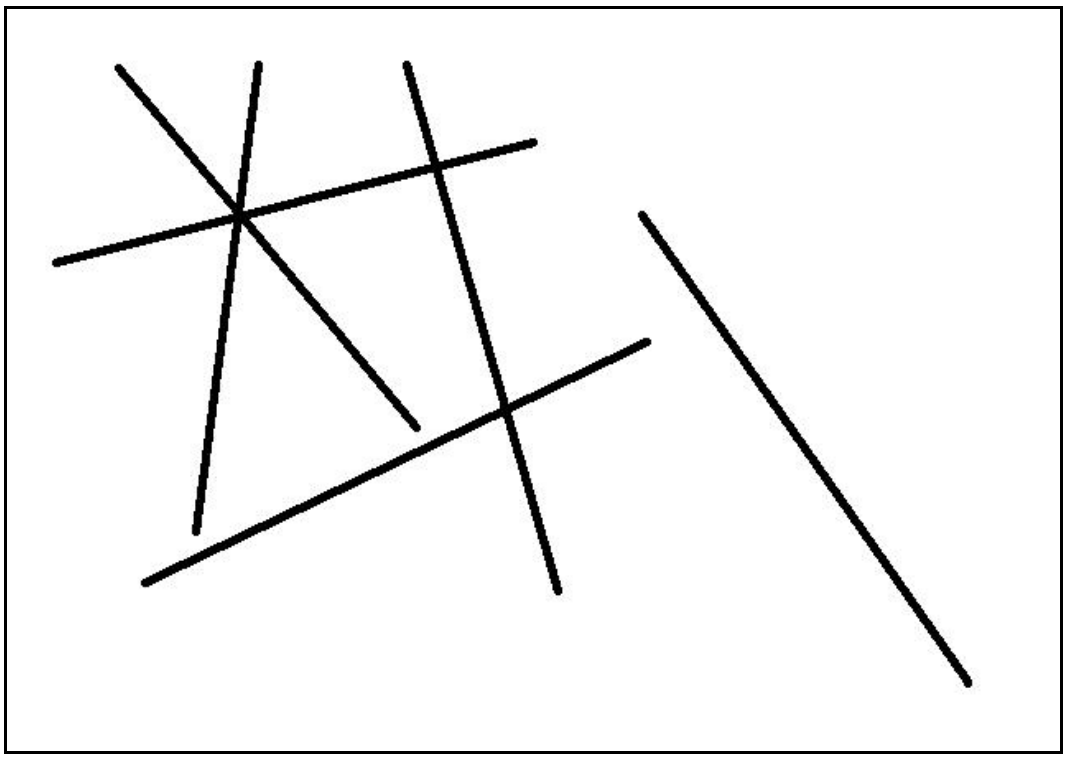
A. Line detection using Hough Transform



