Fusion of Door and Corner Features For Scene Recognition

Mario I. Chacon-Murguia, Rafael Sandoval-Rodriguez, Cynthia P. Guerrero-Saucedo

Abstract:

Scene recognition is a paramount task for autonomous systems that navigate in open scenarios. In order to achieve high scene recognition performance it is necesary to use correct information. Therefore, data fusion is becoming a paramount point in the design of scene recognition systems. This paper presents a scenery recognition system using a neural network hierarchical approach. The system is based on information fusion in indoor scenarios. The system extracts relevant information with respect to color and landmarks. Color information is related mainly to localization of doors. Landmarks are related to corner detection. The corner detection method proposed in the paper based on corner detection windows has 99% detection of real corners and 13.43% of false positives. The hierarchical neural systems consist on two levels. The first level is built with one neural network and the second level with two. The hierarchical neural system, based on feed forward architectures, presents 90% of correct recognition in the first level in training, and 95% in validation. The first ANN in the second level shows 90.90% of correct recognition during training, and 87.5% in validation. The second ANN has a performance of 93.75% and 91.66% during training and validation, respectively. The total performance of the systems was 86.6% during training, and 90% in validation.

Keywords: *scene recognition, robotics, corner detection.*

1. Introduction

The advance of science and technology has motivated new and more complex engineering applications. These new challenges must involve not only the design of adaptive and dynamic systems but also the use of correct information. It is everyday more evident that good multicriteria decision making systems requires the fusion of data from multiple sources. A research area where data fusion has become a fundamental issue is autonomous robot navigation. Making a robot to navigate and perceive its environment requires similar information as the used by a human [1], [2], [3]. This information usually comes from range detection sensors such as ultrasonic, laser, or infrared, and also from image acquisition sensors, such as CCD or CMOS cameras [4]. The information of each sensor must be processed adequately in order to extract useful information for the navigation system of the robot. One paramount issue in autonomous navigation of robots is related to scenery recognition. Recognition of sceneries consists on the identification of a scenario perceived through measurements provided by a sensor. The sensor may be any of the previously mentioned. However, vision sensors are the most frequently used in this task [5], [6], [7]. The advantage of a vision sensor is that it provides compound information that may be separated into useful properties like color, edges, texture, shape, spatial relation, etc. Therefore, it is possible to achieve data fusion with the information of a vision sensor.

This paper presents the design of a hierarchical neural system for scene recognition using information fusion from indoor scenarios provided by a camera. The problem to solve is constrained to the recognition of 10 indoor scenarios shown in Figure 1. The features used in the design of the hierarchical neural networks are related to door position, and corner detection.

The paper is organized in the next sections. Section 2 presents the corner detection method. Door detection through color analysis and segmentation is explained in Section 3. Section 4 describes the design of the hierarchical neural network, and the paper concludes with the results and conclusions in Section 5.

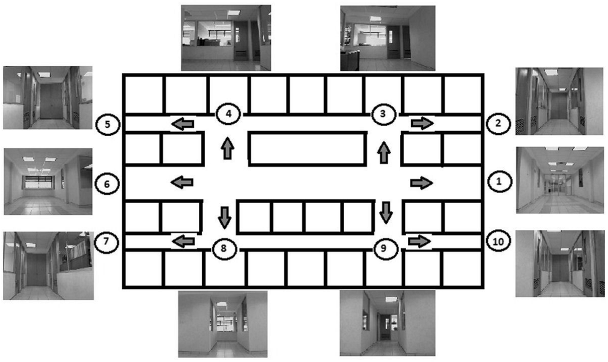

Fig. 1. Scenarios to be recognized.

2. Corner detection method

Corner detection is used in areas such as robotics and medicine. In robotics, it is used for data fusion, navigation, and scene recognition. In medicine, it is applied for image registration such as x-rays, ultrasounds, and medical diagnostics [8]. In other applications, corner detection is used for object recognition, stereo vision, motion detection, among many other usages [9].

Corners are the features more abundant in images of the real world, in contrast to straight lines [10]. For this reason, the use of corners is commonly found in tasks such as image matching. One of the advantages that corners offer is that, if we have images of the same scene, although taken from different perspectives, we will find almost the same corners, which is a good feature for image registration. That in turn will provide information for navigation of mobile robots. There are different definitions for what it is considered a 'corner'. F. Mokhtarian and R. Suomela, in [11], explain that the points of corners in an image are defined as points where the contour of the image has its maximum curvature. Juan Andrade-Cetto, in [12], mentions that corners are one of the most simple features which can be extracted from an image, and define corners as those points in an image where there is change in the intensity in more than one direction. Krishnan Rangarajan *et al.* [13] describe a corner as the point of union of two or more straight lines. In this paper, a corner is considered accordingly to the definition of Rangarajan. Based on this consideration, the proposed method described in this paper locates corners based on corner window analysis.

The methods for the detection of corners can be divided in two groups: those which can accomplish the detection from the image in gray scale, and those which first detect edges and then detect corners. Among the methods of the first group, the most mentioned in the literature are the method of SUSAN [14] and the method of Harris [15]. The method of SUSAN differentiates from other methods in that it does not compute the derivative of the image under analysis and that it is not necessary to reduce the noise that could be present in the image. It uses a circular mask which scans the whole image, comparing the gray levels of the central pixel in the mask and the rest of the pixels inside the mask. All the pixels with a gray level equal to the central pixel level are considered as part of the same object. This area is called USAN (Univalue Segment Assimilating Nucleus). The USAN area has a maximum value when the center is in a plain region of the image, a mean value when it is on an edge, and a minimum value when it is on a corner.

The method of Harris is more sensitive to noise because it is based on the first derivative of the image. However, it is invariant to rotation, translation and illumination, which give it advantages over other methods. This method uses a window which scans the image and determine sudden changes in gray levels which results from rotating the window in several directions.

