A New Corner Detector Approach for Occupancy Grid Map Merging

Carlos Alberto Velásquez Hernández [1], Flavio Augusto Prieto Ortiz [2]

(1) cvelasquez@unal.edu.co
Department of Electrical and Electronic Engineering
National University of Colombia
Bogotá, Colombia

(2) faprietoo@unal.edu.co
Department of Mechanical and Mechatronics Engineering
National University of Colombia
Bogotá, Colombia

A New Corner Detector Approach for Occupancy Grid Map Merging

Keywords:
Corner Detectors; Feature Detector; Computer Vision; Occupancy Grid Map

ABSTRACT

The problem of merging maps in SLAM is one of the most studied in this field, because it allows to extend the SLAM algorithms to Multi-SLAM field. This issue is treated as a problem of merging images. In computer vision, merging images issue, nowadays it has many approaches for solving it, using feature extractors and descriptors. In this paper, we propose and show a new corner detector technique that can be used in map images. The results obtained in our tests show that, our corner detector is reliable and efficient to extract features in images generated by SLAM algorithms. Furthermore, we compared our algorithm with others feature detectors like Harris Corner Detector, Shi-Tomasi Detector, among others. We found out our corner detector has a good and reliable performance, doing the extraction from those kinds of map images.

I. INTRODUCTION

In recent years, the problem of merging maps in SLAM and Multi SLAM field, is one of the most studied in the academic research [1-4], because map merging involves the treatment, at the same time, of many occupancy grid maps (OGM) which are built by mobile robots. In fact, the map merging proposed approaches trying to solve this problem from different points such as, programming real encounters of robots in the environment [5], setting-up initials conditions like knowing the initial position of each robot or sharing the same initial position for all robots.

There are authors [6] trying to solve the map merging problem without programming any encounter among robots and without any initial condition. They try to solve the problem just treating it as an image matching issue. From this point of view, we propose a new corner detector approach that is able to extract the most important features from occupancy grid maps built by SLAM algorithms. In fact, our new approach, tries to solve a concurrent problem for the map merging approaches without any initial condition: extract meaningful features as fast as it can, because one the most important drawbacks in these approaches is the deficiency to apply them in real time operations. Extracting features from maps and matching these features in real time takes long time and it makes impossible to do a scalable map merging algorithm.

In this paper, we report a new corner detector technique that is able to extract features from this kind of maps. We focus just in step of feature extraction and we report that our technique has good properties to perform faster than others known corner and features detectors. We compare our technique with Harris Corner Detector, Shi-Tomasi Detector and Trajkovic-Hedley Detector, those are the most used feature detectors.

Our tests have proven that our new corner detector has a suitable anti-noise performance and good speed to perform the task (corner detection). As our approach was developed for OGMs, tests did not consider the time used by the others algorithms to build an image. We only analyzed the stability, noise and time consumption (complexity) of the algorithms, not the time involved to get the data information because it depends on the algorithms programmer.

Furthermore, we extend our approach to the image processing field because we found it can perform well in other kind of images. A good report about this extension is a future work proposed with the aim to prove the perform of our corner detector algorithm.

Finally, this paper is divided in the following sections: In Section 2 we explain the related work of the corner detector used in our tests, describing the main ideas of the feature detectors. In Section 3 we describes our corner detector. Section 4 describes the configuration we have done to ensure the best perfomance for each method, while Section 5 describes the results obtained from the analysis we have done. Section 6 explains final conclusions and future work related to this work.

II. CORNER DETECTORS

These techniques belong to Feature Extractors and their applications make Corners Detectors the best known and used methods in the field of Image Processing. They differ from another group of extractors (Flat Detectors) because their analysis approach to the process images (pixel by pixel). In [7] a full analysis of the most representative techniques of each group is presented.

Corners detectors, as its name says, are responsible for selecting those image points of interest to the researcher. These points may include corners, edges or characteristic points within the image. They mostly serve to detect the center of well-known forms as squares, crosses, lines, triangles, among others, to detect changes in the image gradient on these forms.

A. Moravec Corner Detector

The first Corner Detector approach was proposed by Moravec in 1977 [8], whose main idea was to use a window or kernel on each pixel of the image and detect, through a threshold, the minimum intensity change, using the neighbors of the analyzed pixel. If any change is detected in any direction on the pixel, it is considered as a homogeneous region. However, if the pixel is located on an edge, it exists an intensity change located in one direction while in the perpendicular one, the image gradient does not have some change. Finally, if the pixel is on a corner, intensity changes should be different in all directions. Commonly these movements are made with the four straight neighbors of the analyzed pixel.

