Object Detection using the Canny Edge Detector

Mathew George[1], C. Lakshmi[2]

[1]SRM University, Department of Computer Science and Engineering, Chennai, India

[2]SRM University, Department of Software Engineering, Chennai, India

Abstract: ***Object detection is one of the applications in computer vision where an image is searched for the presence of an object. An object can be described by its features like its shape or its texture. The shape of a 2-D object is made up of its edges. Many algorithms are currently in use in image processing for detecting the edges in an image. This edge information extracted from the image can be used for object detection. This paper presents an approach for object detection using the Canny edge detector. The approach proceeds by obtaining the edges from the query image which is the image which must be tested for the occurrence of the object and the image of the object to be detected and then employing a template matching procedure.***

Keywords: canny, edge detection, template matching, object detection, computer vision.

1. Introduction

Object detection is an area in computer vision which enables the system to search an image for the occurrence of an object. Object detection can be used in applications such as pedestrian detection, surveillance and industrial inspection to name a few. Object detection involves taking the image of the object and then extracting its features. When given the query image, the feature extraction method is employed on the query image and then a search is carried out for the features of the object that is to be detected, in the transformed query image. If a suitable match is found, then it can be said that the object is present in the query image.

Edges are features that are common to objects. Many edge detection algorithms have been proposed in literature like the Roberts cross-gradient operators (Roberts [1965]), Prewitt operators (Prewitt [1970]), Sobel operators (Sobel [1970]) [1] and the Canny edge detector [2]. Studies have been performed on the performance of these edge detectors and based on them [3], [4]; the canny edge detector is used in the implementation of this paper.

Once the images with edge features are obtained, a template matching procedure is carried out. The auto-correlation of the template image is first calculated. For template matching, the correlation response between the processed query and template images is calculated and the point in the resulting matrix with correlation value above a certain percentage of the maximum of the auto-correlation is understood as the location of the object.

2. Related Work

The main elements in this paper are the use of the edge detector for the extraction of edge features and then the use of correlation for template matching. A comprehensive explanation of correlation for 2-D images is found in [1]. A comparison of various edge detectors like the Roberts cross-gradient operators, Prewitt operators, Sobel operators, the Marr-Hildreth detector and the Canny edge detector are given in [3],[4] and [5] and a statistical study on their performance is given in [6]. According to those studies, the canny edge detector is chosen as the approach for edge detection in this paper because of its ability to produce single pixel thick, continuous edges, ability to detect strong and weak edges and its insusceptibility to noise interference.

A demonstrative application for template matching is found in [1]. An application for automated crater detection on planetary images where edge features are combined with template matching is given in [7].

3. Notes on Techniques Used

3.1. The Canny Edge Detector

The Canny edge detector was introduced by John Canny in [2]. The performance criteria that the canny detector aims at are as follows:-

1. Good detection: There should be a low probability of failing to mark real edge points, and low probability of falsely marking non-edge points.
2. Good localization. The points marked as edge points by the operator should be as close as possible to the center of the true edge.
3. Only one response to a single edge.

As an overview, the Canny edge detection algorithm consists of the following basic steps as in [1]:

1. Smooth the input image with a Gaussian filter
2. Compute the gradient magnitude and angle images.
3. Apply non-maxima suppression to the gradient magnitude image.
4. Use double thresholding and connectivity analysis to detect and link edges.

Figure 1 shows the application of canny detector on an image.

Figure 1: The cameraman image and its edges extracted using the canny edge detection method.

3.2. Cross Correlation and Template Matching

A detailed study of cross correlation and template matching using cross correlation is provided in [1]. It describes correlation as the process of moving a filter mask over the image and computing the sum of products at each location. The maximum value of the correlation coefficient occurs when the template and the region in the input image are identical. This indicates the maximum correlation or the best possible match. The minimum occurs when the template and the region in the input image are least similar.

4. Implementation

Implementation of the paper is done using MATLAB 7.0.1. Images containing the objects to be detected were first obtained. They can be colour or gray scale images. Images where the object is partially occluded can also be taken. The image may contain a single instance or multiple instances of the object. As the first step, all images are converted to gray scale since the implementation of Canny edge detection algorithm in MATLAB 7.0.1 works only for gray scale images. But this is not a disadvantage since the approach is concerned with only the edges in the image. This conversion to gray scale is the pre-processing work done.

