Two Dimensional Polygon Identification Tool

Mr U.Iynkaran[1] , Mr T.Mathilakan[2], Dr.S.Kanaganathan[3]
Vavuniya Campus of the University of Jaffna.
iynkaran86@gmail.com

Introduction

Image processing is a rapidly growing area of computer science. Currently several applications available in all fields and widely use in the medical field such as analyzing the X-Ray reports and use in computer-aided diagnosis.

Here we developed a software tool for the essential need of finds polygons in the given image. This polygon identification tool is mostly used for academic purposes. In this tool we implement basic image processing techniques and some specific edge detection algorithms. Here find the polygons by analyzing the edges throughout the connectivity of adjacent pixels.

> ñIn geometry a polygon is traditionally a plane figure that is bounded by a closed path or circuit, composed of a finite sequence of straight line segmentsò

Tools & Development Environment

This polygon identification tool is developed using Microsoft Visual Studio Dot Net Framework 3.5 (C#). Microsoft Visual Studio 2008 is an Integrated Development Environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications.

Methodology

- ➢Initially import the image to tool from any storage device.(eg jpg, bitmap)
- ➢In this Tool there are techniques available to change the attributes if the user wants then can do the operations
- ➢There are some techniques available to reduce the noise, it also can be applicable. (eg filters).
- ➢RGB image then convert into grayscale image. It makes easier to manipulate the pixels.
- ➢Tool contain several edge detection techniques user can apply those different methods until his/her satisfaction of the edge detections. While same time image can be saving in sequence manner with different edges.
- ➢The polygons are identified through their adjacency and the connectivity between pixels. Here we eliminate curves and the non closed objects from the image.

