The homework is generally split into programming exercises and written exercises. Target due dates will be given for each homework, however you can submit all your homework, without penalty, at the end of the semester. You are encouraged to keep up with the target due dates. You should turn in an electronic copy of your solutions to the homework. Please submit your homework on the course website. You are responsible for submitting clear, organized answers to the questions. Please include all relevant information for a question, including text response, equations, figures, graphs, etc. Please pay attention to the discussion board for relevant information regarding updates, tips, and policy changes.

1 WrittEn ExERCises

1. What is the silhouette of a sphere of radius \( r \) with its center at coordinates \((X_0, 0, Z_0)\) under perspective projection on a planar surface? The perpendicular distance from the pinhole to the image plane is \( f \).

2. Consider the square in Figure 1 viewed under perspective projection.

(a) Solve for the vanishing point for each pair of parallel lines.
(b) Write down an expression for the line at the horizon.

Figure 1: A square viewed under perspective projection.
4. MaSKS Exercise 2.7 (Rotation as a rigid-body motion), p. 38
5. MaSKS Exercise 5.19 (Two physically plausible solutions for the homography decomposition), p. 163. Parts (d) and (e) are optional (for extra credit).

## 2 Programming Exercises

1. **2D Projective Transformations.**
   - (a) Implement MaSKS Algorithm 5.2 (The four-point algorithm for a planar scene) or HZ Algorithm 4.1.
   - (b) Use the four-point algorithm with \( n \geq 4 \) hand-clicked correspondences to remove the projective distortion from three images: `facade.gif`, `floor.gif` and one image of your own choice.

   ![Figure 2: floor.png and facade.png](image)

2. **Projection Matrices and Image Formation.**
   - (a) Write a function to produce a \( 3 \times 4 \) (general) projection matrix \( \Pi \) as defined in MaSKS on p. 56. The inputs to your function should be the intrinsic calibration parameters (i.e. the entries of the \( K \) matrix) and a rigid body motion \( g = (R, T) \in SE(3) \).
   - (b) Write a function to display a planar grid of points under perspective projection. The inputs to your program should be the size of the grid, its depth (the \( Z \) coordinates of the points), and a projection matrix \( \Pi \). Demonstrate your program for an \( 11 \times 11 \) grid of points for three different choices of \( \Pi \). For example, pretend the grid is painted on the wall and your job is to show what it looks like to several observers of varying heights standing roughly in front of it.

3. **Edge detection and Hough Transform for Line Detection.** In this exercise, you will implement your own line detector. You will run your detector on two images, `square.png` and `building.png`. Your algorithm will proceed as follows:
   - (a) Load the image
   - (b) Run a \( 3 \times 3 \) Sobel filter in along the horizontal and vertical axes.
(c) Implement the Hough Transform for line detection as described in Szeliski §4.3.2. How do your results vary with respect to different tuning parameters?

(d) Display the results of your Sobel filter and line detections.

(e) Solve for the vanishing points of square.png and compare the result with your solution of question 2 of the written exercises.

(f) Can you find vanishing points for the lines in building.png?