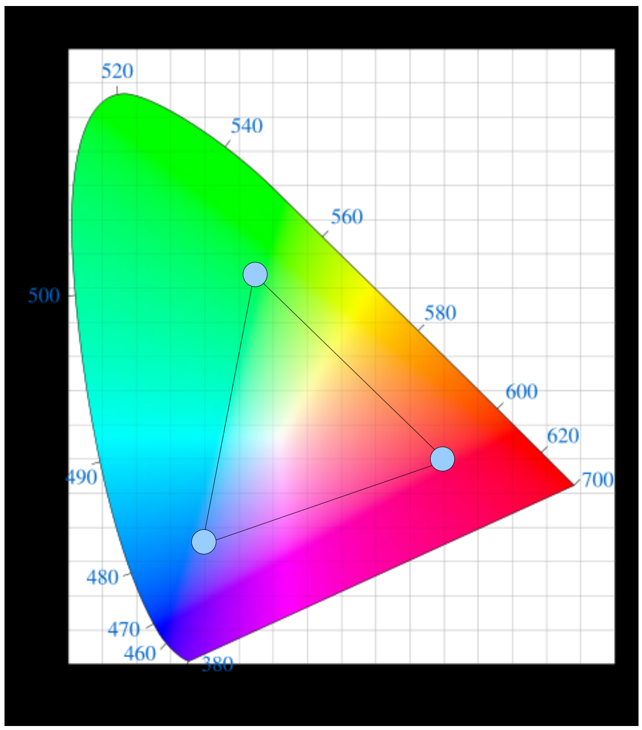
CPSC 314 – Assignment 3 Answers

April 2008

1. Colour Representation

a) Sketch a typical RGB monitor gamut with R, G, B locations:

The exact locations in the R, G, B regions varies but typical monitors have phosphors in each of the three regions and visible colours are generated by mixing these three light sources. As a result, only colours within the triangular region can be reproduced.



b) What is the dominant wavelength of the colour D on the chromaticity diagram? What spectral colour is complementary to D?

This can be found by drawing a line from D to the point C. The intersection of the line with the outer edge of the diagram on either side gives the dominant wavelength (on the closest side) and complementary wavelength (other side). Light with the dominant wavelength is perceived as having a hue identical to the colour represented by the point D.

The dominant wavelength is approximately 575 nm. Blue is complementary to the yellow-orange region (wavelength ~475 nm)

c) Compute and illustrate where the colour (X, Y, Z) = (10, 20, 10) would be located on the CIE chromaticity diagram:

$x = X / (X + Y + Z) = 10/40 = 0.25$

$y = Y / (X + Y + Z) = 20/40 = 0.5$

$z = 1 - x - y = 0.25$

This colour is located at (0.25, 0.5).

d) Visible light consists of a continuous spectrum in the range 400-700 nm. Describe why we can represent colour, as perceived by humans, using only 3 values (RGB, XYZ). Is this true for animals?

Colour perception is based on the response of red, green, and blue cones in the human visual system. Each are responsive to a range of wavelengths and the level of response of the three are combined to generate a single continuous range of perceivable colours. We can therefore generate colours simply by using light suited to each cone type.

Animals can have different numbers of cones sensitive to different wavelengths of light and some cannot see colour at all.

e) Yellow is a spectral colour and cannot be reproduced by mixing other colours. True or false?

Yellow (wavelength ~580 nm) can be produced perceptually by mixing different light sources. Monitors do this by combining red and green light sources. The physical light and overall amount of energy required may vary but the colour itself appears identical. It's the ratios of response between S, M, and L cones that determine the colour that is perceived.

f) What are metamers?

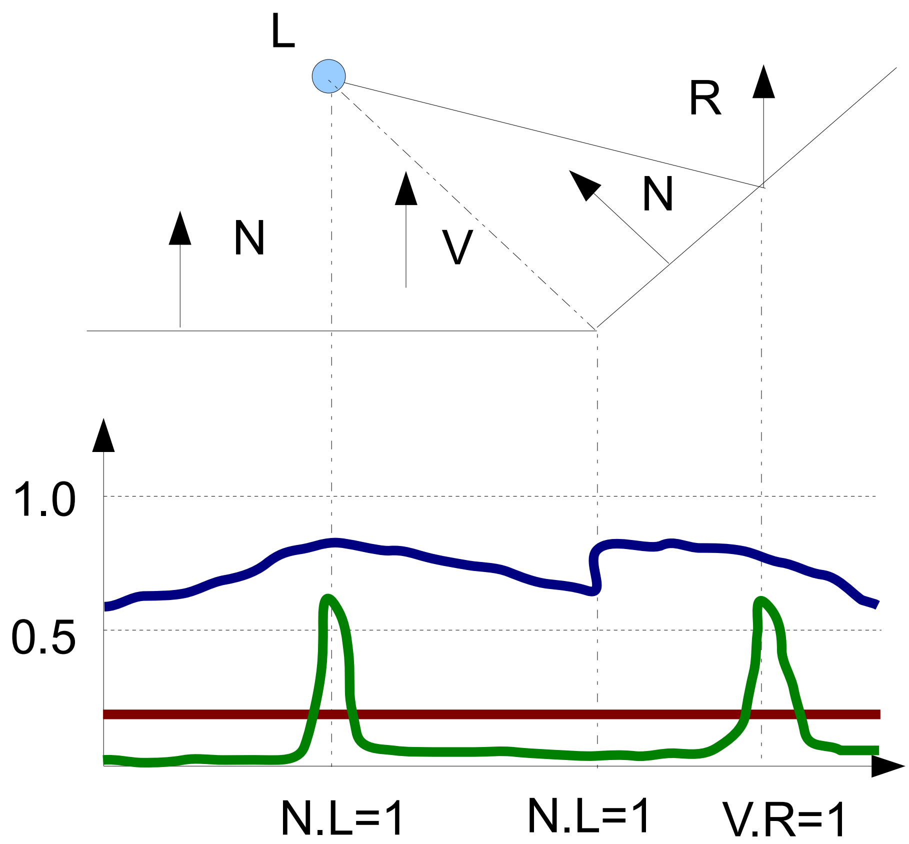
Metamers are perceptually identical colours with different spectral components.

