EMIT
An infrastructure for real-time infrared image generation

Paul Obermeier
Reference: EMIT – An infrastructure for realtime infrared image generation
EMIT – An infrastructure for realtime infrared image generation

This document and the information contained herein is proprietary information of MBDA and shall not be disclosed or reproduced without the prior authorisation of MBDA. © MBDA 2022.
20 years ago I gave my first Tcl presentation at the 3rd EuroTcl in Munich

2002: New image formats for the Img extension

Paul Obermeier
obermeier@poSoft.de
paul.obermeier@lfk.eads.net

Third European Tcl/Tk User Meeting.
Munich, June 2002

2008: First version of EMIT

EMIT - Extensible Multispectral Image Generation Toolset
Using Tcl/Tk for simulation and visualization

Dipl.-Inf. Paul Obermeier
Software Architect for Simulation and 3D Computer Graphics
MBDA Germany

European Tcl/Tk User Meeting 2008, Strasbourg
**EMIT Overview – Infrared Physics**

**Infrared**: Spectral waveband adjacent to the visible waveband

<table>
<thead>
<tr>
<th>Energy Type</th>
<th>Wavelength (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-Rays</td>
<td>0.01 - 10</td>
</tr>
<tr>
<td>UV</td>
<td>0.001 - 0.4</td>
</tr>
<tr>
<td>VIS</td>
<td>0.4 - 0.75</td>
</tr>
<tr>
<td>IR</td>
<td>0.75 - 1000</td>
</tr>
<tr>
<td>Radio Waves</td>
<td>&gt;1000</td>
</tr>
</tbody>
</table>

The emission of IR radiation of solids is described by Planck’s law:

$$ R(\lambda, T) = \frac{2\hbar c^2}{\lambda^5 \left(e^{\frac{\hbar c}{\lambda kT}} - 1\right)} d\lambda $$
Visibility in the infrared spectrum is different from what we are used seeing with our eyes.
The Extensible Multispectral Image Generation Toolset (EMIT) is a modular software library developed at MBDA Germany for the generation of physics-based infrared images in realtime.

It is able to render infrared images in full 32-bit floating point precision using state-of-the-art computer graphics cards and advanced shader programs.

The core modules of the EMIT rendering engine are written in C++ and GLSL, but EMIT also makes heavy use of Tcl/Tk (including several extension packages) for development, maintenance and usage purposes.
EMIT Use Case - Software-in-the-Loop Simulation

- Verification simulation.
- Geotypic databases for development of image processing.
- Thread Enabled
  - Thread for image generation
  - Thread for terrain height queries at high rates
- Generation of seeker images with EMIT
  - Reflections and soft shadows
  - Rolling-shutter investigations

Reference: EMIT – An infrastructure for realtime infrared image generation
This document and the information contained herein is proprietary information of MBDA and shall not be disclosed or reproduced without the prior authorisation of MBDA. © MBDA 2022.
EMIT Use Case - Hardware-in-the-Loop Simulation

- Accredidated verification simulation
- Geospecific databases for flight campaigns
- 5-axis motion table
- SBIR Mirage Infrared Projector
- EMIT Infrared Image Generation
EMIT Overview – Workflow

Workflow for IR image generation needs some additional modeling steps

**Modeling**

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Tool</th>
<th>Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry</td>
<td>Trian3D, Remo3D</td>
<td>Geometry Data</td>
</tr>
<tr>
<td>Atmosphere</td>
<td>ModTran</td>
<td>Atmospheric Data</td>
</tr>
<tr>
<td>Temperature</td>
<td>Thermos, TaiTherm</td>
<td>Thermal Data</td>
</tr>
<tr>
<td>Texture Classification</td>
<td>EMIT-TC</td>
<td>Attributed Data</td>
</tr>
</tbody>
</table>

**Scene Edition**

<table>
<thead>
<tr>
<th></th>
<th>Tool</th>
<th>Scene Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene Edition</td>
<td>EMIT-SE</td>
<td>Scene Data</td>
</tr>
</tbody>
</table>

**Image Generation**

<table>
<thead>
<tr>
<th></th>
<th>Tool</th>
<th>Raw Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Generation</td>
<td>EMIT-IG</td>
<td>Raw Image</td>
</tr>
<tr>
<td>Sensor Modelling</td>
<td>EMIT-IG</td>
<td>Sensor Image</td>
</tr>
</tbody>
</table>

**Atmosphere Balance**

\[
S + R_1 + R_2 + R_3 = W - H - G - V + \Delta Q + Q_W = 0
\]

- **Atmosphere**
- **Heat Balance**
- **Material Classification**
EMIT uses OpenSceneGraph and adds functionality for infrared specific calculations.

