

AN EFFICIENT BINARY CORNER DETECTOR

P. Saeedi, P. Lawrence and D. Lowe

Department of Electrical and Computer Engineering, Department of Computer Science
University of British Columbia
Vancouver, BC, V6T 1Z4, Canada
parvanes@ece.ubc.ca

ABSTRACT

Corner extraction is an important task in many computer vision systems. The quality of the corners and the efficiency of the detection method are two very important aspects that can greatly impact the accuracy, robustness and real-time performance of the corresponding corner-based vision system. In this paper we introduce a fast corner detector based on local binary-image regions. We verify the performance of the proposed method by measuring the repeatability rate under various illumination, scale and motion conditions. Our experimental results show that while the quality of the features are comparable with other conventional methods, ours delivers a faster performance.

1. MOTIVATION

A real-time feature-based motion tracking system can be greatly effected by the efficiency of the feature extraction method. In our previous work [1], we developed a 3D trajectory tracking system suitable for autonomous robot vehicle. The system includes a head with three on-board CCD cameras, which can be mounted anywhere on a mobile vehicle. By processing consecutive trinocular sets of precisely aligned and rectified images, the local 3D trajectory of the vehicle in an unstructured environment can be tracked. First, a 3D representation of stable features in the image scene is generated using a stereo algorithm. Second, motion is estimated by tracking matched features over time. The motion equation with 6-DOF is then solved using an iterative least squares fit algorithm. Finally, a Kalman filter is used to optimize the world representation of scene features. Figure 1 represents a simple block diagram of this system. In this system the computational cost is reduced effectively by processing scene corners using Harris method [2]. However still a great portion of the processing time was spent on extraction of the corners of the stereo images. Not only did this prevent the system from performing in real-time, but due to the lost frames it also impacted the accuracy and robustness of the motion estimation. For this reason the possibility of the a faster corner detection algorithm was studied

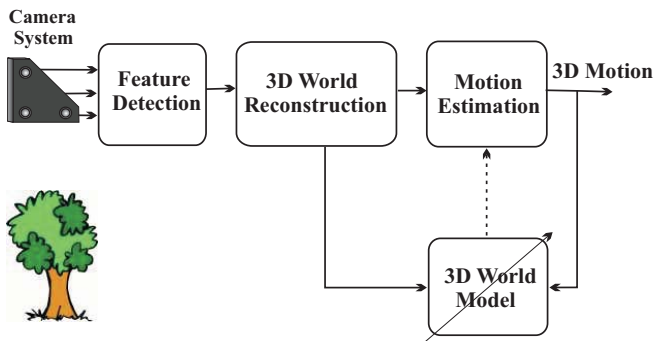


Fig. 1. The overview of the existing 3D motion tracking system.

that led to a simple and fast corner detector with acceptable robustness. In our system the speed of the corner detection is of utmost importance for real-time operation. Improving the speed of the corner extraction algorithm can lead to a higher frame rate processing, which leads to smaller changes between consecutive frames allowing for a better motion estimation.

2. INTRODUCTION

Many vision based tracking systems estimate the camera motion by measuring the relative changes on the projection of the identical scene features in different image frames. Although globally all points in the scene convey some information about the motion, locally not all the pixels of the image carry valuable motion information. For example, edges, occlusions or areas of uniform intensity, locally can convey, at best, only partial information about the motion. For this reason, special attention is paid to selecting regions with higher information content. There have been many different criteria for choosing such regions, mostly based on areas with high second-order derivatives or high frequency content. In spite of the fact that these methods deliver traceable features, there are several characteristics that make some of them more reliable than the others. For instance, good lo-

calization, robustness with respect to noise and illumination changes, and efficiency of the detection can lead to a more accurate and/or faster estimation.

In this paper a binary corner detector is introduced that is more efficient to implement than the existing conventional corner detectors such as Harris [2] and SUSAN [3] and yet quantitatively is comparable with them. The paper is organized as follows. Section 2 presents the details of the binary corner detector. Section 3 explains the measurement for qualitative comparison of the performance. Section 4 and 5 demonstrate the results under various imaging conditions and compare them with those of Harris and SUSAN. Section 6 inspects the computational complexity, and finally, the conclusion is presented in section 7.