However, this technique presents a problem when a corner or edge is located in a different angle as 45°, 30° because it only uses the four straight directions (0°, 90°, 180°, 270°). This technique is not invariant to rotation and depends of the alignment of features with the axis X and Y of the image. Another important drawback with this technique is detection of small changes in intensity according to the threshold, so the response of the technique could give many false features that are not representative to be classified as a real features of the image.

B. Harris Corner Detector

Based on Moravec Corner Detector, in 1988 was proposed another corner detector by Harris and Stephens [9] as a solution to the drawbacks presented in the Moravec technique. This method proposes the use of a matrix tensor (1) built for each image pixel. This tensor included the derivative of the image in X and Y axis. Since the rectangular mask applied by Moravec does not cover the same area when it is rotated, the image is subjected to a convolution with a Gaussian mask.

$$M = \begin{bmatrix} Ix \\ Iy \end{bmatrix} \begin{bmatrix} Ix \\ Iy \end{bmatrix}^T = \begin{bmatrix} I_x^2 & IxIy \\ IyIx & I_y^2 \end{bmatrix} \quad (1)$$

Furthermore, Harris and Stephens proposed a new metric, based on the eigenvalues of the tensor to measure and determine if a pixel is on a characteristics point:

- If $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, the pixel is on a homogenous region.
- If $\lambda_1 \approx 0$ and $\lambda_2 \gg 0$, the pixel is on an edge.
- If λ_1 and λ_2 have a big value, the pixel is on a corner.

However, this analysis has a high computational time, so that is the reason why Harris and Stephens proposed the following measure to optimize the performance of the algorithm [9]:

$$m_h = \det(M) - k * tr^2(M) \quad (2)$$

Where k is a constant proposed by the authors that was tuned between 0,04-0,08. Therefore, the analysis of the image pixels follow the next conditions:

- If $m_h > 0$ and small, the pixel is on a uniform intensity region.
- If $m_h < 0$, the pixel is on an edge.
- If $m_h > 0$ with a high value, the pixel is on a corner.

This technique certainly showed a significant advance in the detection of features within an image; however, it presents a problem of scalability because it could increase the time of image processing and it could lose accuracy in the feature detection because of the data lost from the original image.

C. Shi-Tomasi Detector

This technique is a modification of the Harris Corner Detector. Basically, the authors modify the metric established by Harris to identify whether a point is a corner or not.

While Harris states the following measure [9]:

$$m_h = \det(M) - k * tr^2(M) \quad (3)$$

Shi-Tomasi selects the minimum eigenvalue of the matrix tensor associated to each pixel [10]:

$$m_{shi} = \min(\lambda_1, \lambda_2) \quad (4)$$

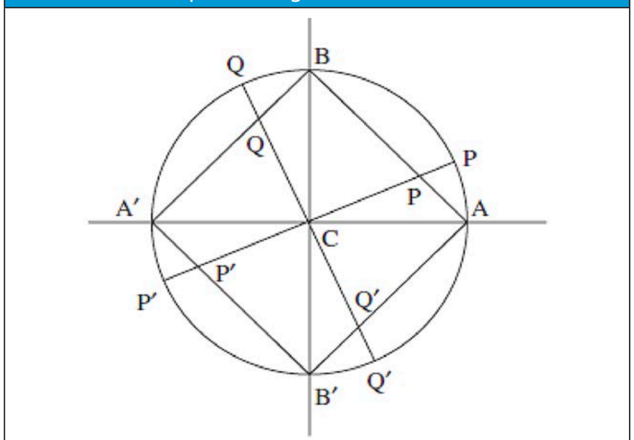
D. Trajkovic-Hedley Detector

This detector was proposed by Miroslav Trajkovic and Mark Hedley in 1998 [11]. Also, it is known as Trajkovic-Hedley corner detector.

This detector is a Feature Extractor technique based on the analysis of the closest neighbors to a point (pixel) by analyzing the grayscale level of each neighbor pixel. Fig 1 shows the analyzed pixel C with its four direct neighbors (A, A', B and B' points) and its four diagonal ones (P, P', Q and Q' points). This technique showed a substantial change in the analysis and feature detection in images, it does not use a window to move through the entire data set of the image.

Additionally, it does not use the concept of image gradient and also, it does not use filters or edge detectors for analysis. The only thing used by this technique is the analysis of the nearest neighbors of an image in low and high resolution. This analysis can include the 4 straight neighbors or the 8 neighbors.

Fig 1. Analysis made by Trajkovic to calculate the corner measure of the C point. Image from [7].



