Chapter 6

Implementation of a Radiometric Simulation in OpenGL

The purpose of this chapter is to investigate the implementation of a practical infrared simulation in OpenGL. It will address a number of the problems identified in the previous chapters and suggest techniques to increase the accuracy of a simulation.

6.1 Dynamic range of the fog colour (path radiance)

In a typical South African scenario, the temperature of the atmosphere varies between 0°C and 40°C. This is equivalent to radiance levels of between 0.64 and 2.95 $Wm^{-2}sr^{-1}$ for a bandwidth of 3-5 μm. The radiance values and their integer representations are shown in Figure 6.1. If these radiance values are to be represented by integers, the radiance from a source at 0°C to a source at 40°C are rounded to only 3 integers. The path radiance and atmospheric background radiance is therefore a potential source of errors in simulations. In detection systems where the pre-amplifier is AC-coupled to the detector, the DC-offset due to the path radiance and atmospheric background radiance is cancelled. These systems include some imaging systems and single-detector systems that use a reticle to chop the irradiance. If the purpose of the simulation is to simulate a DC-coupled system, such as a single detector radiometer, where the image is integrated and projected to a single point, the background and path radiance can lead to significant errors, especially in cases where the background radiance represents a large percentage of the image.

6.2 Simulation setup

A simulation was set up to investigate the effect of the different image components on the total error. The target object was defined as a sphere with a radius of 1m and a radiance value of 12278 $Wm^{-2}sr^{-1}$, representing a surface temperature of 920°C. The binary representation of 12278 is 10111111110110, giving values in all three the "colours" of a synthetic display resolution of 14 bits. The colour of the fog was set to a *RedGreenBlue* value of *(1,0,0)*, representing a temperature of 11°C in the 3-5 μm band. The density of the fog was set to 1.3645×10^{-4}, the value obtained in Figure 5.11. The irradiance on the sensor was determined for a range of distances and compared with theoretically calculated values. The colour of the fog was

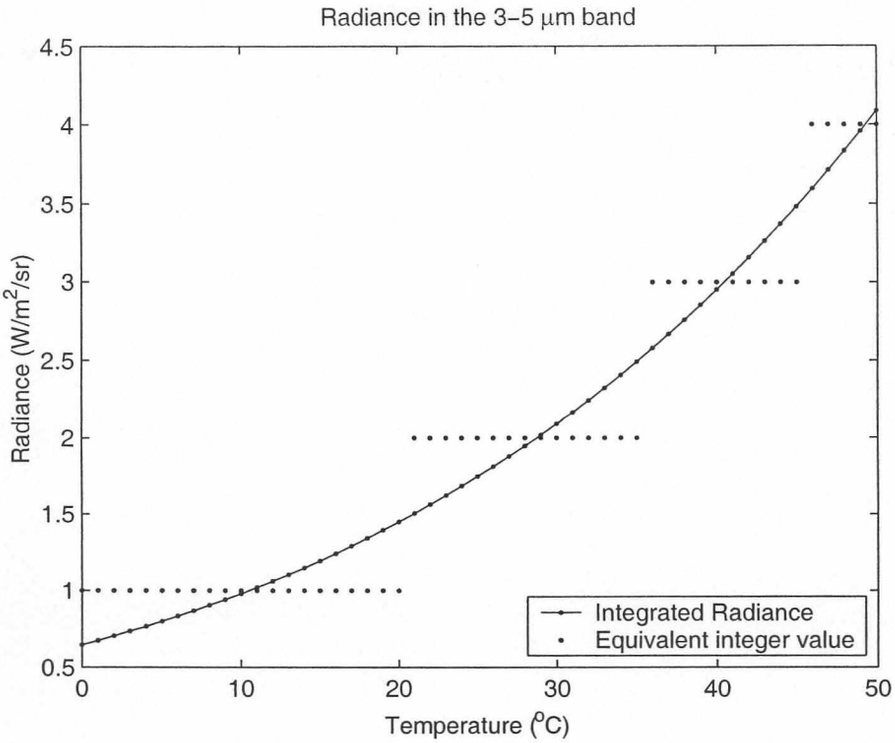

Figure 6.1: Radiance from sources at a range of temperatures

then changed to (*2,0,0*), (*3,0,0*) and (*4,0,0*), equivalent to an atmospheric temperature of 29°C, 41°C and 49°C, and the simulation repeated. The results are shown in Figure 6.2.

