

# Tutorial: Hough transform and line detection

## Note

This tutorial introduces the Hough transform. Line detection operators are implemented.

## 1 Introduction

This tutorial deals with line detection in an image. For a given point of coordinates  $(x, y)$  in  $\mathbb{R}^2$ , there exists an infinite number of lines going by this point, with different angles  $\theta$ . These lines are represented by the following equation:

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta).$$

Thus, for each point  $(x, y)$  (Fig. 1a) corresponds a curve parametered by  $[\theta, \rho]$ , where  $\theta \in [0; 2\pi]$  (Fig. 1b). The intersection of these curves represents a line (in this case,  $y = x$ ).

## 2 Algorithm

The (general and simple) method for line detection is then:

1. Compute contours detections (get a binary image BW).
2. Apply the Hough transform on the contours BW.
3. Detect the maxima of the Hough transform.
4. Get back in the Euclidean space and draw the lines on the image.

Results should look like in Fig. 2.

## 3 Hough transform



Code a function that will transform each point of a binary image into a curve in the Hough space. For each curve, increment each pixel by one in the Hough space.

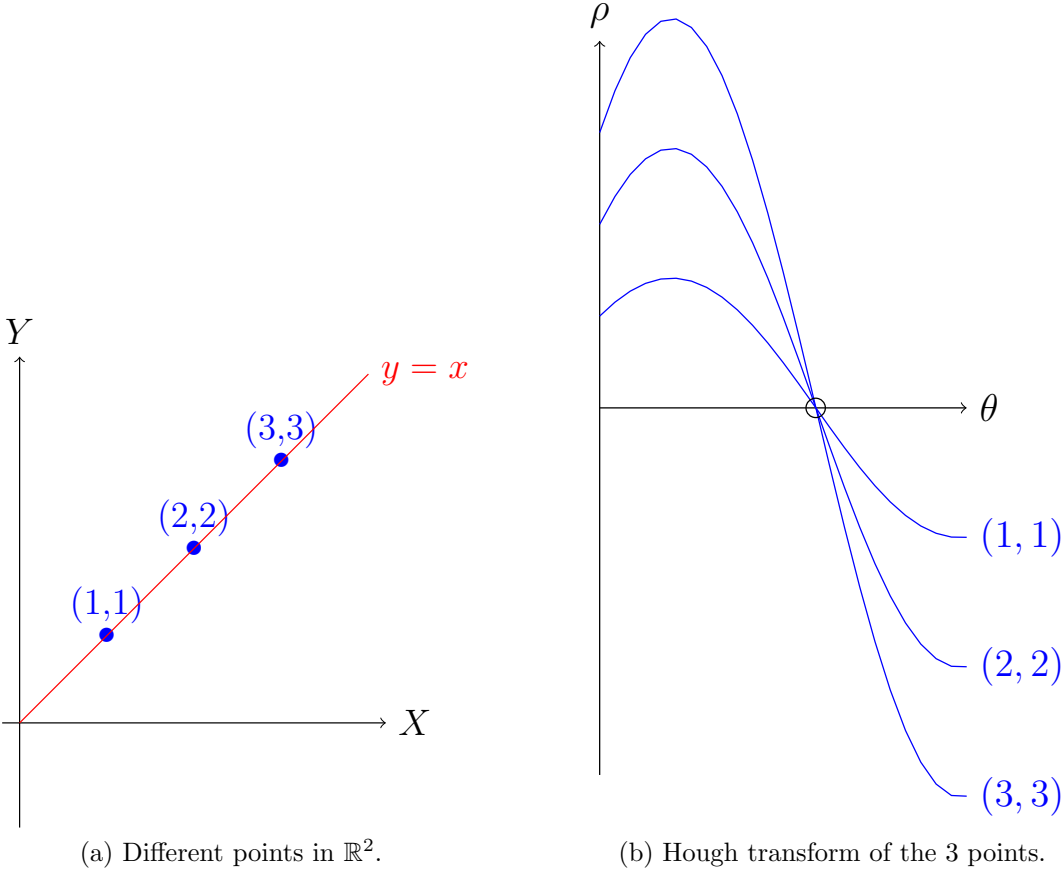
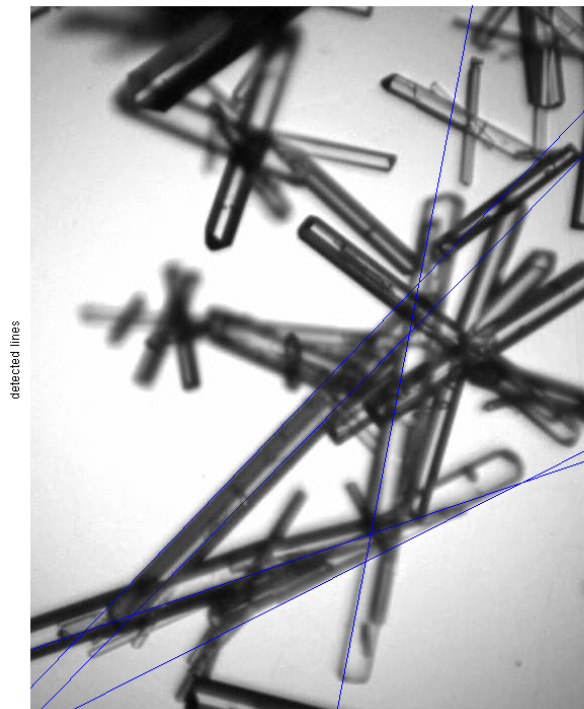


Figure 1: Representation of the Hough transform.



(a) Hough transform and maxima detection. Angles  $\theta$  are represented in abscissa, pixels  $\rho$  are represented in ordinates. The detection of the absolute maxima of this images will lead to the lines.



(b) Line detection.

Figure 2: Results to obtain.

## 4 Maxima detection



Use or code a function to detect maxima (regional maxima). For each maximum, keep only one point.

## 5 Display lines



For each maximum, display the corresponding line above the original image.