A. Feature Measurement

Using the intermeans algorithm with median as initial guess, 113 was chosen as the value of threshold T for test1.bmp.

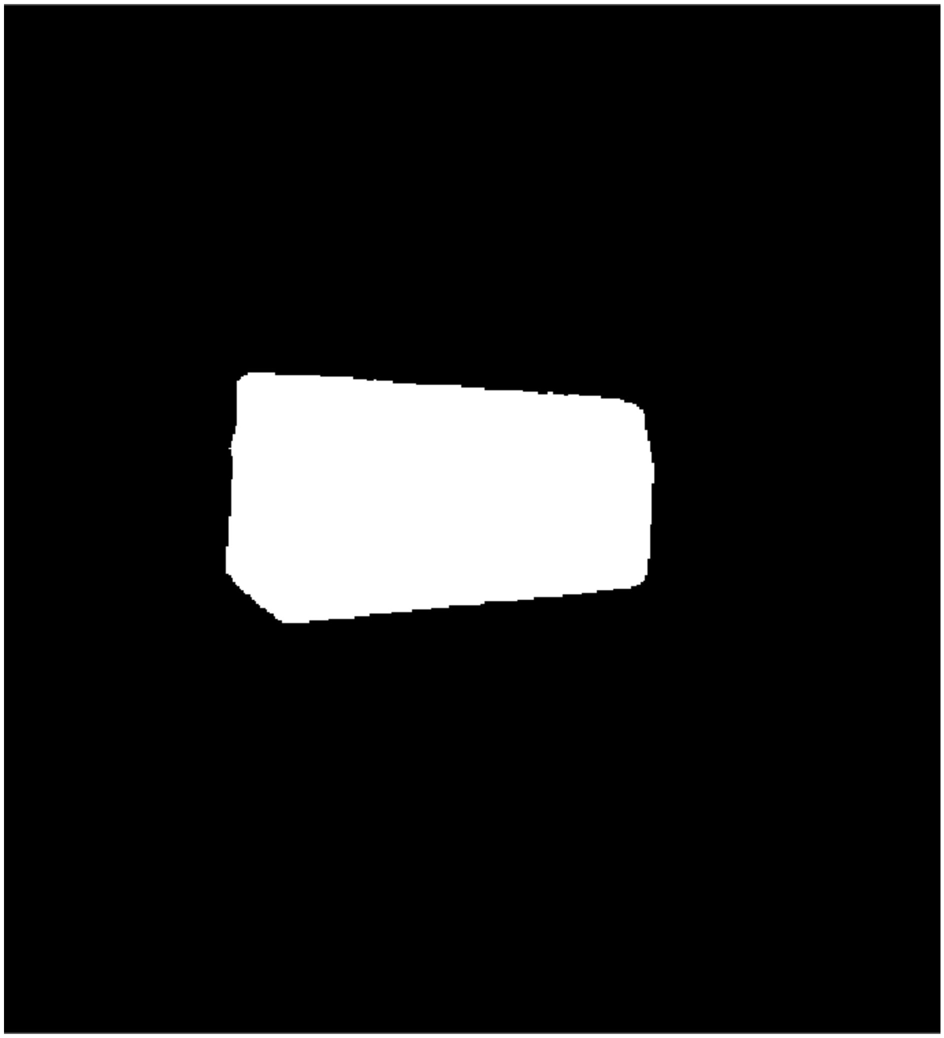

Figure 1: Output image IT of the intermeans algorithm with thresholding applied on test1.bmp.

The values of the features calculated are

1. Perimeter: 553
2. Area: 20016
3. Compactness: 1.2158
4. Centroid: $(200.3693, 251.8056)$
5. ϕ_1: 0.1983

As the image is already binary, `bwperim` can be used to find the perimeter. The default 4-connected perimeter was kept as there is no convenient way to account for diagonal boundaries and multiply them by a factor of $\sqrt{2}$. Choosing 8-connected perimeter results in overcounting.