**EMIT Overview – Architecture**

**Open Interfaces**
- DIS/HLA capable application, ex. DisSim
- CIGI capable application, ex. FlightSim

**EMIT Core**

**Legacy Interfaces**
- SWIG
  - Automatic generation of scripting interfaces
- CDB
- OSG
- EMIT Simulation Interface (ESI)
- EMIT CIGI Interface (ECI)
- EMIT Programming Interface (EPI)
- EMIT Direct Interface (EDI)
- EMIT application, ex. Missile Simulation
- EMIT application, ex. SceneEditor

**Linux (Native or Container)**
- Windows
EMIT and Tcl/Tk – SWIG based Tcl wrapper of EPI

EPI offers a network-transparent interface in C, C++, Python and Tcl for application programming and connection to other simulation tools such as Matlab / Simulink, Labview or TensorFlow.
**EMIT and Tcl/Tk – SWIG based Tcl wrapper of EDI**

**EDI** is the interface used for script-based generation of scenes, realizing the graphical user interface EMIT-SceneEditor and tight integration into other applications.

Available in C++ and Tcl.
Side note – My private Tcl/Tk projects

Tcl extensions to make life at work easier

**Img**
- Additional formats parsers (SUN, SGI, RAW)

**Tcl3D**
- Tcl wrapper for OpenGL and OpenSceneGraph

**CAWT**
- COM Automation With Tcl

**BAWT**
- Build Automation With Tcl
**BAWT based daily builds**

### Operating system
- **Windows**
  - Win 7
  - Win 10

### Compiler
- **VS 2008**
  - 32-bit
  - Project A
- **VS 2013**
  - 32-bit
  - Project B
- **VS 2019**
  - 64-bit
  - Project C

### Platform
- **Windows 7**
  - 32-bit
- **Windows 10**
  - 64-bit

### Library
- **OSG 3.0**
- **OSG 3.4**

### Project
- **Project A**
- **Project B**
- **Project C**
- **Project D**
EMIT and Tcl/Tk – BAWT based daily builds

EMIT depends on a large number of third party libraries. All libraries are compiled with BAWT under Windows and Linux.

<table>
<thead>
<tr>
<th>Tools:</th>
<th>Base Libraries:</th>
<th>Simulation/Graphics Libraries:</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ CMake</td>
<td>✓ zlib</td>
<td>✓ GeographicLib</td>
</tr>
<tr>
<td>✓ SWIG</td>
<td>✓ giflib</td>
<td>✓ KDIS</td>
</tr>
<tr>
<td>✓ Doxygen</td>
<td>✓ libjpeg</td>
<td>✓ Freeglut</td>
</tr>
<tr>
<td>✓ InnoSetup</td>
<td>✓ libpng</td>
<td>✓ Ftgl</td>
</tr>
<tr>
<td>✓</td>
<td>✓ libtiff</td>
<td>✓ Glew</td>
</tr>
<tr>
<td>✓</td>
<td>✓ Freetype</td>
<td>✓ OpenSceneGraph</td>
</tr>
<tr>
<td>✓</td>
<td>✓ libressl</td>
<td>✓ WxWidgets</td>
</tr>
<tr>
<td>✓</td>
<td>✓ Curl</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓ Boost</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓ Eigen</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓ Fftw</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓ Xerces</td>
<td></td>
</tr>
</tbody>
</table>

**Tcl/Tk environment:**

✓ Tcl/Tk (+ appr. 20 Tcl Packages)
EMIT and Tcl/Tk – BAWT based daily builds

Continous Development and Integration

- An automated infrastructure for creating, testing and delivering the various EMIT product versions, including documentation and test protocols.
- Daily Build computer for Windows and Linux (Build, Test, Deploy).
- No additional license costs due to combination of OpenSource (SVN, Tcl) and Microsoft Office (Word, Excel, OneNote) products.