The irradiance values were below the calculated values for fog colours of (*1,0,0*) and (*2,0,0*), whereas a fog colour of (*3,0,0*) gave results similar to the predicted values at longer ranges. A fog colour of (*4,0,0*) gave irradiance values above the calculated values at longer distances. At short distances the calculated irradiance is higher than the rendered irradiance in all the cases. This indicates that the fog density factor is too high.

6.3 A simulation with an 8 bit dynamic range

A temperature range of 0°C to 250°C can be represented using an 8-bit radiance value, for radiance integrated from 3 to 5μm. The quantisation errors limit the temperature resolution in the area close to atmospheric temperatures, as mentioned in Section 6.1. The simulation in Section 6.2 was repeated with the source temperature set at 200°C, equivalent to a radiance value of 113 $Wm^{-2}sr^{-1}$. The fog density was set to 1.4410×10^{-4}. The density value was obtained by calculating the effective atmospheric transmittance for a source at 200°C and finding the equivalent extinction coefficient. The results are shown in Figure 6.4 and Figure 6.5. The fog radiance of *(3,0,0)* gave an error of less than 1×10^{-3} Wm^{-2} in irradiance at distances larger than 1000m. The simulation illustrates that 8-bit resolution can be used to simulate infrared scenarios in cases where the radiance levels are below 256 $Wm^{-2}sr^{-1}$. Careful selection of the input parameters are required to obtain accurate results.

