Yorick de Boer, 10786015
Lucas van Berkel, 10747958

Contents

2 Finding Lines using the Hough Transform

2.1 Hough Transform Theory

In Matlab the formula is used:

$$x^i sin(\theta) - y^i cos(\theta) = \rho \tag{1}$$

The book shows the formula:

$$x^i sin(\theta) + y^i cos(\theta) = \rho \tag{2}$$

The difference is the plus sign instead of the plus. This is because of the matlab coordinates. In matlab, point will descend if x is increased. Because of this problem the space parameters are different from the Lecture Notes.

The reasen ρ can be negative is because of the direction a line is moved after it has been tilled over θ. The direction of a line after being rotated over θ is the same as the line being rotated over $\theta + \pi$, only it is in its negative direction. Because of this the translation over ρ must be also be in negative direction, will result in the same line. This is why ρ can be negative.

2.2 Performing the Hough Transform

Below is a example of a picture with its hough-transform. The hough-transform can be used to calculate sufficient lines over the original picture, in order to obtain the shape in the picture.

Figure 1: Result of houghtransform over the picture shapes

3 Finding the Lines as Local Maxima

In this part we take the hough-transform of the picture and calculate which lines are most efficient. We also dilate the hough-transform in order to filter maxima which are close to eachother.

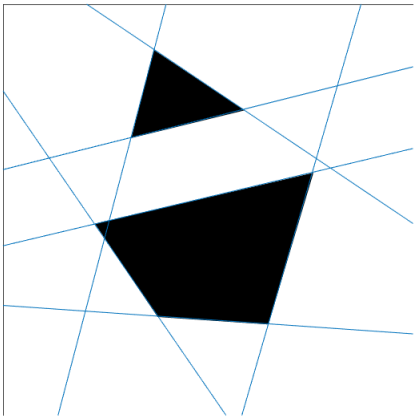

Figure 2: Result of the lines from the hough transform over the picture shapes

5 Optimal Line Estimation

In this part we take each line of the hough-transform and we look which edge-points are nearby. From these edge-points we calculate a new optimal line using the total least squares (TLS), a method similar to the least squares method, the difference is that it takes the orthogonal distance to the line, instead of the vertical distance, which should result into a better line.

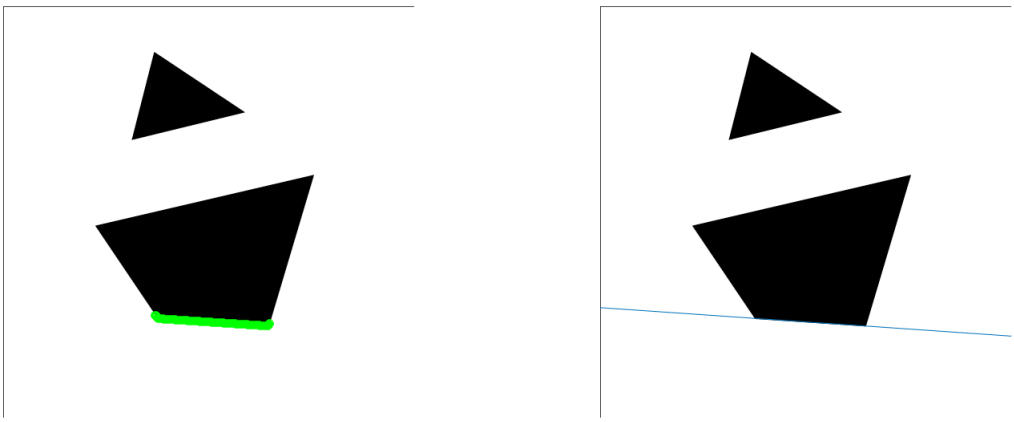

Figure 3: Results finding the edge-points near a line giving by the hough-transform and its fit using TLS

6 Using the Lines

With the optimal line found using the total least squares method, we can now calculate the intersection points of several lines. Calculating the intersection of two lines represented in homogeneous line coordinates is quite simple, you only have to take the cross-product of the two line vectors and normalise the result. The crossproduct is sufficient because the crossproduct calculates the vector which is perpendicular to both vectors(which are now the lines). The formala of the crossproduct is:

$$\vec{u_1} \times \vec{u_2} = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} \times \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} b_1 \cdot c_2 - c_1 \cdot b_2 \\ c_1 \cdot a_2 - a_1 \cdot c_2 \\ a_1 \cdot b_2 - b_1 \cdot a_1 \end{bmatrix} \tag{3}$$

If two lines are parallel, so they never meet, the third argument will be zero, so the homogeneous coordinate becomes a direction instead of a point. In every other situation the third argument is proportional to the sin of the angle between the two lines. Surprising enough the line that connects to points in a 2d space can be calculated using the cross-product once the points are represented in homogeneous form. This is because the cross-product calculates the vector perpendicular to the first to vectors, so the result should go through the two points.
Below is the result of calculating the intersection points of two lines.

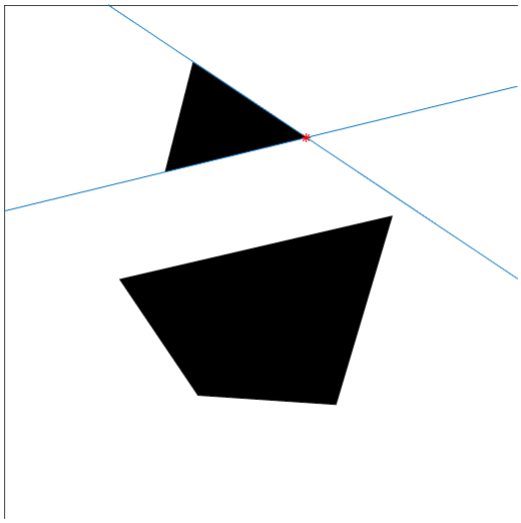

Figure 4: Result of the intersection point from the cross product of two homogenous lines