The formula for compactness is simply

$$\gamma = \frac{\text{perimeter}^2}{4\pi \times \text{area}}$$

The centroid is the point $(\bar{x}, \bar{y})$ defined by

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}} \tag{1}$$

The generic formula for the moments is

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \tag{2}$$

For the central moments, the following derived formulas were applied

$$\begin{aligned}
\mu_{00} &= m_{00} && (3)\\
\mu_{10} &= 0 && (4)\\
\mu_{01} &= 0 && (5)\\
\mu_{20} &= \sum\sum(x - \bar{x}^2)f(x, y) && (6)\\
\mu_{02} &= \sum\sum(y - \bar{y}^2)f(x, y) && (7)\\
& && (8)
\end{aligned}$$

The generic formula for the normalized central moments is

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{9}$$

where

$$\gamma = \frac{p+q}{2} + 1 \quad \text{for } p + q = 2, 3, \ldots$$

Finally, the formula for the first invariant moment is

$$\phi_1 = \eta_{20} + \eta_{02} \tag{10}$$

B. Feature Invariance

For this image, `bwareafilt` is applied to remove artifacts at bottom left and right when calculating its features.

Using the intermeans algorithm with median as initial guess, 79 was chosen as the value of threshold T_2 for test2.bmp.

The values of the features calculated are

1. Perimeter: 631
2. Area: 35412
3. Compactness: 0.8947
4. Centroid: $(219.9672, 189.2160)$
5. ϕ_1: 0.1629

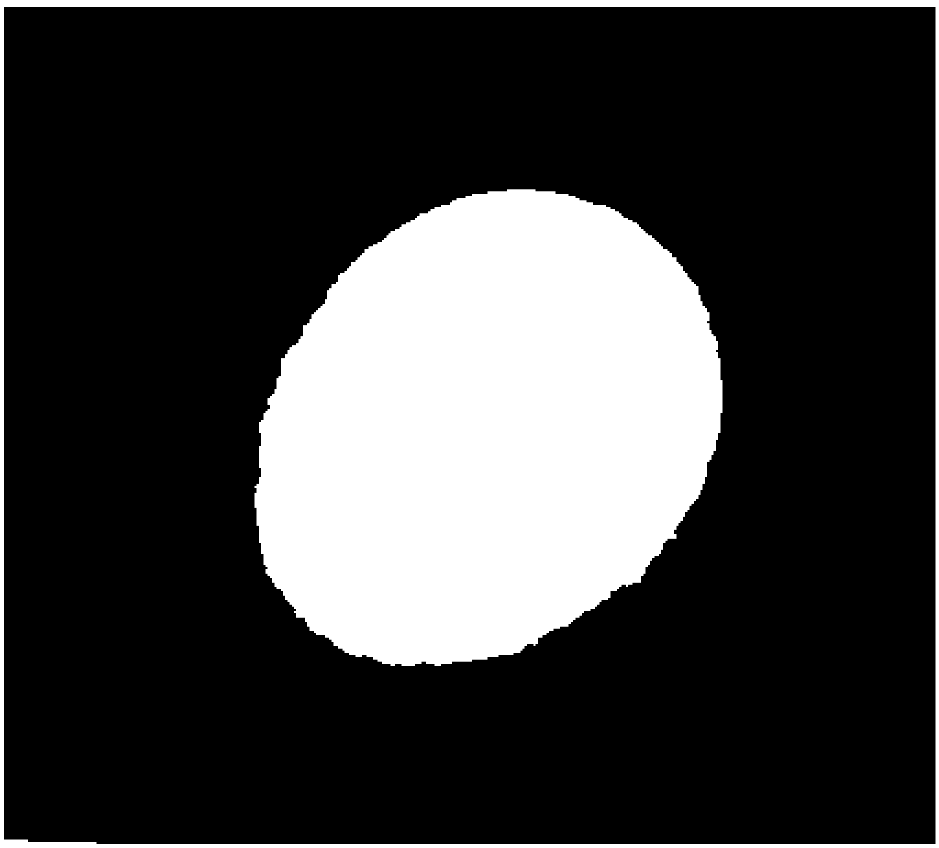

Figure 2: Output image JT_2 of the intermeans algorithm with thresholding applied on test2.bmp.

According to the histogram (Fig. 3), the highest peak is at 29 and there are smaller peaks surrounding it, falling back to relatively normal levels at 32. They can be attributed to the background. From 33 onward, the values are no longer consistently higher than other intensities. The subsequent values around 40 belong to the boundary between the object and the dark background. As a compromise, T_{opt} should be 40.