g) It can be difficult to print a colour image on paper that looks just like the same colour image on the screen. Why?

Printed images are viewed under different lighting conditions. Printer and monitor gamuts are different. Note that it is not necessarily true that printed colours are a strict subset of monitor colours.

2. Local Illumination

The diagram should look (roughly) like this. Note the direction of the viewing vector:



Solving for key points shown above (N parallel to L, R parallel to V, etc.) gives a good idea of the shape of each illumination component:

Ambient = 0.2
Diffuse = 0.8 * (N . L)
Specular = 0.7(R . V)^{200}

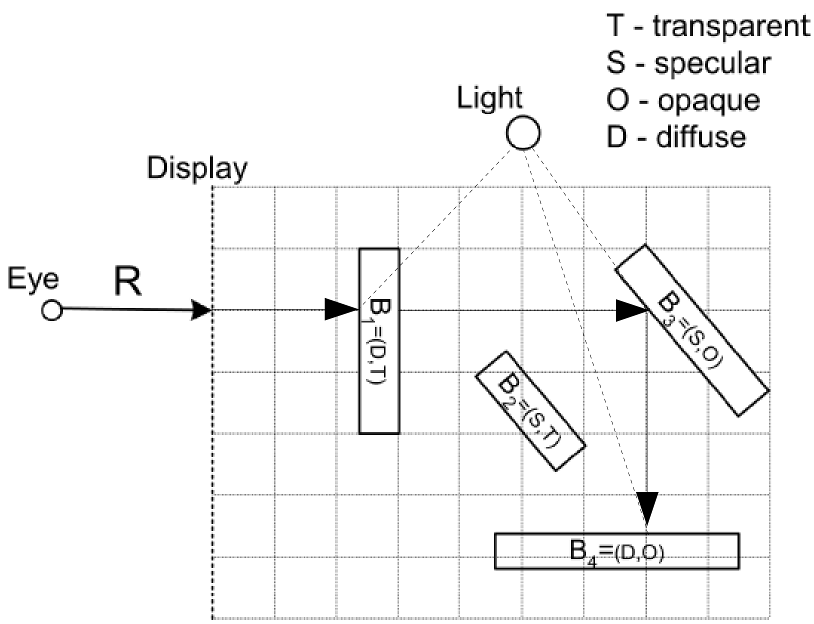
Observe that the dot product of two parallel (normalized) vectors is equal to *cos*(0) = 1. The dot product of two perpendicular vectors is equal to *cos*(pi/2) = 0. This gives us some key points in the graph while the rest can be filled in based on the scaling of the functions and the knowledge that they are periodic, sinusoidal functions (except for the constant ambient term). The specular component rises sharply to two peaks at 0.7 because the function itself is raised to a higher power.

For the total, add up the specular, diffuse, and ambient components. The total may be capped at 1.

Note: an interpretation where the viewing vector was "coming out of the page" is acceptable and results in a specular component of 0.

3. Ray-Tracing:

a) Trace the ray R from the eye to B_1. At this point a shadow ray is cast to the light source. Ray continues straight through B_1 since it is transparent (Snell's Law applies but R is normal to B_1). Continue the ray to B_3 and cast another shadow ray to the light source. B_3 is perfectly specular so the ray is reflected down to B_4.



b) What is the colour returned by the ray tracing algorithm for ray R?

This can be calculated recursively by determining the illumination of the first surface and then combining it with reflected light and transmitted light weighted by the level of specularity and transparency of the surface. This approach yields:

Ray colour = (B_1 local illumination) + 0.5(transmitted ray)
= (B_1 local illumination) + 0.5(B_4 local illumination)

Note that B_3 is perfectly specular and so reflects all of the incoming light from B_4.

