Zoran Popović

Homework 1

Problem 1

(5 points) If you walk with a pinhole camera toward an object, the image (projection) of the object will get bigger. How (exactly) would you adjust the focal length (*f*) to keep the object the same size if you halve the distance to the object? Will objects that are further away appear bigger or smaller than they did before (i.e., before walking toward the front object and adjusting the focal length)?

Problem 2 (16 points)

Each of the "matrices" below is actually a convolution filter for image processing. We have also included a median filter. Write all letters (***if any***) below that describe the behavior of the filter as given. Note that when a description says a filter applies in a given direction, it is understood that it applies *only* in that direction. Possible descriptions include:

A. mean blurring in all directions
B. mean blurring in x direction
C. mean blurring in y direction
D. mean blurring along $y = x$
E. mean blurring along $y = -x$
F. Laplacian
G. gradient in x
H. gradient in y
I. gradient along $y = x$
J. gradient along $y = -x$
K. identity (no effect)
L. shift in x
M. shift in y
P. sharpen in all directions
Q. sharpen in x
R. sharpen in y
S. sharpen along $y = x$
T. sharpen along $y = -x$

(a) $\frac{1}{5}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Description(s): _____

(b) $\frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

Description(s): _____

(c) $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Description(s): _____

(d) $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Description(s): _____

(e) $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

Description(s): _____

(f) $\frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

Description(s): _____

(g) Median filter over 40x40 region

Description(s): _____

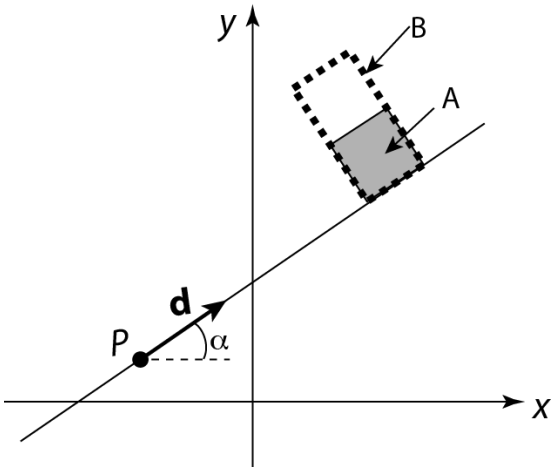
(h) $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Description(s): _____

Problem 3 (10 points)

Consider a line in 2D, $L(t) = P + t\,\mathbf{d}$, passing through point P and along direction **d**, where $t \in (-\infty,\infty)$. Direction **d** makes an angle $\alpha \in [0, 2\pi]$ with the x-axis.

In this problem, you will compute the sequence of transformations that will perform a scale by a factor of 2 in the direction perpendicular to the line. The line, an original unit square (solid filled), the scaled version of the square (now a rectangle, shown as a dotted outline) are shown below to the left. The parameters of the line and the available transformations are shown to the right:



$$P = \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix} \qquad S(a,b) = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{d} = \begin{pmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{pmatrix} \qquad R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T(c,d) = \begin{pmatrix} 1 & 0 & c \\ 0 & 1 & d \\ 0 & 0 & 1 \end{pmatrix}$$

Determine a matrix product that will perform this transformation (from A to B). You do ***not*** need to write out the 3 x 3 matrices. Just write their symbolic references and the arguments they take. You do ***not*** need to justify your answer, but you may do so to help earn partial credit.

Problem 4 (16 points)

Consider the Blinn-Phong shading equation for a single point light source:

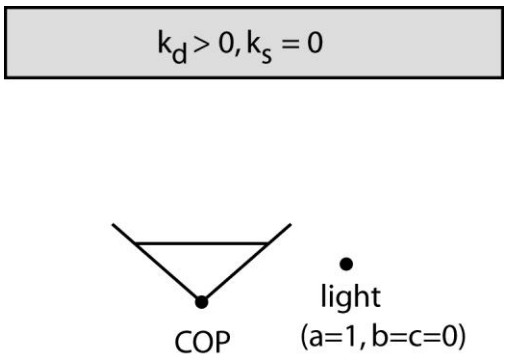
$$I = f(d) I_L B\left[k_d (\mathbf{N} \cdot \mathbf{L}) + k_s (\mathbf{N} \cdot \mathbf{H})_+^{n_s} \right]$$

where $f(d)$ is light distance attenuation defined by:

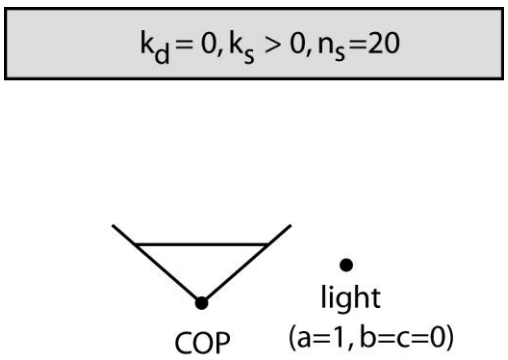
$$f(d) = \frac{1}{a + bd + cd^2}$$

In each of the sub-problems below, you may assume a monochrome world (one color channel corresponding to intensity or brightness).