The edges in both the query image and the template image are now found using the canny edge detection function. The correlation between these two images is next found. The auto-correlation of the template image is found, i.e. the correlation of the template image with itself. The highest value found for the auto-correlation is now taken. This highest value is searched for in the correlation values obtained by the first operation, i.e. between the query image and the template image. If the value is found, then it is concluded that the object is present in the image.

Even though a search for the highest coefficient of the auto-correlation in the correlation values obtained in the operation between the query image and the template image is the theoretical way to detect the presence of the object, practically this is not the case. Due to variations in properties like brightness, colour and slight variations in the shape of the object in the query image compared to the template image, it is desirable to set an offset whereby every point (x,y) having correlation coefficient that is within this offset of the highest possible value is also detected as an object. The offset is set as a percentage of the highest value rather than as a real number so that the algorithm can run for any template.

The next step is to draw bounding boxes around the detected object. The size of the bounding box is the same as that of the size of the template.

Figure 2 and Figure 3 show the screen-shots of the implementation.

Figure 2: The screen shot shows that the method was able to correctly identify the objects in the image.

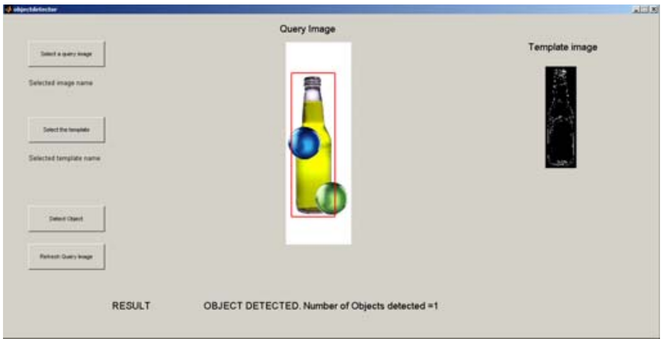

Figure 3: Screen shot shows that the method works even when the object is occluded.

5. Conclusion and Future Work

The proposed approach works well in detecting objects in images. One advantage of template matching using edge images is that the color and to an extent the texture of the object can vary as long as the shape remains the same. The method works fine even when there are partial occlusions.

Therefore the approach permits variations in color and texture and the chances of missing out the object due to shading or lighting effects are decreased. The approach can further be modified such that the algorithm automatically scales the template image to various sizes and searches for its occurrence in the query image. This will allow the detection of the object even if it appears in different sizes. A further modification can be brought about by making the detection rotation invariant. The template has to be rotated 360 degrees in steps and template matching can be done. The proposed approach can be used where given an image; it is needed to count the number of object instances and in industrial scenarios where a product can be inspected to check whether it conforms to the standard shape.

References

[1] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 3rd ed., Pearson Prentice Hall, India, 2009.

[2] John Canny, "A Computational Approach to Edge Detection," In IEEE transactions on pattern analysis and machine intelligence, vol. pami-8, no. 6, 1986.

[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, Vol. 9, Issue 5, No 1, 2012.

[4] Ehsan Nadernejad, Sara Sharifzadeh, Hamid Hassanpour, "Edge Detection Techniques: Evaluations and Comparisons," Applied Mathematical Sciences, Vol. 2, no. 31, p. 1507-1520, 2008.

[5] Li Bin, Mehdi Samiei yeganeh, "Comparison for Image Edge Detection Algorithms," IOSR Journal of Computer Engineering (IOSRJCE), ISSN: 2278-0661, Vol. 2, Issue 6, 2012

[6] Mamta Juneja, Parvinder Singh Sandhu, "Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain," International Journal of Computer Theory and Engineering, Vol. 1, No. 5, 1793-8201, 2009.

[7] Lourenço P. C. Bandeira, José Saraiva, Pedro Pina, "Development of a Methodology for Automated Crater Detection on Planetary Images," in Pattern Recognition and Image Analysis, Third Iberian Conference, IbPRIA, Girona, Spain, 2007.

Author Profile

Mathew George received B-Tech degree in Information Technology from Mahatma Gandhi University, Kerala in 2010. He is currently pursuing M-Tech in Computer Science and Engineering from SRM University, Chennai.

Dr. C. Lakshmi is currently Professor and Head of the department of Software Engineering in SRM University, Chennai. She received her Ph.D from SRM University in 2010. Her research interests are image processing, pattern recognition, software engineering and fuzzy logic.