Corner Detection Using Kernel Windows Matching For Scene Recognition

Mario I. Chacon-Murguia[1], Cynthia P. Guerrero-Saucedo[2], Rafael Sandoval-Rodriguez[2]

[1] DSP & Vision Laboratory, [2]Robotic Laboratory
Chihuahua Institute of Technology, Chihuahua, Mexico
{mchacon, cpguerrero, rsandova}@itchihuahua.edu.mx

Abstract. Corner detection is an important step in applications of object recognition, scene recognition, and stereo vision, among many others. The purpose of this paper is to present a corner detection method for scene recognition in autonomous navigation of mobile robots. The proposed method is based on thirty windows to find possible corners. Some of the proposed windows are taken from fingerprint recognition papers and other are derived in this paper. The set of windows are used through a matching process to detect corners presented in the image under analysis. Results of the proposed method yield better quantitative and qualitative results compared with the Harris and SUSAN methods. These results can be used by a pattern recognition system for scene recognition.

Keywords: Scene recognition, corner detection, robotics.

1 Introduction

Corner detection is used in areas such as robotics, medicine. In robotics, it is used for data fusion, navigation, and scene recognition [1]. In medicine, it is applied for image registration such as x-rays, ultrasounds, and medical diagnostics [2]. In other applications, corner detection is used for object recognition, stereo vision, motion detection, among many other usages [3] [4].

Corners are the features more abundant in images of the real world, in contrast to straight lines [5]. 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 [6], 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 [7], 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. [8], 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 paper is organized in the following sections. In Section 2, some of the more used methods for corner detection are discussed. Section 3 provides a general glance of the method proposed in this paper for corner detection. Sections 4 and 5 describe the main steps of the corner detection method. Results and conclusions are presented in Section 6.

2 Corner detection methods

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 [9] and the method of Harris [10]. 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 [11]. 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. [12]. In a similar way, G. Aguilar et al. [13] compare images of finger prints 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., [14], 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 paper 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 next section explains the corner detection process developed in this paper.

3 General Description of the Corner Detection Method

The process for corner detection is shown in Figure 1. 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 $I_s(x,y)$. A gradient operator and a threshold to determine the edges are applied to the image $I_s(x,y)$. 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.

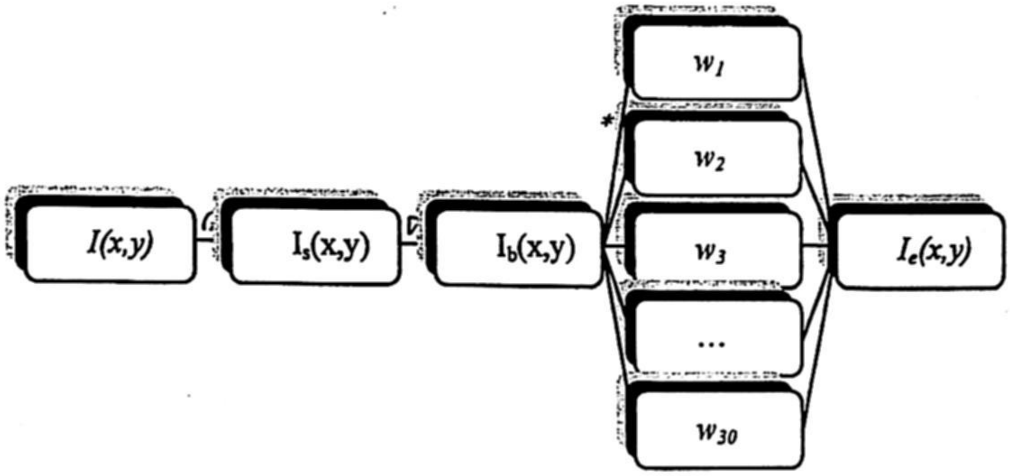


Fig. 1. Corner detection process.

4 Corner Detection Method

4.1 Corner detection windows

The papers from G. Aguilar [13], and W.F. Leung et al. [14], 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 2. 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 finger print recognition works, some windows were chosen to detect corners. These selected windows plus other proposed in this paper make a set of 30 windows. Each corner detection window, w_c, is a 3X3 mask and some of their structures are illustrated in Figure 3.

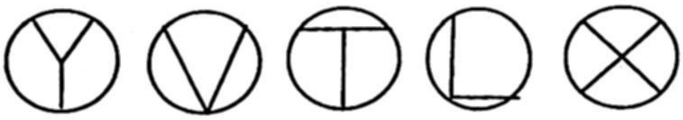


Fig. 2 Type of corners.