Original Image

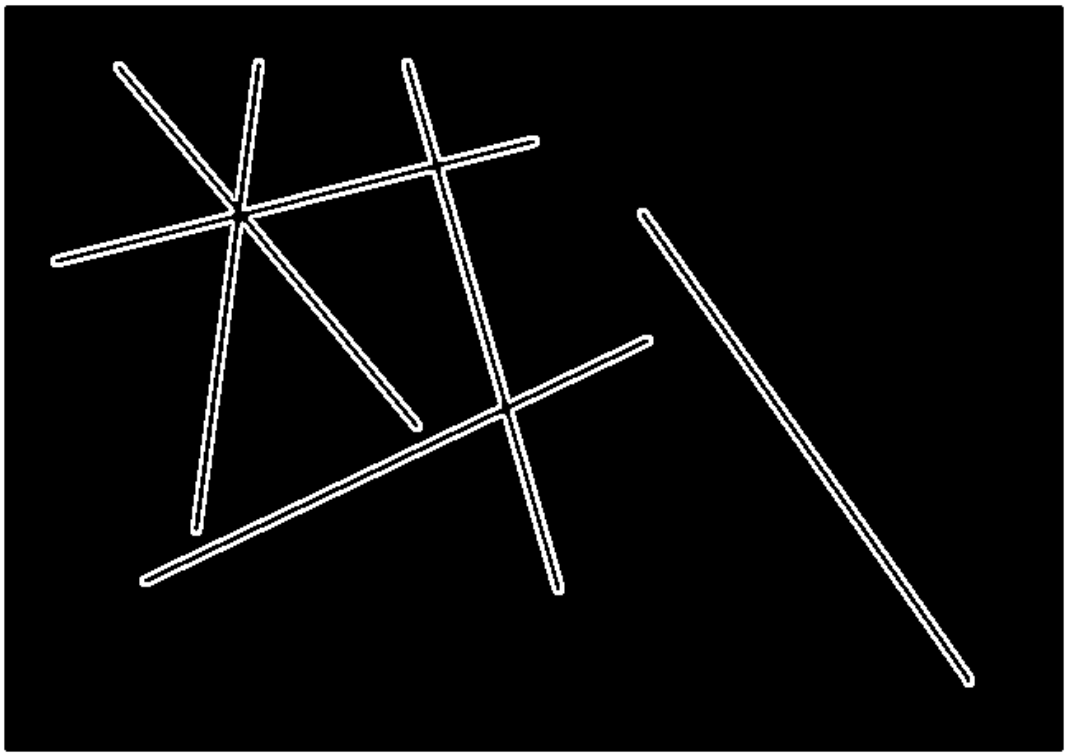


Image after edge detection using Sobel's 