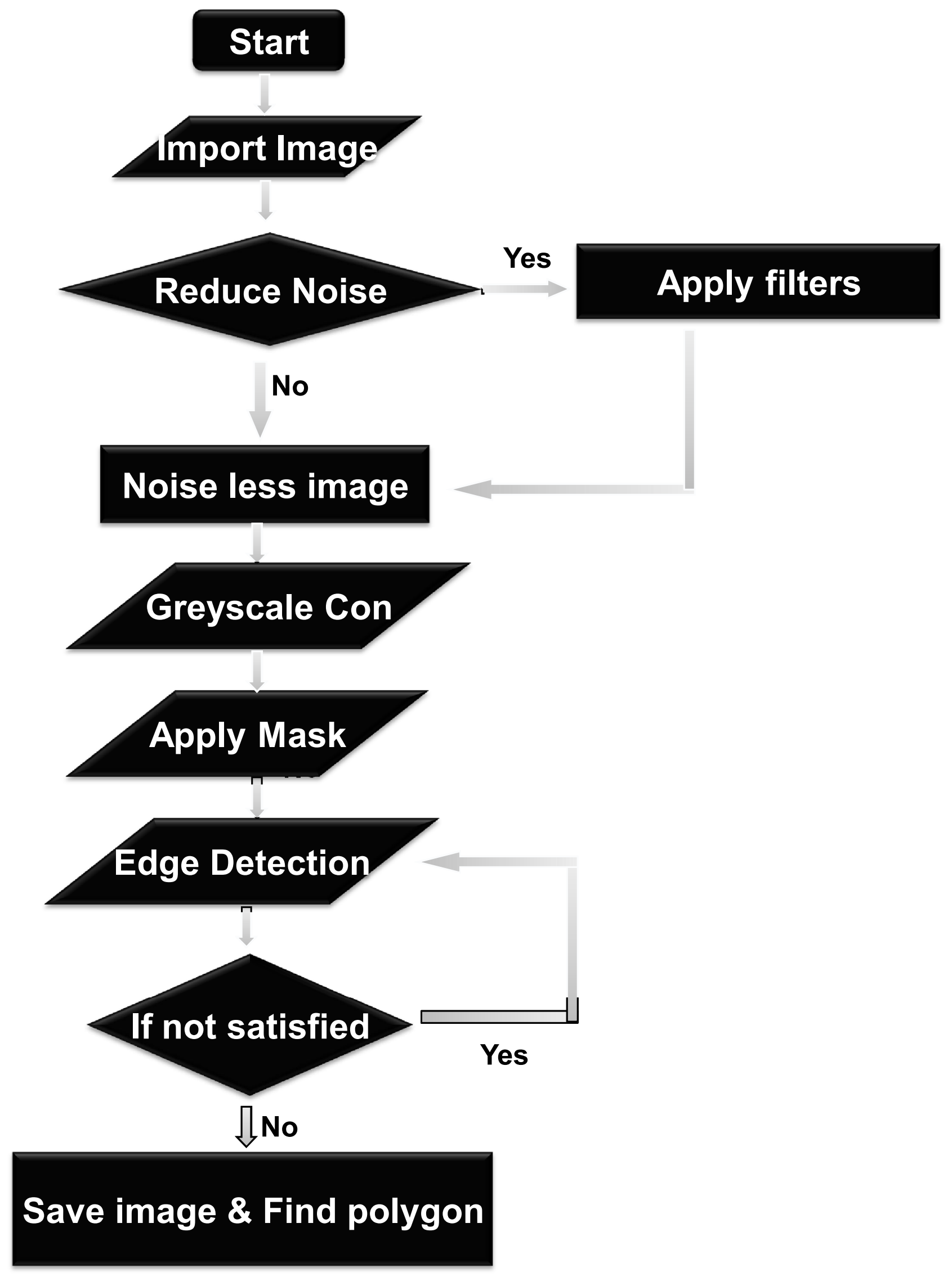

Figure 1: Flow diagram of the proposed method.

Input Image

The pixel format should be in 24-bit it means each pixel contain 8-bit for the primary colors RGB or it may be grayscale image (two color image). The image format may be in bitmap (.bmp), jpeg, or jpg. The other types of images such as GIF, TIFF, PNG or animated images are not suitable for this image processing tool..