3. BINARY CORNER DETECTOR

Much of the work on detection of image 2D structures is dedicated to corners. This is mainly due to their discreteness and invariance with respect to scale, rotation, and point of view. In the next section a brief overview of several corner detection algorithms along with the principal of the proposed method are described.

3.1. Corner detection basis

One simple definition for a corner is a point at high curvature boundaries where the two regions of an image with distinctive intensity values meet. Such a definition has led to an extensive number of edge curvature based corner detectors. Kitchen and Rosenfeld [4] find points in the image where the product of the gradient magnitudes and edge contour curvatures are maximum. Wang and Brady [5] define corners as points with maximum gradient on the direction perpendicular to the edge curvature. Another definition for a corner is points with large intensity variations in all the directions. Moravec [6] presents a method based on computing the intensity autocorrelation on four directions over a small window. Features with the local minimum autocorrelations are declared as corners. Harris and Stephens [2] improve Moravec's method by computing auto-correlation for the squared first order image derivatives. The auto-correlation values over all the possible directions are computed through a transformation whose eigen values are proportional to the principal curvatures of the auto-correlation. Points with high curvature values are declared as corners. Smith and Brady [3] introduce a significantly different low level corner detector by measuring the similarity of the related parts of the image to each individual pixel. They associate each pixel with its local image regions with the similar brightness. Distinctive features are detected by minimizing these image regions.

3.2. Binary corner detector

The main emphasis of this work is on exploiting binary images and substituting arithmetic operations with logicals. To generate a binary image that contains a good low-level information content, first the Laplacian is computed at each point of the intensity image. Horn [7] approximates the image Laplacian value by:

$$\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \approx (I_{i-1,j} + I_{i,j-1} + I_{i+1,j} + I_{i,j+1} - 4I_{i,j}) \quad (1)$$

$I_{i,j}$ stands for the image intensity value at row i and column j . Such an approximation for 2D Laplacian is separable and is implemented efficiently by logical operations. The binary image is then generated by the invariance of the sign of the Laplacian value at each point. Figure 2 shows a typical image and the corresponding binary image.

At this point a circular mask is placed on each point of

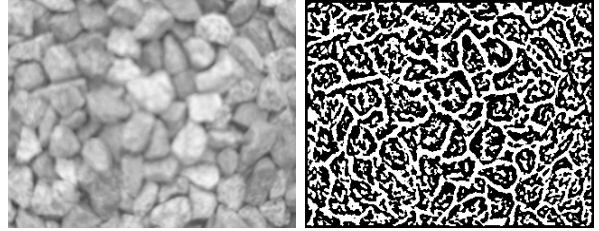


Fig. 2. An image and its binary sign of Laplacian image.

the binary image. The binary value of each point inside the mask is compared with that of the central point.

$$C(p_0, p) = \begin{cases} 1 & \text{if } L(p) = L(p_0), \\ 0 & \text{if } L(p) \neq L(p_0). \end{cases} \quad (2)$$

$L(p)$ represents the binary image value at location $p(x, y)$. Now a total running sum n is generated from the output of $C(p_0, p)$.

$$n(p_0) = \sum_w C(p_0, p) \quad (3)$$

n represents the area of the mask where the sign of the Laplacian of the image is the same as that of the central point. For each pixel to be considered a potential corner, the value of n should be smaller than at least half the size of the mask w . This value is shown as by t in corner response equation (4).

$$R(p_0) = \begin{cases} n(p_0) & \text{if } n(p_0) < t, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

At this point for each candidate with $R(p_0) > 0$, a center of gravity $G(p_0)$ is computed.

$$G(p_0) = \sqrt{g(x_0)^2 + g(y_0)^2} \quad (5)$$

where

$$g(x_0) = \frac{\sum (x_0 - x)}{n(p_0)}, \quad g(y_0) = \frac{\sum (y_0 - y)}{n(p_0)} \quad (6)$$

The center of gravity G , provides the corner direction, as well as a condition to eliminate points with random distributions. Random distributed binary patches tend to have a center of gravity fairly close to the center of the patch. Therefore, all points with close center of gravities are filtered out from the remaining process.

