David Wong
CSE 455
Autumn 2014
HW #2

Finding Callouts in Parts Diagrams

In this homework assignment, we are provided with a few images of parts diagrams. Our goal is to identify the circles displayed on these images and mark them as accurately as possible.

My solution to this problem utilizes the Hough Transform to detect the circles in the images. The magnitude of the gradient is calculated at each pixel location using Sobel operators, and this value is compared with a gradient magnitude threshold. If the calculated gradient magnitude exceeds the threshold, then we estimate the center of the circle using a passed-in radius value. We then store a vote for this (y, x) center location into an accumulator. This accumulator is a 2D-array that maps to the entire image area.

After calculating the gradient for each pixel in the image, we have gathered votes in the accumulator for suspected centers of circles. We iterate over every voted center in the accumulator. If the number of votes for that center location exceeds some passed-in center threshold, then we gather all the votes nearby this location and combine them into one single vote. The new center location for this megavote is based on a weighted average of the center locations of all the votes that it includes. We store this new (y, x) location and vote count back into the accumulator and continue this process for the remaining accumulator votes.

Lastly, we check all the votes remaining in the accumulator and filter out (y, x) locations that are less than some multiple of the passed-in center threshold. This ensures that we are only left with center estimates that are strongly supported by the image pixel data.

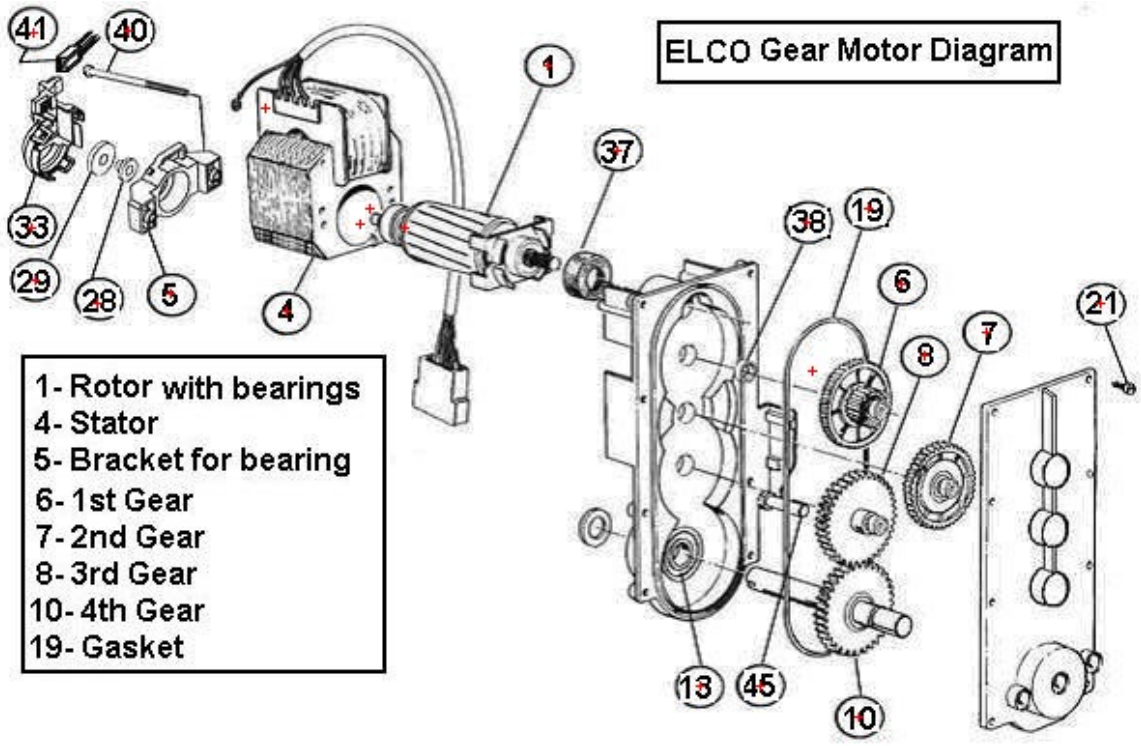
To run the program, merely run the *findcirc.cpp* program while providing the initial *.pgm* file and using the designated *radius* and *center_threshold* values listed below. Pipe the result of that into a *.ppm* file for storage.

```
./findcirc [input_image] [radius] [center_threshold] > [output_image]
```