Among the second group of corner detectors, which use any method of edge detectors, we can mention the one of X.C. He and N.H.C. Yung [16]. They use the method of Canny and indicate the steps to follow for the detection of corners calculating the curvature for each edge. Other authors use windows for corner detection from edge images, such as K. Rangarajan *et al.* [13]. In a similar way, G. Aguilar *et al.* [17], compare images of fingerprints for the identification of persons using 3x3 windows. On those, they propose different bifurcations to be found, which we could call 'corners'. W. F. Leung *et al.* [18], use 23 windows of different bifurcations, and 28 different windows of other type of corners for their detection in the finger print image using neural networks. The method described in this work is based on the second group of corner detectors. Those which first apply edge detection and then detect corners using windows over the edge image.

The general scheme of the corner detection method is shown in Figure 2. The original image $I(x, y)$ is convolved with a Gaussian filter G to remove noise that could be present in the image, yielding the image $Is(x,y)$. A gradient operator and a threshold to determine the edges are applied to the image $Is(x,y)$. These two operations correspond to the edge detection using the method of Canny. The Canny method was used because it yielded better results than the Sobel and other common edge operators. The resulting image $I_b(x,y)$ is convolved (*) with 30 corner detection windows, w_c of order 3x3, to detect the corners present in the image. The resulting image $I_e(x,y)$ contains the corners found.

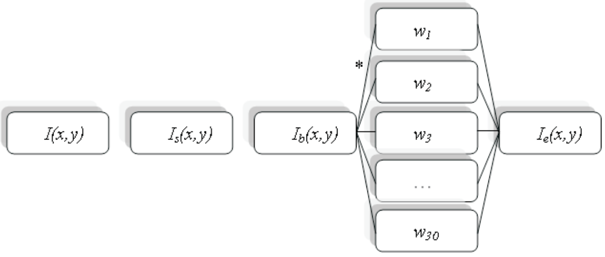

Fig. 2. Corner detection process.

2.1. Edge Detection

The corner definition adopted in this work is the one provided by Rangarajan. It is necessary to find the main line intersections of the scene under analysis. These lines are detected through an edge detection procedure. Among the edge detector operators tested in this work were Sobel, Prewitt, Robert, Canny, and Laplacian. It was decided to use the Canny edge detection method [18], because it generated the best edges in the experiments achieved in this research. It is also one of the most mentioned and used edge detector methods in the literature. The Canny method was developed by John F. Canny in 1986. This method detects edges searching maxima of the image gradient. The gradient is obtained using the derivative of a Gaussian filter. The edges are defined by considering two thresholds related to weak and strong edges which makes the method more robust under noise circumstances. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than others to be fooled by noise, and more likely to detect true weak edges. For a complete description of the method, the reader is encouraged to read [19]. The parameters used for the Canny method were, $\sigma = 1.2$, $\mu H = 0.18$, and $\mu L = 0.18$. These values were chosen after several experimentation results.

2.2. Corner detection windows

The papers from G. Aguilar [17], and W.F. Leung *et al.* [18], coincide in that there are different types of bifurcations or corners that we call them, Y´s, V´s, T´s, L´s, and X´s, accordingly to the form they take, as shown in Figure 3. Based on the similitude of these corners with fingerprint marks, it was decided to investigate the possibility of using a unified theory between fingerprint recognition and scene recognition. Thus, from the fingerprint recognition works, some windows were chosen to detect corners.

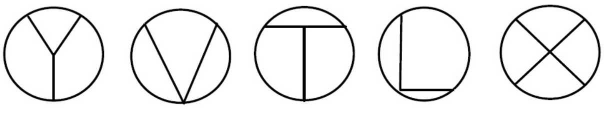

Fig. 3. Type of corners.

These selected windows in addition to other proposed in this work make a set of 30 windows. Each corner detection window, w_c, is a 3x3 mask and their structures are illustrated in Figures 4 and 5. The set w_c of windows is composed as follows. Windows w_1, w_2, w_3, and w_4, are four windows modified from the work of Leung *et al.* [18]. The modification consists on the aggregation of one pixel, because they try to find terminal points, and in our case we look for crossing lines. The extra pixel is darkened in these windows. Windows w_5 to w_{20} were also taken from Leung. The windows w_{17} to w_{20} appear in Aguilar *et al.* [17]. The subset w_{21} to w_{30} are windows proposed in this paper. The proposed windows were defined by analysis of the corners usually found in the set of images considered in this work.

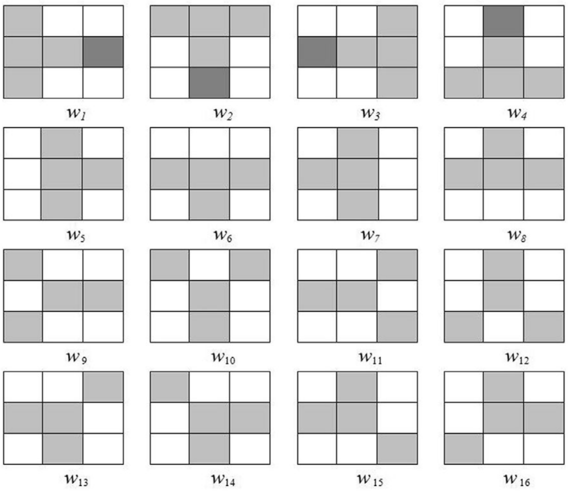

Fig. 4. Windows w_1 to w_{16} for corner detection.

