Introduction to Robotics - CMSC 479/679
Homework #2
Due Wednesday, April 11^{th} at the start of class

1. Kinematics:

(A) **679 students only**: Consider the following claim: *It is possible to build an omnidirectional robot using only two powered Swedish wheels (and possibly some number of unpowered spherical wheels to ensure stability).* Prove this claim to be either true or false. The proof must be rigorous and mathematical.

For the robot to be omnidirectional it must be the case that any desired motion vector in the robot's frame of reference can be achieved by some combination of wheel rotational velocities. Said differently, the equations for inverse kinematics can always be solved for wheel velocities regardless of the desired robot velocity. To prove the claim true, it suffices to find one physical arrangement of wheels resulting in omnidirectionality. To prove the claim false there must be no physical arrangement of wheels resulting in omnidirectionality.

(B) **479 students only:** Design a mobile robot with exactly two Swedish wheels. You may assume, but need not include in your analysis, any number of additional unpowered spherical wheels to ensure stability. If the wheel planes are parallel, you may not use $\gamma = 0$ or $\gamma = \pi/2$. Provide the following:

- All parameters for both wheels (i.e., α, β, l, and γ)
- Equations for forward kinematics (expressing robot velocity as a function of wheel rotational velocities)
- Equations for inverse kinematics (expressing wheel rotational velocities as a function of robot velocity)

2. Perception (all students): Suppose you're given the task of building a robot that will take and fill food orders from students and staff in the ITE building by navigating to the Commons, buying the requested items, and returning to the person who placed the order. Write a short paper (between two and four pages long) about the sensor suite you would choose for this robot. Chapter 4 of the textbook is a good starting point, but feel free to read and reference additional sources. Explain the advantages of the sensing suite you choose. Be sure to specify for each sensor in your suite:

- The type of sensor
- Where it will be located and oriented on your robot
- What type of information the sensor will return
- What part(s) of the task will make use of this information and how

Briefly describe other aspects of your robot (e.g., size, shape, manipulators, drive system) to place your sensor choices in context.

3. Depth from focus (all students): This question asks you to experiment with the idea of passive ranging using depth from focus. Consider the two equations below:

$$s_1 = \sum_{u,v} |I(x+u, y+v) - I(x+u-1, y+v)|$$

$$s_2 = \sum_{u,v} (I(x+u, y+v) - I(x+u-2, y+v-2))^2$$

I is an image, and $I(x, y)$ is the image intensity (e.g., grayscale value) at pixel x, y. Note that s_1 and s_2 measure the "sharpness" of an image around a pixel. Points with high sharpness are in focus, and points with low sharpness are out of focus.

Consider the following image, which is available on the web in both color JPG and grayscale PGM formats here:

- http://www.csee.umbc.edu/courses/undergraduate/479/focus.jpg
- http://www.csee.umbc.edu/courses/undergraduate/479/focus.pgm

You task is to use sharpness to try to separate the foreground, in focus objects, from the background, out of focus, objects. It is probably easiest to use the PGM format, which can be displayed by a wide array of image viewing software. The PGM file contains the following header lines:

```
P5
#LEAD Technologies Inc. V1.01
800 600
255
```

They are, respectively, the PGM format identifier, a comment, the width and height of the image, and the maximum grayscale value. Following that is $800 * 600$ bytes that are the grayscale values. You should write a simple program to read in this data, compute a sharpness image from it, threshold the sharpness, and write out another PGM image (using the same header is OK) with byte values of 255 for those pixels that are above threshold and 0 for those pixels that are below threshold. (You don't have to use these grayscale values, just be sure to use two values that are obviously different.)

Submit your code, the best output image you were able to obtain (i.e., the one that best separates the foreground from the background), and a brief writeup of what you did, if anything, to improve on the results of a simple application of either equation 4.21 or 4.22. (Note: This suggests that you *should* try a few things to improve on your original results.)