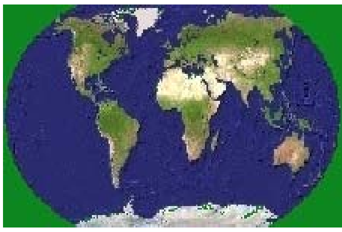
International Journal of Multidisciplinary Research and Development



International Journal of Multidisciplinary Research and Development

Volume: 2, Issue: 6, 99-102
June 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

Shraddha S. Aher
savitribai phule pune university, Department of Entc, RMDSSCOE, Warje, Pune, Maharashtra, India

Correspondence:
Shraddha S. Aher
savitribai phule pune university, Department of Entc, RMDSSCOE, Warje, Pune, Maharashtra, India

Analysis of different algorithms on edge and corner detection

Shraddha S. Aher

Abstract
Many computer vision applications use edge and corner detectors as primary operators before high level processing, such as object recognition and tracking. For instance, the information associated to the edges of an object in an image which is, in many cases, sufficient to identify the object. In this section, we will give an overview on these two components of vision systems, showing their foundation and describing the most common techniques of implementation.

Keywords: Canny edge detection,Harris corner detection,SUSAN detector

Introduction
Feature detection algorithms, such as edge and corner detection, are essential components of many computer vision applications, e.g., image segmentation, object recognition, and feature tracking. The Canny edge detector and the Harris corner detector are the most widely-used feature detection algorithms due to their reliable performance with noisy images. Several techniques have been proposed for both edge and corner detection. As in the case of Canny, the Harris technique is a well-known robust solution for tracking interesting points in a video stream[1].

Edge Detection
Segmentation is a process of distinguishing objects from the background. Hence, Image segmentation is distinguishing or partitioning the image from its background. The four main approaches used for image segmentation are: threshold techniques, edge detection techniques, region-based techniques, and connectivity preserving relaxation methods. Most widely and important amongst these four techniques is “edge detection”. The level of the subdivision has to stop when the object or image of interest have been partitioned. Picking up an appropriate technique for “good” segmentation is a challenging task. Edge being such an essential part in an image, its study becomes important. Some important features can be extracted from an edge of any image (e.g.: corners, lines, curves)[3, 5].

Edge detection is a technique in which the points where image brightness changes sharply or formally are identified. These points are organized under line segments called edges. Edge detection also aims to classify and place discontinuities in an image. Noise and image both have high frequency, hence edge detection becomes difficult. The main objective of studying various edge detection techniques and analyzing their performance is due to problems such as fake edge detection, noisy images, missing edges etc.

Edge: The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The edges extracted from a two-dimensional image of a three-dimensional scene can be classified as either viewpoint dependent or viewpoint independent. A viewpoint independent edge typically reflects inherent properties of the three-dimensional objects, such as surface markings and surface shape. A viewpoint dependent edge may change as the viewpoint changes, and typically reflects the geometry of the scene, such as objects occluding one another. A typical edge might for instance be the border between a block of red color and a block of yellow. In contrast a line can be a small number of pixels of a different color on an otherwise unchanging background. For a line, there may therefore usually be one edge on each side of the line.

Edge detection is a fundamental tool for image segmentation. It partitions a digital image into multiple regions or pixels. There are various techniques for edge detection based on error minimization, fuzzy logic, genetic algorithms, neural networks etc.

Edge: it is a sign of lack of continuity in an image. Any point where discontinuity occurs in an image is a edge. They occur due to significant change in the intensity.

Steps in Edge detection

A. Filtration

Every image is associated with some intensity values, random change in these values can result in noise. Some common noise is: salt and pepper noise, impulse noise etc. Noise can result in difficulties in effective edge detection; hence image has to be filtered in order to reduce the noise content that leads to loss of edge strength. It is also termed as Smoothening.

B. Enhancement

Improving the quality of image is termed as enhancement. It aims to produce an image which is better and more suitable than original. A filter is applied in order to enhance the quality of edge in image.

C. Detection

Several methods are adopted to determine which points are edge points and which a edge pixels should be discarded as noise.