Example:

```
./findcirc images/image1.pgm 11.6 6 > images/image1-result.ppm
```

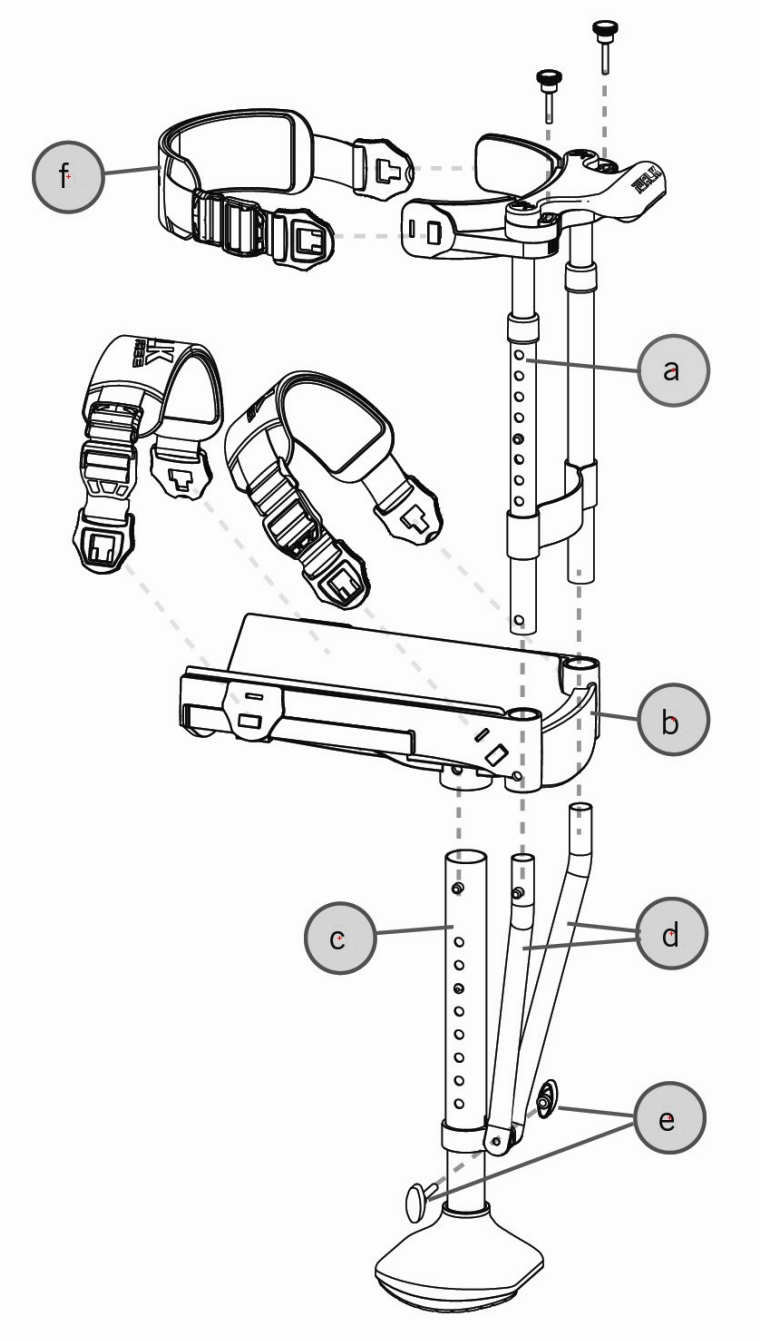
1. **image1.pgm**



radius = 11.6
center threshold = 6

The circle centers are marked pretty well, especially since some circles aren't fully connected due to the numbering (e.g. #28, #29, etc.). However, there are 5 extra marks on the images. These marks are located at areas near curved lines with great pixel density and are unfortunately hard to correct for.