(a) (3 points) In the figure below, a point light source illuminates a gray rectangular surface, and some of the light reflects to the viewer at the center of projection (COP) shown. The field of view of the viewer is indicated by the angled lines emanating from the COP and passing through the edges of the image being recorded, shown as a horizontal line. Assume the surface is diffuse (k_d>0, k_s=0) and the light exhibits no distance falloff (a=1, c=b=0). Which point on the surface would appear the brightest to the viewer? Mark that point with a dot (make sure we can see it!) and draw a line from the light to that point and a line from that point to the COP. **Justify your choice for the brightest point in the margin to the right**.

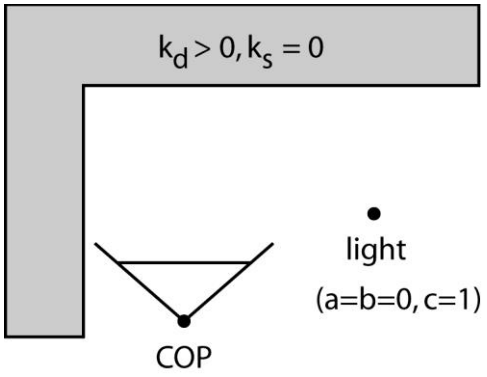


(b) (3 points) Now assume the surface is specular only (k_d =0, k_s >0, n_s =20) and the light exhibits no distance falloff (a=1, c=b=0). As before, indicate the brightest point on the surface as seen by the viewer, indicate the path from the point light source to the viewer, and **justify your answer in the space to the right**.

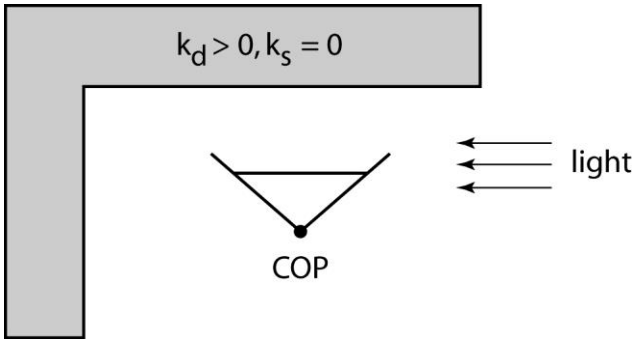


Problem 4 (cont'd)

(c) (3 points) Now consider the L-shaped surface below. Assume the surface is purely diffuse (k_d>0, k_s=0) and the light exhibits a *quadratic* falloff (a=b=0, c=1). As before, indicate the brightest point on the surface as seen by the viewer, indicate the path from the point light source to the viewer, and **justify your answer in the space to the right**. You should ***assume a local illumination model*** (i.e., rays bounce from the light to the surface to the viewer but don't bounce around within the scene).



(d) (3 points) Suppose we move the viewpoint so that it can only see the front-facing wall, and we now illuminate the scene from the right side with a directional light source as shown. As in the previous problem, assume the surface is purely diffuse (k_d>0, k_s=0). Now, ***using a Whitted ray tracer***, describe what you would expect the image to look like, i.e., describe how the brightness would vary across the image, if at all. Assume the surfaces are opaque, and the reflection coefficient is set to k_r=k_s. **Justify your answer in the space to the right**.

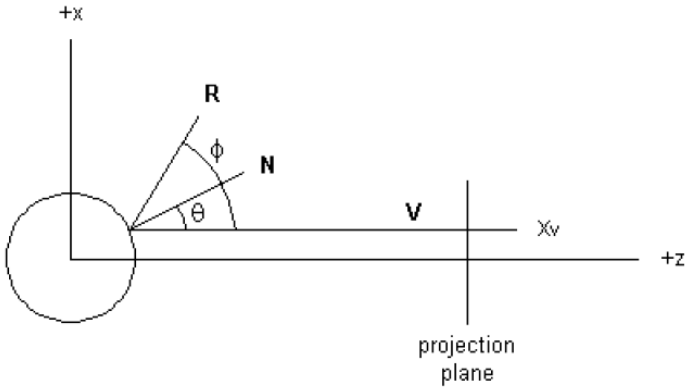


(e) (4 points) Given the scene from (d), how could you modify the ray tracer to render a more realistic image? How would the brightness would vary across the image, if at all? **Justify your answer**.

Problem 5. Environment mapping (20 points)

One method of environment mapping (reflection mapping) involves using a "gazing ball" to capture an image of the surroundings. The idea is to place a chrome sphere in a real environment, take a photograph of the sphere, and use the resulting image as an environment map. Each pixel that "sees" a point on the chrome sphere maps to a ray with direction determined by the reflection through the chrome sphere; the pixel records the color of a point in the surroundings along that reflected direction. You can turn this around and construct a lookup table that maps each reflection direction to a color. This table is the environment map, sometimes called a reflection map.