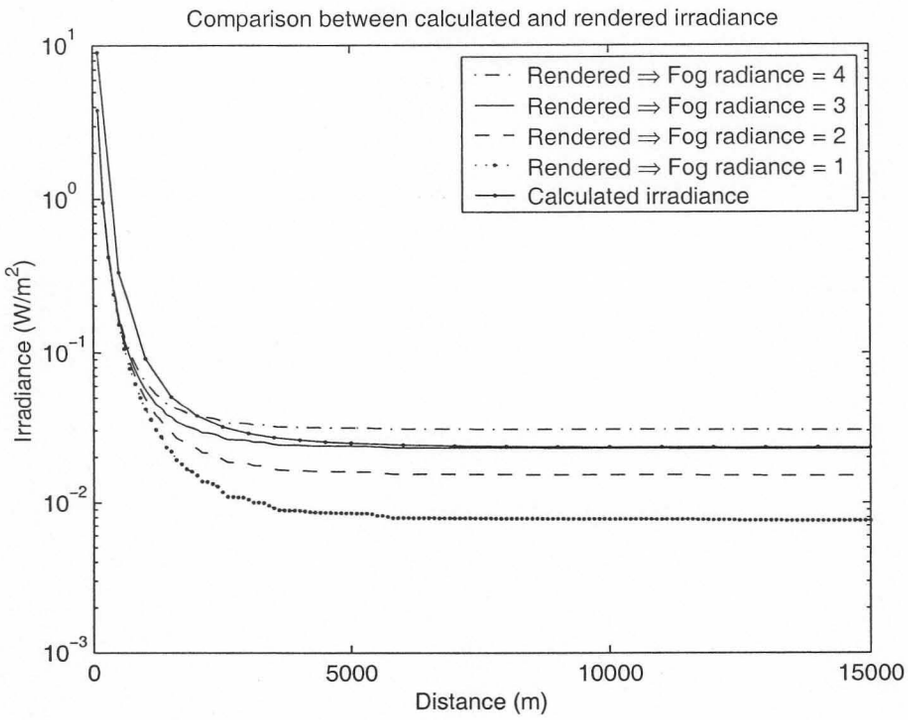

Figure 6.2: Irradiance from scenario with different fog radiances - 14bit

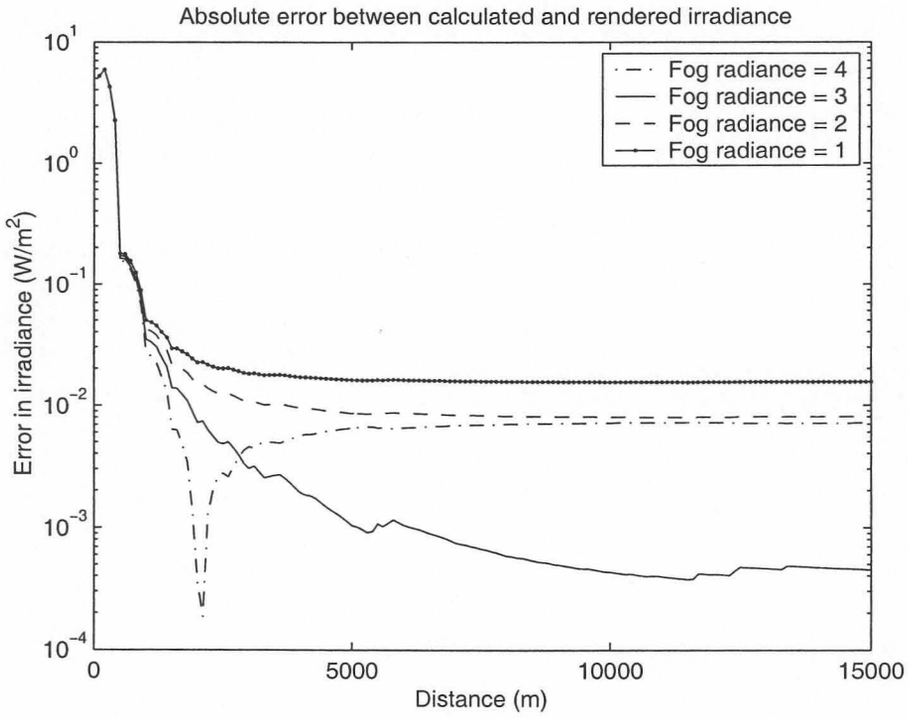

Figure 6.3: Irradiance error with different fog radiances - 14bit

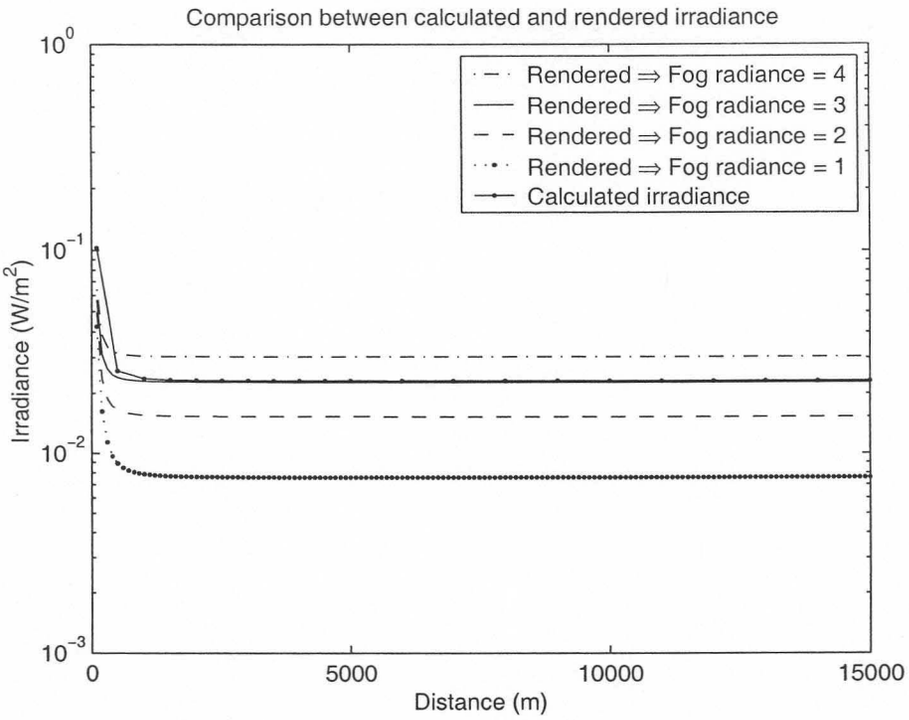

Figure 6.4: Irradiance from scenario with different fog radiances - 8bit

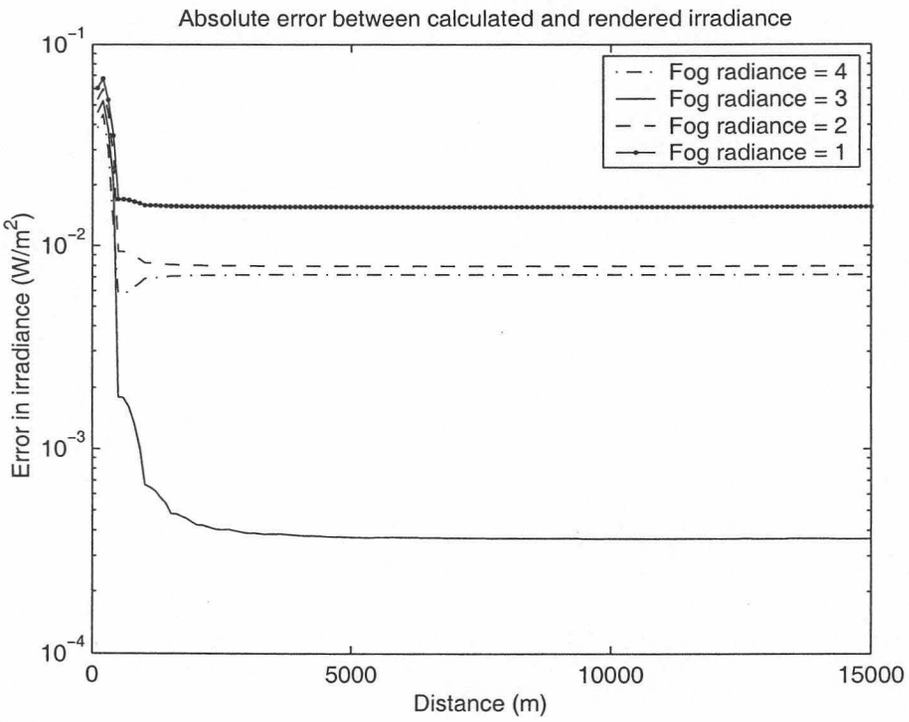

Figure 6.5: Irradiance error with different fog radiances - 8bit

6.4 Techniques to reduce the rendering errors

6.4.1 Super-resolution

A source of errors is that pixels are combined into larger pixels as the distance between the object and the sensor increases. If a single pixel is at a much higher radiance than its neighbours, the irradiance due to that pixel might be lost in the final image after combination with the neighbouring pixels. That can lead to large errors in the irradiance. It is possible to increase the accuracy of the rendered image by rendering the target object at a high resolution and re-sampling the image before adding the target object to the final image.

6.4.2 Mapping the dynamic range of the radiance to hardware's dynamic range

If simulations are limited to low temperatures, for example below 200°C, the input radiance values can be spread over the hardware's dynamic range. This can lead to greater accuracy in the resulting simulation, although an extra processing step is required to convert the image from gray scale to radiance values.

6.4.3 Multi-stage processing of the image