5x5 edge operator

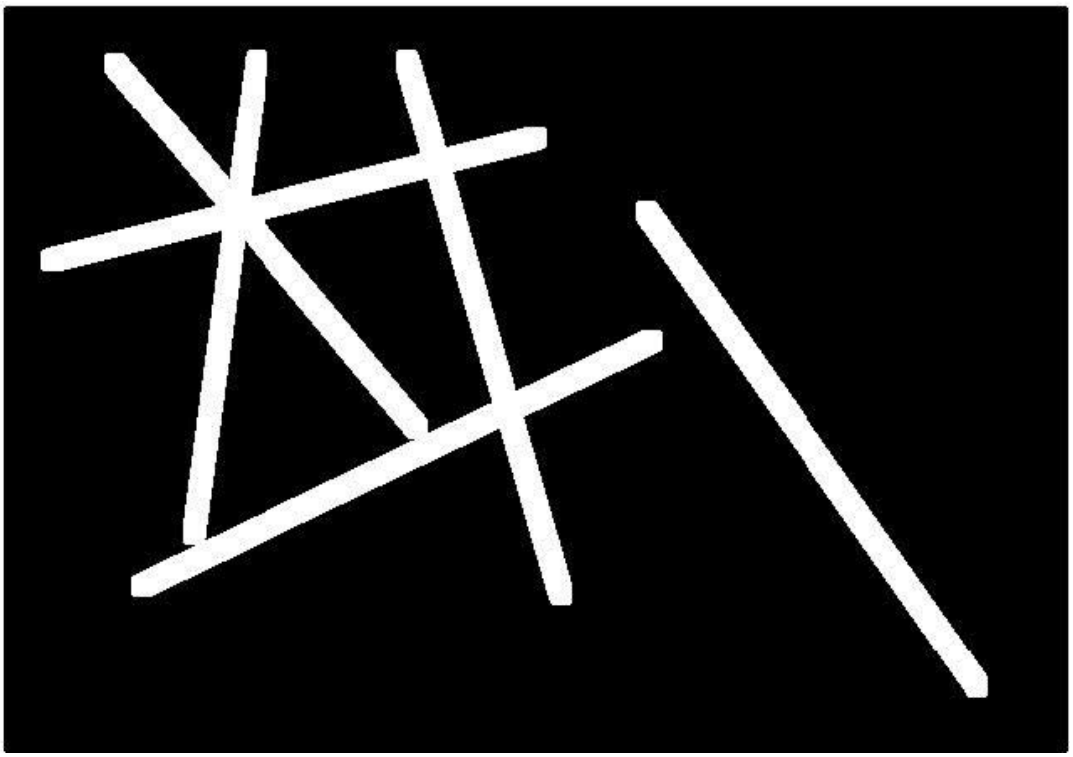


Image after Dilation to fill in gaps

