Revisiting Harris Corner Detector Algorithm: A Gradual Thresholding Approach

Gioacchino Vino and Angel D. Sappa

Computer Vision Center
Edifici O, Campus UAB
08193 Bellaterra, Spain
gioacchinovino@gmail.com, asappa@cvc.uab.es

Abstract. This paper presents an adaptive thresholding approach intended to increase the number of detected corners, while reducing the amount of those ones corresponding to noisy data. The proposed approach works by using the classical Harris corner detector algorithm and overcome the difficulty in finding a general threshold that work well for all the images in a given data set by proposing a novel adaptive thresholding scheme. Initially, two thresholds are used to discern between strong corners and flat regions. Then, a region based criteria is used to discriminate between weak corners and noisy points in the midway interval. Experimental results show that the proposed approach has a better capability to reject false corners and, at the same time, to detect weak ones. Comparisons with the state of the art are provided showing the validity of the proposed approach.

1 Introduction

Low level feature detection, such as key points and edges, represents the first step of many different vision task such as: tracking, localization, SLAM (simultaneous localization and mapping), image matching and recognition and camera calibration (e.g., [1], [2], [3], [4], [5]). Corner detection, in particular, has been largely studied in the literature during last two decades. During all these years different methods have been proposed. They can be roughly classified into three groups: *i*) edge-based; *ii*) direct graylevel-based; and *iii*) graylevel-derivative-based.

The former group uses the changes of the edge direction between two regions to detect a corner and usually requires a great amount of calculation, so this group of methods is seldom used. The second group acts directly on the grayscale image and does not need to do any calculations. Some examples of algorithms belonging to this group are the FAST [3] and the AGAST algorithms [6]. Finally, algorithms in the last group find the corners using their low self-similarity in all directions. Harris algorithm [7] belongs to this group and it identifies a corner by calculating the gradient at each pixel. A function, called response function, is used to compute a value per each point of the image according to the gradients in the two directions: if this value is bigger than a threshold then the point is considered as a corner. In [8] the Harris algorithm is compared with other

M. Kamel and A. Campilho (Eds.): ICIAR 2013, LNCS 7950, pp. 354–363, 2013.

methods of the same category in aspects such as repetition, accuracy and the amount of information contained in detected corners. The authors concluded that Harris algorithm has the best performance. However, the big drawback of Harris corner detector is the need to set a different threshold for each image in order to detect the most important interesting points. If a low threshold value is used, a large amount of points are detected together with noisy data from the image. On the contrary, if a high threshold value is considered, only those strongest interest points will be detected.

Besides, the Harris detector does not work so well with sharp contrast in colour and brightness, since the seminal work [7] several improvements have been proposed in the literature (e.g., [9], [10], [11]). Most of these work are focussed on finding the best threshold for each image that allows detecting the most stable corners. To do this, the image and the response function matrix (R_F) are generally used to extract useful information while estimating the right threshold. An interesting approach has been presented in [9], where the threshold is calculated using the maximum value of R_F, without assuming an empirical value. They noted that using a single threshold for the whole image is not useful because different regions of the image have different brightness and contrast. Hence, in order to improve the quality of the final results, they propose to divide the given image into a set of sub-images that are independently evaluated with different thresholds. In this way, the approach proposed in [9] was able to get a better performance extracting corners when there is a sharp contrast in color and brightness in different parts of the image.

In spite of the good performance in the experimental results presented in [9], the main disadvantage of this approach is the dependence of the threshold from a single value, in this case, from the maximum value of response function in a given sub-image. If this value is too low, due to the fact that in the given sub-image there are not strong corners, the corresponding threshold will be low. Hence, it will result in the detection of many false corners. While, if in the given sub-image there are strong corners, the corresponding threshold will be high and won't be able to discern weak corners. Both, the original Harris and [9] use a single threshold for the given image (or sub-image). In this way, there will be always weak but important corners missed and noisy points detected as corners. Therefore, in the current paper we propose a method that add a new criteria, which gradually finds the best threshold for the given image. Initially two coarsely defined thresholds are used to discern strong corners and noisy points. Then, the region contained in between the two threshold is analyzed with a novel approach that allows to discriminate between strong noisy points and weak corners. This approach is based on the R_F values in the neighborhood of the analyzed points. In this way the method is able to gradually adapt the threshold going from the condition of strong corners to that of noisy points. The manuscript is organized as follow. In Section 2, the original Harris corner detector algorithm is summarized to introduce notation and main concepts. Then, the proposed approach is presented in Section 3. Experimental results and comparisons are provided in Section 4. Finally, conclusions are given in Section 5.