Trajkovic Algorithm has 3 steps. First step is an analysis in low resolution with the aim to extract the most representative points of the image with this condition. In low resolution, it uses the following metric to accept or reject a point as a possible corner:

$$r_A = (f_A - f_C)^2 + (f_{A'} - f_{C'})^2 \tag{5}$$

$$r_B = (f_B - f_C)^2 + (f_{B'} - f_{C'})^2 \tag{6}$$

$$R = \min\,(r_A, r_B) \tag{7}$$

A thresholds *Threshold*1 and *Threshold*2 are defined experimentally based on the knowledge from images to analyze. So, if $R > Threshold1$, the point is accepted as a candidate to be analyzed in the second step. Second step is the same analysis but in high resolution (the original image) and it only analyzes the points extracted in the first step. In this section, the algorithm applies the same metric to accept or reject the analyzed point. However, the threshold used is changed: $R > Threshold2$.

Finally, if a candidate point pass to the last step, the algorithm uses the following equations to calculate the corner measure:

$$C = r_A \tag{8}$$

$$B_1 = (f_B - f_A)(f_A - f_C) + (f_{B'} - f_{A'})\,(f_{A'} - f_C) \tag{9}$$

$$B_2 = (f_B - f_{A'})(f_{A'} - f_C) + (f_{B'} - f_A)\,(f_A - f_C) \tag{10}$$

$$B = \min(B_1, B_2) \tag{11}$$

$$A = r_B - r_A - 2B \tag{12}$$

If $B < 0$ & $A + B > 0$, the analysis continues with:

$$R = C - \frac{B^2}{A} \tag{13}$$

Else,

$$R = \min\,(r_A, r_B) \tag{14}$$

Therefore, a point will be accepted as a corner o feature point if $R > Threshold2$.

In this way, the Trajkovic-Hedley algorithm concludes and, as we can see, it filters the data through the 3 steps explained above until extract the final features points of the image.

III. THE PROPOSED CORNER DETECTOR ALGORITHM

This new feature detector is based on Trajkovic Detector, which analyzes the closest neighbors to the pixel analyzed in order to establish whether or not it belongs to a neighboring corner. This detector works with a basic thresholding on images to facilitate the analysis of potential corners. Since the maps generated by the SLAM algorithm presented three states, they are represented on 3 levels in the grayscale, so use of filters, image gradients and other steps are not necessary. Just a thresholding step is necessary to obtain relevant image information.

Now, through the analysis of the 8 nearest neighbors, a simple filter that speeds up the search on the image (pixels) is set. It is necessary that the central pixel has a level 1 in grayscale (white) because in this way we can infer that, if the pixel is near to a corner, its 8 neighbors must show the required information to extract a real corner from this area of the image. Also, we can ensure that the pixel analyzed is the center of the feature because it is possible to infer that in this point exists the change of the image gradient when we are in presence of a corner.

$$I_{x,y} = 1 \qquad (15)$$

Fig 2. The analyzed pixel is C and the other 8 pixels are the nearest neighbor points of C. A, A', B, B' are the four direct neighbors, while P, P', Q, Q' are the diagonal one. To start the analysis of the point C, this point must have a level 1 in grayscale.

Q	B	P
A'	C	A
P'	B'	Q'

Now, each pixel that presents this first condition (15) is submitted to the analysis of its 8 nearest neighbors through of which it seeks to determine whether or not these neighbors have the characteristic form of a corner. First of all, two measures (r_S, r_D) related to the 4 straight neighbors and the 4 diagonal neighbors must be calculated. Those neighbors are showed in Fig. 2 where it can be noted that this kernel or windows is similar to the Trajkovic one. With the information associated to those pixels, our corner detector works and calculates:

$$r_S = A + A' + B + B' \qquad (16)$$

$$r_2 = P + P' \qquad (17)$$

$$r_4 = Q + Q' \qquad (18)$$

$$r_D = r_2 + r_4 r_D = r_2 + r_4 \qquad (19)$$

As a complement of these measures, the algorithm calculates the following values to increase the accuracy of the method to classify a feature point:

$$r_{Sup} = P + B + Q \qquad (20)$$

$$r_{Der} = P + A + Q' \qquad (21)$$

$$r_{Inf} = P' + B' + Q' \qquad (22)$$

$$r_{esqP} = A + P + Br_{esqP} = A + P + B \qquad (23)$$

Furthermore, it establishes a measure to determine how many pixels have a low value on grayscale or 0 in its information. To do this, it is calculated the following measure:

$$r = r_S + r_D \qquad (19)$$

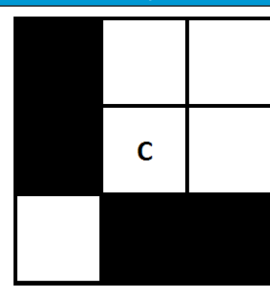
With this information, it starts to perform the analysis of the pixel by applying the selection criteria according to the number of pixels with a low level among its neighbors (measure r):

If r=4, the selection criteria that it is applied is:

° If mod (r_{esqP}) = = 0 & r_S = = 2 & $r_{Sup} + r_{Inf}$ = =3 & r_2 != r_4 , the analyzed pixel is a corner.

