Edge detection for colored images using Gaussian filter

Upneet Kaur

Department of Computer Science & Engg
CEC Landran Mohali, India

ABSTRACT
Edge detection is a problem of fundamental importance in image processing. It is used to enhance the edges and improve the appearance of an original image, so that further processing on the image becomes easy and efficient. In this titled "Edge Detection in Colored Images using guassian filter" the effect of this pre-filter is considered in the edge detection of colored images. This work is mainly motivated by the streaking edge problem persistent in the different edge detectors. Another problem with other edge detectors is that they are not able to suppress the noise efficiently. To suppress the noise and to enhance desired details in the image, filter is used in this paper. Gaussian is a linear filter used for smoothing and noise removal.

Keywords: Edge detection, Roberts detection, prewitt detection, guassian filter, sobel detetction

1. INTRODUCTION

Image processing is reckoned as one of the most rapidly evolving fields with growing applications in all areas of work. It forms a core area of research within computer science and engineering disciplines, but truly speaking, it is multidisciplinary and synergetic effort. Scientists from all fields have contributed to image processing and hence it find umpteen applications in all walks of life- be it scientific, commercial, artistic, medical or military. Edge detection is a problem of fundamental importance in image processing. Edges characterize object boundaries and are therefore useful for segmentation, registration and identification of objects in the scenes. Edges points can be thought of as pixel location of abrupt gray level change. A good application of edge detectors is to enhance the edges and improve the appearance of an original image, so that further processing on the image becomes easy and efficient. Since edge detection is an important part of digital image processing, much work has been done on this topic. Different techniques for edge detection have been employed in the past and the research is still going on. Here is a review of the literature that proved helpful for to understand the problem and carry out this work. Most prominent techniques used for edge detection are by first order and second order derivatives. First order derivative approach searches for the local gradient in a predefined neighborhood, while second order derivative approach goes for the zero crossing. Tamer Peli [1] studied the algorithm for edge detection. The first objective of his work was to gather edge detection methods, which do not assume any priori knowledge of the image, such as edge probabilities or knowledge from a previous set of images, and studied their performance. He examined Roberts' algorithm, Hale's operators and Rosenfeld's algorithms. A large number of edge detection schemes have been discussed in the books by Davies [2], and Pratt [3]. John Canny [4] has studied the criteria for a good edge detector. The main criteria for a good edge detector are: good detection, good localization, and single response to a single edge. In this a computational approach to edge detection is discussed. The success of the approach depends on the definition of comparative set of goals for the computation of edge points. Heath and Sarkar [5] have compared the different edge detectors. They present a paradigm based on experimental psychology and statistics, in which humans rate the output of low-level vision algorithms. Since there is no discrete parameter to measure the performance of a particular edge detector, above approach of comparing the algorithm is a benchmark in the study of edge detection algorithms. Park, Chen, and Huang [6] described a very efficient algorithm to suppress the non-edge points in a local neighborhood. Rosenfeld and Banerjee [7] discussed the problem of edge detection in noisy piece-wise-constant digital signals, using a maximum likelihood approach. M. Ruzon and C. Tomasi [8] described the techniques for edge detection in colored image. Russel C. Hardie and Charles G. Boncelet [9] described the nonlinear edge enhancing prefilters used in preprocessing of the images to enhance the desired details and to hide undesired details. S. Sarkar and K. L. Boyer [10] presented a formal optimality criteria and a complete design methodology for a family of zero crossing based, infinite impulse response (recursive) edge detection filters. D. C. Marr and E. Hildreth [11] gave the theory of edge detection based on zero-crossing approach. The books by A.K Jain [12] and by Gonzalez and Woods [13] provide a very good basic knowledge of digital image processing. These books are benchmark for a beginner in this field. Edge detection is a fundamental step for most of the computer vision application such as:

- Robotics
- Remote sensing
- Fingerprint analysis
- Optical character Recognition
- Medical Imaging

II. EDGE DETECTION

Edge detection is an important issue in image processing. Edges constitute a significant portion of the information contained in images, and thus it is useful to extract these features from a complete image. In most computer vision applications, boundary detection and image segmentation constitute a crucial initial step before performing high-level tasks such as object recognition and scene interpretation. The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries While considerable research, progress has been made in the area of image segmentation, the robustness and the generality of the algorithms on a large variety of image data have not been established. One of the difficulties arises from the fact that most natural images are rich in color and texture, and these features need to be integrated for a good integration. Further image segmentation is an ill-posed problem. For example, scale is an important and context dependent parameter. Consider an image, which contains five different “beans” regions. One might consider each bean, as an individual object and segmentation will be done on this idea, or might consider each “beans” region as a texture and get segmentation according to this different idea.

