Searching for Simple Geometric Shapes in Raster Image

Jiří Venera[*]
Department of Computer Graphics and Multimedia
Brno University of Technology
Czech Republic

Abstract

Looking for a defined geometrical shape, we decided to describe it as a composition of mathematically defined elements. This paper deals with searching for such simplistic elements in a raster image, as a means of robust finding of a defined complex shape.

Keywords: image processing, Hough transformation, Sobel filter, Prewitt filter, edge detection, line detection, circle detection

1 Introduction

The ultimate goal of our work is to develop a tool usable as a potential part of complex image processing and computer vision systems that would detect predefined 2D shapes in grayscale or color raster images. This includes definition of the geometric shape suitable for this purpose.

As the first step, this paper presents effort to detect basic, mathematically defined objects such as lines, circles and ellipses. The final predefined shape would be composed of such elements.

2 Background

The following chapters consider as input a scalar intensity raster, such as a 256-level grayscale image. This kind of input can be acquired either directly from a grayscale camera or by converting the RGB or other image format to grey-scale image by some standard method [2]. Used samples were generated or drawn directly as black-and-white or grayscale images.

Such input images are first pre-processed by edge-detection filters. This class of filters was chosen for its simplicity and potential low consumption of computational resources. There are more complex edge-detection methods (e.g. Roberts cross, Sobel, Canny, Kirsch Mask, Prewitt Mask Edge Detector) [1,4], their kernels are convolution kernels - linear filters. All operators react on transition between two regions of significantly different intensities.

Roberts cross operator is the simplest edge detector, one of the main problems with this operator is sensitivity to noise.

Sobel and Prewitt operators are in fact the same filter with different weight values, these operators are more resistant to noise.

The line and circle detection uses the Hough transformation, which transforms the pre-processed image to a different coordinate space to simplify the search process.

3 Edge Detection in the Source Image

As the input of the edge-detection we require grayscale image, potentially preprocessed and promising the possibility of shape existence.

The purpose of this step is to detect all edges in input image (pixels which are on the boundary between two regions of different intensity). We can use variety of gradient filters or edge masks.

All pictures in this case are negative for printing purposes.

3.1 Sobel Edge Detector

Let us first mention about Sobel edge detector (operator). The Sobel operator performs a 2-D spatial gradient. The operator consists of several 3 x 3 convolution masks (kernels). Other masks we obtain a simply rotation by 45°. For example, the convolution kernel that estimate the partial derivation in the x – direction:

$$S_{0^\circ} = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

All kernels can be combined together to find the magnitude of the gradient.

The gradient may then be estimated for masks in direction x and y as:

RMS

$$S = \sqrt{S_x^2 + S_y^2} \ ,$$

Sum absolute value

$$S = |S_x| + |S_y|,$$

or max absolute value of each mask

$$S = \max\left(|S_x| + |S_y|\right).$$

[*] xvener00@stud.fit.vutbr.cz

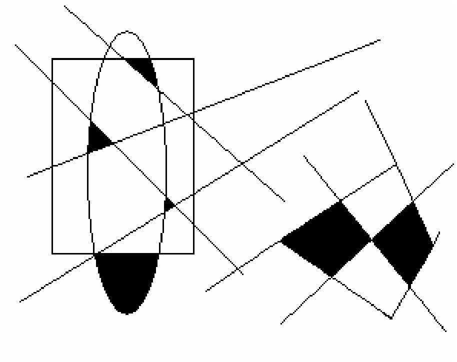

Figure 1. Input image

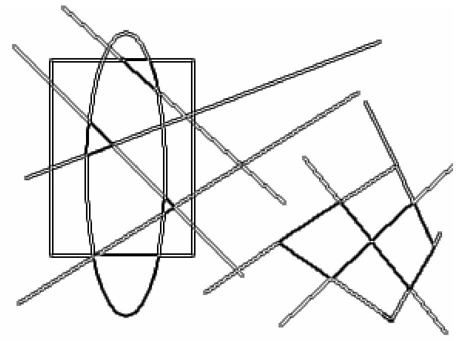

Figure 2. Sobel edge detection operator, evaluated by max absolute values for all direction

If we use a mask e.g. in direction x and mask rotated by 180° (in direction –x, this is only negative response mask in direction x), we get for single line in input image two positive responses (either of the two masks), first for left brink of line and second for right brink. There is a distance between these responses, the final effect are two lines (in Sobel image), which gives the boundary of the line in the input image, but not the original line. Another problem is a passage between two areas of different intensity. If we use mask in direction x, each boundary has two responses, first is edge of split line and the second line representing a boundary between the areas. The final results are two lines next to each other.