This criteria can be analyzed as points with features as shown in Fig 3 where it showed only one case from the 4 possible cases (just turn the feature around the point C to get another feature with the same identity).

Fig 3. Feature extracted with this condition stated above. This is only one case from four possible ones.

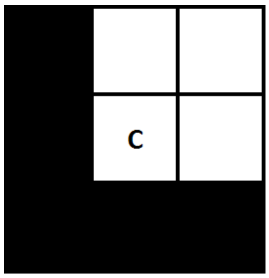


If r=5, the selection criteria that it is applied is:

° If r_S = = 2 & mod(r_{Sup} + r_{Der}) = = 0, the analyzed pixel is a corner.

This criteria can be analyzed as points with features as shown in Fig 4 where it showed only one case from the 4 possible cases. This feature is the best one that our corner detector tries to find and determine, because it shows a complete corner around the point C analyzed.

Fig 4. Feature extracted with the condition stated above. It shows the best corner formed around point C analyzed. With only 5 point disposed as in the image, the algorithm can detect this kind of feature

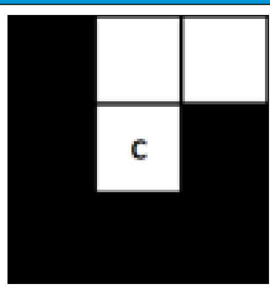


If r=6, the selection criteria that it is applied is:

° If r_S = =3 & r_D = =3, the analyzed pixel is a corner.

In Fig 5, it is showed the extracted feature from this selection criteria. Features with another kind of neighbors configuration cannot be considered because it is considered by the algorithm as a noise.

Fig 5. Feature extracted with this condition. This kind of feature it is a strange one but it appears in irregular map images, so it has to be considered because it show a good point to find out a reliable feature.



With these simple rules, the algorithm designed for feature extraction by detecting corners, makes a fast search by reducing the workspace since it discards pixels with irrelevant information or pixels with a low value on grayscale. However, discarded pixels are analyzed in other iterations if they are near to a real corner, so it is important because discarded pixel can provide us information that helps to define the appearance of a corner in the image.

IV. CONFIGURATION OF CORNER DETECTOR ALGORITM

Since this work presented a comparative analysis of the corner detector techniques, it was necessary to perform a parameter configuration of each corner detector in order to ensure optimal performance of the methods.

Therefore, it was necessary to carry out programming an ideal configuration of the algorithms studied in order to take full advantage of each technique and make a proper comparison among them. Additionally, it noted that the vast majority of techniques were programed using the OpenCV 2.4 package where it was possible to find the official and optimized algorithms for these techniques. However, not all techniques were supported with this package (Trajkovic detector), so it had to be programmed in order to measure their performance. The programming language used was JAVA, which uses a wrapper or interface that allows to call native code from OpenCV; however, in this paper it was not possible to make an analysis of computation time because the techniques did not have an unified programming language and this metric will depend largely on the expertise of the programmer to improve the performance of each one, so this comparison will carry out in future related work with a correct computation time metric.

Once we exposed the framework in which these techniques were programmed, it was necessary to choose the best parameters displayed on the different techniques to improve the performance of each method. Based on earlier works [11-13], the parameters for Harris and Shi Tomasi detectors were set equally because of their similarity. For these two techniques, it was selected a kernel or mask with a size of 3x3 pixels for the Gaussian filter and for the kernel applied to calculate the corner measure. An increase of this window did not show a significant improvement, but it increased the computation time of the technique, as it is also exposed in [12]. Additionally, for Harris Corner Detector the k parameter was set to 0.04 based on the experience and recommendation of the author. The standard deviation for the Gaussian filter applied was set to 1.

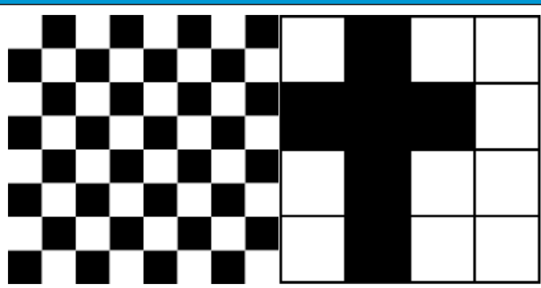
Finally, for the Trajkovic-Hedley algorithm, it was selected the analysis of its four direct neighbors. T1 and T2 thresholds used by the algorithm were experimentally set according to the type of image analyzed in order to optimize the results obtained. In those tests, the results showed that an analysis in low resolution makes the algorithm miss important information because the points in low resolution may not correspond exactly to the same point in high resolution. Therefore, there was a high loss rate of points detected because the algorithm in this step loses accuracy.