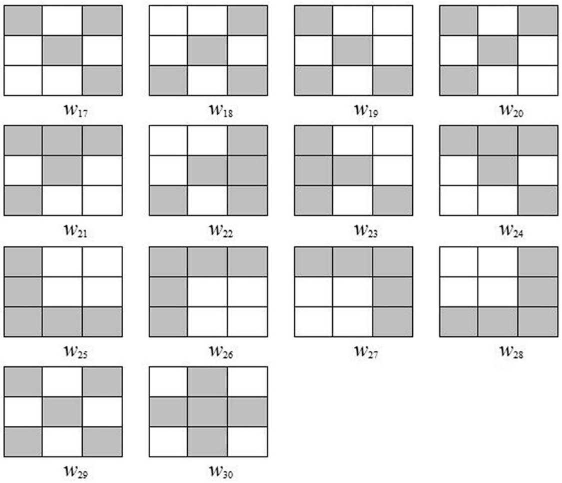

Fig. 5. Windows w_{17} to w_{30} for corner detection.

2.3. Corner detection

Corner detection is achieved through a windows matching process. The image is scanned with the different corner detection windows wc, trying to match the window corner shape with the edge pixels. Assuming a 3x3 neighborhood and two possible values {0,1} for each pixel, the number of permutations is $2^9 = 512$. The values 2^n for n = 0,1,…, 8 can be considered as weighting values to yield a generic weight matrix T_n, starting at the coordinate $p(1,-1)$ as shown in Figure 6.

$$\begin{array}{|c|c|c|}\hline 2^8 & 2^5 & 2^2 \\ \hline 2^7 & 2^4 & 2^1 \\ \hline 2^6 & 2^3 & 2^0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 256 & 32 & 4 \\ \hline 128 & 16 & 2 \\ \hline 64 & 8 & 1 \\ \hline \end{array}$$

Fig. 6. Generic weight matrix, T_n.

Each corner detection window is then associated with an index window B_i

$$T_n : w_c \Rightarrow B_i \qquad \text{for } c,i = 1,\ldots,30 \tag{1}$$

obtained by

$$B_i = w_c \times T_n \tag{2}$$

where the multiplication is element by element and not a matrix multiplication. In this way, each w_c window is related to an index window B_i. In the same way, each index window B_i can be associated to a total weighting factor α_i obtained by

$$\alpha_i = 1 + \sum_{b_i \in B_i} b_i \tag{3}$$

where the b_i corresponds to the weighting factor in B_i.

Corner detection of a scene is accomplished by the next steps. First convolve the binary Canny result image $I_b(x,y)$ with the index matrix B_i

$$I_{ci}(x,y) = I_b(x,y) * B_i + 1 \tag{4}$$

This step yields the possible corners related to each corner window w_c. The next step is to decide which of the possible candidate pixels in each $I_{ci}(x,y)$ is a corner that corresponds to w_c. This process is realized scanning the $I_{ci}(x,y)$ and assigning a pixel value according to

$$p_{ei}(x,y) = \begin{cases} 1 & p_{ci}(x,y) = \alpha_i \\ o & \text{otherwise} \end{cases} \tag{5}$$

to produce a new set of images $I_{ei}(x,y)$, where $p_{ci}(x,y) \in I_{ci}(x,y)$ and $p_{ei}(x,y) \in I_{ei}(x,y)$. The value 1 indicates that the pixel $p_{ci}(x,y)$ is a corner of the type w_c. This process ends up with 30 binary images that indicate the position of the different type of corners. The final step consists on the union of the $I_{ei}(x,y)$ images to produce the final corners

$$I_{FC}(x,y) = \bigcup_{i=1}^{30} I_{ei}(x,y) \tag{6}$$

Experiments of corner detection were performed on images from 16 different scenarios, see Figure 7. Scenarios 1 to 10 represent the environment of interest in this work. They correspond to the real environment where a mobile robot is going to navigate across. Scenarios 11 to 16 were added only for evaluation purposes in other scenarios. Besides, semi-artificial scenarios to compute the quantitative performance were obtained from some of these real scenarios. The purpose of using semi-artificial scenes is to obtain a correct performance measure of the proposed method, which would be hard to compute in real (noisy) scenes. These type of images allow to compute false positives and false negatives detections in a simpler form, and to achieve

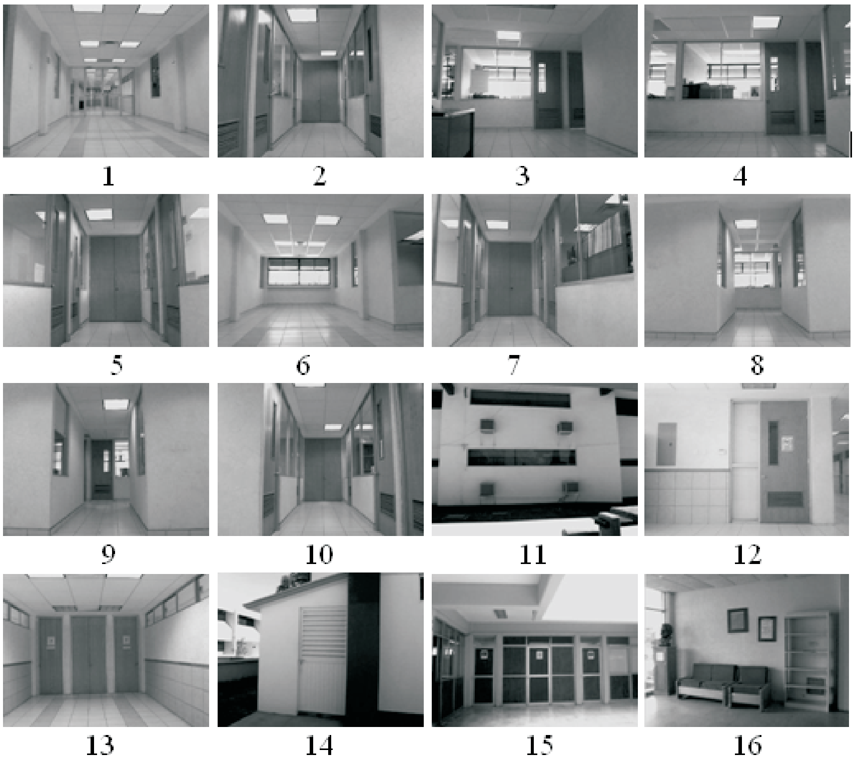

Fig. 7. Scenarios considered to test the corned detection method.

