Chapter 1

Introduction

1.1 Background

Physically realistic imagery is an important part of the design, simulation and evaluation of electro-optical systems. These images can be acquired during field trials or generated using the characteristics of the scenario and objects of interest. The generation of real-time (more than 75 frames per second) images used to be limited to graphics supercomputers such as the SGI Onyx2. These computers utilize dedicated hardware and multiple processors to generate images at the required rate. The development of a new generation of personal computer graphics cards during 1999/2000 made it possible to generate complex images at real-time frame rates, using just a personal computer and a commercial off-the-shelf graphics card. These graphics cards are often called graphics co-processors to indicate that a large part of the graphics processing is handled by the graphics card, instead of the CPU.

The development of the graphics cards are largely driven by the demands of computer game players for realistic imagery to accompany games. The cards are therefore developed to deliver images that are pleasing to the human eye. The generation of two-dimensional representations, as viewed on a computer monitor, of three-dimensional scenarios and objects place large demands on the computer programmer. The programme must transform the three-dimensional objects to two-dimensional representations, add lighting and atmospheric effects and compute surface parameters, all at real-time frame rates. Graphics libraries, such as OpenGL® , were developed to assist with this task.

OpenGL was originally developed to generate visual images, but it is possible to generate infrared images using OpenGL. The infrared parameters such as atmospheric transmittance and source radiance must be mapped to equivalent OpenGL parameters such as fog density and object colour.

Infrared imagery are used for the development of systems and automatic target recognition algorithms and the evaluation of systems. These evaluations include hardware-in-the-loop testing and the evaluation of operational systems prior to their use. Some examples are given in [1], [2] and [3].

1.2 Problem Statement

The generation of high fidelity infrared images from a three-dimensional model of an object would require the following steps:

- Determine the objects that are in the sensor's field of view. These objects are normally described by a collection of three-dimensional polygons. The number of polygons vary with the complexity of the object, but are typically not less than a few thousand polygons per object.

- Calculate the projected area of each surface element or polygon of the object as viewed from the sensor.

- Calculate the spectral radiance (L) of each surface element. The radiance could include reflected, emitted and transmitted energy. The reflected radiance can be from a source such as the sun or another source such as a lamp. The irradiance (E) is influenced by the atmospheric path between the source and the surface element. The radiance is further influenced by the reflectivity, absorptivity and transmissivity of the surface element. These parameters can all vary as a function of wavelength. This would imply that the radiance calculation must be carried out at each wavelength of interest.

- Calculate the spectral atmospheric transmittance between the surface element and the sensor using a model such as MODTRAN [4].

- Calculate the spectral energy emitted by the atmospheric path between the surface element and the sensor.

- Calculate the spectral irradiance as the sum of the irradiance from the object multiplied by the atmospheric transmittance and the irradiance due to path radiance.

- Multiply the spectral irradiance with the sensor system's spectral response and integrate the spectral irradiance to determine the irradiance on the sensor from the specific surface element.

- Repeat the process for all the surface elements in the sensor's field of view. The surface elements could be from terrain, target objects and sky background.

The calculation of each of the items mentioned above require a massive amount of computer power. The computer systems that can carry out such a task operate with multiple processors and costs multiple millions of dollars. Silicon Graphics developed a graphics library, OpenGL, to simplify the calculation of images in the visual band. The library is highly optimised to enable the calculation of images in real-time. The implementation of real-time calculations required that a few short-cuts had to be implemented. One of these was to limit the dynamic range of the final image. This did not have an effect on the results, when viewed by a human observer. Recent developments in graphics technology led to graphics co-processors that could implement graphics libraries such as OpenGL and Microsoft's DirectX in hardware. These graphics co-processors became available on consumer hardware at a fraction of the cost of a graphics super-computer. A shortcut in the development of these systems was to limit the dynamic range of the output images even further, to eight bits per colour for each of the three basic colours.