Convert to Greyscale

In fact a ógrayôcolor is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a image.

p[0] = p[1] = p[2] = (byte)(.299 * red + .587 * green + .114 * blue)

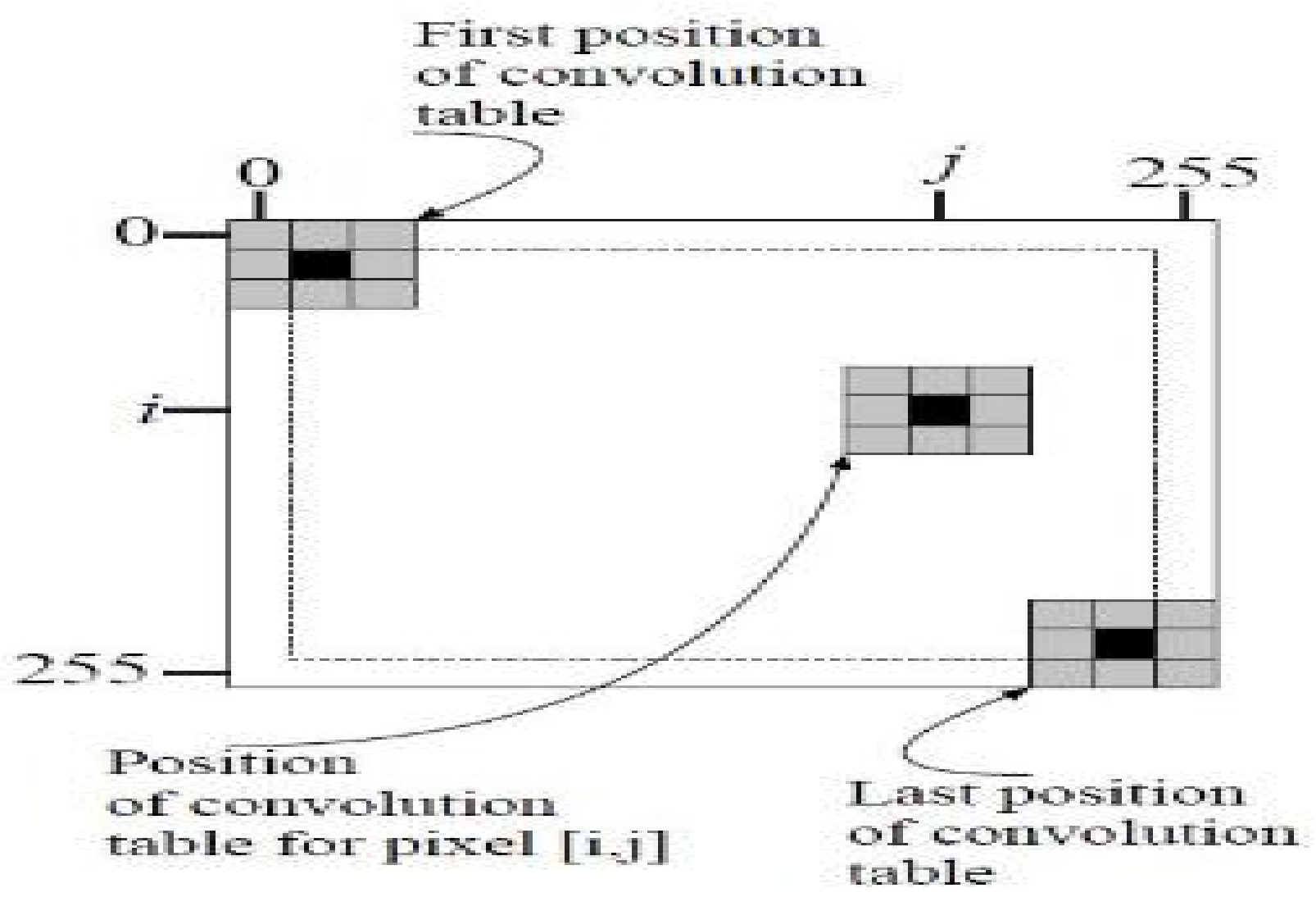
First position of convolution table
0
j
255
0
i
255
Position of convolution table for pixel [i,j]
Last position of convolution table