V. RESULTS

A. Test Images For Corner Detector Analysis

The test images used for this work are divided in two sets. First group images are testing images to analyze the performance of the corner detectors with images with a marked presence of characteristic points. These images are shown in Fig 6.

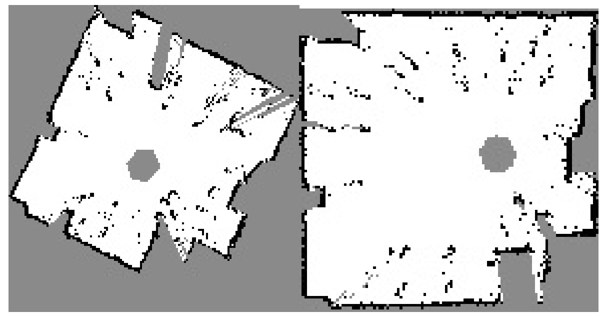
Fig. 6. Test images used for the first group. They correspond to images with 2 values in the grayscale: Black color and White color.



The second set consists on maps obtained from the Simultaneous Localization and Mapping Algorithm. They are low-dimensional images in grayscale. These images for the algorithm proposed here, does not require some pretreatment or filtering algorithm to use. That is, do not require the use of Gaussian filter, edge extractors or thresholding as if it required by other techniques. These images are shown in Fig 7.

Results obtained for each technique are presented. This work takes into account the number of points extracted by each technique, that is, no additional steps are applied. Such as the case of the Shi-Tomasi technique that applies a dispersion of the points in order to avoid concentrating the extracted points on specific areas or zones of the processed image.

Fig 7. Test images used for the second group. They correspond to images with 3 values in the grayscale: Black, Gray 127 and White.



B. First Group of Test Images

As we mentioned above, techniques are analyzed in the images of group 1 as an initial step. This step is ought under this analysis to determine algorithms performance against images with excellent characteristics.

Figs 8, 9, 10 and 11 show the results from these techniques. From these images, all algorithms have an outstanding performance. All were able to detect 100% corners present in the images. Thus, it can be noted that the algorithms discussed here, show a good performance in low-dimensional images in grayscale. And as the only point to highlight, the detection algorithm proposed in this paper, selects the characteristic point not on the edge of the feature but on a side of it. Thus, the algorithm focuses on the center of the extracted feature.

Fig 8. Feature points extracted by Harris Corner Detector. It can be denoted this technique has a good and reliable performance with this kind of images.

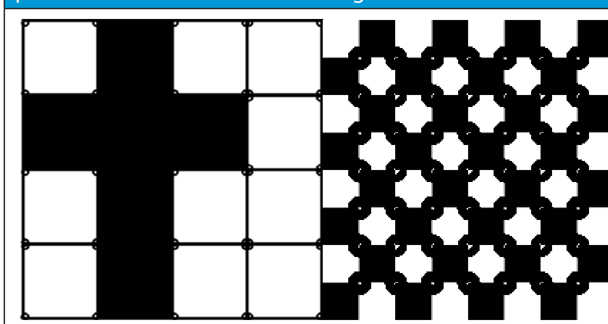


Fig 9. Feature points extracted by Shi-Tomasi Detector. As it was expected, this technique achieved same results of the Harris Corner detector, because Shi-Tomasi is based on Harris.

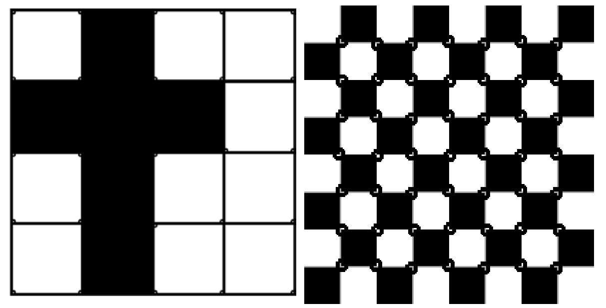


Fig 10. Feature points extracted by Trajkovic-Hedley Detector. Also, Trajkovic-Hedley showed a good and reliable behavior with this kind of images.