2 Original Harris Corner Detector Algorithm

Corners can be defined as points with low self-similarity in all directions. The self-similarity of an image patch can be measured by taking the Sum-of-Squared-Differences (SSD) between an image patch and a shifted version of itself. The Harris algorithm works by computing a response function (R_F) across all the image pixels. Then, those exceeding a threshold, which are also locally maximal, are retained as corners. Lets I be a 2D gray-scale image; for a given shift (x, y) the sum of square differences S_W between values of the image I and its corresponding shifted one (auto-correlation function) is obtained as [12]:

$$S_W(x, y) = \sum_{x_i, y_i} w(x_i, y_i)[I(x_i + x, y_i + y) - I(x_i, y_i)]^2 \tag{1}$$

where w is a windowing function (e.g., a Gaussian). S_W for nearly constant patches will be near to zero, meanwhile for very distinctive patches reaches large values. For small shifts (x, y), we can approximate the shifted image by the first-order Taylor expansion and rewrite it as a matrix equation:

$$S_W(x, y) = \sum_{x_i, y_i} w(x_i, y_i)[I_x(x_i, y_i)x + I_y(x_i, y_i)y)]^2 \tag{2}$$

$$= [xy]H \begin{bmatrix} x \\ y \end{bmatrix} \tag{3}$$

where $\mathbf{H}$ is a 2×2 matrix computed from image derivatives, which will be referred to as the Harris matrix:

$$\mathbf{H}(x, y) = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \tag{4}$$

$$A = \sum_{x_i, y_i} w(x_i, y_i) I_x^2(x_i, y_i) \tag{5}$$

$$B = \sum_{x_i, y_i} w(x_i, y_i) I_{xy}(x_i, y_i) \tag{6}$$

$$C = \sum_{x_i, y_i} w(x_i, y_i) I_y^2(x_i, y_i) \tag{7}$$

The eigenvalues of H are an approximate measure of the image curvature; if both are large, it indicates that there is a peak in the local auto-correlation function and this pixel is a corner. Harris [7] suggested that exact eigenvalues computation can be avoided by computing the response function:

$$\begin{aligned} \mathbf{R_F}(x, y) &= \mathbf{R_F}(\mathbf{H}(x, y)) = \\ &= det(\mathbf{H}(x, y)) - k\ trace^2(\mathbf{H}(x, y)) \end{aligned} \tag{8}$$

where $det(\mathbf{H})$ is the determinant of the local structure matrix $\mathbf{H}$, $trace(\mathbf{H})$ is the trace of matrix $\mathbf{H}$ (sum of elements on the main diagonal), and k is a tunable parameter where values from 0.04 to 0.15 were reported in literature as appropriate [12]. $\mathbf{R_F}(\mathbf{H})$ is large if both eigenvalues of $\mathbf{H}$ are large [3]. Hence, a point is considered as a corner if both: i) its $\mathbf{R_F}(\mathbf{H})$ value is bigger than the given threshold; and ii) there are not greater values in a 3x3 neighborhood window in $\mathbf{R_F}$. The quality of the detected corners depend on the threshold used to discern them. A quite high threshold will detect only very strong corners, while a too low threshold will detect many false corners, which are originated by noisy points. Finding the ideal threshold is a trade-off between the number of non-detected true corners and detected false corners. This threshold depends on the image and on its characteristics, like the brightness for instance.

3 Proposed Method

As mentioned above, one of the main drawbacks of Harris corner detector is to find the ideal threshold for a given image. In the current work we propose a method that does not depend on the fine tuning of a single threshold but uses a criteria that gradually goes from the condition of strong corners to that of noisy points.