The values of the features calculated are

1. Perimeter: 680
2. Area: 43903

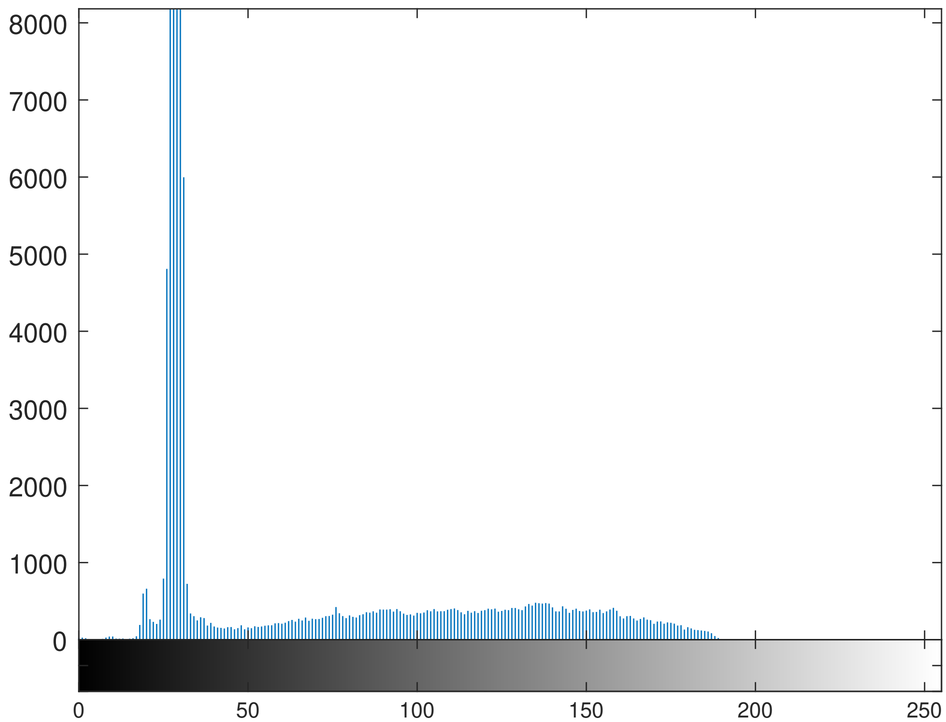

Figure 3: Histogram of test2.bmp.

3. Compactness: 0.8381
4. Centroid: $(215.7495, 179.3796)$
5. ϕ_1: 0.1627

Comparing Fig. 2 (JT_2) and Fig. 4 (T_{opt}), the lower threshold of the latter results in a larger object, with more of the lower values near its boundary being accepted.

The perimeter and area of the object increased with more pixels. The centroid has shifted left, downward, toward the shadowed area (illumination seems to be coming from top right). This is because the increase in pixels from lower threshold value will mostly be in the shadowed area.

The object appears rounder (less elongated), causing compactness to fall.

The first invariant moment, ϕ_1 is invariant to rotation, scale change and translation. A digital image however, is not strictly invariant under image rotation and scale change. In this case, the shape of the object has changed, so it is not surprising to see slight change in ϕ_1 even though it is supposed to be invariant to scale change.

