

## A Corner Detector Algorithm for Feature Extraction in Simultaneous Localization and Mapping

Carlos Alberto Velásquez Hernández\* and Flavio Augusto Prieto Ortiz

*Mechanics and Mechatronic Department – Universidad Nacional de Colombia, Colombia*

Received 8 February 2019; Accepted 11 July 2019

### Abstract

Map merging is one of the most studied problems in Multi-SLAM, since it allows the deployment of teams of robots in unknown environments without any initial condition. The main challenge at dealing with map merging algorithms, who work with graphical SLAM maps, is to become them usable in real-time applications. Since in our work this problem is approached from the computer vision perspective, we initially proposed a new corner detector as the first step of our map merging algorithm. This corner detector showed to be prominent in the field of feature extraction. Thus, in this paper, we present a valuable extension and a deep comparison between our corner detector and the most known feature extractors in computer vision. The results show that the proposed corner detector has an efficient and trustworthy performance at extracting meaningful features, either in graphical SLAM maps or in normal images. The results also show that our detector takes less time than the other analyzed feature extractors, allowing its use in real-time applications.

*Keywords:* Corner detector, SLAM, Occupancy grid map, Feature extractor

### 1. Introduction

Last decade, map merging has been one of the most studied problems in Multi-SLAM (Tungadi et al. [1], Lee [2], Lee et al. [3], Wang et al. [4], Kojima, Okawa and Namerikawa [5], Cortés and Serratosa [6], Lee, Roh and Lee [7]), since it deals with different maps provided by autonomous agents in an environment. The main goal is to get a complete, consistent and unique map of the environment from partial maps. Map merging approaches solve this issue using different strategies such as setting-up initial operation conditions (robots start from a single point of the environment or start from different known points), scheduling real encounters in the environment (Rendez-vous case), like the approach proposed by Tungadi et al. [1], Wang et al. [4], Zhou and Roumeliotis [8], Howard [9], or doing an exhaustive search over the maps to find common areas. The latter strategy is the most challenging, because it does not use any initial condition and it does not depend on real encounters. A map merging method based on the Computer Vision field was proposed in Blanco, et al. [10], they addressed the map merging problem as an image matching problem using graphical maps which are known as occupancy grid maps (OGM). Authors made an analysis of different corner detectors and descriptors, selecting the best ones and combining them with matching, aligning and filtering algorithms to get the best merged map.

From the studies made by Blanco, et al. [10], Blanco et al. [11], we initially proposed a Corner detector approach Velásquez and Prieto [12] that showed having a suitable performance in feature extraction step. First results, reported in Velásquez and Prieto [12], showed that our approach

extracts robust features from OGMs in a shorter time than the other feature extractors, since it was entirely designed with the properties of OGMs. This paper is a reformulation of the initial results obtained with our proposed corner detector. We made important improvements in the properties of the corner detector such as the orientation assignment, the number and quality of the extracted features and a new image analysis in low resolution. We carried out tests with a dataset of 40 OGMs built in real and simulated scenarios and, as an important result, we also tested our corner detector with 20 standard RGB images that it performs well with any type of image.

The proposed corner detector is compared with well-known features extraction techniques such as Harris detector, Shi-Tomasi detector, Trajkovic corner detector and the SIFT keypoint extractor. All feature techniques were programmed in C++ using the official packages provided by OpenCV 3.0.0. We did not modify the code to preserve the original implementation of the authors. Furthermore, we analyze the time consumption of the algorithms, the number and quality of the extracted features. Those parameters can explain how robust a feature extractor is in OGMs and in standard images.

This document is divided as follows: Section 2 briefly explains the corner detectors used in our tests. Section 3 describes extensively our corner detector, showing in detail the improvements and the extracted features. In Section 4, we describe and discuss the results obtained from the tests we carried out. Finally, Section 5 presents some conclusions and future work.

### 2. Corner Detectors

In image processing, Corner detectors are the most known and used methods for feature extraction. Works detailed in Tuytelaars and Mikolajczyk [13], Montero, Stojmenovic and

\*E-mail address: cvelasquez@unal.edu.co

ISSN: 1791-2377 © 2019 Eastern Macedonia and Thrace Institute of Technology. All rights reserved.

doi:10.25103/jestr.123.15

Nayak [14] report tests and comparisons among them. Also, a full and detailed study of the most used and known techniques of feature extraction is detailed in Li and Nigel [15].

Basically, Corner detectors are responsible for selecting the most representative feature points from images. Features points comprise corners, edges or distinctive regions around points, and they always try to detect the center of the feature. Those kinds of features always contain pronounced image gradients, so the extractor (corner detector or feature extractor in this paper) can easily identify where they are in the image. The most difficult issues for feature extractor are to find the correct center of the feature and to extract a feature with a low image gradient when the image has blurred regions or illumination changes. In this section, we present and focus on the most representative Corner detectors in Computer Vision, since they have always been used to make comparisons when a new technique in this field is formally presented.