The proposed approach is based on the information extracted from the $\mathbf{R_F}$ matrix. Hence, since the $\mathbf{R_F}$ matrix is obtained from the given image, somehow the proposed approach is able to adapt to different images. Note that $\mathbf{R_F}$ is a matrix where each element contains a value that gives us information on the type of examined region. If this value is negative, it is very probably that the element is close to an edge; if it is close to zero, the point is within a flat region; otherwise, if the value is big and positive that point is close to a corner. Unfortunately, the use of a single threshold does not allow to discern weak corner from strong noisy points since their $\mathbf{R_F}$ value is similar. Hence, it is evident that some other information is needed to differentiate between these two situations.

In the current work we propose a two step process, which initially select a set of points that can be considered as candidate corners. Then, in the second step, those candidate corners are analyzed and finally labelled as corners, or rejected, using local information. The first step works by using two threshold values; note that these values do not need a fine tuning like in Harris algorithm, just two coarsely tuned values are enough. This first step is intended to speed up the whole process. If the $\mathbf{R_F}$ value is smaller than the lower threshold, the point is not considered, while, if the value is bigger than the higher threshold the point is directly considered as a corner. Actually, these points are strong corners. The points with a $\mathbf{R_F}$ value in between the two thresholds could be noisy data or weak corners. In the current work we empirically define the threshold values as 3×10^6 and 12×10^6, lower and higher respectively, expressing the image in a $[0, 255]$ brightness range.

Once the strong corners and candidate corners have been detected, the second step proceed by analyzing every candidate corner using the corresponding value from the response function as well as the response function values of its surrounding. The proposed approach is based on the use of the weighted neighbourhood sum (WNS) in a 3×3 window centered on the candidate corner. The weight depends on the Euclidean distance: the four closest pixels (indicated with the set Q_1) are weighted by 1, while the other four pixels (indicated with the set Q_2) are weighted by $1/\sqrt{2}$. Note that only positive $\mathbf{R_F}$ values are considered; negative values are not useful to understand the corner strength since they indicate how much the respective region contains an edge.

$$
\begin{aligned}
WNS = & \sum_{(\Delta x, \Delta y) \in Q_1} R_F(x + \Delta x, y + \Delta y) + \\
& \frac{1}{\sqrt{2}} \sum_{(\Delta x, \Delta y) \in Q_2} R_F(x + \Delta x, y + \Delta y)
\end{aligned} \tag{9}
$$

where

$$
\begin{aligned}
Q_1 &= \{(0,1), (0,-1), (1,0), (-1,0)\} \\
Q_2 &= \{(1,1), (1,-1), (-1,1), (-1,-1)\}
\end{aligned}
$$

Now, in order to discern whether the candidate point is a corner or not, a threshold adapted to the local information of the candidate point is computed. It works as follow. Firstly, the candidate point is rejected if its $\mathbf{R_F}$ value is smaller than its corresponding WNS value. Then, the normalized weighted neighbourhood sum ($NWNS$) is computed as:

$$
NWNS(x, y) = \frac{WNS(x, y)}{\mathbf{R_F}(x, y)} \tag{10}
$$

this $NWNS$ value is used to discern whether the candidate point is a corner or not. This final decision is made according to a threshold function, which assumes that the stronger is a corner, the bigger is its response function value and less important is the values of its neighbourhood. This threshold function has been found empirically, and returns the threshold value (Nb_{th}) from the $\mathbf{R_F}$ value of the element $\mathbf{I}(x, y)$.

$$
Nb_{th} = \begin{cases} 2.99 & \text{if } \mathbf{R_F}(x, y) < 4 \times 10^6 \\ k_1 + k_2 \left[\mathbf{R_F}(x, y) - k_3\right]^2 & \text{otherwise} \end{cases} \tag{11}
$$

where the parameters k_1, k_2 and k_3 are set to: 1.99, $1/(49 \times 10^{12})$ and 11×10^6, respectively. These values are fixed and were obtained by fitting the function above to the values of a $\mathbf{R_F}$ discretized into 10 bins. Hence, if $NWNS(x, y) > Nb_{th}$ the candidate point is selected as a corner, otherwise it is rejected. To improve the speed of the algorithm, the sum of the neighbourhood is computed just after have checked that the point is a local maximum and the $\mathbf{R_F}(x, y)$ is between the two thresholds.