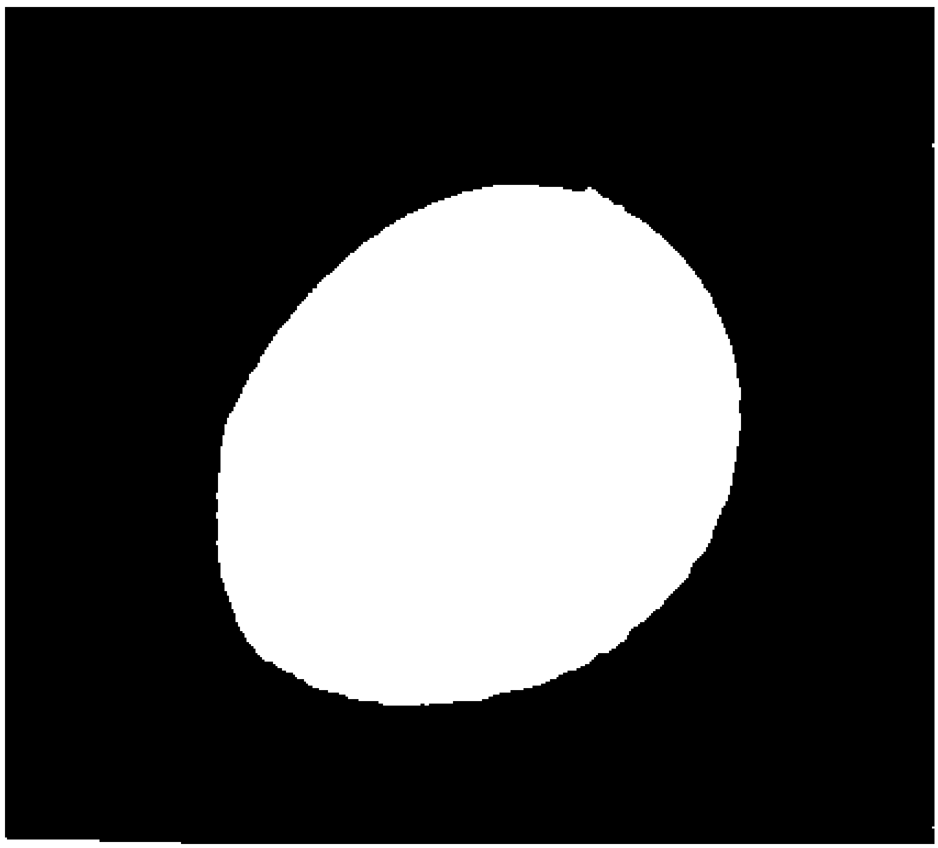

Figure 4: Output image of thresholding with Topt on test2.bmp.

C. Boundary Plot

The centroid may be estimated from the boundary pixels

$$B = \{(x_i, y_i), \quad i = 1, 2, \ldots, N\}$$

The coordinates of the centroid are given by

$$\bar{x} = \frac{1}{N}\sum_i x_i \quad \bar{y} = \frac{1}{N}\sum_i y_i \tag{11}$$

To calculate the distance, the Pythagorean theorem is sufficient. The two argument variant of the inverse tangent function, `atan2`, is necessary avoid ambiguity in calculating the angle when $x < 0$.

At 3.776 44 rad and 3.7805 rad, there are two anomalous points, corresponding to the two white pixels at the bottom left of test3.bmp. `bwareafilt` can be used to remove them.

There are two peaks, at 1.976 89 rad and 5.1282 rad respectively. They correspond to the petiole (stalk) at the top left and the tip at the bottom right of the leaf, which have long thin shapes.

There are three troughs, at 0.010 453 3 rad, 3.851 73 rad and 6.277 12 rad. The first and last are in fact very close together due to the circular nature of polar coordinates. They correspond to the areas where the leaf becomes wider. The part closer to the petiole is significantly wider than the part closer to the tip.

For all the peaks and troughs, it can be seen that they are areas where, at very close angles relative to the centroid, there are pixels at different distances.

