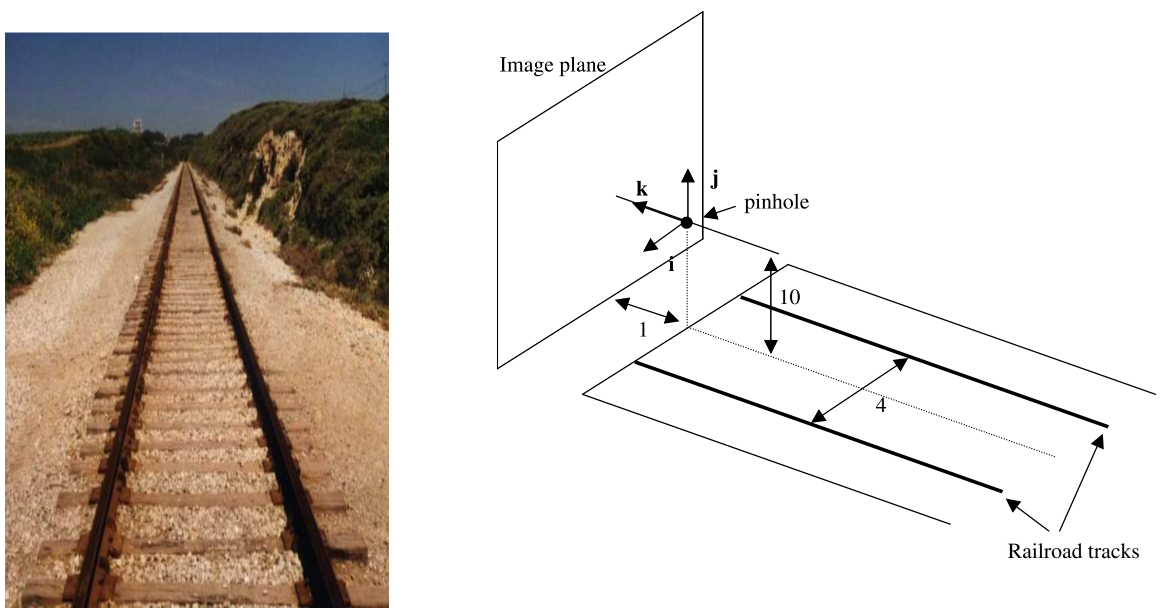
Homework Set One
ECE 161
Department of Computer and Electrical Engineering
University of California, San Diego

Nuno Vasconcelos Spring 2009

Due April 16, 2009

1. You are probably familiar with the *railroad track illusion*, depicted in Figure 1 a): if you stand in the middle of a railroad track and look towards the horizon, it appears that the two tracks converge to a single point. In terms of the perspective projection model, the situation can be modeled as in Figure 1 b): the tracks are two parallel lines perpendicular to the image plane, and the plane that contains the lines has a negative value along the **j** axis. This value is the negative of your height but, to make things simpler, we will assume that it is normalized to 10. We will also assume that the distance from the pinhole to the image plane is $f = 1$, and that the distance between the two tracks is 4.



a) b)

Figure 1: Railroad tracks

a) write down the equations of the 3-D lines associated with the two tracks.

b) write down the equations of the corresponding projection on the image plane.

c) produce a plot that shows, for each 3-D line, the projection of the line segment that starts at a distance of 1 from the pinhole and ends at infinity.

d) does the plot produced in **c)** look like the image in Figure 1 a)? If not, explain the differences?

e) one item that we did not account for in our model were the bars perpendicular to the tracks that always exist in railroads. These bars do not appear to be distorted, in the sense that they are parallel to the **i** axis in the real world and they generate image lines that are parallel, to the lower edge of the image. Why is it that, unlike the tracks themselves, there appears to be no distortion of the parallelism between the lines perpendicular to them?

2. (MATLAB) In this problem we will see how the concepts that we have learned so far, already allow us to perform what you might consider sophisticated computer graphics operations. In particular, we will see how to do *texturemapping* and add a little of lighting to render a realistic artificial image. The idea is the following. We start by collecting a sample of texture in a canonical position. Then, we go back to the studio, create a 3-D scene, and map the texture we collected onto that 3-D scene. We will consider here the simple example illustrated by Figure 2.