### 2.1. Harris corner detector

Based on the initial corner detector developed by Moravec [16], Harris and Stephens [17] proposed a corner detector which solves the deficiencies presented in the Moravec technique. They specially modified the way the image gradient and the corner measure were calculated. This technique computes a matrix tensor  $M$  for each pixel, Eq.1.

$$M = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \quad (1)$$

This matrix tensor includes the image derivatives  $I_x, I_y$  in  $x$  and  $y$  axis, which were previously submitted to a convolution process with a Gaussian kernel  $w$ . Then, the Harris corner measure is calculated, Eq.2:

$$m_h = \det(M) - k \cdot \text{tr}^2(M) \quad (2)$$

Where  $k$  is a constant established between  $0.04-0.08$ .  $\det(M)$  is the determinant of the tensor  $M$ , and  $\text{tr}(M)$  is the trace of the same tensor. Finally, the corner measure  $m_h$  of each image pixel (or image point) is submitted to the next conditions to establish whether the point is a corner or not:

- If  $m_h > 0$  and its value is small, the point is considered to be on a homogeneous area.
- If  $m_h < 0$ , the point is on an edge.
- If  $m_h \gg 0$ , the pixel is considered to be on a corner.

This approach showed an advance in the detection of feature points. However, it presents a scalability problem, since the more image data points are contained in the image, the more image processing time is required and the less accuracy in feature detection is obtained. It also presents a limitation with noisy images, since it still depends on the image gradient information. A possible solution is to increase the Gaussian kernel, but this could increase the image processing time. An improvement of this corner detector is shown by Junxiong and Kai [18].

### 2.2. Shi-Tomasi detector

Tomasi and Kanade [19] introduced a modification to the original corner detector proposed by Harris. Essentially, they changed the corner measure set by Harris: while Harris uses the corner measure showed in Eq.1, Shi-Tomasi makes a selection of the minimum eigenvalue of the matrix (Eq.3), that is:

$$m_{shi} = \min(\lambda_1, \lambda_2) \quad (3)$$

This simple modification increased the corner detector performance, because this corner measure shows better the real image gradient information. In fact, Kenney, Zuliani and Manjunath [20] proved, through an axiomatic study, that the Shi-Tomasi detector is better than the Harris corner detector, because Shi-Tomasi complied with the axioms proposed for their study. Recently, Das, Pukhrambam and Saha [21] showed a real-time application of the Shi-Tomasi detector in face detection and facial expression recognition. However, these modifications do not solve the thresholding problem, because the threshold is responsible for accepting or rejecting points after the calculation of corner measures.

### 2.3. Trajkovic corner detector

Proposed by Trajkovic and Hedley [22], it performs an analysis of the closest neighbors to a point (pixel). It performs the corner detection using the closest neighbors of a pixel in low and high resolution of the image. In Fig. 1, a pixel  $C$  is shown with its 4 direct neighbors (points  $A, A', B$  and  $B'$ ). However, this method has an extension that includes the 8 closest neighbors (points  $A, A', B$  and  $B', P, P', Q$ , and  $Q'$  in Fig. 1).

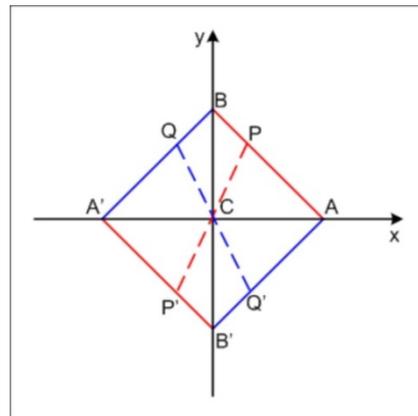


Fig. 1. The closest neighbors associated with a point  $C$  in the image that is being analyzed.

The Trajkovic detector is divided in 3 steps. First step performs a low resolution analysis of the image. In fact, it is common to reduce the image by half of the original resolution to perform the analysis in low resolution. Once the image is obtained in low resolution, it uses the following equations to determine the horizontal  $r_A$  and vertical  $r_B$  intensity variation:

$$\begin{aligned} r_A &= (f_A - f_C)^2 + (f_{A'} - f_C)^2 \\ r_B &= (f_B - f_C)^2 + (f_{B'} - f_C)^2 \\ R &= \min(r_A, r_B) \end{aligned} \quad (4)$$

Where  $f_A, f_B, f_{A'}, f_{B'}$  and  $f_C$  are the intensity image information of points  $A, B, A', B'$  and  $C$ , respectively.  $R$  is the minimum value between  $r_A$  and  $r_B$ . Two thresholds  $T_1$  and  $T_2$  are defined according to the knowledge of the analyzed images. So, if  $R > T_1$ , the point is classified as potential feature point.

In the second step, the detector performs a high resolution analysis with the potential candidates extracted in the first step. This consideration considerably reduces the search space of feature points. In this step, the detector uses the same metrics applied on the first step and it only takes into account the points extracted previously. Nevertheless, the only modification in this step is the use of a different selection threshold:  $R > T_2$ .

In the last step, points that reach the last step are analyzed under the following equations:

$$\begin{aligned} B_1 &= (f_B - f_A)(f_A - f_C) + (f_{B'} - f_{A'})(f_{A'} - f_C) \\ B_2 &= (f_B - f_{A'})(f_{A'} - f_C) + (f_{B'} - f_A)(f_A - f_C) \\ B &= \min(B_1, B_2) \\ A &= r_B - r_A - 2B \end{aligned} \quad (5)$$

$B_1, B_2$  are measures based on the intensity variations around the pixels  $A, B, A', B'$  and  $C$ .  $B$  is the minimum value taken between  $B_1$  and  $B_2$ , and  $A$  is a value related to  $r_A, r_B$  and  $B$ . Then, if  $B < 0 \wedge A + B > 0$ , the detector uses  $R = r_A - \frac{B^2}{A}$ . Else,  $R = \min(r_A, r_B)$ . A point is selected as a feature point if  $R > T_2$ . Finally, a non-maximal suppression step is performed in order to guarantee the quality of the best feature points.