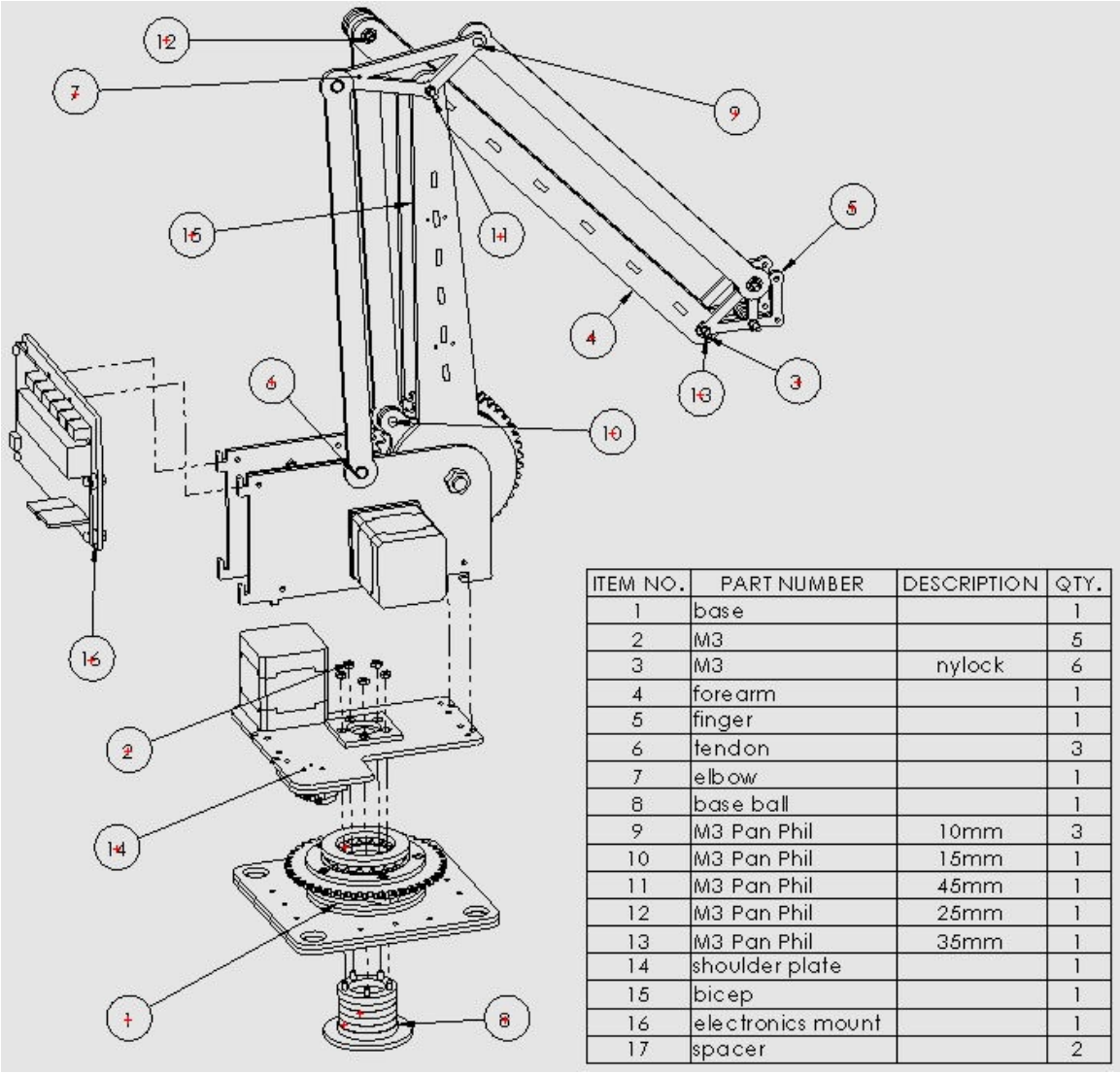
2. image2.pgm



radius = 39
center threshold = 12

It was fairly easy to detect the circles in this image. The diagram was very clear and of high resolution, and each circle was drawn with thick borders. This allowed the program to easily estimate and mark the circle center locations without creating any inaccurate markings.

