Contour Tracing using Harris Corner Detection

1st Shreedeep Kottakota
Department of Electronics and Communication Engineering
Visvesvaraya National Institute of Technology
Nagpur, India
shreedeepkottakota@students.vnit.ac.in

2nd Apurva Rajendra Umredkar
Department of Electronics and Communication Engineering
Visvesvaraya National Institute of Technology
Nagpur, India
apoorv.umredkar@students.vnit.ac.in

3rd Joshin Cheriyan Rexy
Department of Electronics and Communication Engineering
Visvesvaraya National Institute of Technology
Nagpur, India
joshinrexy22@students.vnit.ac.in

***Abstract*—Contours are lines that connect all contiguous points which possess the same intensity. They play a vital role in object detection and analysis of shapes. The goal of this project is to find the most salient object in an image, implement our own proposed Harris Corner algorithm to trace the contour of the salient object using these corner points.**

***Index Terms*—Object Detection, Contour Tracing, Harris Corner, Image Processing**

I. Introduction

Since the evolution of Computer Vision, visual representations have impacted object detection techniques drastically because the border contains a lot of valuable details. It all comes down to whether the contour can be efficiently or almost accurately traced as it helps determine and analyze the behavior of the target object [1].

A digital image is made up of a set of pixels each with a unique value. The set of border pixels of a given pattern P is the pattern's contour. Contour tracing is among one of the preprocessing methods used to extract details about the general framework of digital images. If the contour of a given pattern has been extracted, the pattern's various characteristics can be studied and used as features of pattern classification. As a result, accurate contour extraction would yield more accurate characteristics, increasing the likelihood of precisely identifying a particular sequence [2].

The contour pixels make up a tiny percentage of the overall amount of pixels that constitute a pattern. As a result, when we run attribute extraction algorithms on the contour rather than the whole pattern, the amount of computation is significantly reduced. The feature extraction procedure becomes much more effective when done on the contour rather than the original pattern since the contour shares many of the same characteristics as the original pattern. In a nutshell, contour tracing is often a significant contributor to the reliability of the feature extraction process, which is a critical step in pattern recognition. Since tracing and removing contour pixels is easy and useful for object detection, it is commonly used in smart and wearable image sensor systems [2].

Some of the few Contour Tracing Algorithms are:

Fig. 1: Basic Contour of an Image

- Square Tracing Algorithm
- Moore-Neighbor Tracing

II. Methodology

We will be utilizing two primary libraries to implement our contour tracing algorithm, namely OpenCV and Numpy. To achieve the desired goals we will have to surpass three stages:

1) Finding the salient object in the image
2) Picking the Harris corner points for the salient object
3) Tracing contour of the salient objects using these points

A. Finding the salient object in the image

Initially, we use filter2D method to smoothen the image as we have to strike a balance between retaining the edges of an image and filtering noise. Hence we will discard salt and pepper noise only using this method.

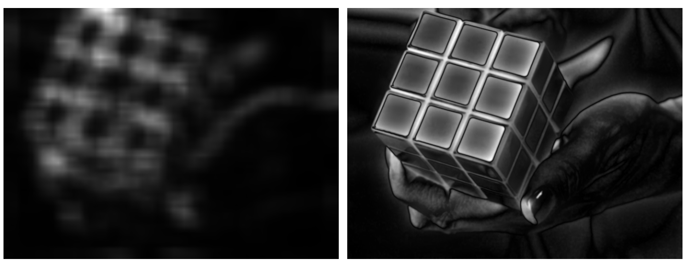
In the next step, we tend to implement a static saliency detection method to find the most prominent object in the image. We have two already implemented algorithms in OpenCV, namely Spectral Residual and Fine Grained. Fine Grained is preferred simply for the fact that the image is of better quality and more distinct than Spectral Residual. We have also provided a comparison of both methods.

Moving ahead, we have applied a Gaussian blur to remove random noise and adaptive thresholding to distinguish foreground pixels from the background. We use adaptive thresholding instead of the customary method since the threshold value is determined for smaller regions and therefore varying threshold values exist for different regions.

B. Picking the Harris corner points for the salient object

In this section, we follow 5 basic steps to find the Harris Corner Points. The algorithm is described below:

1) Compute the gradient of the image over a small region.
$$I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}$$
2) Subtract mean from each image gradient.
3) Calculate the covariance matrix.
$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$
4) Compute the eigenvectors and eigenvalues.
$$Me = \lambda e$$
$$(M - \lambda I) = e$$
5) Use thresholding on eigenvalues to detect the Harris corner points [3].