#### 2.4. SIFT keypoint extraction

As a reference in Image Description and Matching, Lowe [23] created a well-known technique called Scale-invariant Feature Transform (SIFT) for image extraction, description, and matching. This technique proved to be more robust than other techniques, since Lowe achieved to show that his extracted features are "invariant to image scaling, translation, and rotation, and partially invariant to illumination changes", Lowe [23].

This technique starts with the detection of the most distinctive points, also called keypoints, in the space-scale extreme. To do that, Lowe proposed the use of Difference of Gaussian (DoG) at different scales, so DoG can be computed as the difference between two nearby blurred and scale images, which are separated by a constant parameter.

As limitations, the SIFT keypoint extractor requires a long processing time to perform the points extraction, although an improvement of this technique was introduced by Lowe [24] to deal with this issue. Furthermore, there are also works presented in Daixian [25], Wang et al. [26], whose main objective is to make improvements in the computation time of this algorithm. However, it is clear that this method introduced a good approximation to the ideal feature extractor, since it involves the greatest amount of ideal properties for a feature detector.

### 3. The proposed corner detector algorithm for SLAM

Our approach takes the analyzed performed by Trajkovic: it analyzes the 8 closest neighbors to a point to classify it as a corner. This detector can make a basic thresholding on OGM images, since they are composed of 3 levels in the grayscale (0, 127 and 255), or it can use a combination between border filters and thresholding steps to reduce the grayscale on normal images (in this paper, normal images refer to real images or images with a high grayscale). Then, it applies some metrics to find the features shown in Fig. 3, 4, 5. This approach presents an extension to standard images, a greater number of detected features and also an assignment of the feature orientation.

#### 3.1. General algorithm

Let  $I$  be an image and  $I(x, y)$  be the image intensity of the point  $x, y$ . A point is considered if its image intensity is white,  $I(x, y) = 1$  (considering a normalized scale). This assumption ensures that feature points are on the center of a feature, since the image gradient in corners is larger around the point. Fig. 2 shows the analyzed pixel,  $C$ , and the 8 closest neighbors  $A, A', B, B', P, P', Q$  and  $Q'$ .

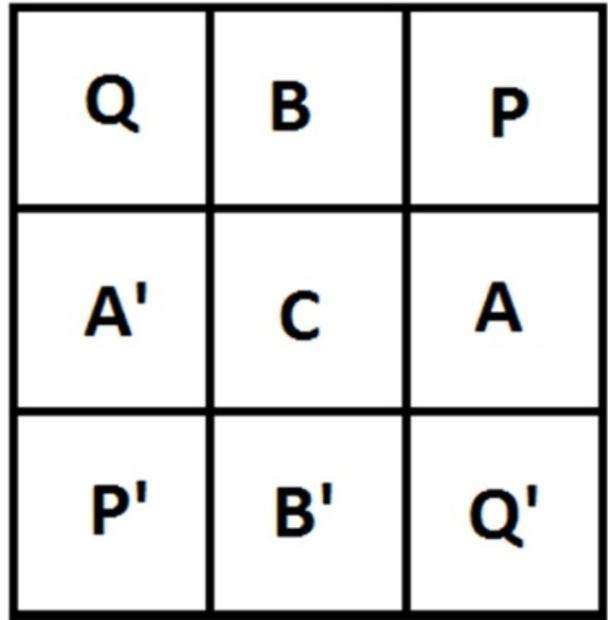


Fig. 2. Analyzed pixel  $C$  with its 8 nearest neighbors ( $A, A', B, B', P, P', Q$  and  $Q'$ ).

$$r_{\text{sup}} = P + B + Q$$

$$r_{\text{der}} = P + A + Q'$$

$$r_{\text{inf}} = P' + B' + Q'$$

$$r_{\text{esqp}} = A + P + B$$

Figure 2. Analyzed pixel  $C$  with its 8 nearest neighbors ( $A, A', B, B', P, P', Q$  and  $Q'$ ). The first 4 neighbors are called direct neighbors, while the last 4 ones are called diagonal neighbors.

The algorithm performs:

- i. Calculate the measures  $r_s, r_d$  as:

$$\begin{aligned}
 r_s &= A + A' + B + B' \\
 r_2 &= P + P' \\
 r_4 &= Q + Q' \\
 r_D &= r_2 + r_4
 \end{aligned}
 \tag{6}$$

ii. Determine a measure  $r$ , which denotes the number of pixels with value 0 on the grayscale,  $r = r_s + r_D$ .

iii. If  $r = 4, 5, 6$ , calculate the following measures:

Analyze the measure  $r$  under the following assumptions:

- If  $r = 4$ , then:  $\text{mod}(r_{esqp}) = 0 \wedge r_s = 2 \wedge r_{sup} + r_{inf} = 3 \wedge r_2 \neq r_4$ . This

criterion selects points with the 4 patterns shown in Fig. 3.

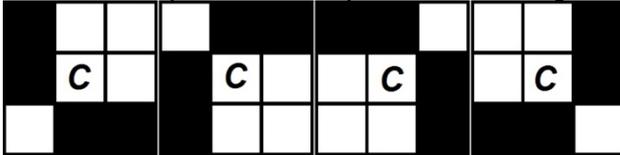


Fig 3. Feature points extracted when  $r = 4$ .