3. image3.pgm

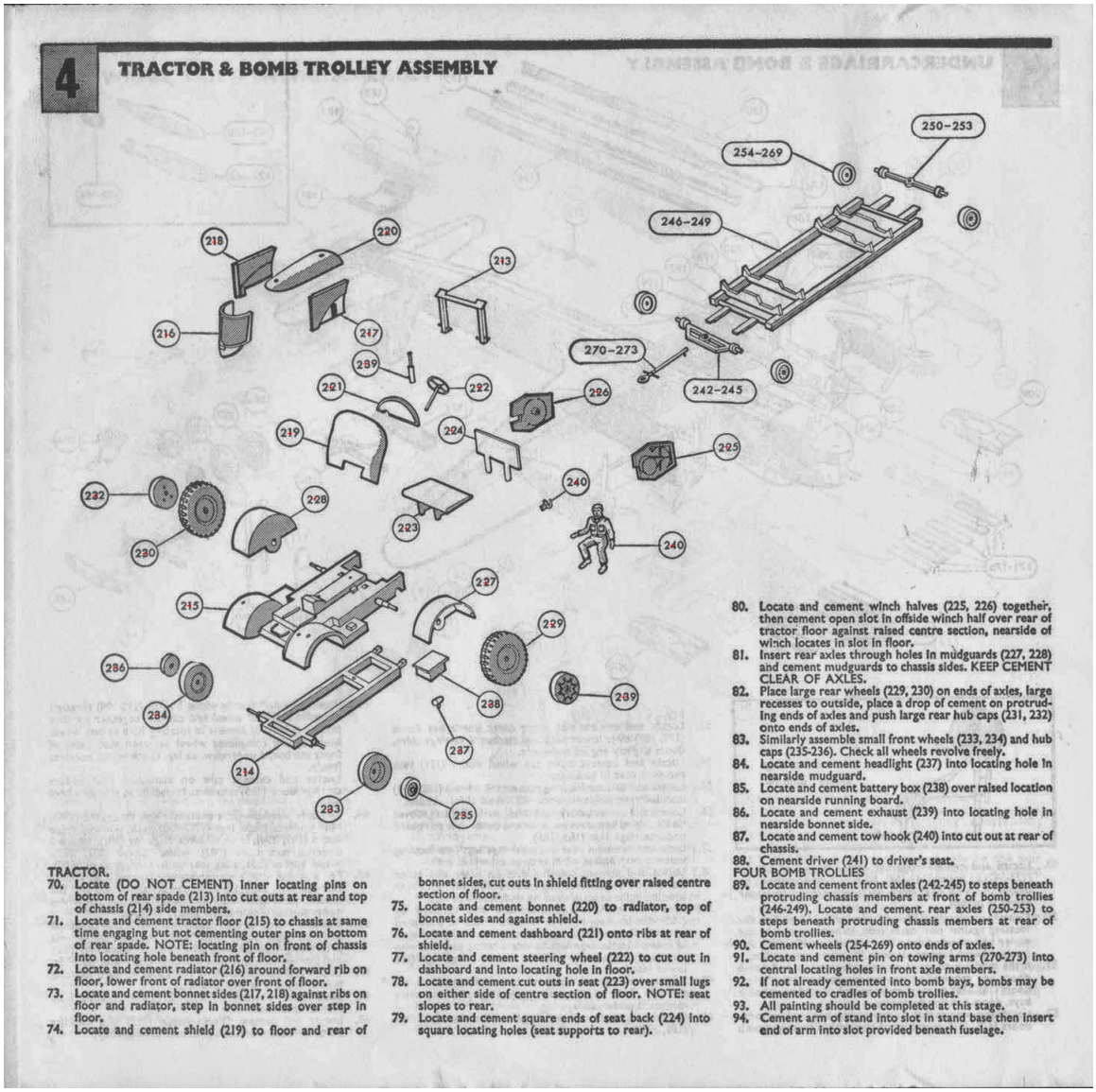


ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	base		1
2	M3		5
3	M3	nylock	6
4	forearm		1
5	finger		1
6	tendon		3
7	elbow		1
8	base ball		1
9	M3 Pan Phil	10mm	3
10	M3 Pan Phil	15mm	1
11	M3 Pan Phil	45mm	1
12	M3 Pan Phil	25mm	1
13	M3 Pan Phil	35mm	1
14	shoulder plate		1
15	bicep		1
16	electronics mount		1
17	spacer		2

radius = 13.5
center threshold = 5

There were 3 additional marks placed on this image. Again, these were located near curved lines that may have provided enough votes to throw off the estimation and filtering process.

4. image4.pgm



radius = 17
center threshold = 11

The circle borders were solid enough that the program was able to successfully detect and label all circles without any excess markings. However, the elongated circles were left unmarked, presumably due to an insufficient number of votes. I assume this is the expected behavior.