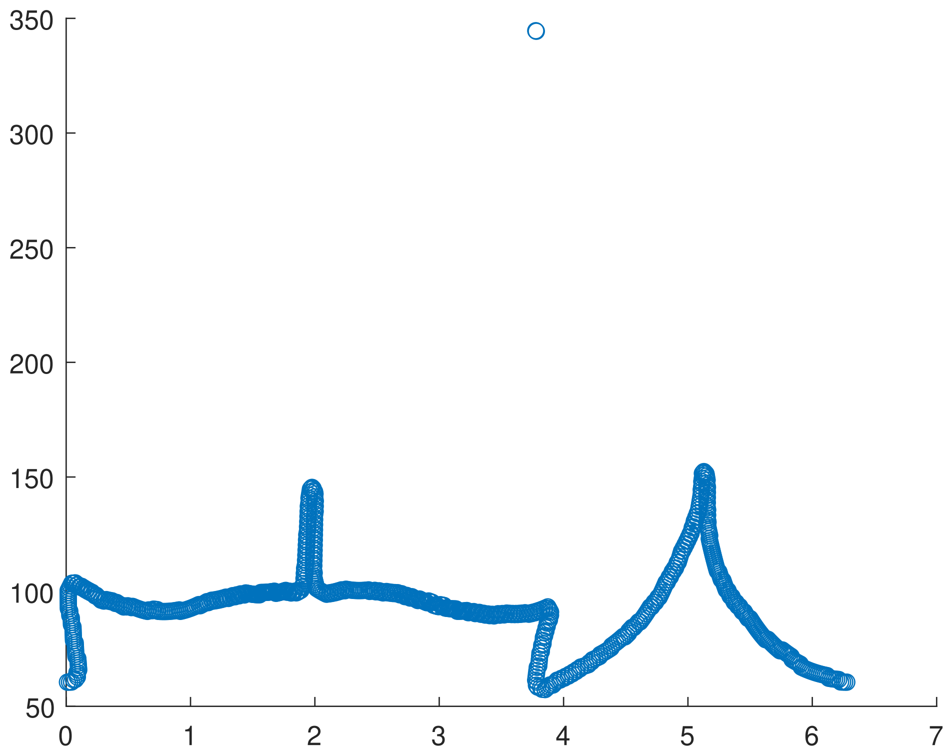

Figure 5: r–θ values of test3.bmp.

D. Hough Transform

It appears that for straight line detection in letter.bmp, the simpler methods taught in this module provide better results than, for example, the Canny method, the most powerful edge-detection method MATLAB provides.

The L-shaped object is metallic, but primarily this question is interested in the straight lines of the L-shape. The texture of the fabric background and the metallic shin of the object are considered noise. The `edge` function has additional parameters but Prewitt, Sobel and Roberts all work as intended by default. The gradient magnitude alone is sufficient in this case. Sobel, the default method, was chosen.

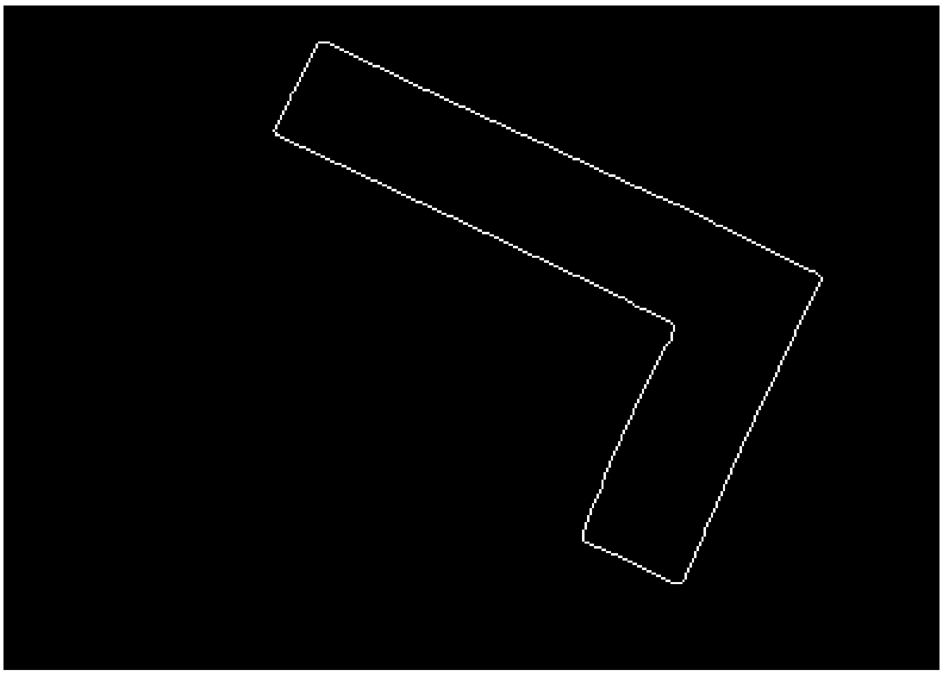

Figure 6: Edge map of letter.bmp.

The normal representation of the line is chosen to allow the detection of any vertical lines.

$$x \cos\theta + y \sin\theta = \rho \tag{12}$$

With regard to the bin count, θ is restricted to $-90° \leq \theta < 90°$. This follows the convention set by MATLAB's `hough` function and avoids overlap for signed distance. Meanwhile, ρ is restricted to $-481.9647 \leq \rho \leq 481.9647$. With origin at the top-left, the maximum absolute value of ρ is the diagonal of letter.bmp, calculated using the Pythagorean theorem. The dimensions of letter.bmp are 393 and 279. The number of different integer values for ρ is chosen to be two times the diagonal rounded down, 963. The resulting matrix is 963 by 180.