These steps should be followed carefully in order to detect the edges effectively, as the next steps are dependent on edges detected.

Types of edge detection algorithm

A. First order edge detector

1. Canny edge detector

B. Classical edge detector

1. Robert operator
2. Prewitt operator
3. Sobel operator

C. Second order edge detector

1.Marr hildrith edge detector

A. First order edge detector

1. Canny Edge Detection:

Canny edge detector is one of the most commonly used image processing tools. It detects edges in a very robust manner.

Unlike Roberts Cross and Sobel, the canny operation is not very susceptible to noise. It takes less time than Roberts cross. It is one of the most important methods to find the edges by separating noise from input image. The algorithm is adaptable to various environments.

It is a better method because it extracts the features in an image without disturbing its features. There are certain criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed. The second criterion is that the edge points be well localized i.e. the distance between the edge pixels as found by the detector and the actual edge should be minimum. A third criterion is to have only one response to a single edge**.**

1.1 Importance of Canny:

Despite of number of edge detection techniques available canny algorithm is considered because it contains a number of adjustable parameters which can affect the computation time and effectiveness of the algorithm.

a) The size of the Gaussian filter: The smoothing filter used in the first stage directly affects the results of the detection of small, sharp lines. A larger filter causes more blurring, smearing out the value of an given pixel over a larger area of image.
b) The use of two thresholds with hysteresis allows more flexibility than in a single-threshold. A threshold set too high can miss important information. On the other hand, a threshold set too low will falsely identify irrelevant information (such as noise) as important.

The edge detection in this technique is optimized with regard to the following criteria

a) Maximizing the signal-to-noise ratio of the gradient.
b) Edge localization for ensuring the accuracy of edge.
c) Minimizing multiple responses to a single edge.

The steps are as follows

a) Smoothing: Blurring of given image to remove noise. It is done by convolving the image with Gaussian filter. A suitable mask is calculated and Gaussian smoothing can be performed using standard convolution methods.

Convolve image f(r,c) with a Gaussian function

$$F^\wedge (r,c)=f(r,c)*G(r,c) \qquad (1)$$

Where G(r,c) is the Gaussian function

b) Finding Gradients: After smoothing the image next step is to find the strength of edge by taking the gradient of the image. Sobel operator is used to perform 2-D spatial gradient measure on an image. The edges should be marked where the gradients of the image has large magnitudes, finding the gradient of the image by feeding the smoothed image through a convolution operation with the derivative of the Gaussian in both the vertical and horizontal direction.
c) Non-maximum suppression: Only local maxima should be marked as edges. Finds the local maxima in the direction of the gradient, and suppresses all others, minimizing false edges. A. The “non-maximal suppression” step keeps only those pixels on an edge with the highest gradient magnitude. These maximal magnitudes should occur right at the edge boundary, and the gradient magnitude should fall off with distance from the edge.
d) Double Thresholding: Potential edges are determined by thresholding, Instead of using a single static threshold value for the entire image, the Canny algorithm introduced hysteresis thresholding, which has some adaptively to the local content of the image. There are two threshold levels, th, high and tl, low where th > tl. Pixel values above the th value are immediately classified as edges.
e) Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very strong edge.

A simple threshold may actually remove valid parts of a connected edge, leaving a disconnected final edge image. This happens in regions where the edge’s gradient magnitude fluctuates between just above and just below the threshold. Eliminating pixels whose gradient magnitude D falls below some threshold removes problem, but it introduces a new problem. Hysteresis is one way of solving this problem. Instead of choosing a single threshold, two thresholds thigh and tlow are used. Pixels with a gradient magnitude D<tlow are discarded immediately. However, pixels with tlow ≤ D < thigh are only kept if they form a continuous edge line with pixels with high gradient magnitude (i.e.,above thigh).

Hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1.

B. Classical edge detector

1. Robert's cross Operator:

This is somewhat similar to Sobel and Prewitt Operator. It is a 2-D spatial gradient measurement of an image. The pixel value represents the absolute magnitude of spatial gradient of input at that point. Operator has 2x2 convolution kernel. Kernel responds to maximize edges running at 450 to each pixel grid. For each of the perpendicular orientations there is one corresponding kernel.