For the feature orientation, the assignment is made on the free diagonal of these patterns, *i.e.*, on the white pixels. The orientation values correspond to the angles  $45^\circ$ ,  $135^\circ$ ,  $225^\circ$  and  $315^\circ$ . This assignment considers the coordinate system of the image, so the user can define which are points  $A$  and  $B$  on the straight neighbors. Point  $A$  corresponds to the  $X$  axis, while point  $B$  corresponds to  $Y$  axis. The direction of the axes is given by the vectors  $\overline{CA}$  and  $\overline{CB}$ .

- If  $r = 5$ , apply the following selection criteria:  $r_s = 2 \wedge \text{mod}(r_{sup} + r_{der}) = 0$ . This criterion selects points with the 4 patterns shown in Fig. 4.

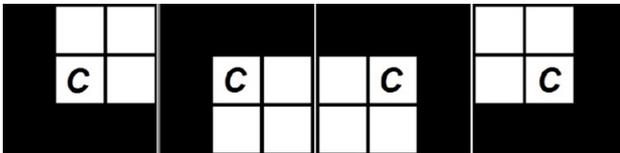


Fig. 4. Feature extracted when  $r = 5$ . It shows the ideal corner around the analyzed point  $C$ .

Again, the assignment is made on the white pixels of the 8 neighbors: the free pixel of one of the diagonals is taken into account. These angles correspond to the value of  $45^\circ$ ,  $135^\circ$ ,  $225^\circ$  and  $315^\circ$ . The coordinate system is the same as the one explained above.

- If  $r = 6$ , apply the following selection criteria:  $r_s = 3 \wedge r_D = 3$ . Fig. 5 shows the patterns extracted with this criteria. Features with other configuration are rejected, because they are interpreted by the algorithm as noise.

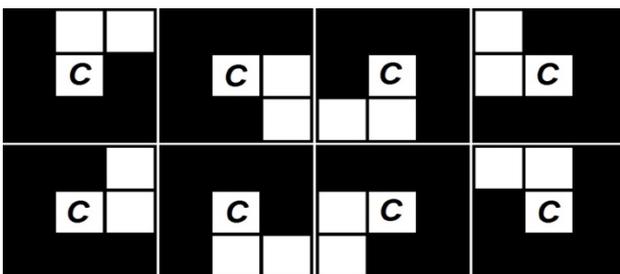


Fig. 5. Features extracted when  $r = 6$ . They look strange, but they appear in irregular map images, so they were considered as features.

For assigning the feature orientation, the algorithm considers the only two free (or white) pixels that are on the edge of the region. These pixels, together with the pixel  $C$ , form opening angles of  $22.5^\circ$ . These angles correspond to the values of  $22.5^\circ$ ,  $67.5^\circ$ ,  $112.5^\circ$ ,  $157.5^\circ$ ,  $202.5^\circ$ ,  $247.5^\circ$ ,  $292.5^\circ$  and  $337.5^\circ$ .

iv. Points that comply with one of the above conditions are considered as Corner, while the others are discarded.

With these rules, our corner extraction algorithm makes a fast search over the workspace by reducing the number of analyzed pixels, since it only takes into account white pixels. However, discarded pixels or black pixels are only analyzed when they are close to a real corner, *i.e.*, they are considered when they form a feature around an interest point.

Finally, this algorithm can be configured by using all the conditions stated in the step iv. It means, it is possible to enable or to disable the 3 possibilities of  $r$  ( $r = 4, 5, 6$ ), and it is also possible to include another assumption when  $r = 7$ , but it is necessary to be aware of this condition, because the algorithm may detect too many features. Tests proved that using the last condition in high scale gray images performs well, while in SLAM images it does not provide extra detected points, since this kind of condition is not common in OGM.

## 4. Results

We carried out different tests with the aim to evaluate our proposed corner detector. We used 2 different kind of images. The first group is composed of 40 OGMs (or SLAM images) with 3 levels of gray. This kind of images are known to have a large quantity of noisy information, since SLAM builds maps through a probabilistic method applied to sensors, localization methods and map reconstruction algorithms. The second group is composed of 20 standard images taken from different places.

We tested each algorithm 20 times to get the average computation time of each algorithm. In this manner, we gathered information to analyze which algorithm presented the best computation performance, since the principal goal of this work is to select the best feature extractor algorithm applied to OGM in real-time applications. It is also important to mention that the number of extracted features was also gathered to do a qualitative analysis.

### 4.1. Test image bank

The image bank used in this work is divided into 2 sets. The first set consists on 40 maps obtained from SLAM, *i.e.*, occupancy grid maps (or OGM). These images do not require any filtering, since the information represented in the image may correspond to obstacles in the environment. Fig. 6 shows 6 OGM images as examples of the complete dataset used in this work.

The second set is composed of 20 standard images. As we said, our proposed corner detector was originally designed for OGM images; however, it was found out that the corner detector is capable to perform, in a good way, the feature extraction with this kind of images. Fig. 7 shows 5 images used in this group.

The results obtained for each technique are presented below. They take into account the computation time and the number of the extracted features by each technique. No additional post-processes are either applied or taken into

account in the analysis, just the steps and processes applied by each technique.