A typical infrared scenario consists of target radiance, background radiance, atmospheric transmittance and atmospheric path radiance. The results in Chapter 4 show that it is possible to render an object with reasonably high accuracy at short distances, using techniques such as extending the dynamic range. The results in Chapter 5 show that errors in calculating the atmospheric transmittance and atmospheric path radiance occur and can lead to large errors. The limited dynamic range of the rendering hardware is an important source of errors in calculating the path radiance.

The optimal method of rendering images would then be to render the target objects at a short distance using OpenGL. The spatial characteristics of the target are taken into account and the result is a bitmap viewed from the correct azimuth and elevation. The image can be converted from gray scales to radiance if the technique in Section 6.4.2 was used. The bitmap is modified using super-resolution to compensate for the distance between the target and the sensor in the actual scenario. The atmospheric transmittance is calculated using Equation (5.4) or a $e^{-\alpha r}$ format in cases where the simulation is for narrow bands. The target bitmap is scaled with the value of the atmospheric transmittance. The path radiance is calculated and added to the value of the target bitmap. The atmospheric background radiance is pre-calculated using MODTRAN and assigned to a bitmap. The two bitmaps are combined into a final rendered image. This image consists of 32-bit floats. Adding the atmospheric background radiance to the target would introduce large errors in cases where the target to background ratio is close to one. The correct technique would be to only add the background radiance values to the parts of the target image where the pixel values are zero. This would increase the time required to execute the simulation. The process is shown in Figure 6.6.