There are only 6 lines on an L, but every non-background point has to be considered and the lines are relatively thin, so having one bin for each integer value of ρ is necessary for precision despite the cost of detecting the same line multiple times with slight variation in ρ and θ in the output if the threshold is set too low. With too many bins, closely-related lines are put into separate bins by the algorithm.

With threshold value at 0.54 times the maximum value in the accumulator, I was able to obtain three lines for the three longest sides of the L-shaped object (Fig. 8), but further reducing the value to obtain more straight lines results in multiple lines appearing for the three sides already found.

As a voting technique, when there are lines that are much longer than the others and there is no perfect alignment due to the limitations of the edge detection algorithm, Hough transform tends to find multiple slight variations of the longer lines before finding the shorter ones.

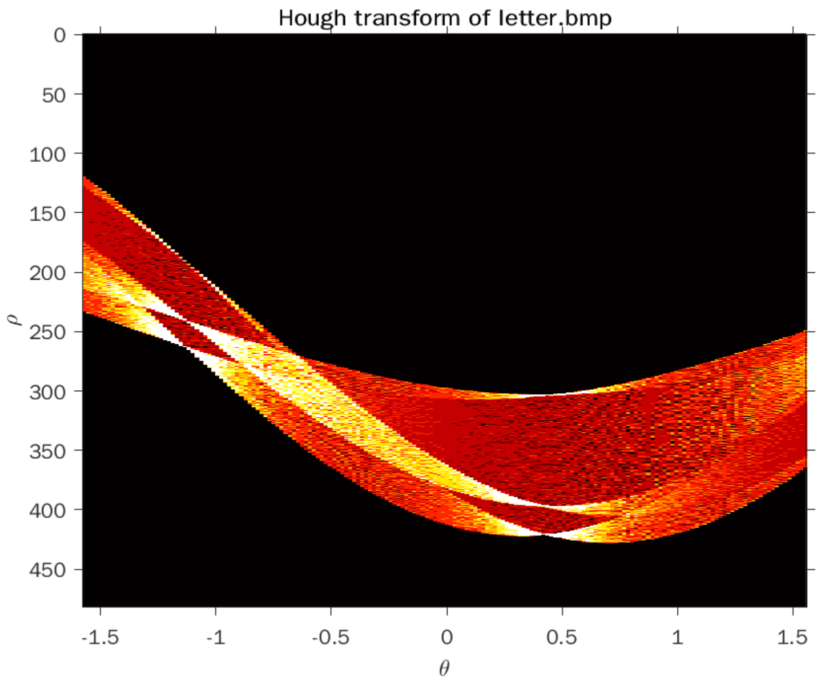

Figure 7: Hough transform of letter.bmp.

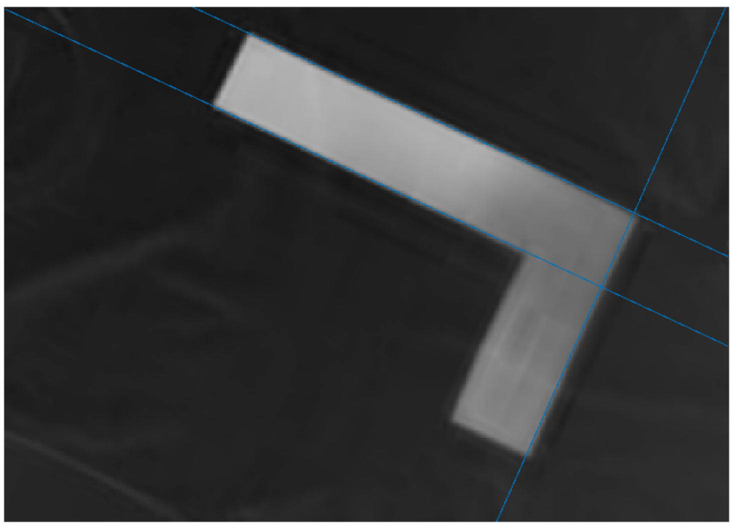

Figure 8: Detected lines of letter.bmp.