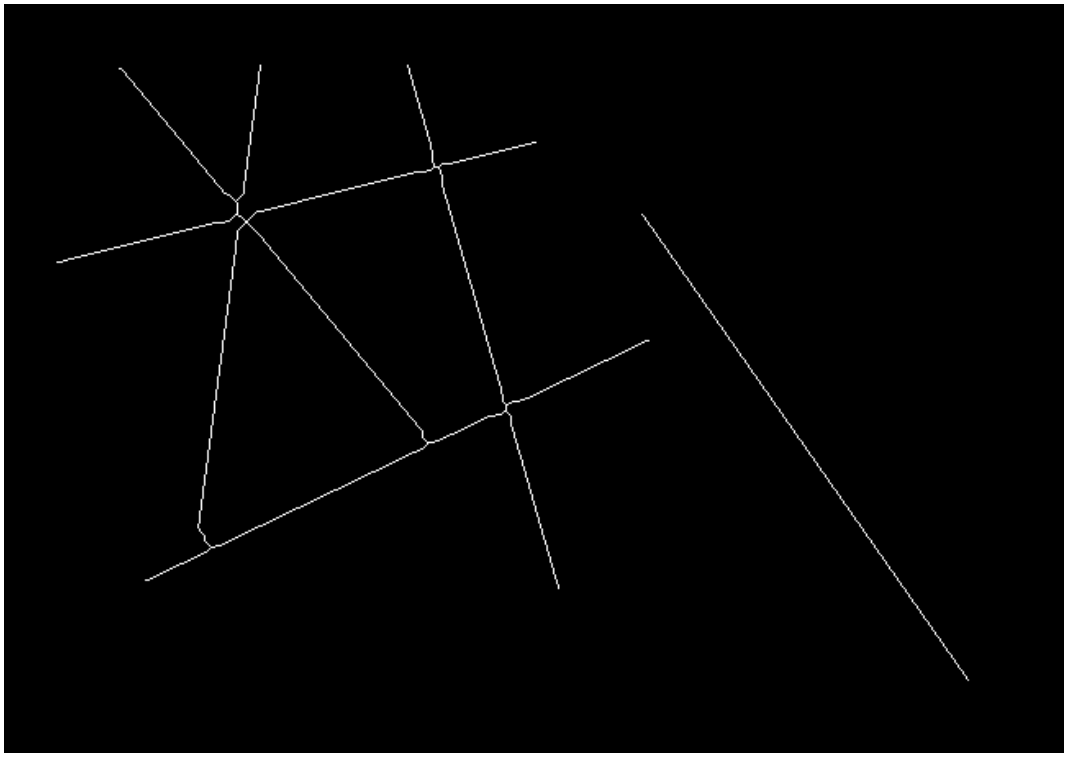
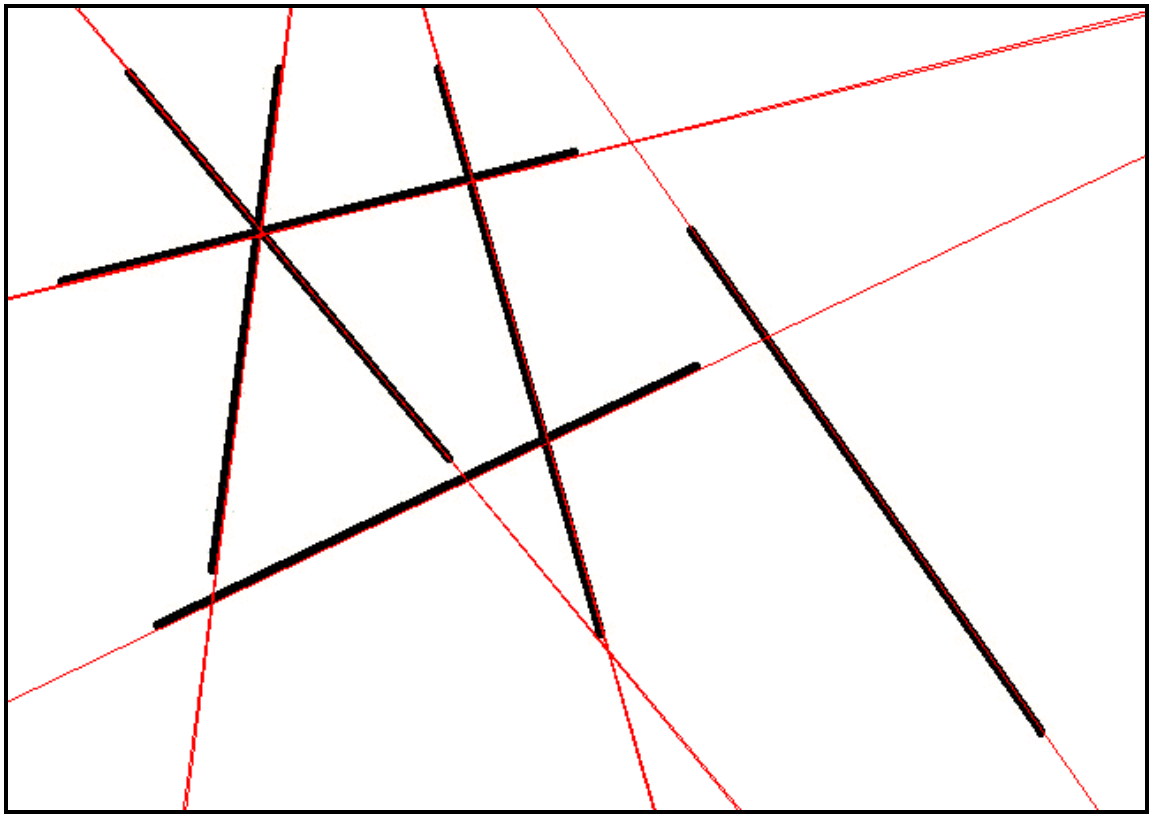