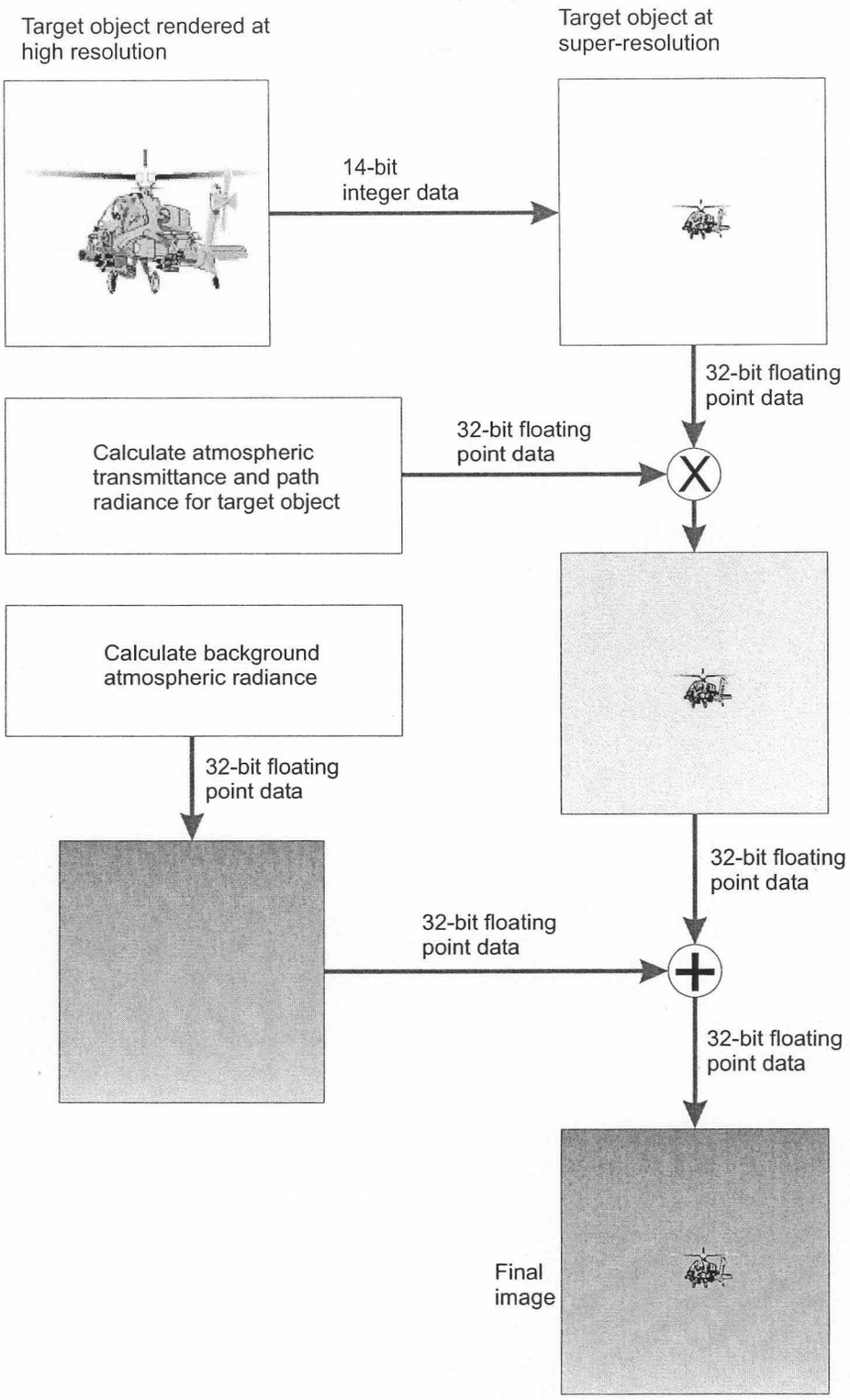

Figure 6.6: Rendering an image to obtain maximum accuracy

6.5 Benchmarks

A series of benchmarks were carried out to evaluate the frame rates that can be achieved in simulations. The test system was a 667 MHz Pentium III machine with a GeForce 2 GTS graphics card running Windows NT4. The following tests were carried out:

- **Render images** The GeForce card was used to render OpenGL images. The test object was a sphere consisting of 10000 polygons. The card rendered 123 frames per second.
- **Read images from the frame buffer** An image was rendered on the GeForce card and a 256×256 sub-image was read out using the *glReadPixels* command. It was possible to read out 14.7 sub-images per second.
- **Multiply a matrix with a constant** A 512×512 floating point matrix was defined in memory and the elements in the matrix were each multiplied with a floating point value. It was possible to repeat the operation 6.9 times per second.
- **Calculate 256×256 super-sampled image** A 256×256 floating point matrix was defined in memory and values were assigned to 5×5 elements in the centre of the matrix. The rest of the matrix was set to zero. The super-sampled equivalent of the matrix was calculated at various distances. The operation was repeated 26.8 times per second.
- **Render image, read result and calculate super-sampled image** A 512×512 image was rendered using the GeForce card, a 256×256 sub-image was read out from the card and the super-sampled equivalent of the sub-image was calculated. The frame rate was 5.4 frames per second.