$$G(p_0) > |r_g| \quad (7)$$

The two conditions in (4) and (7) by themselves do not provide enough stability. Therefore a third inspection is performed by computing the directional derivative along the corner direction for the remaining candidates. Once again points with small directional intensity variations are eliminated. This condition is shown by:

$$|I(p_0) - I(p)| > I_t \quad (8)$$

I_t represents the brightness variation threshold. Figure 3.a to 3.e illustrate the principal of the binary corner detector through different cases. Figure 3.a shows a circular mask

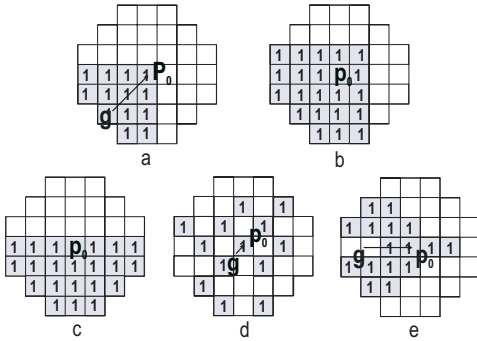


Fig. 3. Illustration of different cases for a corner.

on an ideal corner. The response $R(p_0)$ for this corner is smaller than half the mask size and the center of gravity g is located in a fair distance from center of the mask p_0 . Furthermore the intensity variation of the mask center along $\vec{gp_0}$ is large. Figure 3.b and 3.c demonstrate cases where the mask centers are located either on an edge or on an area of uniformity. For such cases, $n(p_0)$ fails in equation (4). Figure 3.d represents an example where pixels on the mask have a random distribution. Although the initial verification in equation (4) may be satisfied, the condition on a distant center of gravity in (7) fails. Finally, Figure 3.e shows an instance where the first two tests are fully fulfilled, but the intensity variation along vector $\vec{gp_0}$ fails to provide a sufficiently large value, condition (8).

4. EVALUATION METRIC

Schmid *et.al.* [8] [9] introduce an evaluation metric to verify the performance of several feature detectors. This measurement is defined as the percentage of the total detected corners that are repeated in the two images, and is called the repeatability rate. A scene point P with projection p_i in image I_i is repeated, if the corresponding point p_j is detected in image I_j . In computing this rate, points that are seen in one image but due to the existing transformation, H_{ij} , between the two frames are out of the other one, are not counted. Also since the corresponding point to p_i , p_j , is usually not detected at the exact predicted position p_j but rather in a neighborhood of ϵ , the repeatability rate is defined as a function of such a neighborhood by:

$$r_i(\epsilon) = \frac{|D(\epsilon)|}{n_i} \quad (9)$$

where

$$D(\epsilon) = \{(p_i, p_j) | dist(p_i, H_{ij}p_j) < \epsilon\} \quad (10)$$

n_i represents the number of the corners in image I_i that can be potentially observed in image I_j .

5. REPEATABILITY RESULTS

In this section, the performance of the algorithm under different imaging conditions is presented. A reliable computation of repeatability in this part involves a careful registration of image frames. For this purpose first the registration parameters between each image pair are roughly estimated by solving the registration equations for some manually found correspondences. Second, through a recursive least square minimization, the registration parameters are refined until an error residual smaller than half a pixel is attained.

5.1. Camera noise

Camera noise effect is studied by processing images of a static scene that are captured under identical conditions but at different moments. The results have been shown in Figure 4. In this experiment Harris delivers the highest robustness and is followed closely by binary method. SUSAN method however does not retain a good rate mainly due to the elimination of the Gaussian smoothing.

5.2. Rotation

The behavior of the method under rotation is studied by processing a set of images that are acquired while the camera was rotated near to its optical axis. Due to the limited rotation mechanism the transformations between the frames

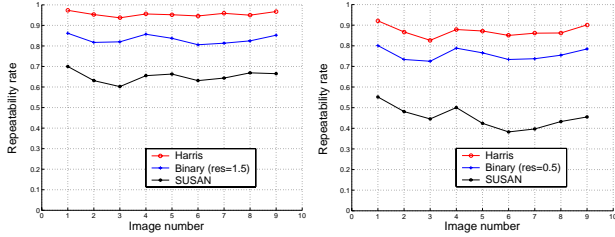


Fig. 4. Repeatability result for camera noise.

were not purely rotational and included translation as well. Figure 5 shows the repeatability results for neighborhoods 0.5 and 1.5 pixels. This experiment covers rotations between 0° to 180° with steps of 10° . As the rotation angle increase the repeatability of Harris and binary become closer to each other. Since in many real-time applications the rotation between the two consecutive frames is small, another experiment has performed to test the methods for smaller rotations covering a range of 0° to 10° , Figure 7.