The magnitude is given by the same formula as of the Sobel but the orientation of the angle is given by: $O = \arctan(GX/GY) - 3\pi / 4$.

2. Prewitt Operator:

It is similar to the Sobel Operator and is used to detect vertical and horizontal edges in an image.

3. Sobel Operator:

It is 3x3 convolution kernels. One kernel is simply the other rotated by 900. It is a row edge detector.

This figure shows the masks used by Sobel operator. The kernel can be applied separately to input image for obtaining gradient component in each orientation i.e. GX and GY. The magnitude is given by:

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

$|G|=\sqrt{GX2 + GY2}$ (2)

its approximation is done by:

$|G|=|GX| + |GY|$ (3)

The orientation of angle is given by:

$O = \arctan(GX/GY)$ (4)

C. Second order edge detector

1. Marr hildrith edge detector:

It first smoothes the image and then computes the Laplacian. This yields in double edge image; hence for finding the edge the zero crossing between the double edges is taken.

Corner Detection

Corner detection is a popular research area in image processing and therefore many corner detectors have been presented. Some of them are widely used in industries. such as Harris detector and SUSAN detector[7].

Types of corner detector:

Common corner detection methods can be divided into three groups:

1) Template based corner detection;
2) Contour based corner detection;
3) Direct corner detection.

As presented in template based corner detection methods use different representative templates to match the image. Correlations between templates and the image are used to detect corners. However, this category of methods has several drawbacks. For example, the representative templates cannot cover all possible corner situations. Therefore, the detection performance highly depends on the choice of appropriate templates. Furthermore, after the correlations between the templates and the image are determined. An appropriate threshold should be carefully chosen to determine the existence of corners. Contour based corner detection methods are based on edge detection. In this category of methods, edges in the image are detected first. Then, the corner is detected along the contour. Direct corner detection methods use mathematical computations to detect the corner. This category of methods usually applies some statistical operations to the image first. Then, corners are detected based on statistical information.

Some famous corner detection methods include SUSAN detector, Harris detector, wavelet based detector and blob detector. Harris detector is based on local auto-correlation function. It is also a combined edge and corner detector. The core idea of Harris detector is calculate the Eigen values and eigenvectors of a small region. Then, use the largest two Eigen values to calculate some functions. Finally, use the function value and a threshold to detect the corner.

SUSAN corner detector does not require derivative. That is why it can work well when the noise is present. The main idea of SUSAN is the usage of a mask to count the number of pixels having the same brightness as the center pixel. By comparing the number of pixel having the same brightness as the center pixel with a threshold, the detector can determine whether the center pixel is a corner.

Since SUSAN detector and Harris detector is the two most famous corner detector, we can compared the performance of those two detectors in terms of complexity, stability, execution time, and so on. The simulation results showed that Harris detector is better than SUSAN detector. As explained in SUSAN uses a fixed global threshold instead of an adaptive threshold. Moreover, the anti-noise capability of SUSAN detector is worse than Harris detector.

A corner is defined as an area that exhibits a strong gradient value in multiple directions at the same time. The Harris operator uses this premise to find Corners in an image. The first step is to obtain the first partial derivative of the image function I(u, v) in both directions, horizontal and vertical, based on the approximations. With the values of Ix and Iy, it is possible to calculate the elements of the matrix M, described in equations,

M= [A C]
[C B] (5)

$A= Ix^2 \times w$ (6)

$B= Ix^2 \times w$ (7)

$C= (Ix \times Iy) \times w$ (8)

Where ω is a smoothing circular operator, e.g. a Gaussian filter. The final step is to obtain the Harris operator response R as in equation.

$R = Det [M] - k \cdot Tr2 [M]$ (9)

Where R is positive in corner regions, negative in edge regions, and is very small in flat regions, and k is a coefficient that, in practice, is a fixed value in the range of 0.04 to 0.06. This step can also be obtained by analyzing the eigenvalues of the matrix M.