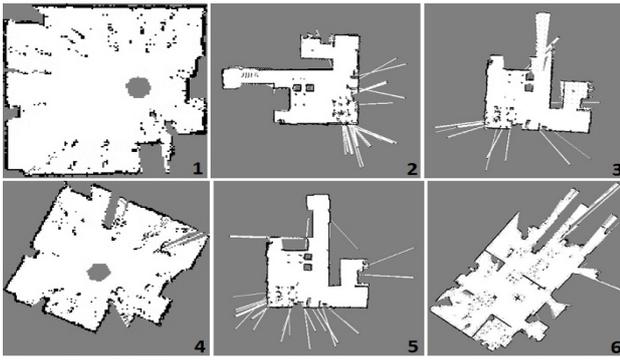


Fig. 6. Set of 6 OGM images used for the tests. These images have 3 levels in the grayscale: 0, 127 and 255.

#### 4.2. Programming and setting considerations



Fig. 7. Second image set composed of images in high-dimensional grayscale, such as those obtained with common RGB cameras.

Finally, for the Trajkovic corner detector, the analysis was performed with its 4 closest neighbors. The  $T_1$  and  $T_2$  thresholds were experimentally set according to the type of the group of images. The SIFT keypoint extractor did not need to be configured, because the OpenCV Contrib package has the best implementation of this technique, so it was not necessary to do any extra modification.

#### 4.3. First group of images - OGM images

OGM images, as it was mentioned above, are generated by SLAM algorithms in 3 levels of the grayscale: black color that represents occupied state (0 in grayscale), gray color that represents unknown state (127 in grayscale) and white color that represents free state (255 in grayscale).

Fig. 8, 9, 10, 11 and 12 show the results obtained with images from Fig. 6. The techniques have an outstanding performance with these images, except for the Trajkovic-Hedley detector. As mentioned in Velásquez and Prieto [12], the topology and the high level of noise contained in the OGM images make the Trajkovic detector extract all the edges as characteristic points. Since OGM images are threshold images, the gradient of the image is very large at the edges due to high changes in image intensity, so it is normal for this technique to make detection in these regions.

Techniques were programmed in C++ or by using their implementation in OpenCV 3.0 and OpenCV Contrib packages. The Integrated Development Environment (IDE) was QT, which uses an interface or wrapper to call native code from OpenCV. The computer used in these tests had a 2.2 GHz processor with turbo boost, 8GB Ram and Ubuntu 14.04 LTS 64 bits.

Since this work aims to make a comparative analysis of different corner detector algorithms, it was performed a parameter configuration, since we wanted to guarantee an optimal performance of each analyzed algorithm. Based on earlier works done by Trajkovic and Hedley [22]; Bargrowski and Luckner [27]; Chen, et al. [28] and Velásquez and Prieto [12], the parameters for Harris and Shi-Tomasi remained the same, since they have similar steps. For these techniques, the Gaussian filter was set to 1 and the kernel size was established in a window of  $3 \times 3$  pixels, since an increase in the kernel size did not improve the performance of the algorithms, but it increased the computation time.

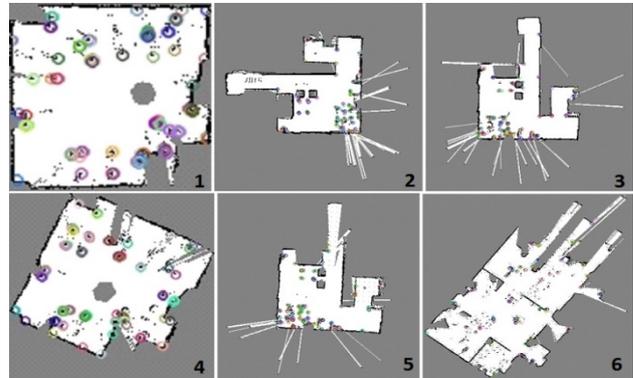


Fig. 8. Results obtained from the Harris corner detector in OGM images.

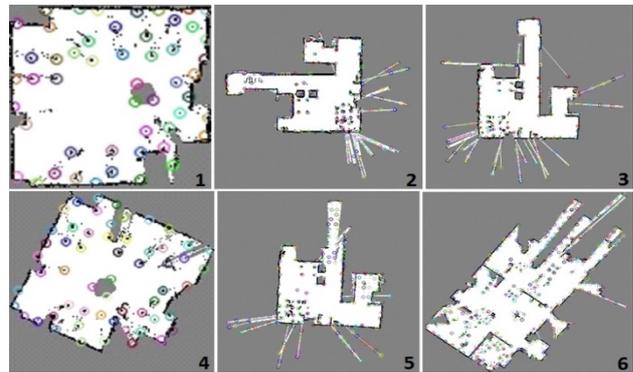


Fig. 9. Results of the feature extraction step made by Shi-Tomasi detector in OGM images.

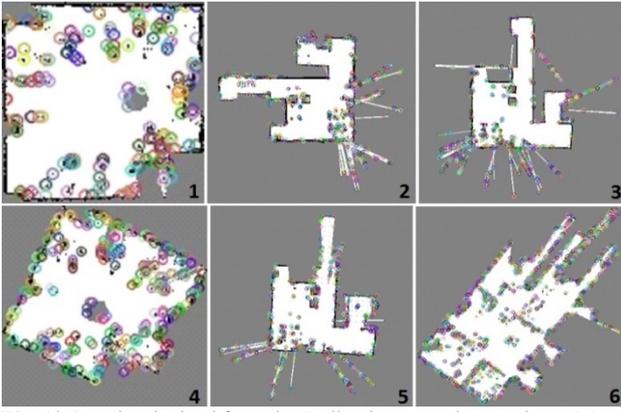


Fig. 10. Results obtained from the Trajkovic corner detector in OGM images.