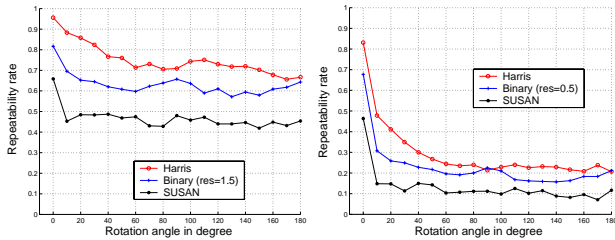


Fig. 5. Repeatability rate under rotation.

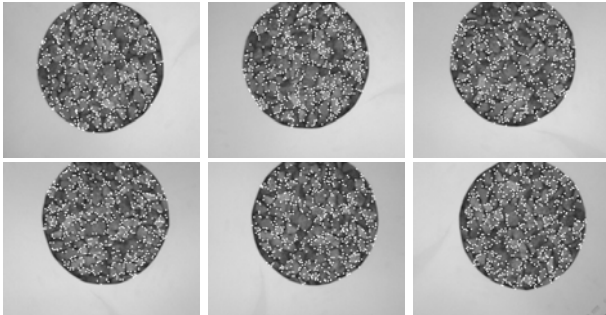


Fig. 6. Images of a scene under different rotations.

5.3. Scale

The scale change is accomplished by moving the camera in the direction perpendicular to the image plane. The scale change over 1.8 is not experimented due to the high sensitivity and poor performance of the methods. Figure 8 represents the repeatability rate for the set of images shown in

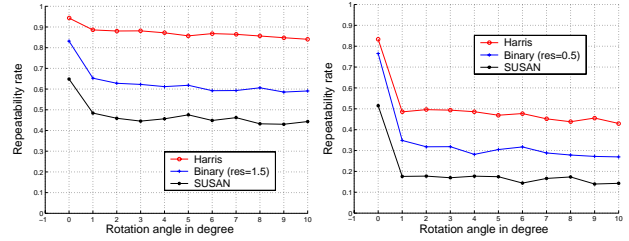


Fig. 7. Repeatability rate under small rotations.

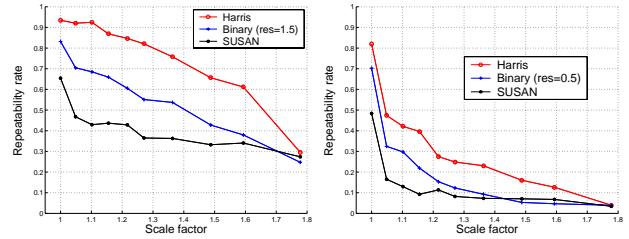


Fig. 8. Repeatability rate for different scales.

Figure 9. The results show that all the methods perform better in a larger neighborhood, ($\epsilon = 1.5$). This can be explained by the fact that at different scales identical features of the scene project onto areas with different resolutions. Therefore, although corners are considered invariant to scale change, their positional precision and their repeatability are highly scale dependent.

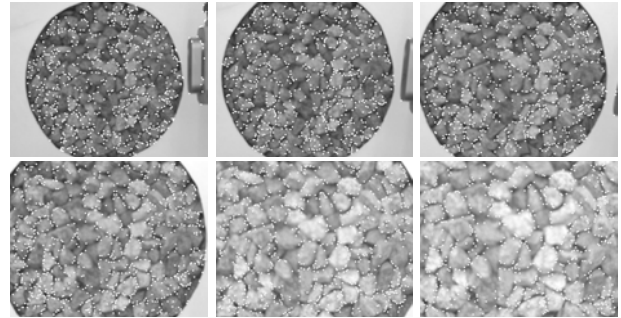


Fig. 9. Images of a scene under different scales.

5.4. Illumination

In this part the sensitivity of the proposed method on image intensity variation is studied.