5. image5.pgm

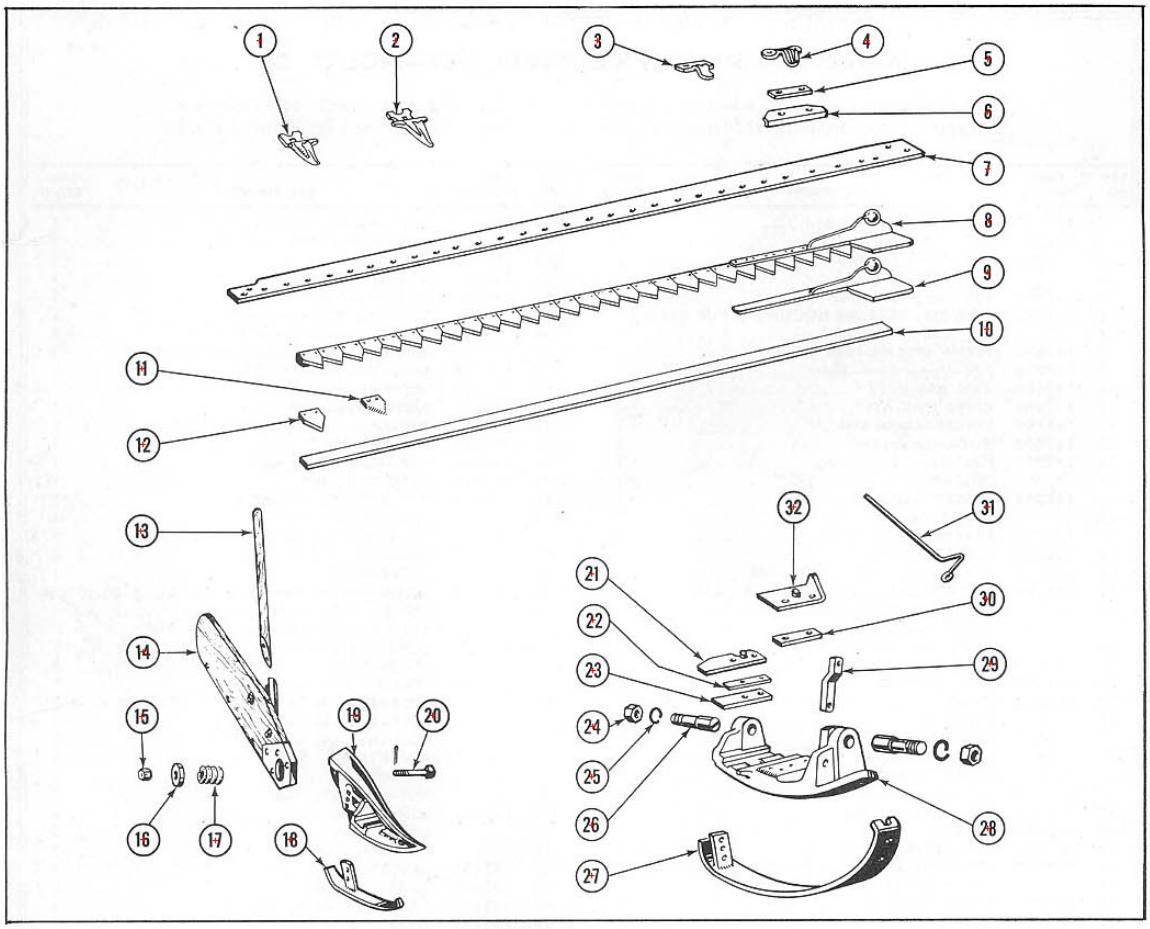
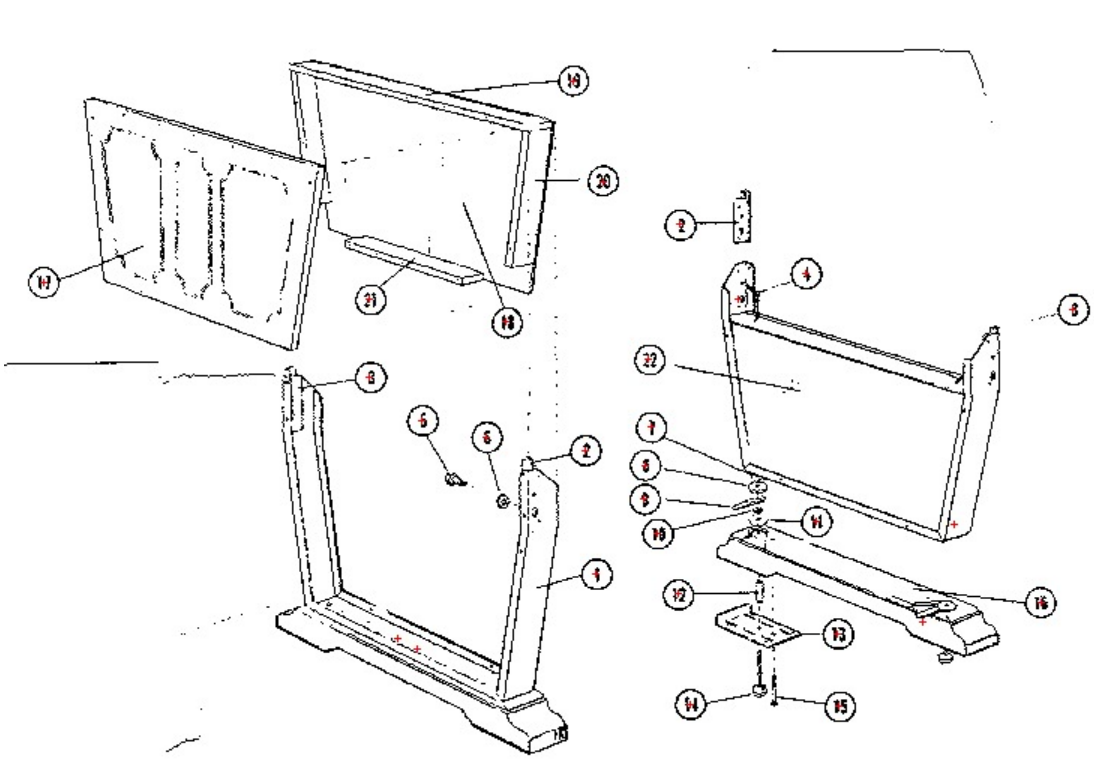


Figure 29

radius = 13.5
center threshold = 8

Thanks to well-defined circles and the lack of many/any other circular shapes of the same radius, the program was able to successfully mark only the 32 circles in the parts diagram.

6. image6.pgm



radius = 9.5
center threshold = 4

The provided image was not of high resolution, and so the circles depicted are fairly pixelated and not perfectly round. The lack of resolution meant that we had to use a fairly low center threshold for this diagram, and the program unfortunately could not label all the circles without including 5 extra markings.