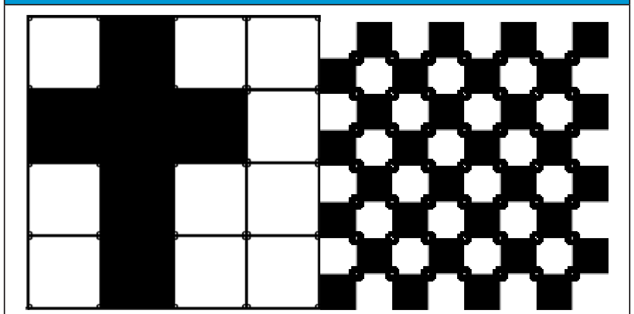
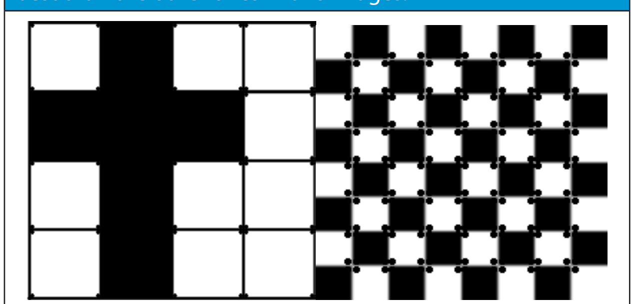


Fig 11. Feature points extracted by our corner detector technique. It can be denoted how our corner detector approach extracts more features from the image (two features per corner). It shows that our algorithm is certainly best than the other ones in this images.



C. Second Group of Test Images

These images, it was explained in previous section, are generated by SLAM algorithm in low dimension in grayscale. These images have this feature since SLAM algorithm works with three states: Gray color corresponds to unknown state, Black color to occupied state and White color to free state.

Figs 12, 13, 14 and 15 show results obtained with the image from Fig 7. Techniques have a remarkable behavior except Trajkovic-Hedley detector with these images, because of its high noise content, detected all edges of the image as feature points. This shows that in low-dimensional images in grayscale, this technique does not have an adequate performance. The main reason is Trajkovic detects around corner if their neighbors have an opposite level of gray color to the level of gray color from analyzed point or pixel.

Fig 12. Feature points extracted by Harris Corner Detector in images generated by SLAM algorithm.

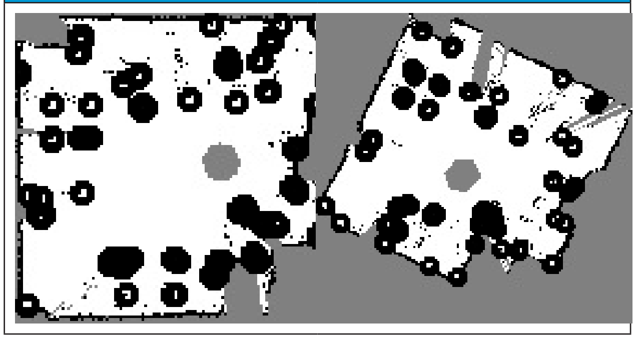


Fig 13. Feature points extracted by Shi-Tomasi Detector in images generated by SLAM algorithm.

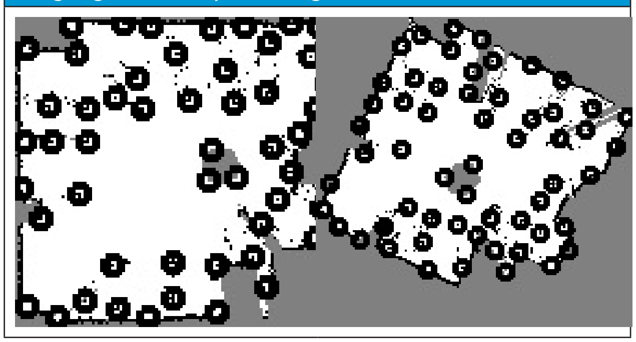
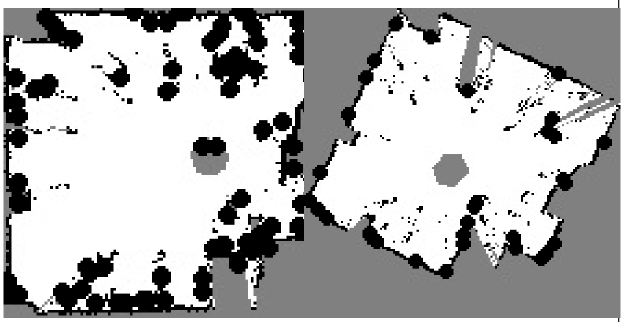


Fig 14. Feature points extracted by Trajkovic-Hedley in images generated by SLAM algorithm.

Fig 15. Feature points extracted by our corner detector technique in images generated by SLAM algorithm.