It is possible to partially eliminate this effect (supposing the original line or edge plane has high intensity and locality): If the response is positive (in direction x), the original line is one pixel on the right side and then we write this result on position on right side. The mask rotated by 180° will have positive response on other side of edge line (this is two pixels on right), but not for area with similar intensity (the second response is in reaction to a point which belongs to original line and in the output image we have recorded a detection of this passage by previous algorithm). We turn back one pixel left and we can compare this value with value on this position in Sobel image and write higher of these two values or rewrite with new value. In analogous manner we can proceed in other direction. We get an image with the original lines.

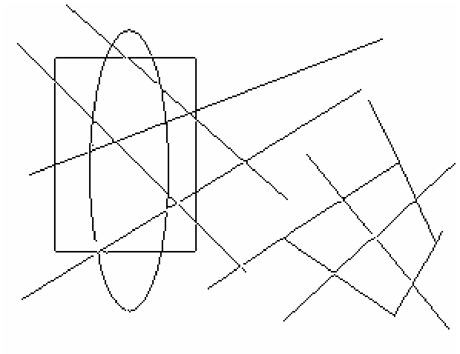

Figure 3. Modified Sobel filtered image

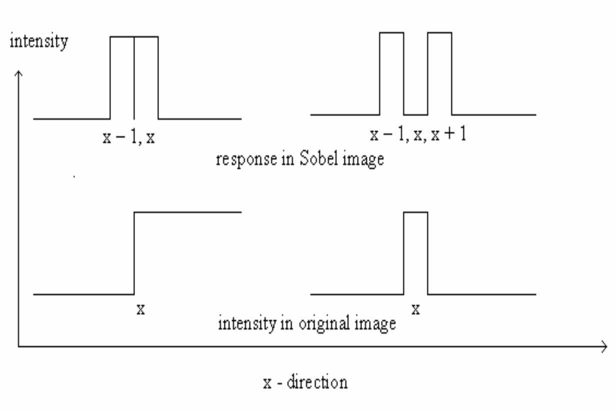

Figure 4. Sobel edge detection operator, reconstruction of original line objects (in contrast to finding the outlines)

3.2 Kirsch edge mask (template matching)

The template producing the highest correlation determines the edge magnitude. Each of the masks represents an ideal step in the corresponding orientation. The great response may be estimated as max of absolute values in each direction (or use other techniques).

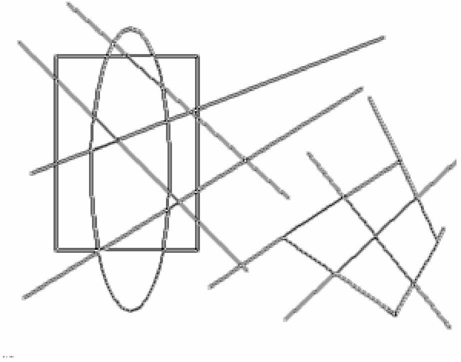

Figure 5. Kirsch edge mask evaluated by max of absolute values.

3.3 Adjusting filtered image for subsequent processing

We can reduce the number of not important pixels in filtered image for next acceleration compute of Hough transform.

As first step we divide image on region 16x16 pixels. For each region we create histogram. If the highest value arbitrary region is less than the highest value of whole image, we resample this histogram (the highest value of arbitrary region is mapped on the highest value of whole image). Then we use threshold for removing redundant pixels (with small intensity).

The second step is to remove single pixels, which do not belong to edges in image and we thin the edges of objects in the image. The result is an image in binary representation. See Figure 6.

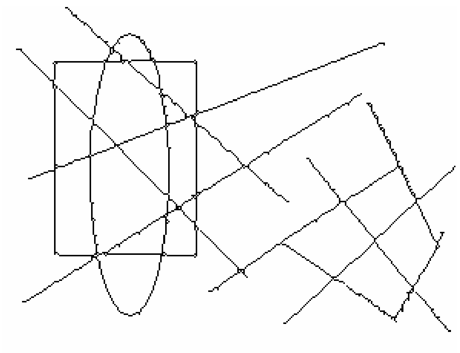

Figure 6. Image obtained by described technique in above text from Figure 5.