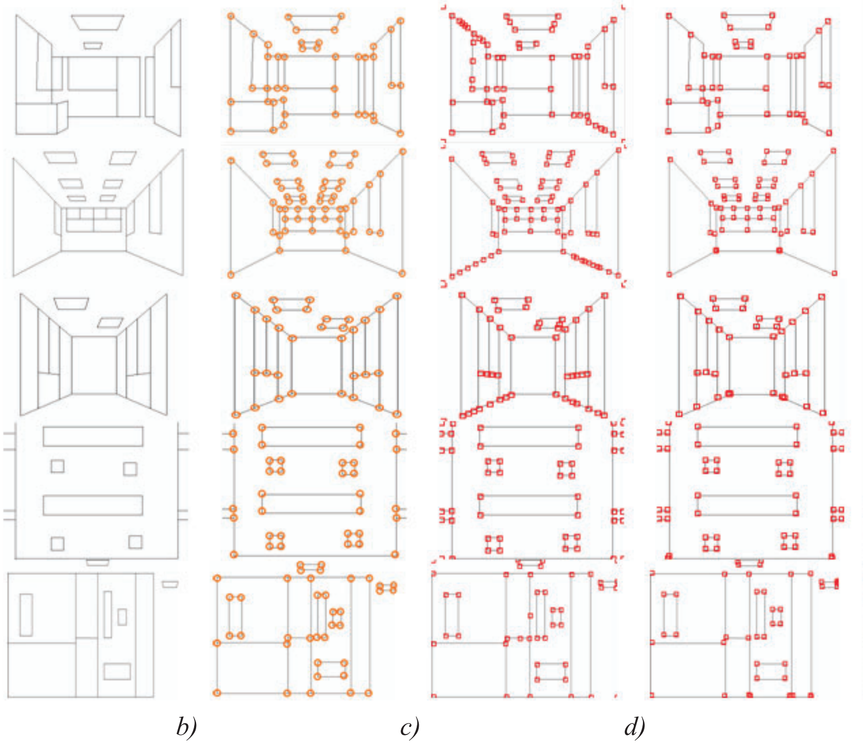

Fig. 8. a) Semi-artificial scenarios, 3, 6, 10, 15,16. b) corners to detect, c) Harris detection, c) SUSAN detection, d) detection with the proposed method.

an analysis of these cases. A semi-artificial image is a simplified image extracted from a real scene. Results of the application of the proposed method to the scenarios are shown in Figure 8. Figure 8a is the semi-artificial scenario obtained from a real one; the corners to be detected are in 8b. The detected corners with the Harris and proposed methods are illustrated in 8c and 8d, respectively. A summary showing the performances of both, the proposed and the Harris methods are shown in Tables 1 and 2. The detection of real corners is very alike in the two methods, 99% and 98%, respectively. Where there is a more noticeable difference is in the false positives, where the proposed method has 13.43%, while the Harris has a 25.86%. Comparison with the SUSAN algorithm is not possible because it requires multi gray level information. In the case of the Harris method, it was assumed two-gray-level images. A qualitative comparison will be given over the original images later on. Our explanation for the false negatives and false positives is as follows. False negatives are mainly due to corners that do not match exactly to any w_c. For example, two false negatives of scene 2 are indicated with circles in Figure 9a. A close view of these cases is illustrated in Figure 9b; it can be observed that the missed left corner does not match exactly with the w_{27}. On the other hand, false positives tend to appear as a consequence of multiple corner detections. This is because more than one w_c make a match with the image edge structure. Figure 10 shows multiple detections in scene 1. It also shows the corner structures w_5, w_8, w_9, w_{28} that make a match with the w_5, and w_8 windows.

Figure 11 shows the corners detected by Harris (a), SUSAN (b), and the proposed method (c). It can be observed that, in general, Harris and SUSAN tend to detect more corners than the proposed method. However, the false positive rate is assumed to be very high, as proved with the semi-artificial images using the Harris method. Considering that corner information is used for robot navigation, high rate on false positives may lead to complicate more the scene recognition than the lack of some corners.

Table 1. Performance of the proposed method.

Scenario	Real corners	Corner detected	Hits	False positives	False negatives
3	42	40	40/95.23%	0/0%	2/5%
6	55	61	55/100%	6/11%	0/0%
10	32	36	32/100%	4/12%	0/0%
15	34	46	34/100%	12/35%	0/0%
16	38	43	38/100%	5/13%	0/0%
Total	201	226	199/99%	27/13.43%	2/1%

Table 2. Performance of the Harris method.

Scenario	Real corners	Corner detected	Hits	False positives	False negatives
3	42	55	41/97.6%	14/33%	1/2.4%
6	55	70	52/94.5%	18/33%	3/5.5%
10	32	43	32/100%	11/34%	0/0%
15	34	42	34/100%	8/24%	0/0%
16	38	40	38/100%	1/2.6%	0/0%
Total	201	250	197/98%	52/25.86%	4/2%

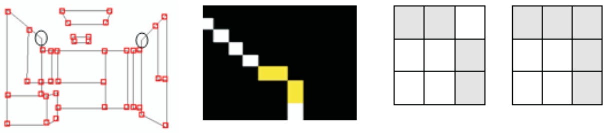

Fig. 9. a) False negatives, b) One false negative zoomed out, c) Pixel view, d) Window w_{27}.