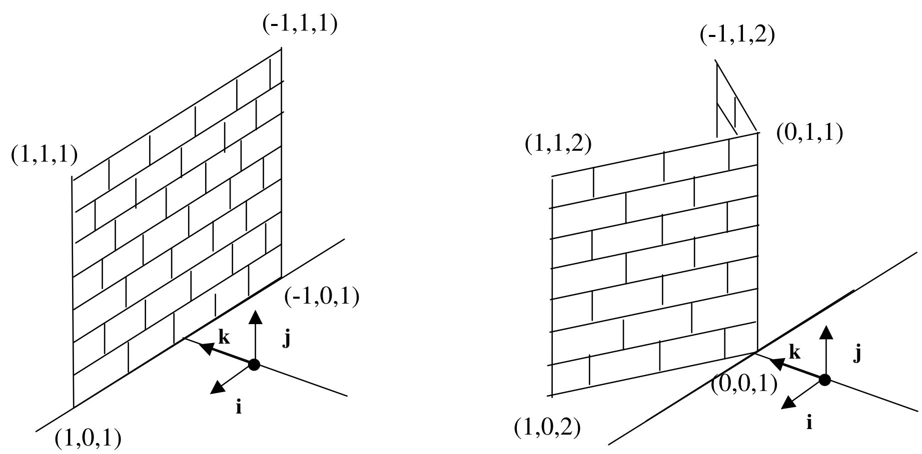


Figure 2: Texture mapping

We collected a sample of texture from a real brick-wall, with the imaging set-up shown on the picture in the left. The goal is to map the texture into the brick-wall shown in the right picture. In the both pictures, we show the coordinates of the 3D points required to completely specify the scene (note that we make Z positive in the direction away from the image plane, which is equivalent to using the virtual image plane that we discussed in class). In both cases, we used $f = 128$ and the texture sample that was collected is provided in the file `brickwall.jpg`. In all that follows let (X, Y, Z) be a 3D point in the original scene, (x, y) the corresponding image point, (X', Y', Z') a point in the synthetic scene and (x', y') the corresponding image point.

a) write down the equations for the wall plane in the original image (i.e. $Z = f(X, Y)$) and the two planes that make up the new wall in the synthetic image (i.e. $Z' = f(X', Y')$).

b) write down the equations for the mapping from scene to image coordinates for each of the scenes. Invert each of these in order to obtain scene coordinates as a function of image coordinates (note that you must consider two cases in the synthetic scene).

c) if (X, Y, Z) is mapped into (X', Y', Z'), then (x, y) is mapped into (x', y'). Use this observation and **b)** to obtain the equations for the mapping $(x, y) = f(x', y')$ (once again you will have to consider the two planes in the artificial scene separately).

d) on MATLAB, read the file `brickwall.jpg`. Map it into the the synthetic scene (apply one of the mappings derived in **c)** to each half of the image), with the function `interp2`. Hand in a plot of both the original and the synthetic scene. (Note: matlab code for implementing the parts of this question which are not related to the understanding of the material is available from the course web page).

e) the scene obtained so far is correct from a geometric point of view, but does not "feel" right. The reason is that we did not add any lighting yet. For this we are going to assume that the wall is a Lambertian surface with constant albedo. We will also assume that lighting is due to a point source at infinity. Show that, while it is impossible to recover the value of the radiance, it is possible to obtain the ratio between the radiances of the wall when illuminated from two different orientations $\mathbf{s}_1$ and $\mathbf{s}_2$.

f) we next assume that the texture sample was taken with a source that is aligned with the wall normal, i.e. $\mathbf{s} = (0, 0, 1)$. Using **e)**, determine the radiance for each of the facets of the synthetic wall, when the synthetic scene is illuminated with a source of direction $\mathbf{s}' = \alpha \mathbf{n}_1 + (1 - \alpha)\mathbf{n}_2$ where $\mathbf{n}_1$ is the normal to the plane in the region $X \geq 0$ and $\mathbf{n}_2$ the normal to the plane in the region $X < 0$. Repeat for $\alpha \in \{0, 0.1, 0.2, 0.5, 0.8, 0.9, 1\}$ and hand in the plots of the synthetic image for each case.