4 Detecting Lines, Circles and Ellipses by the Hough Transform

To detect mathematically defined objects in the image preprocessed by the means described in section 3, we use the Hough transform [1, 3 and 4].

4.1 Hough transform for line detection

The analytical equation of a straight line is given:

$$x \cdot \cos(\varphi) + y \cdot \sin(\varphi) = r\,,$$

where r is the length of a normal to the line from point (0,0) in the image and φ is the angle of normal line [1, 3, 4]. If we plot values r and φ, for some point in input image (x,y), into output image (Hough space), we get a sinusoidal curve in Hough space (r, φ space), where r and φ are variables. The Hough transform for input image is shown on Figure 7, where $\varphi \in \langle 0°, 180° \rangle$ and $r \in \langle -a, a \rangle$ where a is length of the diagonal of the input image. The φ is sampled by $1°$ and r is sampled by 1 pixel.

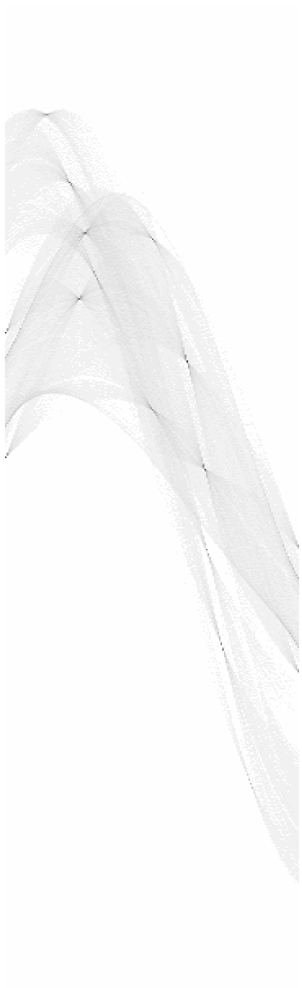

Figure 7. Hough transform – Hough's space of the input image on Figure 1, (x-axis representing φ and y-axis representing r).

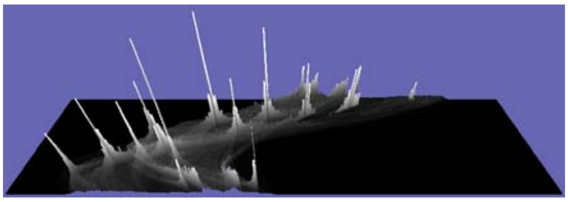

Figure 8. Hough transform – Hough space of previous image (see Figure 7.) in 3D. (x-axis representing r, y-axis φ and z-axis representing intensity)

Every local extreme or peak represents a combination of angle and length of a normal to the line from the origin, which determines a line in the source image.

There are several approaches to searching for these extremes.

The most simplistic is using a fixed threshold and collapsing detected areas “above the threshold” to one pixel – producing one line. This one proved to be useful after extending in the following way. For each pixel above the threshold, the algorithm looks in close surroundings (several pixels) for an extreme value.

Having these extremes, we can locate the extents of the boundary line by comparing with pixels in the filtered image (Sobel, Prewitt…).

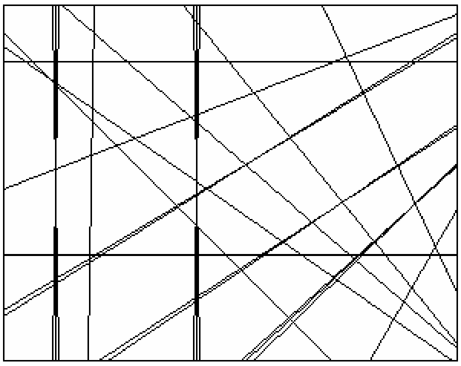

Figure 9. Detected lines from the Hough transform, virtually infinite in size

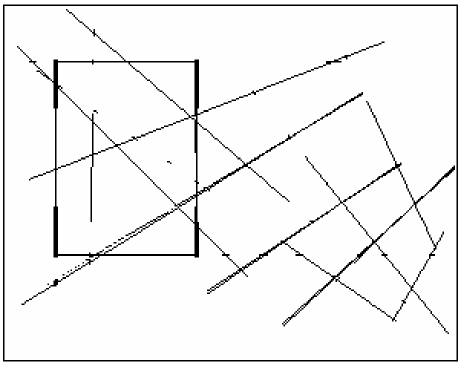

Figure 10. Detected lines with detected extents

4.2 Hough transform for circle detection

The equation of a circle is

$$(x-a)^2+(y-b)^2=r^2$$

where (a,b) is axis of center circle and r is the radius. In this case we need a 3-dimensional Hough space [1, 3, 4]. The first dimension for a, second for b and third for radius. If we presume that the center of circle lies in image, then a, b are equal to the width and height of the input image and the extent of the radius we choose as the expected size of circle or define the interval where the radius may lay.