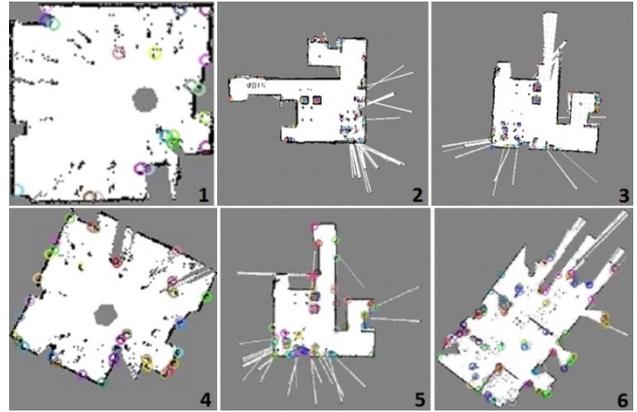


Fig. 12. Feature points obtained with our corner detector method in OGM images.

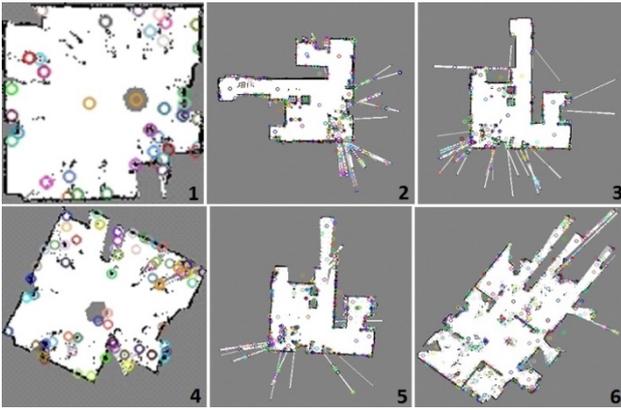


Fig. 11. Feature points obtained with the SIFT keypoint extractor in OGM images.

From Fig. 8, 9, 10, 11, 12. It is also notable the quality of the feature points extracted by our technique, since they were detected correctly in the most representative areas of corners, showing robustness to detect points in OGM images.

Robustness to noisy data is an important property in OGM merging, since noise will always be present in those images due to the stochastic nature of SLAM algorithms, so the map fusion should not depend on feature points extracted from noisy data. For instance, if we analyze the feature points extracted by the Harris and Shi-Tomasi techniques, they tend to detect feature points in areas with poor or noisy data in the map. This problem is also present in the results obtained from the SIFT keypoint extractor. In fact, in all OGM images, the SIFT keypoint extractor detects points in homogenous areas (with no relevant information), so this unsuitable behavior is not good, since it would make lose processing time to the map merging algorithm. Results obtained from Trajkovic show that this technique is also noisy, so its behavior is not suitable to perform in OGM images.

Table 1. Number of corners and average computation time obtained from the first group of images.

Technique	Map 1		Map 2		Map 3		Map 4		Map 5		Map 6	
	Corner #	Time (μs)										
Harris detector*	96	536	123	5990	158	5447	93	960	138	6137	102	8296
Shi-Tomasi detector	52	475	152	5151	160	4292	57	803	166	4605	261	6929
Trajkovic detector**	153	347	493	2519	512	2341	266	643	578	2616	1425	6765
SIFT keypoint	51	6503	390	66667	285	62197	93	10120	288	72341	455	75719
Our corner detector	20	113	59	930	42	767	28	157	55	857	81	964

\*threshold=140

\*\*  $T_1 = 8000$   $T_2 = 4000$

On the other hand, the time consumption spent by each technique was calculated and it is shown in Tab. 1. In this group of images, it is clear that our corner detector had the best (lowest) average time consumption from all images. In this table, we can also note a difference between the Shi-Tomasi detector and the Harris corner detector. The best time consumption is reached by Shi-Tomasi detector, despite the noisy point detection. The number of extracted points is less than the ones extracted by the Harris corner detector. As we stated, the SIFT keypoint extractor does not have a good time consumption, *i.e.*, it is higher than the others algorithms,

because it uses a space-scale calculation to extract key points and that step takes too much time to do a real-time feature detection in applications such as map merging in Multi-SLAM.

From the results, we can also infer that the dispersion step applied by Shi-Tomasi does not help to do a good extraction, because this step increased the false feature points detected and, as we stated above, the map merging algorithm could make a wrong fusion map if the detected features are not reliable or consistent with the environment they represent. This situation is showed in Fig. 13.

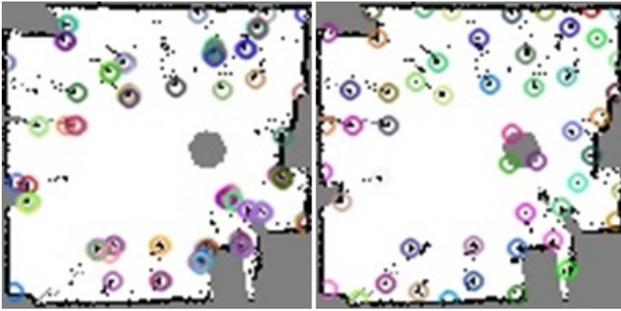


Fig. 13. The left image shows the results obtained by the Harris corner detector, while the right image the results obtained by the Shi-Tomasi detector. It could be noted how the Shi-Tomasi technique makes bad detections in the middle of the map (gray circular region) due to its dispersion step.