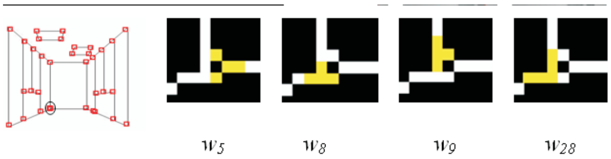

Fig. 10. False positives in scene 1.

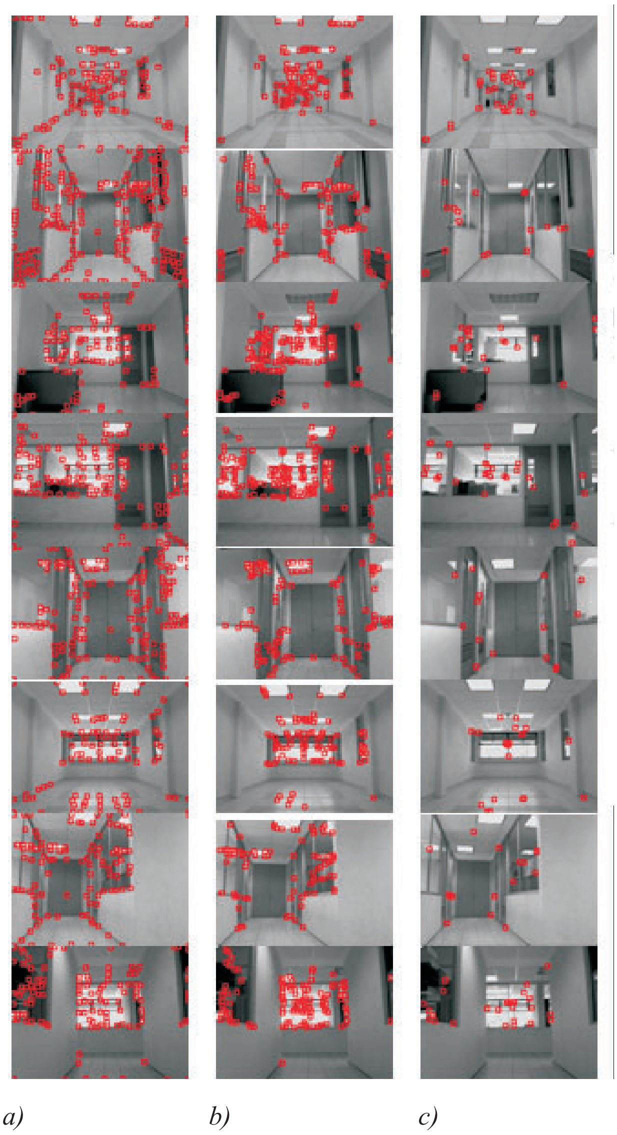

Fig. 11. Corner detection by a) Harris b) SUSAN and c) Proposed method.

3. Scene Segmentation

Interior environments are highly structured scenarios, because they are designed under specific conditions related to size, shape, position and orientation of their components like doors, walls, aisles. Therefore, finding landmarks of these scenarios are important hints for autonomous robot navigation systems. Z. Chen and S.T. Birchfield [19], state that doors are particular landmarks for navigation since they indicate input and output points. Besides, doors provide stable structures and they are semantically significant.

This section describes the segmentation process to obtain features related to the doors found in the scenarios under analysis. The process is shown in Figure 12. The RGB image is transformed to the HSV color space to be more tolerant to illumination changes [19]. Then detection of the doors is achieved by color analysis.

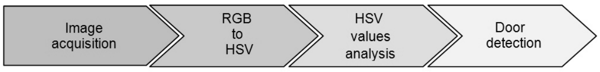

Fig. 12. A block diagram of the process for door detection.

3.1. RGB to HSV color space

The HSV color space is one of the spaces commonly used in color analysis. Schwarz, Cowan and Beatty [20], performed an experiment to compare five color-space models. In this experiment they found significant differences. In particular, they found that the RGB model is fast but it is imprecise. On the other hand, the HSV is not as fast as the RGB but it is very precise for color analysis.

The color space transformation from RGB to HSV is obtained by the following equations

$$V = \mathrm{maxi}\,(R, G, B) \tag{7}$$

$$S = \begin{cases} 0 & \mathit{if}\ \max(R,G,B) = 0 \\ 1 - \dfrac{\min(R,G,B)}{\max(R,G,B)} & \mathit{otherwise} \end{cases} \tag{8}$$

$$H = \begin{cases} \dfrac{1}{6}\dfrac{G-B}{\max(R,G,B)-\min(R,G,B)} & \mathit{if}\ \max(R,G,B)=R \ \mathit{and}\ G<B \\ \dfrac{1}{6}\dfrac{B-R}{\max(R,G,B)-\min(R,G,B)}+1 & \mathit{if}\ \max(R,G,B)=R \ \mathit{and}\ G\geq B \\ \dfrac{1}{6}\dfrac{B-R}{\max(R,G,B)-\min(R,G,B)}+\dfrac{1}{3} & \mathit{if}\ \max(R,G,B)=G \\ \dfrac{1}{6}\dfrac{R-G}{\max(R,G,B)-\min(R,G,B)}+\dfrac{2}{3} & \mathit{if}\ \max(R,G,B)=B \end{cases} \tag{9}$$

3.2. HSV component analysis

Statistics of the HSV component values where determined by a sampling process. The sampling consisted on a set of 11samples from each door in the scenarios, see Figure 13. Each sample corresponds to a window of 5x5 pixels. The process involved the computation of the mean and variance of the mean distributions of the windows samples over the HSV values. Table 3 shows the mean distributions over the scenarios, while Table 4 presents the statistic values of the means.