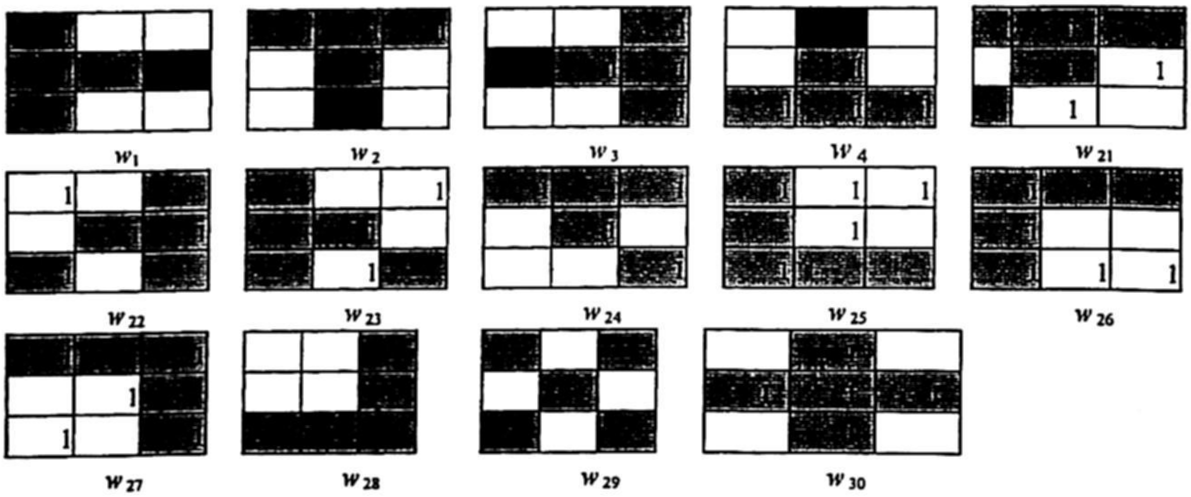


Fig. 3 Proposed corner detection windows.

The set of w_c windows is composed as follows. Windows w_1, w_2, w_3, and w_4, are four windows modified from the work of Leung at al. [14]. 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. [13]. 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.

4.2 Corner detection

Corner detection is achieved through a windows matching process. The image is scanned with the different corner detection windows w_c, 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 generate a generic weight matrix T_n, starting at $p(1,-1)$ as shown in Figure 4.

2^8	2^5	2^2			256	32	4
2^7	2^4	2^1	=		128	16	2
2^6	2^3	2^0			64	8	1

Fig. 4. Generic weight matrix, T_n.

Using the generic weight matrix T_n each w_c can be associated with an index window B_i

$$T_n : w_c \Rightarrow B_i \qquad for \quad c,i = 1,..,30 \tag{6}$$

obtained by

$$B_i = w_c \times T_n \ . \tag{7}$$

where the multiplication is an element by element multiplication and not a matrix multiplication. In this way, each w_c window is related to a 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 equation (8).

$$\alpha_i = 1 + \sum_{b_i \in B_i} b_i \ . \tag{8}$$

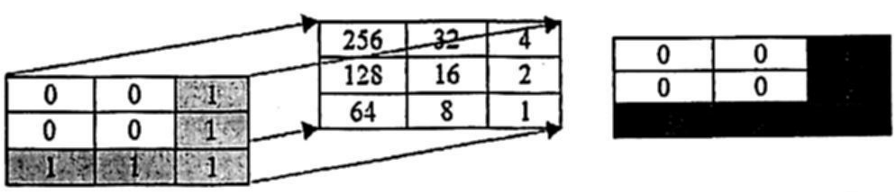
where the b_i correspond to the weighting factor in B_i. Figure 5 illustrates the computation of the index window B_{28} and the total weighting factor α_{28} of the window w_{28}. Once the main definition of the proposed methodology has been defined, the steps to detect the corners are now described.

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

$$I_{ci}(x,y) = I_b(x,y) * B_i + 1 \ . \tag{9}$$

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

$$p_{ei}(x,y) = \begin{cases} 1 & p_{ci}(x,y) = \alpha_i \\ o & \text{otherwise} \end{cases} \tag{10}$$



w_{28} T_n B_{28} $\alpha_{28} = 80$

Fig. 5. Computation of the index window B_{28} and the total weighting factor α_{28} of w_{28}.

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

$$I_{FC}(x,y) = \bigcup_{i=1}^{30} I_{ei}(x,y) \ . \qquad (11)$$

5 Results

The proposed method was tested in semi-artificial scenes as well as with a set of real scenes. The purpose of using semi-artificial scenes is to obtain a correct performance measure of the proposed method, which will be hard to compute in real (noisy) scenes. These kind of images allows to compute false positives and false negatives detections in a simpler form and to achieve an analysis of these cases. A semi-artificial image is a simplified image extracted from a real scene. Figure 6 shows the semi-artificial version of three scenes.