#### 4.4. Second group of images

These images are images in high dimension of gray levels. These images present the challenge to obtain the best and

higher number of key points, since they contain a lot of information with noise, blur and other common properties. In these images, our corner detector requires a common edge extractor like the sobel border filter. This extra step was required because the algorithm was developed specially for low grayscale images (such as OGMs). Nevertheless, this extra step does not imply any considerable increment in computation time.

Fig. 14, 15, 16, 17, 18 show the results obtained with images from Fig. 7. From these results, it is possible to deduce that the Shi-Tomasi and our proposed corner detector have the best performance in terms of the scattering of extracted points because they do not concentrate on specific areas. They can explore the entire space and extract the vast majority of features. In this case, the step of dispersion applied by Shi-Tomasi helps it to have a good performance in detection, but as it was stated above, this property is not useful in all images. The user has to know the type of images to be treated in their application.



Fig. 14. Results obtained from the Harris corner detector in standard images.



Fig. 15. Results of the feature extraction step made by Shi-Tomasi detector in RGB images.



Fig. 16. Results obtained from the Trajkovic corner detector in RGB images.



Fig. 17. Feature points obtained with the SIFT keypoint extractor in standard images.



Fig. 18. Feature points obtained with our corner detector method in standard images.

Another important point is the difference between the Shi-Tomasi detector and the Harris corner detector. From Fig. 14 and 15, it is clear that Harris only concentrates its detection in areas with large gradient changes, so it tends to form clusters of points, while Shi-Tomasi disperses the point detection over the whole image. This behavior has a possible explanation: the corner measure applied by Harris Corner and Shi-Tomasi. Meanwhile, the first one uses a measure based on the complete Hessian Matrix (it uses the determinant and trace of

the matrix), the second one only uses the minimum eigenvalue of this matrix.

On the other hand, Trajkovic corner detector has a good performance at detecting corner and edges with a large gradient change, but, as can be noticed, it is not capable of detecting edges from blurred or noisy areas. That is the case of the image of the woman, where it could not detect points in the background of the image.

Table 2. Number of corners and average computation time obtained from the second group of images.

Technique	Img 1		Img 2		Img 3		Img 4		Img 5	
	Corner #	Time (μs)	Corner #	Time (μs)	Corner #	Time (μs)	Corner #	Time (μs)	Corner #	Time (μs)
Harris detector*	1157	21112	1841	21448	1793	19819	395	20004	893	12363
Shi-Tomasi detector	1035	16220	840	16211	551	16848	607	16477	477	11057
Trajkovic detector**	1014	6210	1547	9528	791	5244	1214	8081	937	6711
SIFT keypoint	2000	222554	2000	229727	1266	217639	1294	221016	606	142823
Our corner detector	927	3554	871	4486	398	3415	607	3652	384	2049
	* Threshold: 150 ** $T_1 = 4000$ $T_2 = 2000$		* Threshold: 93 ** $T_1 = 3019$ $T_2 = 1509$		* Threshold: 131 ** $T_1 = 2038$ $T_2 = 1019$		* Threshold: 165 ** $T_1 = 4000$ $T_2 = 2000$		* Threshold: 188 ** $T_1 = 4000$ $T_2 = 2000$	

Tab. 2 presents the number of feature points extracted by each technique and also the average time consumption. Despite the number of points detected, our proposed corner detector preserves the lowest time consumption: twice faster than Trajkovic and, approximately, 4 times faster than Shi-Tomasi. Therefore, our detector seems to be a good feature technique for detecting points in real-time applications. Finally, it can be noticed again that SIFT keypoint had the

highest average time consumption due to the space-scale analysis applied to images.

## 5. Discussion

From the results shown above, it is notable that the proposed technique and the Shi-Tomasi detector are the best techniques

in terms of number/quality of extracted feature points. Both techniques detect features on the same areas, but there is an evident difference when noisy data is on the image: our proposed corner detector performs more robustly than the Shi-Tomasi detector, since it has been designed to deal with this kind of images.

Despite Shi-Tomasi works well in images in low dimension in the grayscale, its dispersion parameter influences the quality of the extracted feature points, since the algorithm is forced to accept points that are located in poor regions. It is also important to mention that the proposed technique has a notable performance with images in high dimension in the grayscale, despite using an edge extractor that can increase its time consumption. In Fig. 13, it is notable how in some gray zones, the Shi-Tomasi detector extracts points, while our proposed corner detector does not, showing robustness to noisy areas.

On the other hand, despite the results obtained with the Trajkovic-Hedley detector and the Harris corner detector, they are not suitable for extracting feature points from images in real-time applications, since they require a parameter tuning which depends on the type of the source image and the noise present in the data. Tab. 1 and 2 show how these parameters change from one group to another. To deal with this, it must be necessary to do an automatic tuning, which means to add an extra process to the feature extraction step. Furthermore, as it was shown in Section 4, the Trajkovic corner detector does not have a suitable performance in OGM images due to its sensibility with large image gradients.

## 6. Conclusions

From the results explained above, two detectors presented a notable performance in feature detection taking into account the quality, number and time consumption: the Shi-Tomasi detector and our proposed corner detector. These detectors showed a good performance in different type of images (low and high dimension in the image grayscale) and both detectors do not require any additional parameter configuration if the image data changes. So, those properties enable them to be used in real-time applications such as object tracking, OGM merging algorithm and SLAM techniques.