Windows Daily Build

| 1. Checkout sources |
| 2. Build 3rd Party libs |
| 3. Build EMIT libs and apps |
| 4. Build EMIT documentation |
| 5. Run acceptance tests |
| 6. Build deployment files |
| 7. Copy logs & deployments |
| 8. Send success/failure mails |

Linux Daily Build

| 1. Checkout sources |
| 2. Build 3rd Party libs |
| 3. Build EMIT libs and apps |
| 4. Build EMIT documentation |
| 5. Run acceptance tests |
| 6. Build deployment files |
| 7. Copy logs & deployments |
| 8. Send success/failure mails |

- Continuous Development and Integration
**EMIT and Tcl/Tk – CAWT based generation of documentation**

**Input**
- Requirements List (Excel)
- Specification Template (Word)
- Assessment Template (Word)
- UserManual Template (Word)
- UserManual Pictures (PowerPoint)
- Doxygen Reference (HTML files)

**Tcl-Scripts**
- Gen. Assessment Doc.
- Execute Tests (RunAndCheck)

**Output**
- Specification Document Word
- Assessment Instruction Word
- User Manual Word
- User Manual PDF

---

Page 19  
Reference: EMIT – An infrastructure for realtime infrared image generation

This document and the information contained herein is proprietary information of MBDA and shall not be disclosed or reproduced without the prior authorisation of MBDA. © MBDA 2022.
EMIT and Tcl/Tk – Scripted test procedures

Test Input

- SpecifySceneFile “Chessboard-Animation.csgt”
- SpecifySensorFile “PP-Animation/PP-Animation_Path.esc”
- SpecifyWindow 1 1 10 20
- SpecifyImageOutput “PP-Animation_Path.jpg” # RAW
- SpecifySceneType FlatEarth
- SpecifyLocal01
- SpecifyLogging 2 “PP-Animation_Path.log”
- Set camPos1 {expr {(-10000.0 - 44.278)}
- SpecifyLocation Pos_tBox 0.0 0.0 0.0 0.0 0.0 0.0
- SpecifyLocation Pos_tBox1 0.0 0.0 0.0 0.0 0.0 5.0

# New location
set locationName “Pos_tBox”

Checkimage $testName $locationName $testWidth $testHeight
# Check border of C01. Object width must be 50 pixels.
Checkpoint “eq” $testName $locationName “C01 LeftBorder” 200 205 1.0
Checkpoint “eq” $testName $locationName “C01 RightBorder” 290 205 2.0
Checkpoint “eq” $testName $locationName “C01 RightBorder” 290 205 0.0
Checkpoint “eq” $testName $locationName “C01 RightTerrain” 280 205 1.0

Test Scripts

- RunAndCheck
- Generate Report

Report Template

- Word
The EMIT Scene Editor builds up an EMIT renderable scene by:

- combining components created in the modelling step (Players, Terrain)
- animating dynamic players
- defining the environment
- configuring virtual sensors

The EMIT Scene Editor is written in Tcl by using the wrapped EMIT functionality supplied by the EDI interface.

Additionally a separate Viewer class allows the incorporation of the OpenSceneGraph based EMIT 3D window into a Tk widget.
Overview of the Graphical User Interface

Tcl/Tk specialities:

- Togl widget
- Tk events -> OSG
- Rollups
- Dials
- Tablelists
Tcl/Tk specialities:

- C++ Code-Generation for EntityTypes with Tcl script
EMIT and Tcl/Tk – EMIT SceneEditor

Render Modes & Sensor Management

Tcl/Tk specialities:

- ukaz
- tdom
Replay Functionality

Tcl/Tk specialities:

- Render log files are written as Tcl scripts and can directly be interpreted.
Atmospheric and Thermal Calculations

Tcl/Tk specialities:

- Ukaz
- Tablelists
- Sqlite binding
Main features:
• Realtime Image Generation
• Realtime sensor effects
• Realtime special effects (ex. fire, smoke)
• Flexible, network transparent interface
• E/O and IR mode
• Linux and Windows

Tcl usage:
• Tcl/Tk based graphical user interface
• C++ interfaces wrapped for Tcl with SWIG
• CD/CI workflow based on Tcl