1. Harris Corner detection Algorithm

The Harris corner detector implementation is divided into five steps.

Color to grayscale

The first step of this implementation is identical to the one presented in the previous section for the Canny implementation.

Spatial derivative computation

This step computes the first derivatives Ix (u, v) and Iy(u, v) of the input image f(u, v) by applying the approximations.

Building the matrix M

In this step, the values A, B, and C, defined in previously eq(6),(7),(8) are computed to build the matrix M, (5)defined in previous equation. Three sub-pipelines are applied in parallel to perform these computations. Each sub-pipeline is formed by a multiplier, a 5×5 NE block, and a Gaussian filter.

Harris response

The Harris response operator computes the values of R, defined in (9). To keep the pixel stream within an 8-bit resolution without losing weak corner values, R is truncated at 255. This approach can create large regions around the corner spot with saturated values, making difficult the following NMS process. To solve this, a threshold block eliminates low R values that do not represent corners followed by an extra Gaussian filter to blur these saturated regions, producing a maximum spot at the center of these regions.

Non-maximum suppression

The final step is to select the best values representing corners. To do this, a 9×9NMS block analyses a region (window) and marks the maximum value as a detected corner.

Conclusion

I have fully studied about the references about edge and corner detections. However, for the constraints on the pages and my purpose, I only wrote a proportional summarization but not the detail histories and algorithms about edge and corner detections. I believe that I have completed the fundamentals of the detections. On the later research, there are still many enhanced algorithms to develop. Future works may still keep on finding the latest algorithms in the color image. Since the definition for an edge of a color image would be in several ways and pixels have not only intensity component but also hue and saturation components that form the vector differences between R, G, B.

References

1. Paulo Ricardo Possa, Sidi Ahmed Mahmoudi, NaimHarb, Carlos Valderrama, “A Multi-Resolution FPGA-Based Architecture for Real-Time Edge and Corner Detection”, IEEE Transactions on Computers, January 2013.
2. G. Anusha, Dr. T. JayaChandra Prasad, Dr. D. SatyaNarayana, “Implementation of SOBEL Edge Detection on FPGA,” International Journal of Computer Trends and Technology, 2012, Vol. 3 Issue3, pp. 472-475.
3. G.T. Shrivakshan, Dr. C. Chandrasekar, “A Comparison of various Edge Detection Techniques used in Image Processing,” IJCSI International Journal of Computer Science Issues, September 2012, pp. 269-276.
4. Jyoti Malik, Ratna Dahiya, G. Sainarayanan, “Harris Operator Corner Detection using Sliding Window Method”, International Journal of Computer Applications, May 2011, Vol. 22,No.1, pp. 28-37.
5. Mamta Juneja, Parvinder Singh Sandhu, ”Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain”, International Journal of Computer Theory and Engineering, December, 2009, Vol. 1, No 5, pp. 614-621.
6. Poonam Dhankhar, NehaSahu, “A Review and Research of Edge Detection Techniques for Image Segmentation,” International Journal of Computer Science and Mobile Computing, July 2013, Vol.2 Issue 7, pp.86-92.
7. Reddy Sekhar K, Mahesh M, “Performance Evaluation of Corner Detectors: A Survey,” IJCSMC, July 2013, Vol. 2, Issue. 10, pp. 226 – 233 October 2013.
8. Rashmi, Mukesh Kumar, RohiniSaxena, “Algorithm And Technique On Various Edge Detection: A Survey,” Signal & Image Processing: An International Journal (SIPIJ), June 2013, Vol.4, No.3, pp. 65-75.
9. Zhenxing Luo,” Survey of Corner Detection Techniques in Image Processing”, International Journal of Recent Technology and Engineering (IJRTE), May 2013, Vol. 2, Issue 2, pp. 184-185.
10. Mohammad Awrangjeb, Guojun Lu, and Clive S. Fraser, “ Performance Comparisons of Contour-Based Corner Detectors”, IEEE Transactions On Image Processing, September 2012, Vol. 21, No. 9, pp. 4167-4179.