Image after edge thinning

Polar Hough Transform: Accumulator Space of Rho (vertical axis) vs. Theta (horizontal axis)



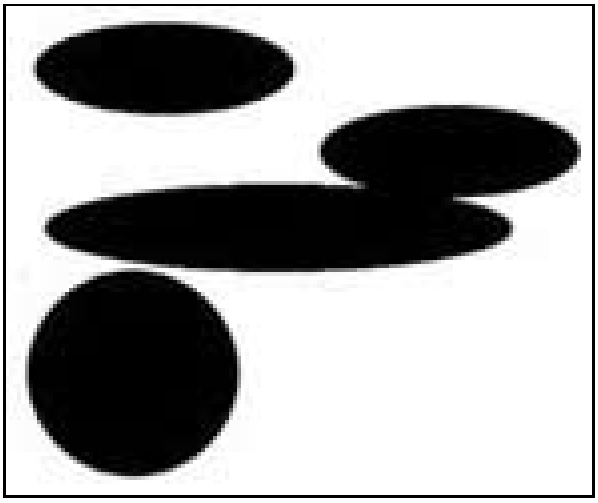
Lines plotted from Accumulator after finding Local Maxima and Thresholding of over 150 votes

B. Application of the Hough Transform for line detection

Original Image

Lines with over 90 votes detected
using Polar Hough Transform

C. Ellipse detection using Hough Transform



Original Image

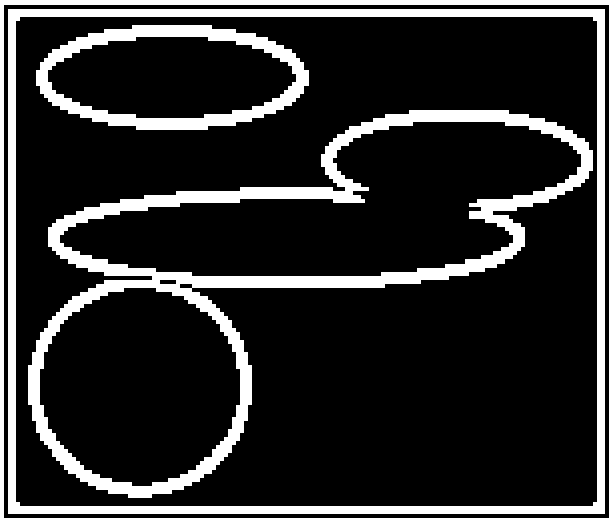


Image after edge detection using
Sobel's 5x5 edge operator

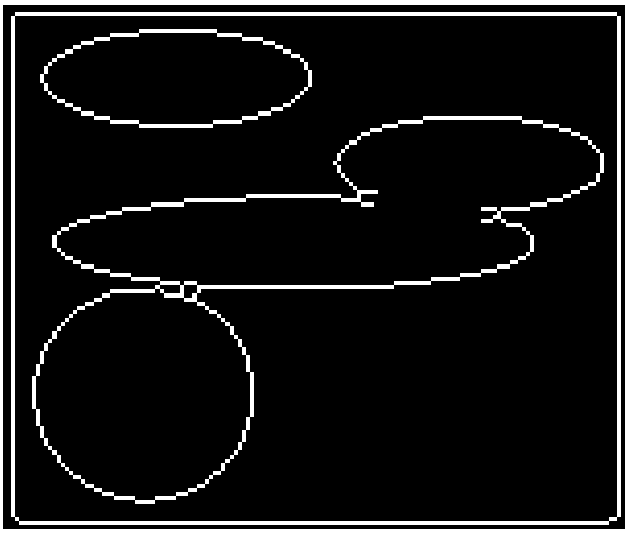
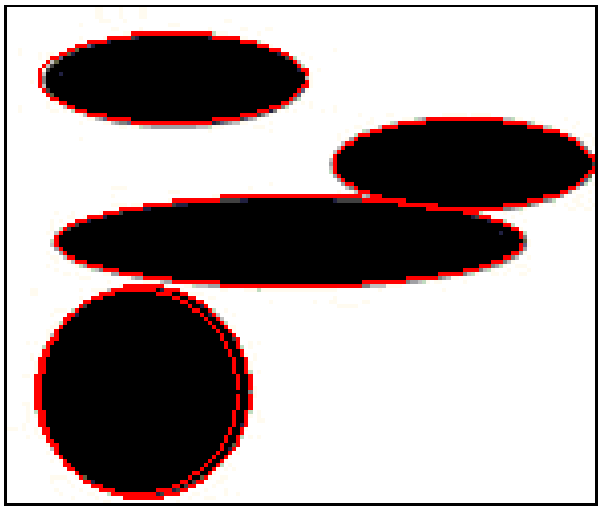


Image after edge thinning



Ellipses plotted from 4 dimensional accumulator
after finding Local Maxima and Thresholding for 23 votes

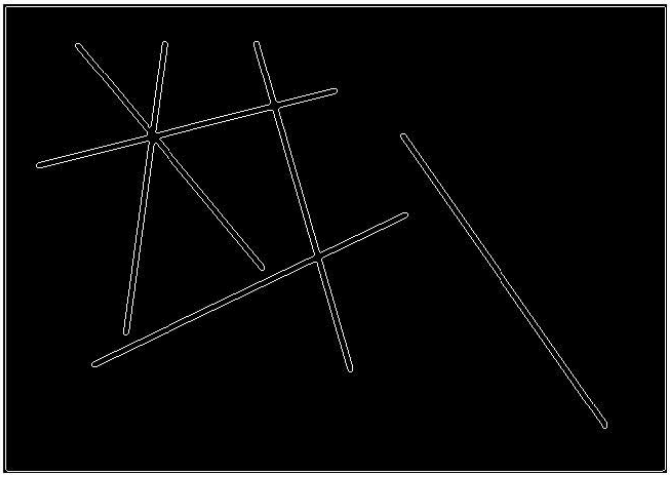
Note: Due to "Out of Memory" exceptions in Matlab the given image (390 x 325) was re-sized to (147 x 123)