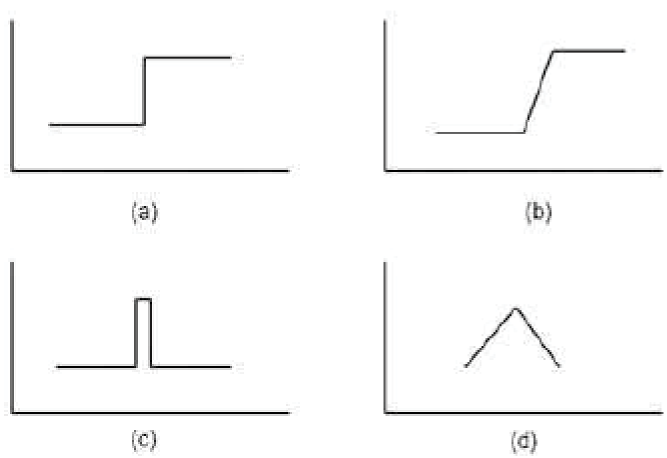

Fig.1. Type of Edges (a) Step Edge (b) Ramp Edge (c) Line Edge (d) Roof Edge

An Edge in an image is a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity. Discontinuities in the image intensity can be either Step edge, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or Line Edges, where the image intensity abruptly changes value but then returns to the starting value within some short distance [3]. However, Step and Line edges are rare in real images. Because of low frequency components or the Smoothing introduced by most sensing devices, sharp Discontinuities rarely exist in real signals. Step edges become Ramp Edges and Line Edges become Roof edges, where intensity changes are not instantaneous but occur over a finite distance.

A. Steps in Edge Detection

Edge detection contain three steps namely filtering, Enhancement and Detection. The overview of the steps in edge detection is as follows:

1) Filtering: Images are often corrupted by random variations in intensity values, called noise. Some common types of noise are salt and pepper noise, impulse noise and Gaussian noise. Salt and pepper noise contains random occurrences of both black and white intensity values. However, there is a trade-off between edge strength and noise reduction. More filtering to reduce noise results in a loss of edge strength.
2) Enhancement: In order to facilitate the detection of edges, it is essential to determine changes in intensity in neighbourhood of a point. Enhancement emphasizes pixels where there is a significant change in local intensity values and is usually performed by computing the gradient magnitude.
3) Detection: Many points in an image have a nonzero value for the gradient, and not all of these points are edges for a particular application. Therefore, some method should be used to determine which points are edge points. Frequently, thresholding provides the criterion used for detection.

B. Edge Detection Methods

Three most frequently used edge detection methods are used for comparison. These are (1) Roberts Edge Detection, (2) Sobel Edge Detection and (3) Prewitt edge detection.The details of methods as follows:
1) The Roberts Detection: The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It thus highlights regions of high spatial frequency which often correspond to edges. In its most common usage, the input to the operator is a gray-scale image, as is the output. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point.
2) The Prewitt Detection: The prewitt edge detector is an appropriate way to estimate the magnitude and orientation of an edge. Although differential gradient edge detection needs a rather time consuming calculation to estimate the orientation from the magnitudes in the x and y-directions, the compass edge detection obtains the orientation directly from the kernel with the maximum response. The prewitt operator is limited to 8 possible orientations, however experience shows that most direct orientation estimates are not much more accurate. This gradient based edge detector is estimated in the 3x3 neighbourhood for eight directions. All the eight convolution masks are calculated. One convolution mask is then selected, namely that with the largest module.
3) The Sobel Detection: The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input gray-scale image.