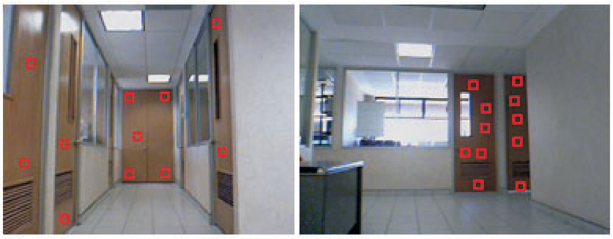

Fig. 13. Color door sampling in two scenarios.

Table 3. Mean distributions over the scenarios.

Scenario	H Mean	S Mean	V Mean
2	0.078	0.361	0.357
3	0.059	0.471	0.205
4	0.084	0.605	0.252
5	0.075	0.393	0.360
7	0.099	0.367	0.361
9	0.075	0.576	0.243
10	0.078	0.308	0.372

Table 4. Statistic values of the means.

Component	Mean	Standard deviation
H	$H = 0.078$	$\overline{H}_\sigma = 0.018$
S	$S = 0.440$	$\overline{S}_\sigma = 0.132$
V	$V = 0.307$	$\overline{V}_\sigma = 0.077$

3.3. Door segmentation

Door detection in autonomous robot navigation is an important issue because they appear in many interior environments [21]. Thus, doors are important landmarks that can be used for scene recognition. Door detection is achieved in this work by analysis of the HSV components implemented in the following condition

$$\text{if } |\overline{H} - H_p| < T_h \,\mathit{and}\, |\overline{S} - S_p| < T_s \,\mathit{and}\, |\overline{V} - V_p| < T_v$$

$$\mathit{Then}\ p(x,y) \text{ is a door pixel} \tag{10}$$

where H_p, S_p, and V_p are the HSV components of the pixel p at coordinates (x,y). The thresholds T_h, T_s, and T_v are

$$T_h = |\overline{H} - H_\sigma|,\ T_s = |\overline{S} - S_\sigma|,\ T_v = |\overline{V} - V_\sigma| \tag{11}$$

After the classification of the color pixels, blobs of less than 300 pixels are eliminated since they are not considered doors. Figure 14 illustrates some examples of doors detected by the previous method.

4. Recognition of scenarios

The recognition of the scenarios is achieved with a hierarchical neural network, HNN. This type of architecture was selected due to the similarity of some of the scenarios. The HNN is composed of two levels, Figure 15. The first level is composed by one neural network and the second by two neural networks.

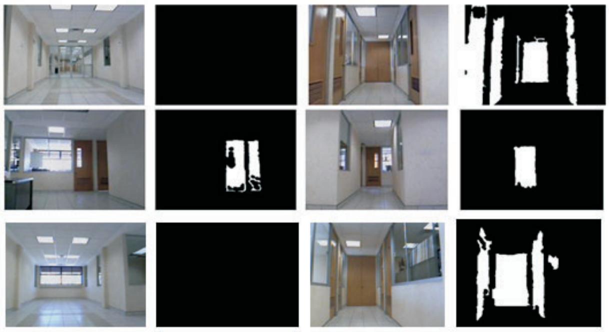

Fig. 14. Door segmentation.

The idea of this HNN is to separate the scenarios into 4 classes, and then use the second level to resolve more specific cases.

The first neural network is a feedforward backpropagation network with 32 inputs, 32 neurons in the hidden layer, and 4 output neurons, sigmoid tangent activation functions in the hidden layer, and sigmoid logarithmic functions in the output layer. This first neural network is trained to classify the four classes, see Figure 16, using the next feature vector

$$X_1 = \begin{bmatrix} C_{Pxi} \\ C_{Pyi} \\ h_i \\ a_i \end{bmatrix} \tag{12}$$

here C_{Pxi} and C_{Pyi} are the centroids of the blobs that corresponds to the doors, while h_i and a_i are the normalized height and width, respectively, of those blobs. Figure 17 presents examples of the door blobs with their respective centroids.

The neural network of the second level is trained to classify classes 2 and 4 into their corresponding scenarios, as shown in Figure 18, using the next feature vector

$$X_{2,3} = \begin{bmatrix} C_{Ex} \\ C_{Ey} \\ N \end{bmatrix} \tag{13}$$

where C_{Ex} and C_{Ey} are the centroids of the corner coordinates and N is the number of corners found in the scenario. Figure 19 presents examples of the corners with their respective centroids.

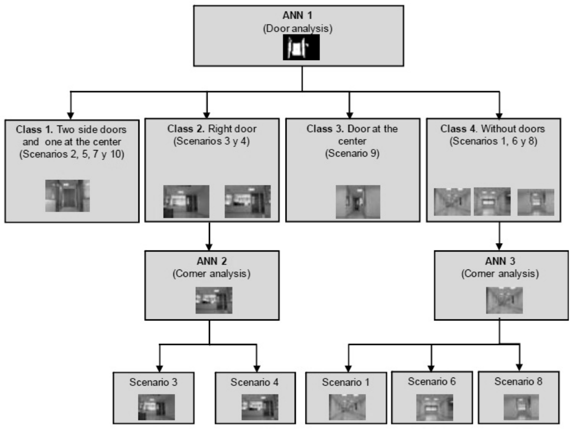
ANN 1
(Door analysis)
Class 1. Two side doors and one at the center (Scenarios 2, 5, 7 y 10)
Class 2. Right door (Scenarios 3 y 4)
Class 3. Door at the center (Scenario 9)
Class 4. Without doors (Scenarios 1, 6 y 8)
ANN 2
(Corner analysis)
ANN 3
(Corner analysis)
Scenario 3
Scenario 4
Scenario 1
Scenario 6
Scenario 8

Fig. 15. Hierarchical neural network scheme.