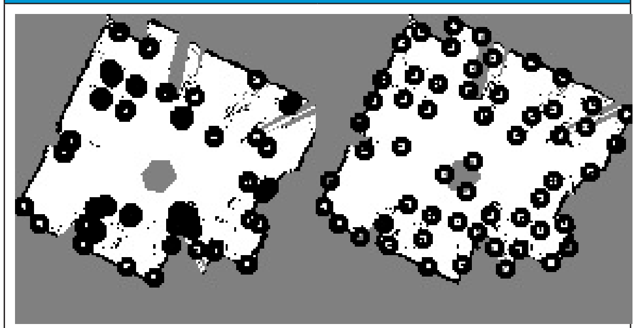


Another analysis that can be deduced from results shown in Figs 12, 13, 14, and 15, is the detection of corners or feature points extracted by the proposed technique. Notice that the extracted points are properly on the areas of greatest presence of edges, showing robustness to noise in the images. This noise is very common in this type of pictures, due to the SLAM algorithm in itself estimates the probabilistic map information from the sensor system of the robot, as it is known, this information carries a high level of noise. So, if the results obtained by the Harris and Shi-Tomasi techniques are observed, these detected in areas whose characteristic information for the map corresponds to poor or noise information in the image. This idea is important because with the aim of developing a map merging algorithm, it will not improve the matching results between characteristic points of different maps, because this information, to be product of the sensor noise, cannot be present in other maps analyzed by the algorithm. Thus, a match on these points would not be expected and therefore, time consumption of the algorithm will increase finding matches on points with noisy information.

Finally, results presented in this work, you may notice that, although the Shi-Tomasi technique is based entirely on the Harris detector, it has different results in terms of the dispersion of the results obtained. It may be noted that, while Harris results concentrated on certain regions, Shi-Tomasi, from the official OpenCV implementation, scattered over the results without creating concentrations of the detected points. This can be beneficial for images with little information on their pixels, because it can be spread effectively, but dangerous in images with high level of noise, because the detector tends to extract points with no representative image information.

From Fig 16, it can notice how the Shi-Tomasi algorithm tends to detect points in the center of it, where there is not characteristic information. Also, it can be seen how Shi-Tomasi detected several unimportant points. This technique detects the point as a feature point, but as stated above, this point can really noisy correspond to information that was leaked on the SLAM algorithm.

Fig 16. a) Results obtained by Harris Corner Detector. b) Results obtained by Shi-Tomasi Detector. In the middle of the images, our corner detector does not detect noisy features. Situation that it is opposite from Shi-Tomasi algorithm.



D. Discussion

With the results shown in the last section, it can be denoted that the proposed technique is equivalent to the Shi-Tomasi technique in terms of the performance for both groups of images. Both techniques detect features in the same areas, but it exists a short different with noisy areas as it is showed in Fig 16, our corner detector is more robustness than Shi-Tomasi method. Although Shi-Tomasi working properly for pictures with a low change in the image gradient, it does not work so well well with images with some degree of noise. Its dispersion parameter can be a point of analysis to make it more robust against this type of drawbacks. However, this is the best performing technique of those algorithms analyzed here. But, our corner detector has an outstanding performance in images with some degree of noise (figures from images group 2). That was the best conclusion of our research.

Furthermore, the Trajkovic-Hedley, although this technique presents some important results, always requires a tuning of thresholding parameters T1 and T2. Therefore, for a real-time application, requires an optimal strategy that automatically tune them once processed images change in the information contained in the pixels. Furthermore, as shown in the previous section, the Trajkovic-Hedley technique has unsuitable performance in low dimensional images in grayscale (figures from images group 2). Therefore, for a merging map algorithm, this technique is discarded by the results obtained in this work.

TABLE I. COMPARISON TABLE OF CORNER DETECTORS

Comparison and Analysis of Corner Detectors

	Images Group 1		*Images Group 2*	
Technique	*# Extracted Points*	*# Extracted Points*	*# Extracted Points*	*# Extracted Points*
Harris Corner Detector	122	673	106	110
Shi-Tomasi Detector	25	49	57	50
Trajkovic-Hedley Detector	41	196	987	859
Our Corner Detector	41	196	66	84

An additional point to note is the homogeneity of the Trajkovic-Hedley detector and our corner detector, results shown in the Table I, with the number of feature points extracted in the image group 1: both detectors extracted the same number from extracted points. This can be explained, because both techniques perform an analysis of the four/eight nearest neighbors around the analyzed pixel. So it is possible to state, those pixels have the same information for both methods and their corner measure is similar. That is the reason which causes both algorithms detect the same points in the image.

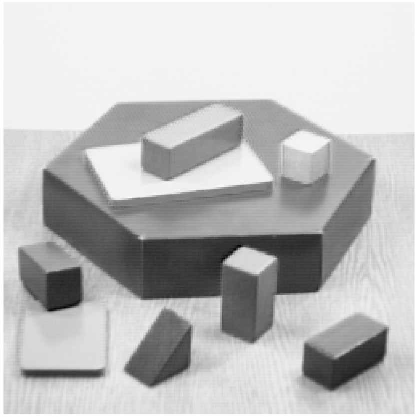
Also, from Table I, it is possible to conclude that our corner detector and Shi-Tomasi algorithm are more efficient in feature detection, because they are more accuracy than the other techniques.