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

Gx

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

Gy

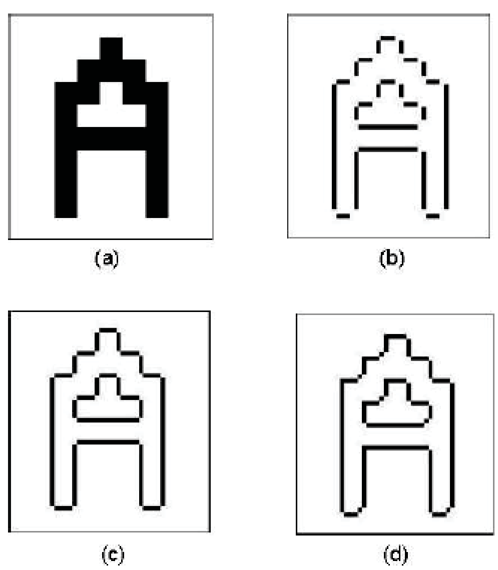

Figure2. The comparison of the edge detections for the example image.(a) Original Image (b) using Prewitt Edge Detection (c) using Roberts Edge Detection (d) using Sobel Edge Detection.

III. DESIGN AND IMPLEMENTATION

Our main goal will be to improve the results of edge detection using prefilters. So as we discuss in image enhancement, preprocessing will have to be done on the image. The image to be edge detected may be having noise, so this noise must be removed.

6.1 Algorithm: Following algorithm is used to detect the edges in a colored image:
1. Separate the three RGB components of the image.
2. Convolve the image with a Gaussian mask to smooth it.
3. Calculate the gradient (magnitude and direction).
4. Apply Non-Maximal Suppression algorithm
5. Apply thresholding and edge following algorithm
6. Combine the edges of different parts.
7. Superimpose the edges on the image.

Now each part of the algorithm is discussed in detail.

1. Separating the RGB components of an image:As we know, the image data is stored pixel wise, and at each pixel the RGB values are there. Three color components are separated and are stored in individual array, so that further processing becomes easy.

2. Smoothing with Gaussian Filter:At this step of the algorithm I implemented, the Gaussian mask to smoothen the image. 2-dimensional Gaussian is implemented by two orthogonal 1-dimensional Gaussians. Firstly 1-dimensional Gaussian is applied in x-direction and then in y-direction, thus giving the effect of a 2-dimensional Gaussian.

3. Calculation of Gradient: To calculate gradient, Prewitts operators are applied, and corresponding magnitude and direction of gradient is calculated according to the equations given below. Magnitude

$$(\nabla \mathbf{f}) = [G_x^2 + G_y^2]^{1/2} \quad \text{and } \operatorname{ang}(\nabla \mathbf{f}) = \tan^{-1}(G_y / G_x)$$

4. Non-Maximal Suppression: This is a new term that we are facing in this paper first time. This method is used in this algorithm of edge detection for removing unnecessary edges by suppressing the nonmaximal magnitude in each cross-section of the edge direction in their local neighborhood.
The following steps summarize the algorithm:
(a): Do the following steps for every edge pixel p in the image.
(b): Find two nearest neighboring pixels n1 and n2, in the edge direction at the center pixel p. Compare the actual edge direction at n1 and n2 with the edge direction at p: $|0(n1) - 0(p)| \leq \alpha$ and $|0(n2) - 0(p)| \leq \alpha$ then do step(c); otherwise goto step(a), where 0(x) is the edge direction at x and α is a predefined angular tolerance.
(c): Find two nearest neighboring pixels q1 and q2, perpendicular to the edge direction at center pixel p. Compare the edge magnitude of q1 and q2 with edge magnitude of p: If $G_m(p) < G_m(q1)$ or $G_m(p) < G_m(q2)$, then set the flag as NOEDGE at pixel p; otherwise mark it as POSSIBLE EDGE and return to step(a).