Figure 2: Mask Applying sequence.

Apply Mask to Image

In computer graphics, when a given image is intended to be placed over a background, the transparent areas can be specified through a binary mask. This way, for each intended image there are actually two bitmaps: the actual image, in which the unused areas are given a pixel value with all bits set to 0ôs, and an additional mask, in which the correspondent image areas are given a pixel value of all bits set to 0ôs and the surrounding areas a value of all bits set to 1ôs.

Edge Detection Techniques

Sobel Edge Detection

The Sobel Edge Detection Operator is a 3x3 Spatial mask. It is based on the differential operation [1 0 -1] and an averaging operator [1 2 1], Convolving these operators we get the 3x3 spatial mask for Sobel:

$$\begin{aligned}\mathbf{Gx[i, j] = Im[i+1, j-1] + 2 * Im[i+1, j] + Im(i+1, j+1)}\ &- \\ \mathbf{Im[i-1, j-1] + 2 * Im[i-1, j] + Im(i-1, j+1)}\end{aligned}$$

$$\begin{aligned}\mathbf{Gy[i, j] = Im[i-1, j+1] + 2 * Im[i, j+1] + Im(i+1, j+1)}\ &- \\ \mathbf{Im[i-1, j-1] + 2 * Im[i, j-1] + Im(i+1, j-1)}\end{aligned}$$

Prewitt edge detection

Prewitt edge detection produces an image where higher grey-level values indicate the presence of an edge between two objects. The Prewitt Edge Detection filter computes the root mean square of two 3X3 templates.

$$\mathbf{X = -1 * a1 + 1 * a3 - 1 * a4 + 1 * a6 - 1 * a7 + 1 * a9}$$

$$\mathbf{Y = 1 * a1 + 1 * a2 + 1 * a3 - 1 * a7 - 1 * a8 - 1 * a9}$$

Prewitt gradient = ã$\mathbf{X^2 + Y^2}$

Polygon Detection

In polygon detection we developed an algorithm to find adjacency of the each and every edge pixel through 8-connected method then find the closed polygons and remove the non closed objects & also the curves in the image and redraw the find polygons in a blank bitmap file.

