**Problem #1: Sobel Edge Detection (30%)**
See Problem 10.10 (a) and (b) of the textbook (Gonzalez and Woods)

**Problem #2: Application of Hough Transform (30%)**
See Problem 10.15 of the textbook (Gonzalez and Woods)

**Problem #3: Programming -- edge linking (40%)**

Download the color image ‘rubberband cap.png’ from the course web site. Its resolution is 661 pixels x 471 pixels.

We want to extract the outer boundary of each object contained in this image. Such boundary information is useful in many applications, such as object recognition and mensuration.

First, use edge() function to obtain the edge map. Then use imview() function to manually locate one pixel located on the outer boundary of each object. For example, (x=145, y=158) specifies one boundary pixel of the upper left rubberband object.

Next, apply any technique covered in the course to trace the boundary edge pixels. Try to get a boundary as complete and accurate as possible. Plot the final boundary you are able to detect and trace for each object. Discuss the success, failure, and possible ways for improvement.

Note boundary tracing is a non-trivial problem, involving many complicated issues such as noise, broken boundaries, ambiguous locations and directions, among others. Thus, don’t worry if you are not able to get a ‘perfect’ result – any best-effort results will be considered.

Do not use the ‘bwtraceboundary()’ function provided by Matlab image processing library. The idea is for you to get familiarity of the algorithm through the experiments in this homework.