Results of the application of the proposed method to the scenarios are shown in Figure 7. The first figure indicates the corners to be detected and the second the detected corners by the proposed method. There are 32 corners to detect in scene 1. The method detects 36 corners, therefore there are 4 false positive detections. The scene 2 contains 42 corners. The method found 40 corners, and there are 2 false negatives corners. In the scene 3 the method detected 61corners, 6 of them are false positives. These results yield 98.45% of real corner detections, 7.75% of false positive detection, and 1.56% of false negatives. The results of the Harris method using the same scenes are 96.9% of real corner detections, 33.33% of false positive, and 3.1% of false negative. Comparison with the SUSAN algorithm is not possible because it requires gray level information. A qualitative comparison will be given over the original images later on.

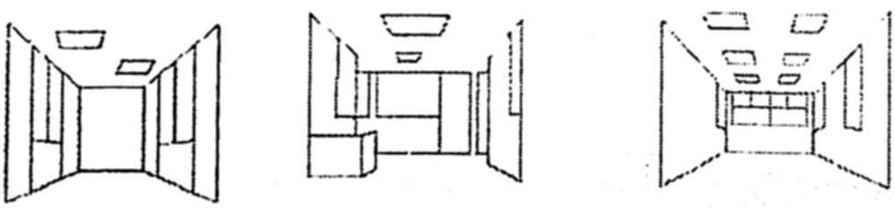


Fig. 6. Semi-artificial scenes, 1, 2 and 3.

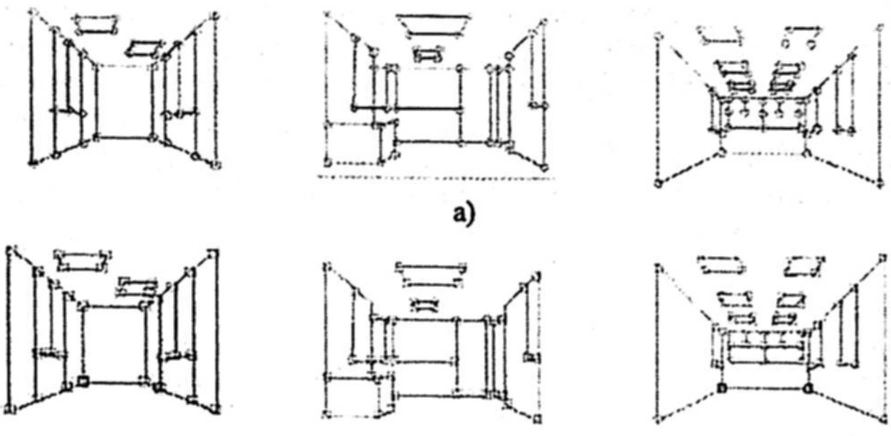


Fig. 7. a) Corners to be detected and b) the corresponding detected corners by the proposed method for scenes, 1, 2 and 3.

False negatives are mainly due to corners that do not match exactly to any w_c. 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 close to the corner. Figure 8 shows two multiple detection due in scene 1. Figure 8b illustrates the corner structures that make a match with the w_5, and w_8, windows.

The final results commented in this work are related to the comparison of the proposed method with the Harris and SUSAN methods applied in real scenes. Figure 9 show the corners detected by Harris (a), SUSAN (b) and the proposed method (c).

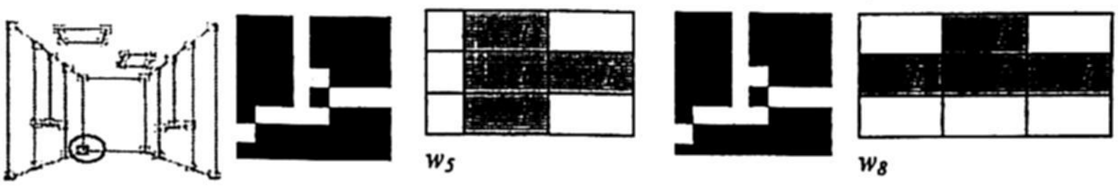
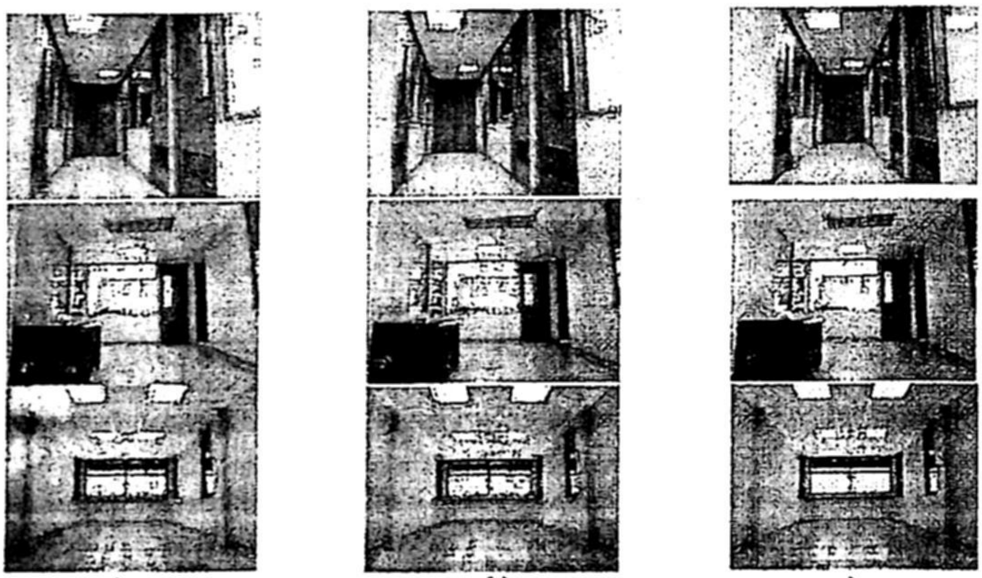