1.3 Methodology

The objective of this study is to investigate the implementation of a real-time infrared image generation system on a Pentium III class computer. Generating infrared imagery on a system that was developed to calculate visual imagery presented three problems:

- The graphics library was developed for visual images. It therefore requires the extension of the library to the infrared band. This can be accomplished by defining a mapping of spectral infrared parameters such as radiance and atmospheric transmittance to parameters that can be implemented in OpenGL or DirectX.
- The dynamic range of graphics accelerators is not sufficient to calculate high dynamic range infrared images. A technique is therefore required to "extend" the dynamic range of the graphics accelerator.
- The implementation of solutions to the first two problems might be so time consuming that the frame rate of the resulting images is too low to be useful. The solution to this problem will require the development of some additional techniques to reduce the number of required calculations.

1.4 Research Areas

To reach the objective the following areas were investigated:

- **Radiometry**
  The mapping of spectrally variant parameters such as atmospheric transmittance, surface emissivity and radiance to single value parameters that can be implemented as an equivalent model and used in an OpenGL environment. The results are compared with theoretically calculated data to determine the fidelity of the technique.
- **Rendering physically realistic imagery on low-cost computer platforms**
  The references found in the literature search all used high-end SGI machines with built-in hardware acceleration to generate images at real-time frame rates. Elements of a radiometric simulation were implemented on a personal computer with a graphics co-processor. Images were rendered using this system and the results are compared with theoretically calculated values. The study was limited to thermal self-emission (radiance) to reduce the number of variables in a specific scenario.
- **Increasing the dynamic range of commercial graphics accelerators**
  According to Makar *et al.* [5], the dynamic range of high-end image rendering systems, such as the SGI InfiniteReality2 is 12 bits per colour. The equivalent dynamic range on a personal computer is 8 bits per colour. The dynamic range limits the range of radiance levels that can be presented in a simulation. A technique to extend the dynamic range, proposed by Olsen *et al.* [6], was investigated.

1.5 Literature Survey

A literature search using the DIALOG system yielded four references related to the proposed research area.

Lorenzo *et al.* [7] describe a system that was developed to generate synthetic imagery for sensors such as FLIRs. The system implements OpenGL to generate the imagery, but runs on SGI hardware. The authors do not describe the accuracy of the method nor the correspondence to actual scenarios. Anding *et al.* [8] developed a product called SensorVision that uses OpenGL to generate infrared imagery. The authors do not describe the accuracy of the method nor the correspondence to actual scenarios. The simulation runs on SGI hardware.

Sundberg *et al.* [9] and Crow *et al.* [10] are using OpenGL to generate infrared images based on parametric databases.

1.6 Contribution

The literature survey shows that current infrared image generators runs on multi-million dollar graphics computers. There is a limited description of the techniques required extend OpenGL to generate infrared images. The contributions flowing from this study will be:

- It will be shown that it is possible to generate infrared imagery on a consumer personal computer.
- OpenGL will be extended to render infrared images on a personal computer.
- The accuracy of the mapping of infrared parameters to the visual band will be investigated and problem areas defined.
- It will be shown that a technique to extend the dynamic range of SGI machines is also applicable to personal computers.
- A technique will be developed that would enable the rendering of high-fidelity infrared images on a personal computer.

1.7 Dissertation Outline

The following are addressed in this document:

- **Chapter 2** is an overview of the radiometric principles that are relevant to this study.
- **Chapter 3** is an overview of the elements of computer graphics that are relevant to this study.
- **Chapter 4** describes an investigation of the technique proposed by Olsen *et al.* [6] to artificially increase the dynamic range of a graphics accelerator card. The difference between the rendered and calculated radiance of a test object is determined for different

positions on the object. The difference between the rendered intensity and the calculated intensity of the object is determined for different separation distances between the test object and the sensor.

- **Chapter 5** is a description of the determination of equivalent single parameter entities for atmospheric transmittance and path radiance.
- **Chapter 6** describes the practical implementation of a radiometric simulation in OpenGL. Problems with the implementation are highlighted and techniques are suggested to implement an accurate simulation.