D. Analysis

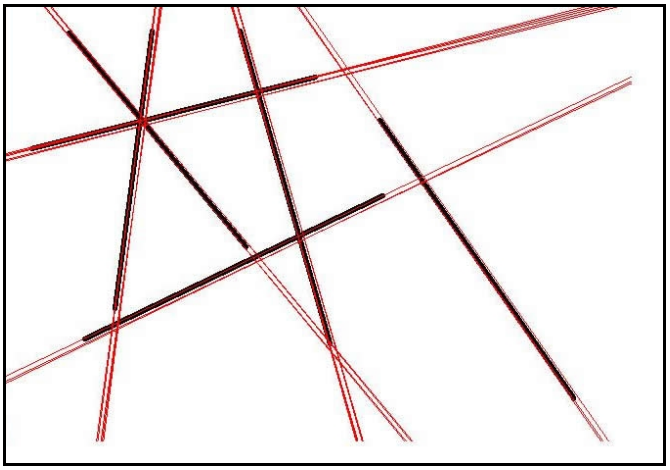
Dilation and Erosion

While implementing the Hough Transform for detecting lines, it may be useful to incorporate dilation before erosion to remove the gaps in parallel lines. Not only will the reduced edges offer better results but will also drastically reduce the number of computations to be carried out.

Here are the results I got without dilation as compared to what is presented above:



Thinning without dilation

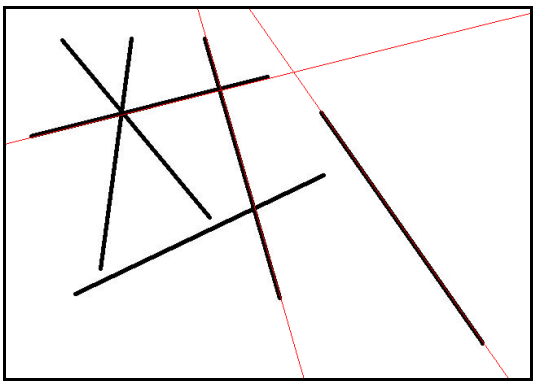


Edge detection gives parallel lines in spite of filtering by Local Maxima and a Threshold

Precision of the Parameters

While calculating the accumulator, increasing the precision of the parameters (at the cost of time) ended up giving significantly better results.

Initially, I was incrementing the value of theta (line detection) by 1 degree which gave me an accumulator with bins that had many neighbors in the same range. However, by increasing the steps to 0.1 degrees, the number of bins having high values decreased but there were more points with higher values resulting in clearer, brighter points in the accumulator plot.



With a step size of 1 degree for theta, a threshold of 150 votes gave just 3 lines. Conversely with a step size of 0.1 degree for theta, a threshold of 150 votes plots all 6 lines.

Cartesian vs. Polar Co-ordinate Systems

In the case of the Cartesian parameterization m can actually take an infinite range of values, since lines can vary from horizontal to vertical. To avoid the complications of handling these conditions, I decided to use the alternate Polar Co-ordinate system for calculating the bins.

Principal Axis parallel to X-Axis

While implementing the Hough Transform it helps to reduce the number of possible combinations of the parameter space by applying constraints that may be available. In this case, we were told that the principal axis of the ellipses would always be parallel to the X-axis. In other words, a >= b at all times.

This greatly reduced the number of computations as the range of 'a' was curtailed depending upon the value of 'b'. The code in Matlab was:

```
for b=1:maxB
    for a=b:maxA
        …
```

Using the Gradient Direction

A number of calculations in the parameter space can be reduced by utilizing the gradient direction in the image space that denotes the direction of the greatest change.

Although, I was able to use the gradient direction in detecting the ellipses, I could not find a way to actually reduce the number of parameters as hinted in various textbooks.

References

1. Handbook of Computer Vision Algorithms in Image Algebra, 2nd Ed - Gerhard X. Ritter
2. Feature Extraction in Computer Vision and Image Processing - Mark S. Nixon
3. Computer Vision A Modern Approach - Forsyth & Ponce
4. Computer Vision - Ballard & Brown