4 Results

The proposed approach has been evaluated and compared with the original Harris algorithm, using both low and high threshold values respectively. Additionally, the proposed approach has been compared with [9] showing a better performance when different images are presented. In this section experimental results using the image showed in Fig.1 are presented.

The outcomes of Harris with low and high threshold values and from the proposed approach are shown in Fig.2 and Fig.3. The amount of detected corners

Fig. 1. Original image used as a test bed by the different algorithms

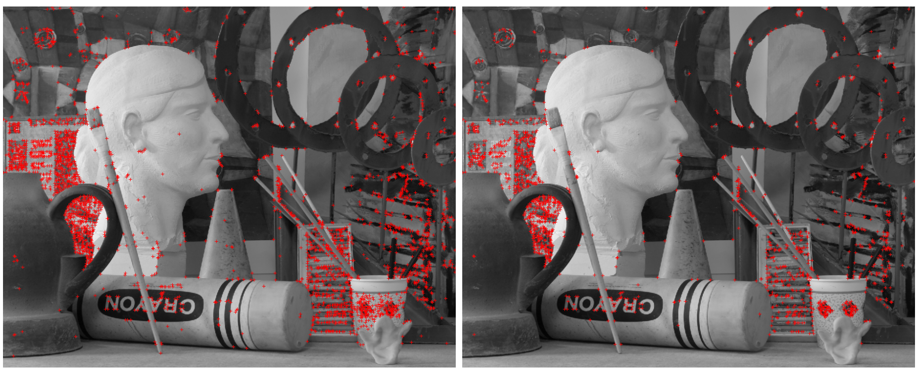

Fig. 2. Results from Harris algorithm with: (*left*) **low threshold value** and (*right*) **high threshold value**

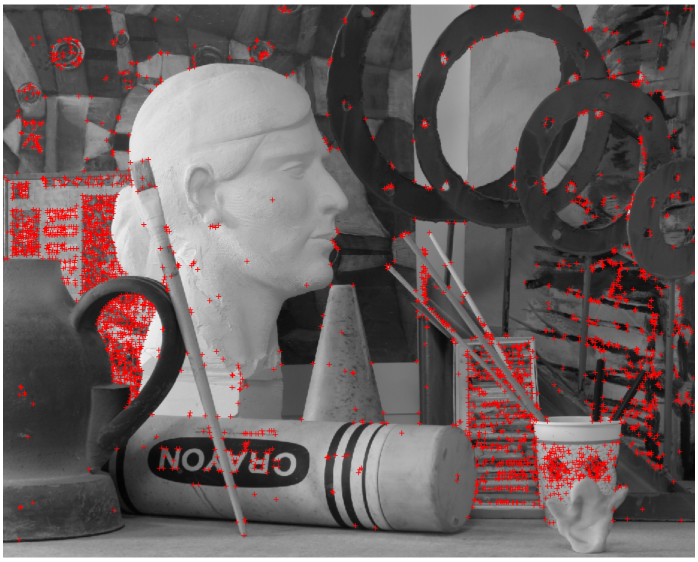

Fig. 3. Result from the proposed gradual thresholding approach

are 3007, 1587 and 2834, respectively. It can be observed that in Fig.2($left$), original Harris algorithm with a low threshold, a larger amount of corners were detected some of them corresponding to noisy data. On the contrary, when the original Harris algorithm with a high threshold value is used, only stronger corners are detected (see Fig.2($right$)). Finally, Fig.3 shows the result from our gradual threshold Harris algorithm; it can be appreciated how it succeeds to reject false corners while detecting weak but important corners, for instance see at the corners on the traffic cone.