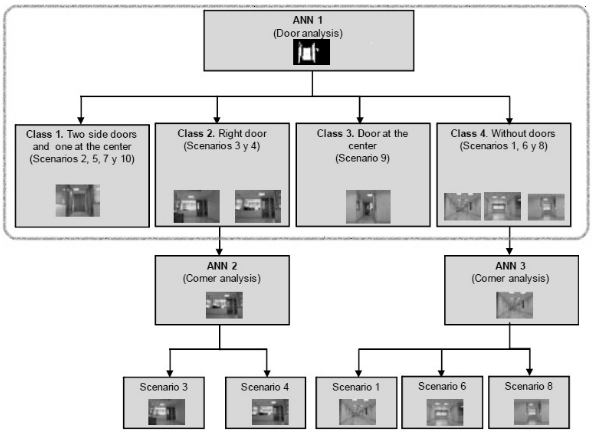
ANN 1
(Door analysis)
Class 1. Two side doors and one at the center (Scenarios 2, 5, 7 y 10)
Class 2. Right door (Scenarios 3 y 4)
Class 3. Door at the center (Scenario 9)
Class 4. Without doors (Scenarios 1, 6 y 8)
ANN 2
(Corner analysis)
ANN 3
(Corner analysis)
Scenario 3
Scenario 4
Scenario 1
Scenario 6
Scenario 8

Fig. 16. First level of classification.

The neural network of the second level is a feedforward-backpropagation, with 3 inputs, 26 neurons in the hidden layer, 2 output neurons, sigmoid tangent activation functions in the hidden layer, and sigmoid logarithmic functions in the output layer.

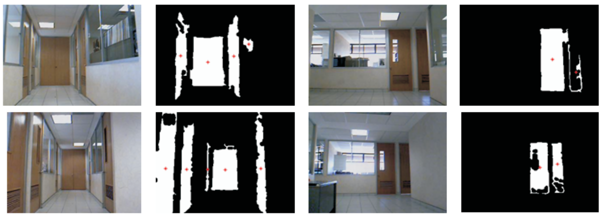

Fig. 17. Examples of door centroids.

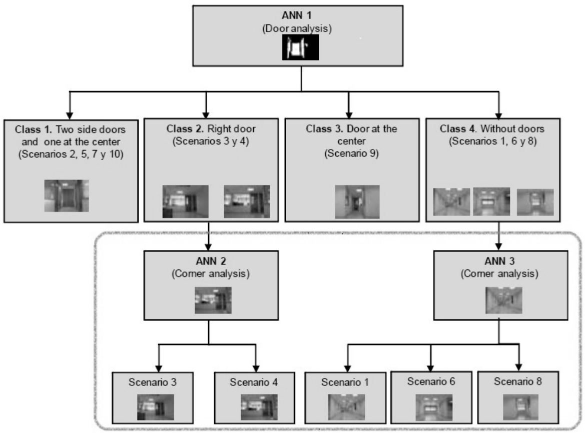
ANN 1
(Door analysis)
Class 1. Two side doors and one at the center (Scenarios 2, 5, 7 y 10)
Class 2. Right door (Scenarios 3 y 4)
Class 3. Door at the center (Scenario 9)
Class 4. Without doors (Scenarios 1, 6 y 8)
ANN 2
(Corner analysis)
ANN 3
(Corner analysis)
Scenario 3
Scenario 4
Scenario 1
Scenario 6
Scenario 8

Fig. 18. Second level of classification.

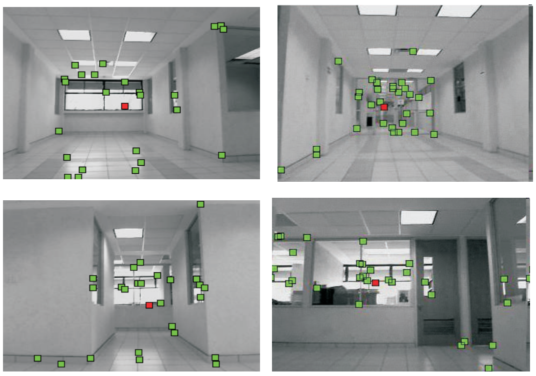

Fig. 19. Examples of corner centroids.

Results of the scenario recognition are commented next. The performance by levels of the HNN over the 10 scenarios considering the two levels is illustrated in Figure 20. The first classification was achieved by door detection using the centroids, height and area of the door blobs. The ANN of the first level was trained to classify the 10 scenarios into 4 classes. This ANN had 90% of correct classification during training, and 95% in validation. Class 1 that contains the scenarios 2, 5, 7, and 10 was considered as one type of scenario because of the high degree of similarity among the four scenarios. This similarity turns to be hard to resolve even for human observers. Class 3 was not reclassified because it only contains images of scenario 9. Regarding the classification of scenarios in the classes 2 and 4, in the second level, it was performed by using corner detection information as well as the number of corners. ANNs 2 and 3 were trained with this information. The ANN 2 separated class 2 into scenario 3 and 4 wih a performance of 90.90% in training and 87.5% during validation. The neural network 3 that classifies class 4 into the scenarios 1, 6, and 8 has a performance of 93.75%

in training, and 91.66% in validation.

The total performance of the systems considering all the scenarios was 86.66% for training, and 90% in validation.

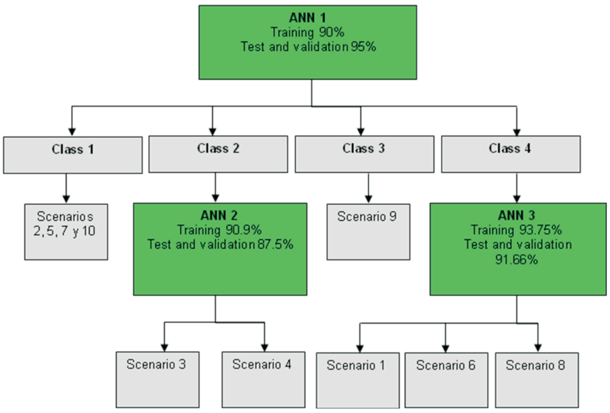

Fig. 20. System performance for levels.