Finally, taking into account that our detector was initially designed to deal with OGM images, the results in this paper showed that our detector also has a notable performance in both OGM and normal images, so the initial results presented in Velásquez and Prieto [12] were also checked and ratified in this study. Furthermore, these results extend the use of our corner detector to other applications, since it showed to be more robust and faster than the other analyzed techniques. As a future work, it is necessary to explore other properties related to the field of Corner Detectors such as scale-space invariance and robustness to illumination changes. It is also proposed to make an analysis of border extractors when our corner detector performs with standard images.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License



## References

1. Tungadi, F., Lui, W.L.D., Kleeman, L., Jarvis, R.: *Robust online map merging system using laser scan matching and omnidirectional vision*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 7-14.
2. Lee, Heon-Cheol, Lee, Beom-Hee: *Improved feature map merging using virtual supporting lines for multi-robot systems*, Advanced Robotics, vol. (25), 2011, pp. 1675-1696.
3. Lee, H., Lee, S., Choi, M., Lee, B.: *Probabilistic map merging for multi-robot RBPF-SLAM with unknown initial poses*, Robotica, vol. (30), 2012, pp. 205-220.
4. Wang, K., Jia, S., Li, Y., Li, X., Guo, B.: *Research on map merging for multi-robotic system based on RTM*, IEEE International Conference on Information and Automation, 2012, pp. 156-161.
5. Kojima, T., Okawa, Y., Namerikawa T.: *Multi-robot SLAM for large scale map building using relative information of local maps*, The SICE Annual Conference 2013, 2013, pp. 164-169.
6. Cortés, X., Serratos, F.: *Cooperative pose estimation of a fleet of robots based on interactive points alignment*, Expert Systems With Applications, vol. (45), 2016, pp. 150-160.
7. Lee, H.C., Roh, B.S., Lee, B.H.: *Multi-hypothesis map merging with sinogram-based PSO for multi-robot systems*, Electronics Letters, vol. (52), 2016, pp. 1213-1214.
8. Zhou, X.S., Roumeliotis, S.I.: *Multi-robot SLAM with unknown initial correspondence: the robot rendezvous case*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 1785-1792.
9. Howard, Andrew: *Multi-robot simultaneous localization and mapping using particle filters*, Jet Propulsion Laboratory, NASA, 2006, pp. 1-8.
10. Blanco, J.L., González, J., Fernández, J.A.: *A new method for robust and efficient occupancy grid-map matching*, 3rd. Iberian Conference on Pattern Recognition and Image Analysis (IBPRIA'07), 2007.
11. Blanco, J.L., González, J., Fernández, J.A.: *A robust, multi-hypothesis approach to matching occupancy grid maps*, Robotica, vol. (31), 2013, pp. 687-701.
12. Velásquez, C., Prieto Ortiz, F.A.: *A new corner detector approach for occupancy grid map merging*, Memorias IV Congreso Internacional de Ingeniería Mecatrónica y Automatización - CIIMA 2015, 2015, pp. 60-71.
13. Tuytelaars, Tinne, Mikolajczyk, Krystian: *Local invariant feature detectors: A survey*, Foundations and Trends in Computer Graphics and Vision, 2008.
14. Montero, Andres S., Stojmenovic, M., Nayak, A.: *Robust detection of corners and corner-line links in images*, IEEE 10th International Conference on Computer and Information Technology (CIT), 2010.
15. Li, J. A., Nigel M.: *A comprehensive review of current local features for computer vision*, Neurocomputing, vol. (71), 2008, pp. 1771-1787.
16. Moravec, Hans: *Towards automatic visual obstacle avoidance*, Proceedings of the International Joint Conference on Artificial Intelligence, vol. (2), 1977, pp. 584-584.
17. Harris, Chris, Stephens, Mike: *A combined corner and edge detector*, Alvey Vision Conference, 1988, pp.147-151.
18. Junxiong, Z., Kai, Y.: *Fast Harris corner detection algorithm based on image compression and block*, 10<sup>th</sup> International Conference on Electronic Measurement & Instruments (ICEMI), 2011.
19. Tomasi, C., Kanade, T.: *Detection and tracking of point features*, International Journal of Computer Vision, 1991.
20. Kenney, C.S., Zuliani, M., Manjunath, B.S.: *An axiomatic approach to corner detection*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 2005, pp. 191-197.
21. Das, A., Pukhrambam, M., Saha, A.: *Real-time robust face detection and tracking using extended haar functions and improved boosting algorithm*, International Conference on Green Computing and Internet of Things (ICGIoT), 2015.
22. Trajkovic, M., Hedley, M.: *Fast corner detection*, Image and Vision Computing, vol. (16), 1998, pp. 75-87.

23. Lowe, David G.: *Object recognition from local scale invariant features*, International Conference on Computer Vision, vol. (2), 1999, pp. 1150-1157.
24. Lowe, David G.: *Distinctive image features from scale invariant keypoints*, International Journal of Computer Vision, vol. (60), 2004, pp. 91-110.
25. Daixian, Zhu: *SIFT algorithm analysis and optimization*, Image Analysis and Signal Processing (IASP), 2010.
26. Wang, L., Ma, S., Xue, H., Hou, Z.: *An improved SIFT algorithm for feature points matching of dairy cow images*, World Automation Congress (WAC), 2010.
27. Bagrowski, Grzegorz, Luckner, Marcin: *Comparison of corner detectors for revolving objects matching task*, Artificial Intelligence and Soft Computing, vol. (7267), 2012, pp. 459-467.
28. Chen, Jie, Zou, Li-hui, Zhang, Juan, Dou, Li-hua: *The comparison and application of corner detection algorithms*, Journal of Multimedia, vol. (4), 2009, pp. 435-441.