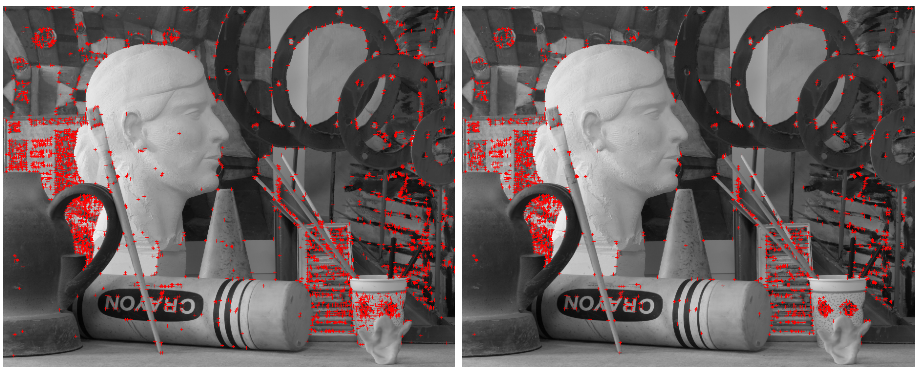

Fig. 4. Results by tuning Harris algorithm according to [9], considering 9 sub-images and setting: ($left$) $p = 0.005$ and ($right$) $p = 0.015$ respectively

Finally, we show the results obtained using the Autothreshold algorithm proposed in [9]. In [9], the authors propose to evaluate the local threshold multiplying the maximum value of the response function in a sub-image by a parameter (p), which have a value in between 0.005 and 0.015. In Fig.4 the results obtained using these limits are depicted. In these cases, the amount of detected corners are 5231 and 3330, respectively. Note that in this case, even if the algorithm works fine, it is necessary to select the right value of the p parameter, although it gets lower effects than the selection of the threshold in the original Harris algorithm. Using the maximum value for the p parameter, some corners are not detected. Indeed, it can be appreciated in the image in Fig.4($right$) that some corners in the upper part of the traffic cone are not detected. Additionally, as mentioned above, this algorithm uses a threshold that depends on the maximum value of response function in a given sub-image. So, the quality of the obtained result depends on the amount of sub-images used to split up the given image; if this value is too low, it could result in a large number of false corners, in the case that the resulting sub-image does not contain strong corners. It could also happen that if in a given sub-image there is a very strong corner, the resulting threshold will be too high to discern weak corners. The authors of [9], to prevent this effect, advise to use a small number of sub-images, e.g. nine. But in real images nobody knows how big is a flat region and to illustrate this behaviour we show results obtained when the number of sub-images is increased. Figure 5 shows the results obtained when the given image is split up into 9 and 81 sub-images, respectively, and using $p = 0.005$.

It can be seen that this algorithm does not properly work when flat regions are present in the given image. Moreover, using a large set of sub-images, a large number of false corners are detected in flat regions (see the top right region in Fig.5). Although the results are better when 9 sub-images are considered, there still are several false corners detected in the region close to the drumsticks (see Fig.5($left$)). The number of corners detected in these cases are 11649 and

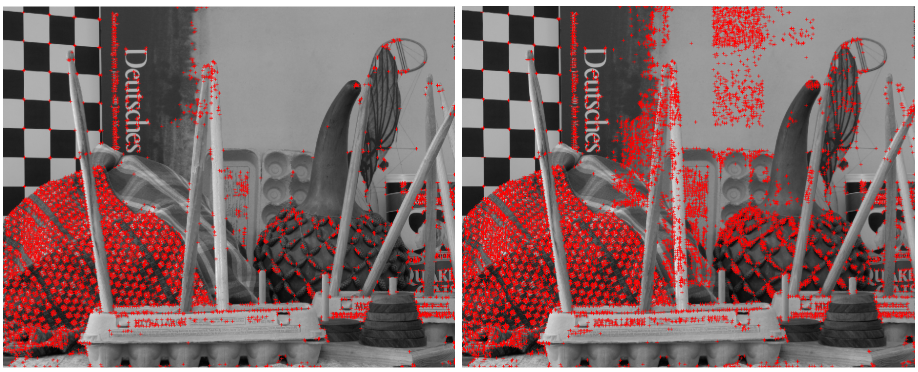

Fig. 5. Result by tuning Harris algorithm according to [9], setting $p = 0.005$ and considering: ($left$) 9 sub-images and ($right$) 81 sub-images respectively