5. Thresholding and Edge Following: This is also a very important step of the main algorithm. In this combined approach of thresholding and edge tracking, edge tracking is controlled by two threshold values, that is, first upper threshold, T_H, and second lower threshold T_L ($T_L < T_H$). An edge detector does better than selecting one threshold by selecting two –an upper and a lower threshold. Following two steps summarize the procedures at this step.
Step a: Repeat the following two steps for every pixel p in the image.
Step b: If pixel p is marked as POSSIBLE EDGE and the magnitude of the pixel is greater than the upper threshold, T_H, mark it as EDGE and go to step(c), otherwise return to step (a).
Step c: Find in the 3 X 3 Neighborhood of pixel p, the pixels, which are marked as POSSIBLE EDGE and magnitude greater the lower threshold T_L. Mark them as EDGE. Do step(c) considering all these pixels as p. It means go on finding all the pixels, which are marked as POSSIBLE EDGE and having magnitude greater than T_L. Mark all of them as EDGE.

6. Combining all the Edge parts of the three parts of the image: At this step of the algorithm, the three edge maps are logically Red to give the combined edge map of a colored image.

7. Superimposing the edges and the image: At this step of the algorithm, a color is chosen to plot the edges on the image. Then edge map is scanned and wherever the edge pixel is there, that pixel in the image is plotted with the color chosen.

IV. RESULTS AND CONCLUSION

Now the results of this algorithm will be analyzed. Consider the image shown in figure 6.1. Firstly the edge detection process is applied on this image without smoothing (as pre-processing). The result is shown in figure 6.2. It was found that that result had some edges due to noise present in the image. So the smoothing is performed before edge detection process. The result is shown in figures 6.3. The values of T_L (Lower threshold) as 0.3 and T_H (Upper threshold) as 0.85 and the sigma value (The control parameter for the width of the Gaussian mask) as 1 are taken. It was deduced from the result that this image has good edge detection and the noise present in the image has no effect in the result. Thus this Mask (Gaussian mask) nullifies the noise.

Figure 6.1

Figure 6.2

Figure 6.3

V. FUTURE SCOPE OF WORK

During this work the effect of guassian filter is studied. With the use of this filter edge detection is improved. But till the results are not up to mark. There is still some possibility of improvement. Some other filters may be considered for the improvement of results without changing the whole algorithm. Other edge linking algorithms may be considered. So there is scope of further improvement in the results.

VI. REFERENCE

[1]Tamer Pali, Davis Malah, "A study of edge detection algorithms", *Computer Graphics and Image Understanding, 20, 1- 21,1982*

[2]E. Davies, "Machine Vision: Theory, algorithms and Practicalities", Academic Press.

[3]W. K. Pratt, " Digital Image Processing", Wiley, New York, 1977.

[4]J. Canny, "A computational approach to edge detection", *IEEE Transactions, Pattern Analysis and Machine Intelligence.* PAMI-**8**, 1986, 679-698.

[5]M. Heath, S. Sarkar, "Comparison of edge detectors- A methodology and initial study", *Computer Vision and Image Understanding.* Vol. 69, No. 1, January, pp 38-54, 1998.

[6]Park, Chen, Huang, "A new gray edge thinning method", *Dept of Computer Sc. And Engineering, The University of Alabama, Tuscaloosa.*

[7]Rosenfeld, S. Banerjee, "Maximum-Likelihood edge detection in digital signals", *CVGIP: Image Understanding. Vol. 55, No.1 January, pp. 1-13, 1992.*

[8]M. Ruzon, C. Tomasi, "Color edge detection with the compass operator", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Ft. Collins, CO, Vol. 2, pp. 160-166, June, 1999.*

[9]Russel C. Hardie, Charles G. Boncelet, " Gradient-Based edge detection Using nonlinear edge enhancing prefilters", *IEEE Transactions on Image Processing, Vol. 4, No. 11, November 1995.*

[10]S. Sarkar, K. L. Boyer, "Optimal infinite impulse response zero crossing based edge detectors", *CVGIP: Image Understanding. Vol. 54, No. 2, September, pp. 224-243, 1991.*

[11D. C. Marr, E. Hildreth, " Theory of edge detection", *Proceedings Royal Society Landon, Vol. B 207, pp. 187-217, 1980.*

[12]Anil K. Jain, " Fundamentals of Digital Image Processing", *PHI Pvt. Ltd*

[13]Rafeal C. Gonzalez and Richard E. Woods, " Digital Image Processing", Pearson Education Asia