5. Results

Two important results are derived from this work. The first one is related to the proposed corner detector method and the second to the recognition for scenarios. In regards the corner detector method we can mention that the proposed method has similar hit performance in semi-artificial scenarios as the Harris detector, 99% and 98%, and false negatives 1% and 2%, respectively. However the proposed method is better with respect to false positives, 13.43% and 25.86%. On the other hand, the use of combined information, door and corner information, provide important discriminative data validated by the hierarchical ANN. The ANN findings present an adequate overall performance of 86.66% for training, and 90% in validation.

6. Conclusions

In conclusion, it can be said that the proposed corner detector method shows good corner detection in semi-artificial as well as in real indoor and outdoor scenarios as it was validated in the corner detection experiments performed in this research. Besides, the corner detection method provides correct information that is validated with the performance achieved in the ANNs 2 and 3.

Another important conclusion is that the proposed solution to the scene recognition problem based on fusion of color and corner features proved to be effective based on the experimental results obtained in this work.

Results shown in this research confirm that complex problems like scene recognition for robot navigation are well faced with information fusion where different type of information complements each other.

ACKNOWLEDGMENTS

The authors thanks to Fondo Mixto de Fomento a la Investigación Científica y Tecnológica CONACYT - Gobierno del Estado de Chihuahua, by the support of this research under grant CHIH-2009-C02-125358. This work was also supported by SEP-DGEST under Grants 2173.09-P and 2172.09-P.

AUTHORS

Mario I. Chacon-Murguia*, Rafael Sandoval-Rodriguez, Cynthia P. Guerrero-Saucedo - Visual Perception on Robotic Applications Lab Chihuahua Institute of Technology, Chihuahua, Mexico. E-mails:
{mchacon, rsandoval, cpguerrero}@itchihuahua.edu.mx.
* Corresponding author

References

[1] Kemp C., Edsinger A., Torres-Jara D., "Challenges for Robot Manipulation in Human Environments", *IEEE Robotics & Automation Magazine*, March 2007, pp. 20-29,.

[2] Durrant-Whyte H., Bailey T., "Simultaneous Localization and Mapping (SLAM): Part I*", IEEE Robotics & Automation Magazine*, June 2006, pp. 99-108.

[3] Bailey T., Durrant-Whyte H., "Simultaneous Localization and Mapping (SLAM): Part II", *IEEE Robotics & Automation Magazine*, September 2006, pp. 108-117.

[4] Addison J., Choong K., "Image Recognition For Mobile Applications". In: *International Conferences on Image Processing ICIP 2007*, 2007, pp. VI177-VI180.

[5] DeSouza G., Kak A., "Vision for Mobile Robot Navigation: A Survey", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 24, no. 2, February 2002 , pp. 237- 267.

[6] Kelly A., Nagy B., Stager D., Unnikrishnan R., "An Infrastructure-Free Automated Guided Vehicle Based on Computer Vision", *IEEE Robotics & Automation Magazine*, September 2007, pp. 24-34.

[7] Srinivasan M.V., Thurrowgood S., Soccol D., "Competent Vision and Navigation Systems", *IEEE Robotics & Automation Magazine*, September 2009, pp. 59-71.

[8] Antoine Maint J.B., Viergever M.A., "A Survey of Medical Image Registration", *Medical Image Analysis*, 1998, vol. 2, no. 1, pp. 137.

[9] Zitova B., Flusser J., "Image Registration Methods: a survey", *Image and Vision Computing*, vol. 21, 2003, pp. 977- 1000.

[10] TissainayagamP.,Suter D., "Assessing the Performance of Corner Detectors for Point Feature Tracking Applications", *Image and Vision Computing*, vol. 22, 2004, pp. 663-679.

[11] Mokhtarian F., Suomela R., "Curvature Scale Space for Robust Image Corner Detection". In: *International Conference on Pattern Recognition*, Brisbane, Australia, 1998.

[12] Andrade J., *Environment Learning for Indoor Movile Robots*. PhD Thesis, Universidad politecnica de Catalunya, 2003.

[13] Rangarajan K., Shah M., van Brackle D., "Optimal Corner Detector", *2^{nd} International Conference on Computer Vision*, December 1988, pp. 90-94,

[14] Smith S.M., Brady J.M., "SUSAN - A New Approach to Low Level Image Processing". *Int. Journal of Computer Vision*, vol. 23, no. 1, May 1997, pp. 45-78.

[15] Harris C.G., Stephens M., "A combined corner and edge detector". In: *Proceedings of the Alvey Vision Conference*, Manchester 1988, pp. 189-192.

[16] He X.C., Yung N.H.C., "Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region

of Support". In: *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, August 2004, pp. 791- 794.

[17] Aguilar G., Sanchez G., Toscano K., Salinas M., Nakano M., Perez H., "Fingerprint Recognition". In: *2nd International Conference on Internet Monitoring and Protection - ICIMP 2007*, July 2007, pp. 32-32.

[18] Leung W.F., Leung S.H., Lau W.H., Luk A., "Fingerprint Recognition Using Neural Network, Neural Networks for Signal Processing". In: *Proceedings of the 1991 IEEE Workshop*, 30th Sept.1st Oct. 1991, pp. 226-235.

[19] Chen Z., Birchfield S.T., "Visual detection of lintel-occluded doors from a single image", In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops 2008*, pp. 1-8.

[20] Schwarz M.W., Cowan M.,W., Beatty J. C., "An experimental comparison of RGB,YIQ, LAB, HSV, and opponent color models", *ACM Transactions on Graphics*, vol. 6, issue 2, 1997, pp. 123-158.

[21] Cariñena P., Regueiro C., Otero A., Bugarín A., Barro S., "Landmark Detection in Mobile Robotics Using Fuzzy Temporal Rules", *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, August 2004, pp. 423-235.