To evaluate, apply the Phong illumination model:

B_1 Local ~ (1,0,0) * (0.5, 0.5, 0.5)(N • L) ~ (0.5, 0, 0) * cos45 ~ (sqrt(2)/4, 0, 0) ~ (0.35,0,0)

B_4 Local = (0, 1, 0) * (N • L) = (0,1,0) * cos(7 / sqrt(53)) ~ (0,1,0).

Therefore,

Ray colour ~ (0.35, 0, 0) + 0.5(0, 1, 0) = (0.35, 0.5, 0)

Note that this assumes local illumination of B_1 since the surface is partly transparent. A solution where the local illumination of B_1 is 0 is also acceptable and in that case the R component of the result would be 0. Also note that it is acceptable not to evaluate the $N \cdot L$ terms, but the solution must clearly indicate which angle is associated with which surface.

4. **Visibility**:

a) For simple back-face culling a polygon is culled iff its normal is facing away from the viewing direction (>90 degrees). f, g, h, c, and d would be culled. a is borderline but a line drawn from the eye point to the a-b vertex shows that it would be slightly visible and would not be culled.

b) There are many possible optimizations for rendering the statue. The simplest way of determining whether or not the statue is in the FOV is to calculate a bounding region of some kind and then figure out whether or not it's inside the viewing frustum. Optimizations such as back-face culling are not really what the question asked for since we want to decide whether or not to even bother rendering the statue.

c) False. The order matters if there are vertices with equal Z value. Multiple behaviours are possible for equal Z-values (in OpenGL you can actually set environment variables to determine this).

d) A scene with a large number of occluded polygons would be more efficiently rendered by a ray-tracing algorithm. In this case light rays reflect off the first surface visible so others do not enter into the ray-tracing calculations. The Z-buffer algorithm would still draw everything.

e) This is false because the Z values for each vertex of a triangle need not be the same. Imagine for example two intersecting triangles in three dimensional space.

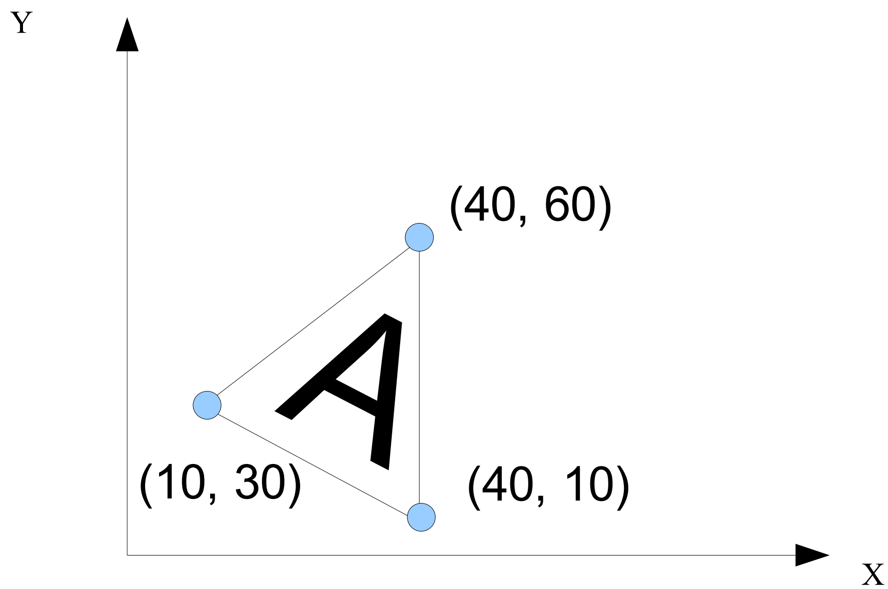
f) A BSP tree recursively subdivides a space using hyperplanes, or lines in this two-dimensional version of the problem. To build the tree, consider the polygons in alphabetical order, extend them to lines bisecting the scene. Remaining polygons on the "+" side will become leaves on one side while other polygons end up on the other.

Traversal consists of an in-order walkthrough of the tree, where nodes farther away from the viewpoint are visited first.

The correct order is: f-b-d-c-d'-e-a-i-j-k-h-g, where d/d' are the line segments subdivided by c (d' is closer to the eye point).

5. Texture Mapping

a) Each pair of calls establishes a correspondence between a vertex in world space and where it is located in texture space. The triangular portion {(0.3,1), (0, 0.5), (0.5, 0.5)} of the ABCD texture is drawn onto the triangle {(40,60,0), (40,10,0), (10,30,0)}:



b) MIP-maps are precomputed sets of textures down-sampled to multiple lower resolutions. They are useful as a rendering optimization because small or distant objects may not need to be rendered with a higher resolution texture. For example, a polygon taking up only an 8x8 region of the screen does not need 64x64 texturing.

c) This question is about the distinction between linear interpolation in screen space (affine texture mapping that does not account for perspective foreshortening) and perspective correct texture mapping done in three dimensional space. Both are useful in different situations -- linear interpolation is fast and can look decent with polygon subdivision while perspective correct mapping generates the desired results for the general polygon case.