Let's examine this in two dimensions, using a "gazing circle" to capture the environment around a point. Below is a diagram of the setup. In order to keep the intersection and angle calculations simple, we will assume that each viewing ray **V** that is cast through the projection plane to the gazing circle is parallel to the z-axis. The circle is of radius 1, centered at the origin.



a) (5 points) If the x-coordinate of the view ray is x_v, what are expressions of the (x,z) coordinates of the point at which the ray intersects the circle? What is the unit normal vector **N** at this point?

b) (3 points) What is the angle between the view ray **V** and the normal **N** as a function of x_v?

c) (5 points) Note that the angle φ between the view ray **V** and the reflection direction **R** is equal to 2θ, where θ is the angle between **V** and the normal **N**. Sketch a plot for φ versus x_v. In what regions of the image do small changes in the x_v coordinate result in large changes in the reflection direction?

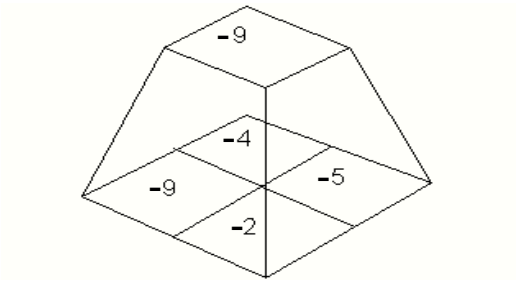
d) (4 points) We can now treat the photograph of the chrome circle as an environment map (for a 2D world). When ray tracing a new chrome object, we compute the mirror reflection direction when a ray intersects the object, and then just look up the color from the environment map. (If the computed reflection direction lands between directions stored in the environment map, then you can use bilinear interpolation to get the desired color.) Would we expect to get exactly the same rendering as if we had placed the object into the original environment we photographed? Why or why not? In answering the question, you can neglect viewing rays that do not hit the object, assume that the new object is not itself a chrome circle, and assume that the original environment is some finite distance from the chrome circle that was originally photographed.

e) (3 points) Suppose you lightly sanded the chrome circle before photographing it, so that the surface was just a little rough.

- What would the photograph of the circle look like now, compared to how it looked before roughening its surface?
- If you used this image as an environment map around an object, what kind of material would the object seem to made of?
- If you did not want to actually roughen the object, what kind of image filter might you apply to the image of the original chrome circle to approximate this effect?

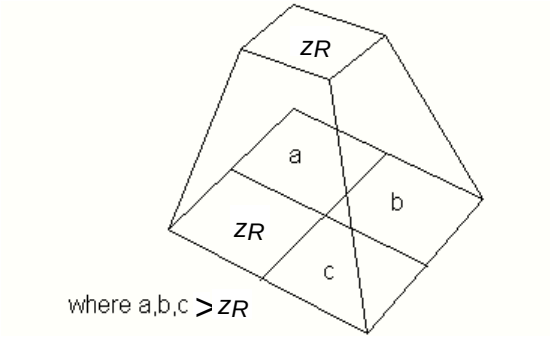
Problem 6. Z-buffer (14 points)

The z-buffer algorithm can be improved by using an image space "z-pyramid." The basic idea of the z-pyramid is to use the original z-buffer as the finest level in the pyramid, and then combine four *z*-values at each level into one *z*-value at the next coarser level by choosing the farthest (most negative) z from the observer. Every entry in the pyramid therefore represents the farthest (most negative) *z* for a square area of the z-buffer. In this problem, assume the image is always square with side length that is a power of 2. (Handling non-square, non-powers-of-2 images is a simple generalization of this.) Before each primitive is rendered, the z-pyramid is updated to reflect the current state of the z-buffer. When you have a new primitive to draw, you are then testing against the current, up-to-date z-pyramid. A z-pyramid for a single 2x2 image is shown below:



a) (3 points) At the coarsest level of the z-pyramid there is just a single *z* value. What does that *z* value represent?

Suppose we wish to test the visibility of a triangle **T**. Let z_T be the nearest *z* value of triangle **T**. **R** is a region on the screen that encloses the triangle **T**, and is the smallest region of the z-pyramid that does so. Let z_R be the *z* value that is associated with region **R** in the z-pyramid.



b) (3 points) What can we conclude if $z_R < z_T$?

c) (3 points) What can we conclude if $z_T < z_R$?

If the visibility test is inconclusive, then the algorithm applies the same test recursively: it goes to the next finer level of the pyramid, where the region **R** is divided into four quadrants, and attempts to prove that triangle **T** is hidden in each of the quadrants of **R** that **T** intersects. Since it is expensive to compute the closest *z* value of **T** within each quadrant, the algorithm just uses the same z_T (the nearest *z* of the entire triangle) in making the comparison in every quadrant. If, at the bottom of the pyramid, the test is still inconclusive, the algorithm resorts to ordinary z-buffered rasterization to resolve visibility.

d) (5 points) Suppose that, instead of using the above algorithm, we decided to go to the expense of computing the closest *z* value of **T** within each quadrant. Finding the closest value amounts to clipping the triangle to each region and analytically solving for the closest z. This approach also applies to the finest level of the pyramid, where the pixels are abutting square regions. Would it then be possible to always make a definitive conclusion about the visibility of **T** within each pixel, without resorting to rasterization (effectively intersecting the viewing ray with the triangle)? Why or why not?