5.4.1. Uniform change

Uniform illumination change is simulated for this part due to the existing limitation on changing our camera's aperture. For this purpose first a set of images from a static

scene with an identical point of view at different moments is acquired, Figure 11; then, the average grey level of each image is manually changed. Figure 10 displays the computed

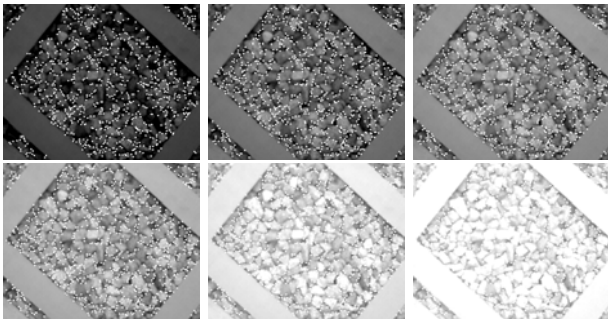


Fig. 10. Images of a scene with linear illumination change.

repeatabilities. In these graphs reveals that as the relative gray level with respect to the reference image changes, the number of repeating features decreases in these methods.

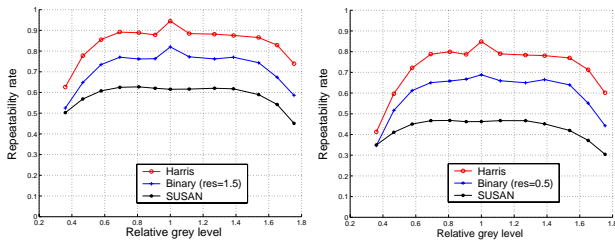


Fig. 11. Repeatability rate under uniform illumination change.

5.4.2. Directional change

One of the important features of a good corner detector is its robustness under a more complex lighting condition. Therefore, the method's stability is evaluated under directional illumination variations in which the light source lightens the scene from different directions. Figure 12 represents some of the processed image taken under the above condition. The light source is moved with a step of 10° and covers a range of 0° to 90° . As the light source moves from a direc-

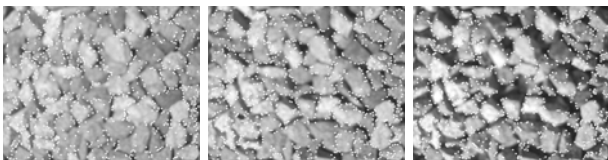


Fig. 12. Images of a scene under changing illumination.

tion perpendicular to the camera plane (0°), the left image in Figure 12, to its side (90°), the right image in Figure 12, shadow effect becomes stronger. Such shading effect causes

object boundaries to move toward the light source. This notably impacts the repeatability by a great amount specially for small neighborhoods. Figure 13 represents the results of this experiment.

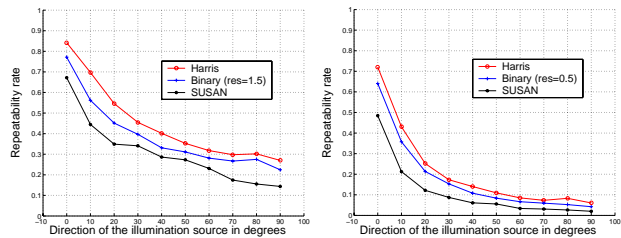


Fig. 13. Repeatability rate for changing illumination.

6. DISCUSSION

As presented in the previous section, in average the repeatability of the Binary method is a bout 20% less than Harris and 15% more than SUSAN. This might seem to be a problem at first but as will be discussed in here for many vision based applications including our trajectory tracking system, it is effecting the reliability. There are two important aspects that make the Binary corner detector still suitable as following:

- If the acquired images are processed faster, the transformation between the two consecutive frames is smaller. A smaller difference between the two frames can partially compensate for the lower repeatability rate of the Binary method.
- One of the questions that we tried to answer in our previous work [1] was that how many corners are needed for a reliable motion estimation. It was observed that the motion parameters can be determined from a minimum of 30 to 40 corners. However a number between 50 and 100 guarantees a reliable motion estimation. For a typical outdoor image Harris detects about 400 corners. Losing 20% of the corners in Binary method results in 300 corners that is still plenty to provide sufficient information for a reliable motion estimation.