Fig. 8. False positives in scene 1.

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.

6 Conclusions

The proposed method has a better quantitative performance on semi-artificial scenes than the Harris method; and a better qualitative performance, with real scene images, than the Harris and SUSAN methods. Scene recognition using features obtained from the detected corners presents preliminary acceptable results in a neural network scene recognition system like in [15]. The performance of this system is around 85% of correct recognition considering 10 different scenes considering different distance and orientations, four of them very alike, which makes the recognition harder. Future work includes more feature extraction experimentation over the detected corners, and the design of new scene recognition systems.



a) b) c)

Fig. 9. Corner detection by a) Harris b) SUSAN and c) Proposed method.

Acknowledgement

This work was supported by SEP-DGEST under Grants 2173.09-P and 2172.09-P.

7 References

[1] Se, S., Lowe, D., Little, J.: Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. International Journal of Robotics Research, pp. 735–758. (2002).

[2] Antoine Maintz, J. B., Max A. Viergever.: A Survey of Medical Image Registration. Medical Image Analysis, vol. 2, number 1, pp. 1–37, Oxford University Press (1998).

[3] Zitova, B., Flusser, J.: Image Registration Methods: a survey. Image and Vision Computing 21, pp. 977–1000, (2003).

[4] Bradski, G., Kaehler, A.: Robot-Vision Signal Processing Primitives. IEEE Signal Processing Magazine, pp. 130- 133, January (2008).
[5] Tissainayagam, P., Suter, D. .: Assessing the Performance of Corner Detectors for Point Feature Tracking Applications. Image and Vision Computing, vol. 22, pp. 663-679, Elsevier (2004).
[6] Mokhtarian, F., Suomela, R. .: Curvature Scale Space for Robust Image Corner Detection. Int'l Conf. Pattern Recognition, Brisbane, Australia (1998).
[7] Andrade, J.: Environment Learning for Indoor Movile Robots. PhD Thesis, Universidad politecnica de Catalunya, (2003).
[8] Rangarajan, K., Shah, M., van Brackle, D.: Optimal Corner Detector. Second International Conference on Computer Vision, pp. 90 – 94, December (1988).
[9] Smith, S.M., Brady, J.M.: SUSAN - A New Approach to Low Level Image Processing. Int. Journal of Computer Vision, vol. 23 (1), pp.45-78, May (1997).
[10] Harris, C.G., Stephens, M.: A combined corner and edge detector. In proceedings of the Alvey Vision Conference, pp. 189-192. Manchester (1988)
[11] He, X.C., Yung, N.H.C.: Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support. Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, pp: 791- 794. Aug. (2004).
[12] Zitová B., Flusser J., Kautsky J., Peters G.: Feature Point Detection in Multiframe Images. Proceedings of the Czech Pattern Recognition Workshop (Svoboda T. ed.), Czech Pattern Recognition Society, pp. 117-122, Praha (2000).
[13] Aguilar, G., Sanchez, G., Toscano, K., Salinas, M., Nakano, M., Perez, H.: Fingerprint Recognition. Second International Conference on Internet Monitoring and Protection, ICIMP 2007, pp. 32 – 32, July (2007).
[14] Leung, W.F., Leung, S.H., Lau, W.H., Luk, A.: Fingerprint Recognition Using Neural Network, Neural Networks for Signal Processing [1991], Proceedings of the 1991 IEEE Workshop, pp. 226 – 235, 30 Sept.-1 Oct. (1991).
[15] Wang, S., Lei, S., Chang, F.: Image registration Based on Neural Network. Proceedings of the 5th , International Conference on Information Technology and Application in Biomedicine, in conjunction with The 2nd International Symposium & Summer School on Biomedical and Health Engineering, pp 74-77, Shenzhen, China, May (2008).