortiz, G. Oliver. *Visual Navigation for Mobile Robots*: A Survey. J. Intell. Robotic Syst, Vol. 53, No. 3, 2008, p. 263–296.

[15] Resa Nekovei, Ying Sun. *Back-propagation network audits configuration for blood vessel detection in angiograms*. IEEE Transactions on neural networks, Vol. 6, No.1, 1995.

[16] A. Ghosh, KP. Sankar. *Neural Network, self organization, an object extraction.* Pattern Recognition Letters, Vol. 13, No.5, 1992.

[17] D.E. Rumelhart, JMc-Clelland & PDP. Research group. *Parallel Distributed Processing. Explorations in the Microstructure of recognition.* Vol. 1, 1986, MIT Press, Cambridge.

[18] D. Cremers, M. Rousson, R. Deriche. *A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape*. International Journal of Computer Vision, Vol. 72, No.2, 2007, p. 195-215.

[19] S. Young, P. Scott, NM. Nasrabadi. *Object recognition multilayer Hopfield neural network.* IEEE Transations on image processing, Vol. 6, No.3, 1997.

[20] O. Djekoune. *Incremental hough transform: an improved algortithm for digital device implementation.* Real Time Imaging. Vol. 10, No.6, 2004, p. 351-363.

[21] Mayer, G. Kaufmann, U. Kraetzschmar, G. & Palm, G. Neural *Robot Detection in RoboCu.* In: Biomimetic Neural Learning for Intelligent Robots, Wermter, S. & Palm, G. & Elshaw, M. (Ed.), Springer, Heidelberg, 2005, p. 349-361.

[22] D. Wang, C B. Low. *Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective.* IEEE Trans. on Robotics, Vol. 24, No. 3, 2008, p. 676–687.

[23] R.K. Satzoda, S. Suchitra, T. Srikanthan. *Parallelizing the Hough Transform Computation.* IEEE Signal Processing Letters, (2008), Vol. 15, No.1, 2008, p. 297-300.

[24] Y. F. He, Z. Zivkovic, R. Kleihorst, & Al. *Real-time implementations of Hough Transform on SIMD architecture.* ICDSC'08, ACM/IEEE. 2nd International Conference on Distributed Smart Cameras, Stanford University, USA, 7-11, September 2008, p. 1-8.

[25] N. Baklouti, A. Alimi. *The geometric interval type-2 fuzzy logic approach in robotic mobile issue.* IEEE International Conference on Fuzzy Systems, 2009, p. 1971–1976.

[26] S. Suchitra, Sathyanarayana, R.K. Satzoda, T. Srikanthan. *Exploiting Inherent Parallelisms for Accelerating Linear Hough Transform.* Vol. 18, No.10, 2009, p. 2255-2264.

[27] B. Ommer, JM. Buhmann. Learning the Compositional Nature of Visual Object Categories for Recognition. IEEE Transactions Pattern Analysis and Machine Intelligence. TPAMI, Vol. 32, No.3, 2010, p. 501-516.

[28] F. Bonin-Font, A. Ortiz, G. Oliver. A visual navigation strategy based on inverse perspective transformation. Int. Robot Vision Eds, 2010, p. 62–84.

[29] Jaedo Kim, Youngjoon Han, Hernsoo Hahn. Character Segmentation Method for a License Plate with Topological Transform. World Academy of Science, Engineering and Technology, No.56, 2009, p. 39-42.

[30] K. Parasuraman, P. Vasantha Kumar. *An Efficient Method for Indian Vehicle License Plate Extraction and Character Segmentation.* IEEE International Conference on Computational Intelligence and Computing Research, 2010.

[31] M. Antón_Rodríguez, D. González_Ortega, F. J. Díaz_Pernas, M. Martínez_Zarzuela, I. de la Torre_Díez, D. Boto_Giralda, and J. F. Díez_Higuera. *Bio-inspired computer vision based on neural networks. P*attern Recognition and Image Analysis, Vol. 21, No.2, 2011.

Mahfoud Hamada was born in Tebessa, Algeria in 1960. He received the Electronics Engineer degree in 1985 from University of Annaba and his Magister on Industrial Electricity (Master) in 2000 from University of Batna. Ph.D candidate at the Department of Electrical Engineering, he is currently research member in LEB Laboratory. His research interests include Image Processing, Pattern Recognition, Real Time Embedded Systems, Robotics and their Applications.

Abdelhalim Boutarfa was born in Lyon (France) in 1958. He has graduated from University of Constantine (Algeria) in Physics in 1982. He obtained the Electronic Engineer Degree from the Polytechnic School of Algiers in 1987, a Magister (Master) in 2002 and a Doctorate (PhD) in 2006 in "Control Engineering" at the University of Batna. In 2007 he received a postdoctoral degree in "Habilitation of conducting research in Control Engineering" from the same University where he is currently full professor and research member in the Advanced Electronics Laboratory (LEA). He is also, since October 2010 the Project Manager of the National Center for Technology Transfer at the University of Setif (Algeria). His research interests include applications of neural networks to pattern recognition, robotic vision, and industrial processes.