Next section provides a discussion of the efficiency.

7. COMPUTATIONAL COMPLEXITY

As it is customary in existing research, we compare the computational complexity of all the methods by comparing their running time under the same conditions. We use images of 240×320 pixels and average the running time for one thousand executions on a 1.4 GHz AMD Athlon(tm)

Processor[®]. The results of this comparison are presented in Table 1. As shown in this table SUSAN algorithm has the

Table 1. Comparison of the time complexity of the three methods.

Method	Execution time (msec)
Binary	23.293
Harris	37.674
SUSAN	168.452

slowest performance. Details of the SUSAN corner detector suggest a simple computational complexity. This is due to the fact that most of the computations and evaluations are performed on a selection of pixels that are considerably less than the total pixels of the image. However the initial inspections for nominating such candidates include a considerable number of the operations, causing SUSAN algorithm to have the slowest performance.

Harris method performs significantly faster than SUSAN, however 60% of its time is spent on the Gaussian smoothing of each image and its corresponding squared first order derivatives. This is in spite the fact that all the 2D Gaussian filters are approximated by two 1Ds and a large number of the arithmetic operations are approximated by logical operations through a sigma of 0.8.

Binary corner detector performs 1.64 times faster than Harris and 7.23 times faster than SUSAN. Moreover, about 70% of all the employed operations can be substituted by bitwise operations for an even faster implementation.

8. CONCLUSIONS

In this paper we described a fast corner detector for selecting informative pixels of an image. We evaluated our method by computing the repeatability rate and comparing them with those of Harris and SUSAN corner detectors. All the experiments are performed on images of outdoor scenes including natural geometric shapes. The faster performance of this method may allow for a better real-time performance of feature based vision systems. On average, the repeatability rate for our method is about 20% less than Harris. As mentioned earlier, the estimated motion is concluded from a small subset of the image points, usually about 10% to 5% image size. A decrease of 20% in that number still provides more than enough points and delivers the same motion parameters. Also as system performs faster, the changes between consecutive frames are less that cause a higher repeatability rate. Moreover those features that are not repeated in the next image may appear in the later frames. Through the use of Kalman filtering these points in the world can be maintained for a while even if they are not physically in the scene or just

simply missed from the detection algorithm. Furthermore generating the binary images, creates another opportunity for accelerating the process of similarity measurement. As presented in [10], substitution of an expensive SSD with a binary XOR-correlation can speed up the comparison at least by a factor of 3.63, while maintaining a 92.7% accuracy. The future work includes inspecting the improvement of the accuracy and real-time performance of the existing 3D motion tracking system while employing the proposed method.

9. ACKNOWLEDGMENT

This work was supported by the Canadian IRIS/PRECARN Network of Centers of Excellence.

10. REFERENCES

- [1] P. Saeedi P. Lawrence and D. Lowe, "3D motion tracking of a mobile robot in a natural environment," *IEEE Int. Conf. Robotics and Automation*, pp. 1682–1687, 2000.
- [2] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceeding 4th Alvey Vision Conference*, pp. 147–151, 1988.
- [3] S. M. Smith and J. M. Brady, "SUSAN- A new approach to low level image processing," *J. Computer Vision*, pp. 45–78, 1997.
- [4] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters*, pp. 95–102, 1982.
- [5] H. Wang and M. Brady, "A practical solution to corner detection," *IEEE Int. Conf. Image Processing*, pp. 919–923, 1994.
- [6] H. Moravec, "Visual Mapping by a Robot Rover," *Proc. Int. Conf. Artificial Intelligence, Tokyo, Japan*, pp. 599–601, 1979.
- [7] Berthold K.P. Horn, *Robot Vision*, The MIT Press, Massachusetts, 1992.
- [8] C. Schmid and R. Mohr, "Local Greyvalue Invariants for Image Retrieval," *IEEE Int. Conf. Pattern Analysis and Machine Intelligence*, pp. 530–535, 1997.
- [9] R. Mohr C. Schmid and C. Bauckhage, "Comparing and Evaluating Interest Points," *Int. Conf. Computer Vision*, pp. 230–235, 1998.
- [10] P. Saeedi, "A fast similarity metric for matching features in image sequences," Tech. Rep., University of British Columbia, 2001.