

HW2: Finding Callouts in Part Diagrams

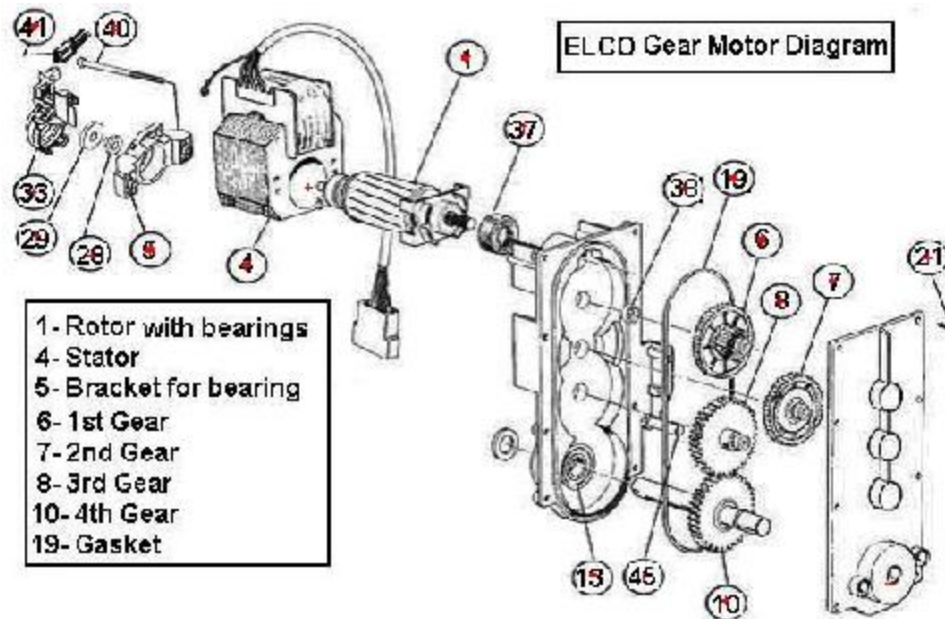


Diagram Image with Callouts Marked in Red

What to Do

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

1. input a floating point radius (ie. 11.6 for this diagram).
2. try radii in a small range around it (ie. 11.0 to 12.2)
3. for each pixel of the input image
 - a. use the Sobel operator to find horizontal and vertical gradients
 - b. if the gradient magnitude is above a threshold
 - compute gradient angle
$$r0 = i - R * \sin(\text{angle});$$
$$c0 = j + R * \cos(\text{angle});$$
 - i is the row, j is the column, and R is the radius
 - use it to vote for the center $(r0,c0)$ of the circle
4. after all pixels have voted, find peaks (using the center threshold) in the accumulator array
5. do some merging of adjacent bins that really represent the same circle
6. use the provided routine **drawcenters** to mark the circles on the output

Turn In (by 11:59pm October 14)

1. your code for finding circles, inserted into the skeleton code and, as before, well commented, so that the grader can compile them to working binaries.
2. your report including:
 - Very brief description of the problem being solved
 - Details of your solution, including the Hough Transform, the data structure used, and, in particular, how you did the center clustering
 - Results shown on the images and discussed in words
 - What went right, what went wrong, etc.
 - Instructions on how to run the program.

Logistics

- Contents of hw2.zip package:
 - **images** : folder containing 6 test images
 - **utils.cpp, utils.h** : methods for reading/writing images
 - **findcirc.cpp** : skeleton codes for you to complete
 - **CMakeLists.txt** : Build rules for CMake

Evaluation: 10 pts

Working Program: 4 points

Circle Finding Quality: 3 points

Center Clustering: 2 points

Report: 1 point