(a) Spectral Residual Saliency Detection (b) Fine Grained Saliency Detection

Fig. 2: Saliency Detection

C. Tracing contour of the salient objects using these points.

Coming onto the ultimate section we follow a series of steps again. They are described as follows:

1) Adjusting the Harris points by dilation, erosion, and closing morphological transform
We first implement the idea of dilating, eroding, and closing morphological transforms to adjust the Harris Corner points obtained in the red region to make it more prominently visible.
2) Extracting the boundary contours.
Here a function is deployed that will verify if a pixel is a boundary pixel or not. The N8-Neighbourhood points are examined to see if there is any existence of a black pixel. An occurrence of the same indicates that the pixel is present at the edge and it is a boundary pixel.
3) Mapping the contours onto the input image.
Initializing an empty contour map to map the points and using these contours to map over the input image with a little bit of dilation thus enhancing the contour tracing.

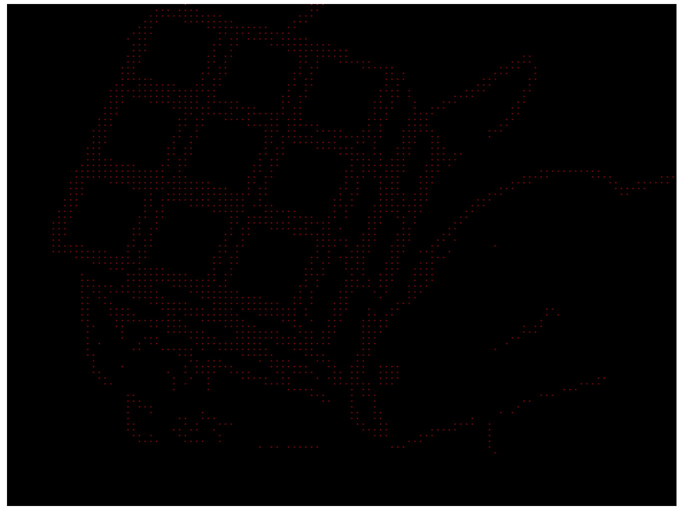


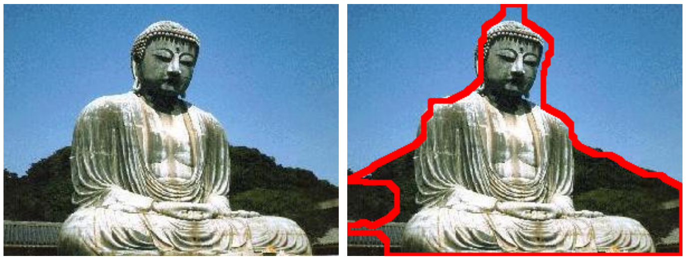
Fig. 3: Harris Points for the input image

III. Results

All the results are produced and obtained by ourselves to the very best knowledge. The inbuilt function was used only for the salience detection stage. Determining the Harris corner points and tracing contours were implemented by our algorithms. The result sections contain 4 test case contours on the right in comparison to the original input image on the left.

(a) Input Image (b) Output Image

Fig. 4: Test Case 1



(a) Input Image (b) Output Image

Fig. 5: Test Case 2

(a) Input Image (b) Output Image

Fig. 6: Test Case 3

(a) Input Image (b) Output Image

Fig. 7: Test Case 4

IV. Discussions

This project came out to be successful after few obstacles we faced. The most challenging part we encountered was in the last stage where we had to develop an algorithm to trace the contour of the image without using inbuilt functions. We initially tried to study the Square Tracing algorithm but it did not give an efficient performance and some of the test cases had failed. We then tried to implement the Radial Sweep Algorithm. The technique was quite easy to comprehend but for some reason, the contours went to revert back the same path after hitting some edges of the salient object. Lastly with the help of boundary conditions we were able to plot the boundaries of the dilated image and draw its contours. We then compared the obtained outputs from our algorithm with the outputs obtained using inbuilt contour functions and the result came out to be quite satisfactory.

V. Conclusions

The applications of object detection are vivid in the field of computer vision. In terms of engineering technology, new projects like buildings, roads, canals, etc. perform a contour survey to select the most economical and appropriate location. Certainly, this sector has a huge scope for even more progress as today's rapid advancement in Artificial Intelligence or Deep Learning with the combination of Image Processing can be very deadly.

References

[1] J. Shi and C. Guo, "The Extraction of Circle Contour Based on Improved Boundary Tracing Algorithm," 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, Nanchang, China, 2012, pp. 104-107, doi: 10.1109/IHMSC.2012.121. IEEE

[2] Contour Tracing. (2000). Contour Tracing. http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour

[3] Kris Kitani, "Harris Corners", 16-385 Computer Vision, Carnegie Mellon University