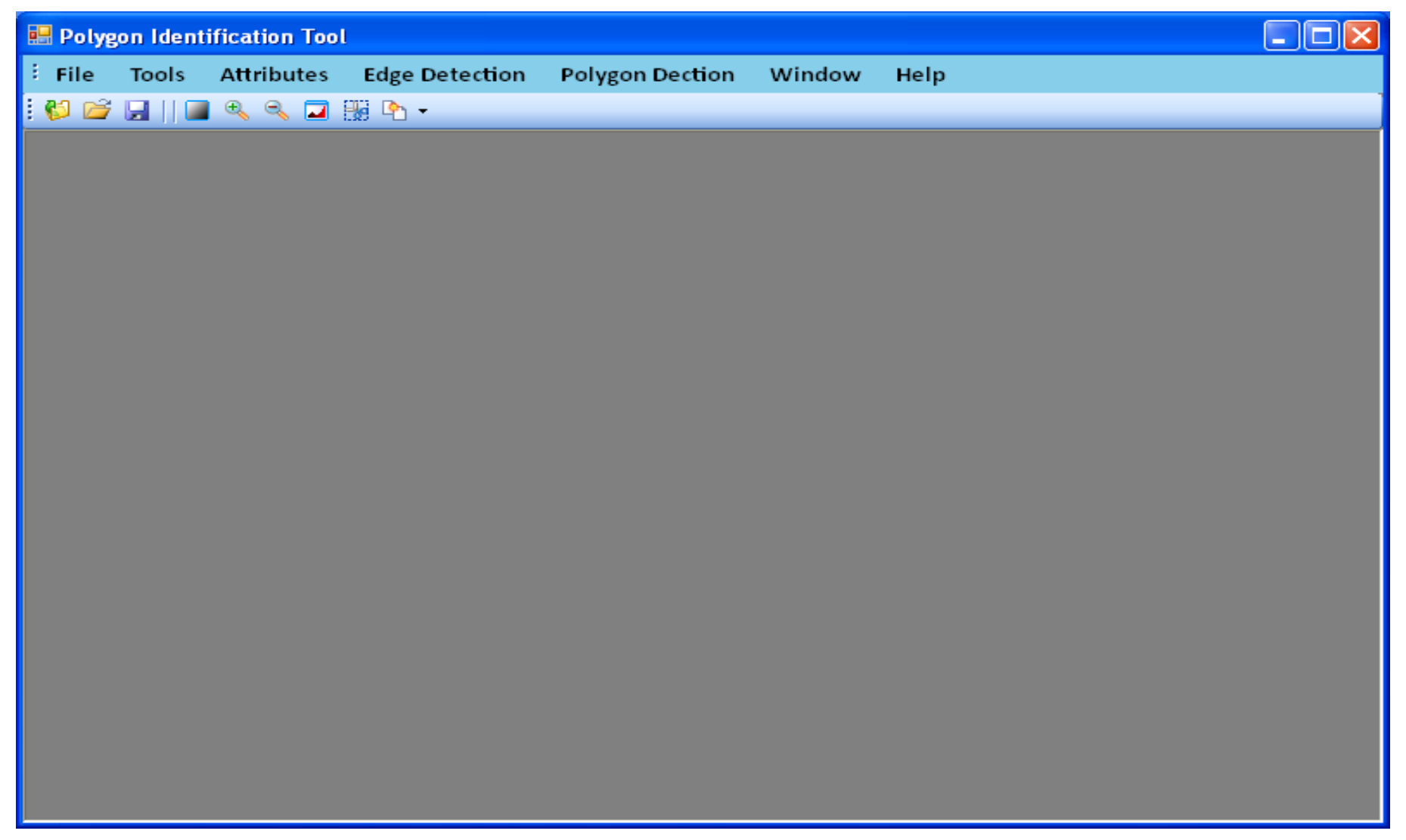

Figure 3: shows main window of the tool

Results

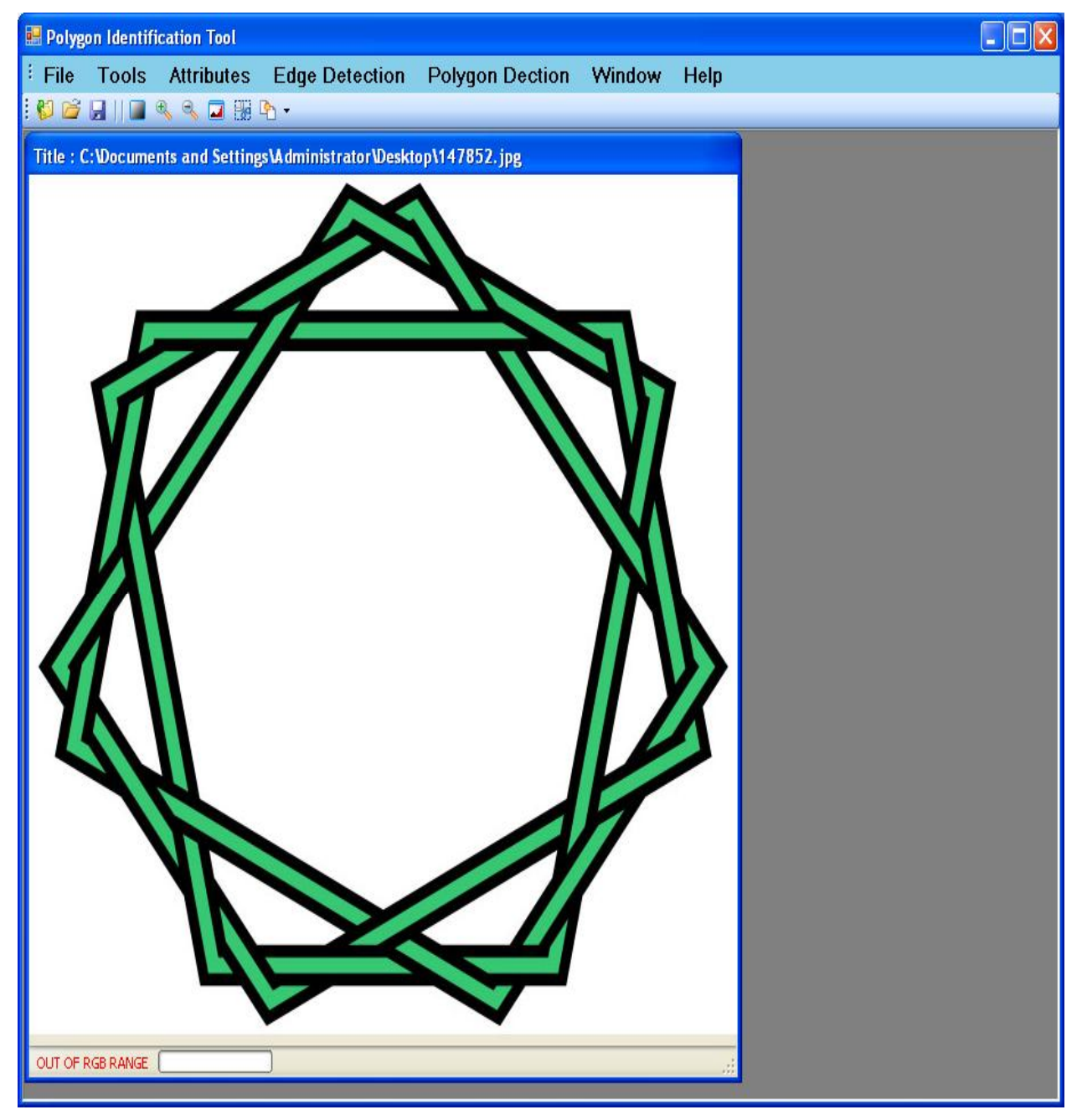

Figure 4: Shows the initial image.(here we use well defined image for easy understanding)

