Chapter 10 Solutions

10.1 Suppose that we move across a scanline left to right starting on the outside of a polygon. Assume that 0 corresponds to the background color and 1 is the edge and fill color. As we move across the scanline we replace each point by the exclusive or of its value with the value of the point on the left (which has already been processed). If we start on the outside of any polygon, we can assume the first point is a 0. Thus as long as we remain outside the first polygon we generate $0 \otimes 0 = 0$. When we encounter the first edge, a 1, we compute $1 \otimes 0 = 1$ (that is we have a 0 from the last point XORed with the edge). As we proceed inside the polygon we compute $1 \otimes 0 = 1$ because we have a 1 from the last XOR and a 0 from the unfilled point. When we encounter the second edge, we compute $1 \otimes 1 = 0$, thus returning to our initial situation

10.3 The major problem is that the environmental map is computed without the object in the scene. Thus, all global lighting calculations of which it sould be a part are incorrect. These errors can be most noticable if there are other reflective objects will now not show the reflection of the removed object. Other errors can be caused by the removed object no longer blocking light and by its shadows being missing. Other visual errors can be due to distortions in the mapping of the environment to a simple shape, such as a cube, and to errors in a two stage mapping.

10.5 Consider a single sine wave that is sampled at just faster than the Nyquist rate with the first sample where the sine is zero. The next sample will be slightly greater than zero, the following slightly greater (in magnitude) than the second, and so until we get samples about equal to the maximum of the sine. Subsequent samples will get smaller and smaller until we get to the minumum of the sine, and then the values will slowly increase. This pattern will repeat, each cycle taking multiple periods of the original sine. Visually, this pattern looks like a modulated sine wave. This visual appearence is due to us seeing the energy of the original samples and those of the first replica which appears just over the Nyquist rate. Our eye is a low pass but imperfect filter that lets both pass through. If we add the two sinusoids using the trigonemtric identity we see the sum and difference frequencies which explains this modulation. The higher frequency (the sum) gives the inner frequency while the difference generates the slowly varying envelope. These patterns describe the Moire patterns we see when we move past twofences, one behind the other, when driving in a car.

10.7 Once aliasing occurs, it is very difficult to remove because the data that has been aliased at another frequency is mixed with any unaliased data at that frequency. Thus, if we remove energy at that frequency, we remove both aliased and unaliased information. On the other hand, if we remove data at frequencies above the Nyquist frequency before sampling occurs, although we loose information, we are left with only unaliased data after samling occurs.

10.9 The basic problem is that when we use subtractive colors and filters, the colors modulate each other and there is a multiplicative rather than additive relationship. Thus, if we have two CMY colors. (C_1, M_1, Y_1) and (C_2, M_2, Y_2), the resulting color is (C_1C_2, M_1M_2, Y_1Y_2). OpenGL supports this kind of blending by allowing us to set a source or destination blending factor that is itself a color, Thus if the source blending factor is the destination color and destination blending factor is zero, we can create a filter.

10.13 Suppose that the histogram of the image is a function $f(x)$ where x is the luminance. The lookup table formed from the function $\int_0^x g(x')dx'$ will create an image with a flat histogram. A simple discrete example is illustrative. Suppose we have a image which is 1024 x 1024 and has 256 values of luminance. If the image we want to create is to have a flat histogram, there should be 4096 pixels with each luminace value. If we have the histogram of the original image, we can use this histogram to find which original luminace values we must use to obtain the lowest 4096 values to assign to 0, the next lowest 4096 to assign to 1, and so on. The function that describes this process is the integral of the histogram curve.