Both methods detected a lower quantity of corners (66-64 and 57-50 points, respectively), but, from Fig 13 and 15, those points are richer in reliable information than the other ones from the others methods. Both algorithm are quite accuracies and fast.

VI. CONCLUSION

From the different techniques results presented, two detectors have better performance: Shi-Tomasi detector and the corner detector proposed in this paper. These two techniques did not require additional configuration or changes in parameters for extracting points. This demonstrates robustness of the two techniques analyzed here, when they extracts features in real time images. So, they can be used in real-time applications like video processing because they do not need any configuration if image conditions change. Also, they can be used in applications related to map merging algorithms or SLAM techniques.

Fig 17. High dimension image in grayscale. This image uses the entire grayscale, so it is a good candidate to be analyzed in future works. Image taken from [14].



Finally, taking into account the focus of this analysis, the proposed technique is the most prominent feature detector to be used as the initial step for a merging map algorithm to develop, since it presents the best performance in the techniques analyzed here. However, in initial tests with images in high dimension in grayscale (images not considered in this paper), the programmer must choose a suitable threshold and an edge extractor (like Sobel Filter) to prepare the image before using the technique proposed in this paper. Fig 17 shows an example of this kind of images used in these initial tests. We think our algorithm would improve its performance if we tune and chose the correct parameters for the edge extractor and filters, so results obtained with this type of images and filters will be present as a future work.

VII. FUTURE WORK

As a future work, we have to carry out other tests to state, with more reliability, the performance of our proposed corner detector. In fact, we want to extend this technique to normal images because we found out that our corner detector could be applied to low and high resolution images which use the entire gray scale. Furthermore, we think our algorithm would improve its performance if we tune and chose the correct parameters for the edge extractor and filters. Those works will be present in future related works.

REFERENCES

[1] Xun S. Zhou, Stergios I. Roumeliotis. "Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case", International Conference on Intelligent Robots and Systems. Beijing, China. 2006.

[2] Sebahattin Topal, Ismet Erkmen, Aydan M. Erkmen. "A Novel Map Merging Methodology for Multi-Robot Systems", Proceedings of the World Congress on Engineering and Computer Science, Vol. 1. San Francisco, USA. Octuber 2010.

[3] Xun Zhou, Stergios I. Roumeliotis. "Multi Robot SLAM Map Alignment with Rendezvous", Department of Computer Science & Engineering, University of Minnesota. Minnesota, USA. June 2005.

[4] Sebastian Thrun, Yufeng Liu. "Multi-Robot SLAM with Sparse Extended Information Filers", Department of Computer Science, Standford University and Deparment of Physics, Carnegie Mellon University. USA.

[5] A. León, R. Barea, L.M. Bergasa, E. Lopez, M. Ocaña, D. Schleicher. "SLAM and Map Merging", Journal of Physical Agents, Vol 3. January 2009.

[6] Jose L. Blanco, Javier Gonzalez, Juan A. Fernandez M. "A New Method for Robust and Efficient Occupancy Grid-Map Matching", IbPRIA '07 - Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis, part II, pp. 194-201. 2007.

[7] Jing Li , Nigel M. Allinson. "A comprehensive review of current local features for computer vision", Neurocomputing 71, pp. 1771– 1787. 2008.

[8] H. Moravec. "Towards automatic visual obstacle avoidance", Proceedings of the International Joint Conference on Artificial Intelligence. 1977.

[9] C. Harris, M. Stephens, A combined corner and edge detector. Alvey Vision Conference, pp. 147–151. 1988.

[10] Tomasi, C., Kanade, T. "Detection and tracking of point features", International Journal of Computer Vision. 1991.

[11] Trajkovic, M., Hedley, M. "Fast corner detection", Image and Vision Computing, vol. 16, n° 2, pp. 75 – 87.1998.

[12] Grzegorz Bagrowski, Marcin Luckner. "Comparison of Corner Detectors for Revolving Objects Matching Task". Artificial Intelligence and Soft Computing, ICAISC 2012 - Proceedings of the 11th International Conference, pp 459-467. Zakopane, Poland. April 29-May 3, 2012.

[13] Jie Chen, Li-hui Zou, Juan Zhang, Li-hua Dou. "The Comparison and Application of Corner Detection Algorithms", Journal of Multimedia, vol. 4, n° 6. 2009.

[14] S.M. Smith, J.M. Brady. "SUSAN—a new approach to low level image processing", International Journal Computer Visison, vol. 23, n° 1, pp. 45-78. 1997.