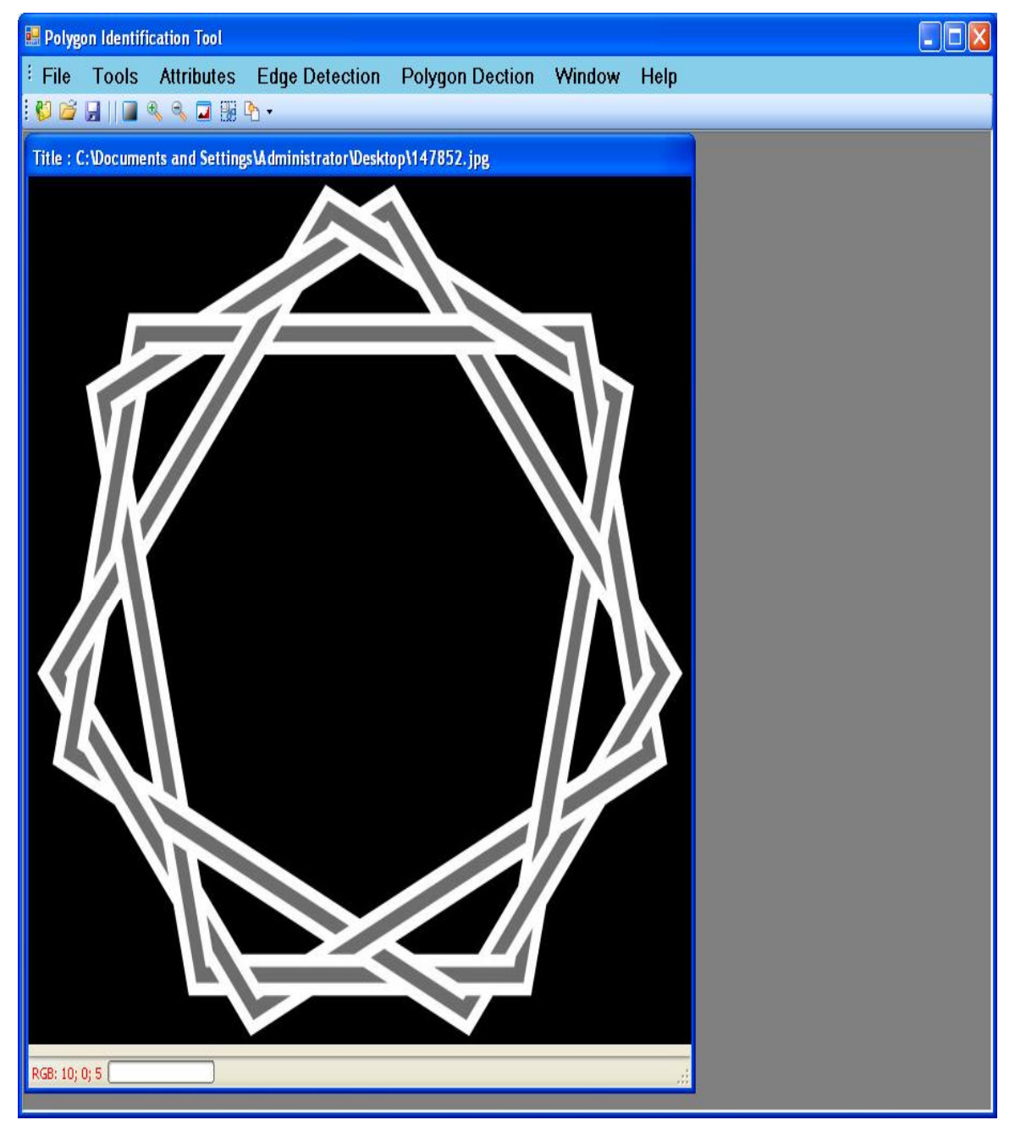

Figure 5 : After applying the Filters & Grayscale and edge detection algorithms

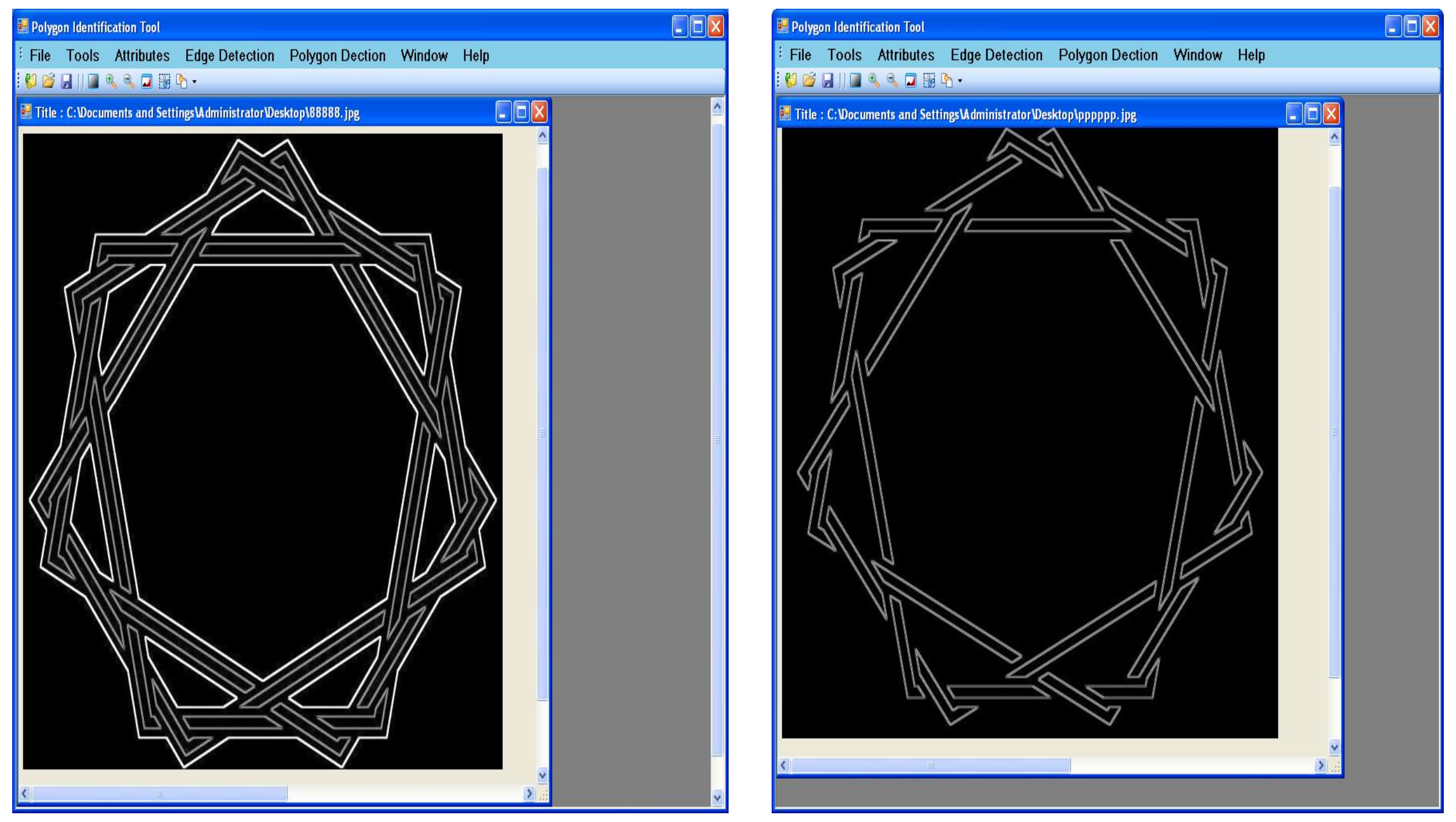

Figure 5(a),(b):The Above two windows shows different polygon detections. In the second window we remove white edges in the image.

Discussion & Conclusion

☞ The effective detection of the polygon is depending on the quality or noise less of the image pixels.

☞If the user finds edges accurately then only he can identified the whole polygons in the image.

☞ Here while eliminating the Curves according to the precision sometimes it may causes to miss the connectivity of the edges.

☞The user must have some knowledge about image processing to handle this tool.

Reference

[1]. Al Bovick, The Essential Guide To Image processing.
[2]. Torsten Seemann, Journal Faculty of Information Technology Monash University Australia
[3]. Rafael C.Gonzalez , Richard E.Woods Digital Image Processing Second Edition.