For each important point in the filtered image (e.g. Sobel, Prewitt, …) compute possible axis of center circle for actual radius (having a point with coordinates (x,y) and radius is equal r, then the possible center lays on circle with center (x,y) and radius r), then add into accumulator actual value this point.

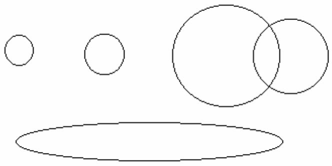

Figure 11. Original image with circles to be detected.

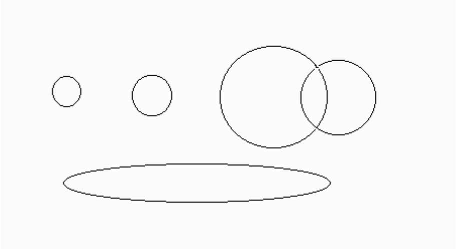

Figure 12. Modified Sobel (for description see section 3) filtered image

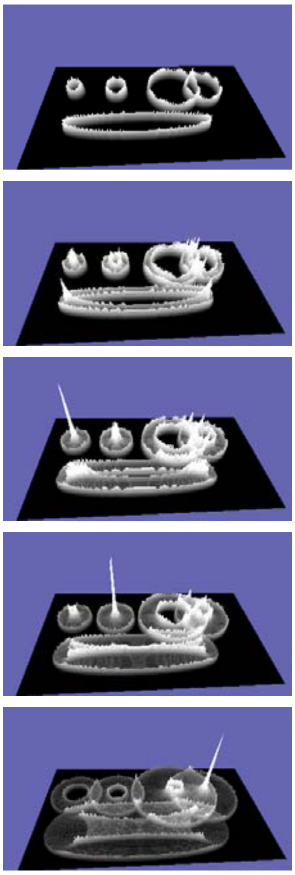

Figure 13. Hough's space for increasing radius (descending) in 3D. (x-axis representing a, y-axis b and z-axis representing intensity).

In the Hough space for circle detection, one can see the peaks / local extremes which represent the axis of possible centers of circles. Previous figures evince how the peaks grow (for increasing radius), for individual possible center of circles in input image on figure 11. At first, for small values of searched radius (compared to the original radius), the extremes are cyclic or elliptic shaped. Raising the radius meets the best representation of the original circle with a thin peak. Higher Hough-transformed radius values loose the extreme quickly. Extremes detected in this iterative way can be verified by comparison with the original image, leading to accurate parameters of detected circles and even ellipses.

5 Conclusion and Future Work

This paper presents experiments and implementational results of searching for mathematically defined shapes in a raster image. We achieved good reliability in detection of lines, linear edges of solid areas and circles, having good chance of detecting also ellipses.

These detection sub-routines allow us to start experimenting with design of a shape, which will be finally looked for in raster image. This shape needs to be reliable, insensitive to rotation and possibly to some degree of affine transformation. Other challenge in the shape definition is to allow detection and quantification of such (rotation, projection) deformations.

References

[1] David Vernon. *Machine Vision, Automatic Visual Inspection and Robot Vision*. Printed in Great Britain by Cambridge Press 1991

[2] Jiří Žára, Bedřich Beneš, Petr Felkel, *Moderní počítačová grafika.* Computer Press 1998.

[3] Joohi Lee, *Hough transform*. For a presentation in Computer Vision - Spring 2000 http://www.cs.unc.edu/~lee/class/comp290_75/HoughTransform/HoughTransform.html

[4] Bob Fisher, Simon Perkins, Ashley Walker and Erik Wolfart, *HyperMedia Image Processing reference.* Department of Artificial Intelligence University of Edinburgh UK 1994 http://www.cee.hw.ac.uk/hipr/html/hipr_top.html