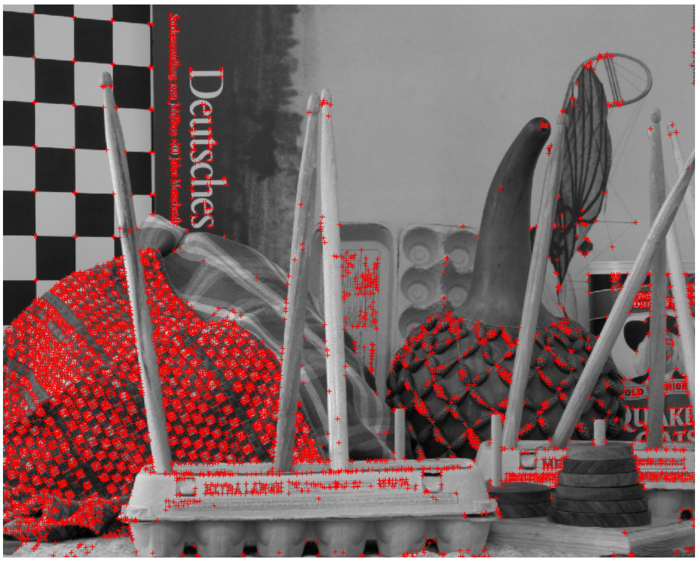

Fig. 6. Result from the proposed gradual thresholding approach

8957, respectively. The result obtained with the proposed approach is presented in Fig.6. It can be appreciated that the proposed approach works fine also in presence of flat regions. In this case only 5917 corner points have been detected. Note that this result has been obtained without changes in the parameters used by the proposed approach (the same setting that the one used to obtain the result in Fig. 3 is considered).

5 Conclusions

This paper presents a novel approach that overcome the problem of the original Harris corner detector algorithm to find the best threshold for each image, in order to detect the most important corners. The approach does not use a single threshold but uses a criteria that discriminate between corners and noisy points in a gradual way. The obtained results show that the proposed approach has a better capability to reject false corners and, at the same time, to detect some weak corners since a local analysis is performed. Further work will be focused on studying other possibilities to discern between strong noisy points and weak corners in the midway interval. Another improvement will be based on the use of color images.

Acknowledgements. This work was partially supported by the Spanish Government under Project TIN2011-25606.

References

1. Alessio, D., Beoldo, A., Regazzoni, C.: Multitarget tracking with a corner-based particle filter. In: Proceedings of the International Conference on Computer Vision Workshops, Kyoto, Japan, pp. 1251–1258 (2009)
2. Sappa, A., Dornaika, F., Gerónimo, D., López, A.: Registration-based moving object detection from a moving camera. In: IROS 2008 2nd Workshop on Perception, Planning and Navigation for Intelligent Vehicles, Nice, France, pp. 65–69 (September 26, 2008)
3. Rosten, E., Porter, R., Drummond, T.: Faster and better: A machine learning approach to corner detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 105–119 (2010)
4. Zhang, X., He, G., Yuan, J.: A rotation invariance image matching method based on harris corner detection. In: International Congress on Image and Signal Processing, Tianjin, China, pp. 1–5 (2009)
5. Sappa, A., Dornaika, F.: An edge-based approach to motion detection. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3991, pp. 563–570. Springer, Heidelberg (2006)
6. Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 183–196. Springer, Heidelberg (2010)
7. Harris, C., Stephens, M.: A combined corner and edge detector. In: 4th. Alvey Vision Conference, pp. 147–151 (1988)
8. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. International Journal of Computer Vision 37, 151–172 (2000)
9. Shen, S., Zhang, X., Heng, W.: Auto-adaptive harris corner detection algorithm based on block processing, pp. 1–4 (2010)
10. Yuan, X., Pun, C.: Invariant digital image watermarking using adaptive harris corner detector. In: International Conference on Computer Graphics, Imaging and Visualization, pp. 109–113 (2011)
11. Bruce, N., Kornprobst, P.: Harris corners in the real world: A principled selection criterion for interest points based on ecological statistics. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2160–2167 (2009)
12. Svoboda, T., Kybic, J., Hlavac, V.: Image processing, analysis and machine vision: A matlab companion. In: Cengage (2008)