

# EMIT - EINE INFRASTRUKTUR ZUR INFRAROT-BILDGENERIERUNG

P. Obermeier, Dr. P. Schätz, R. Wiedemann  
LFK-Lenkflugkörpersysteme GmbH  
Landshuter Str. 26, 85716 Unterschleißheim  
Deutschland

## Zusammenfassung

Moderne Lenkflugkörpersysteme verwenden Infrarot-Sensoren zur Zielerfassung und für die Navigation. Der Entwicklungs- und Validationsprozess solcher Lenkflugkörper benötigt hochgenaue Simulationen, die auch eine Abbildung der Umwelt anhand von synthetischen Szenarien erlauben.

EMIT (**E**xtensible **M**ultispectral **I**mage **G**eneration **T**ool) ist ein bei der LFK-Lenkflugkörpersysteme GmbH entwickeltes Werkzeug zur Erzeugung von synthetischen, physikalisch korrekten Infrarot-Bildern in Echtzeit.

EMIT wurde bereits in diversen Projekten bei Design, Entwicklung und Validation von Lenkflugkörpersystemen verwendet. Aufgrund seiner modularen und erweiterbaren Software-Architektur kann es in folgenden Anwendungsfeldern zum Einsatz kommen:

- Man-in-the-Loop Simulationen
- Software-in-the-Loop Simulationen
- Hardware-in-the-Loop Simulationen

Die Veröffentlichung gibt einen Überblick über den typischen Workflow und die dabei verwendeten Werkzeuge beim Einsatz einer Infrarot-Bildgenerierung zur Stimulation von Infrarot-Suchköpfen. Insbesondere wird auf die Anforderungen und zusätzlichen Aufwände (atmosphärische und thermische Modellierung) im Vergleich zu klassischen „VisSim“-Applikationen eingegangen.

Anhand der System-Architektur von EMIT werden die grundlegenden Hardware- und Software-Technologien dargestellt, die eine echtzeitfähige Infrarot-Bildgenerierung ermöglichen: Programmierbare Graphikkarten und SceneGraph-Software.

Zum Abschluss wird ein Ausblick auf aktuell laufende Forschungs- und Entwicklungsarbeiten zur qualitativen und quantitativen Verbesserung der Bildgenerierung gegeben.

## 1. EINLEITUNG

LFK-Lenkflugkörpersysteme GmbH verwendet seit vielen Jahren Simulationen mit Infrarot-Bildgenerierung zur Stimulation von realen und virtuellen Infrarot Suchköpfen.

Um einen möglichst großen Bereich der klimatischen und atmosphärischen Bedingungen abdecken zu können, wird dabei mit synthetisch generierten Szenarien (künstlichen Umgebungen) gearbeitet.

Für die Nachbildung von Suchkopf- und Visiereingangssignalen werden IR-Bilder benötigt, die mit entsprechender Rate generiert werden können und die eine kontrollierte Bewegung im Szenario erlauben (d.h. keine IR-Videos).

Aufgrund der Erfahrungen mit kommerziellen Infrarot-Bildgeneriersystemen wird bei der LFK GmbH seit 2003 eine eigene Infrastruktur zur Infrarot-Bildgenerierung entwickelt, um unabhängig von Herstellern und etwaigen Export-Restriktionen zu sein, sowie durch die volle

Kontrolle über den Bildgenerierprozess flexibel auf neue Anforderungen aus den Projekten und internen Entwicklungen reagieren zu können. Ein weiterer wichtiger Grund für eine Eigenentwicklung war der Einsatz des Systems als Nachweismittel für die Leistungen der entwickelten Lenkflugkörper. Durch die Möglichkeit der Code-Inspektion („White-Box“ vs. „Black-Box“) ist eine Verifikation und Validation der verwendeten Modelle deutlich einfacher zu realisieren.

Die erste, nicht echtzeitfähige Version der Infrarot-Bildgenerierung (EMIT-Ray [4]) wurde als reine Software-Lösung auf Basis der Ray-Tracing Technologie realisiert. Eine echtzeitfähige Version der Bildgenerierung (EMIT-Shade [5]) wurde der bestehenden Infrastruktur hinzugefügt, als handelsübliche Graphikkarten die für Infrarot-Bildgenerierung erforderlichen Leistungen (z.B. durchgängige Verarbeitung von 32-bit Fließkommazahlen) anboten.

Hardware-in-the-Loop Simulationen zur Stimulation von Infrarot-Suchköpfen stellen die höchsten Anforderungen an eine Bildgenerierung, nämlich hohe und konstante

Frame-Raten mit einer Vielzahl von Szenarien zu erbringen.

Die bisher bei LFK GmbH eingesetzten kommerziellen Infrarot-Bildgeneriersysteme werden sukzessive durch die neue Infrastruktur EMIT ersetzt.

## 2. WORKFLOW UND WERKZEUGE

Der Arbeitsablauf (Workflow) bei der Infrarot-Bildgenerierung ist dem traditionellen Arbeitsablauf im Bereich der Computer-Animation oder der Erzeugung von „VisSim“-Applikationen sehr ähnlich.

Im Wesentlichen umfassen die Abläufe dabei 4 Phasen:

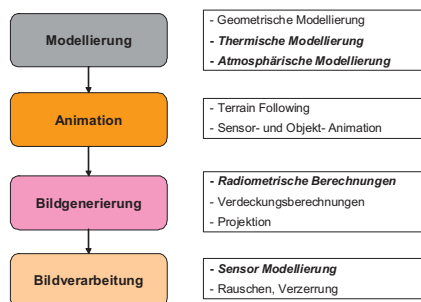


BILD 1. Workflow der Infrarot-Bildgenerierung

In der Modellierungs-Phase werden die Landschaften (Terrain, statische Objekte wie Häuser und Bäume), sowie die benötigten dynamischen Objekte (Autos, Flugzeuge) erzeugt. Zusätzlich zu den geometrischen Modellierungsanteilen müssen für eine physikalisch korrekte Berechnung der Strahlungsverteilung atmosphärische und thermische Berechnungen durchgeführt werden.

In der Animations-Phase werden die dynamischen Objekte platziert, sowie deren Bewegungsabläufe festgelegt und in einer Szenarienbeschreibung gespeichert.

Die Bildgenerier-Phase erzeugt aus der Szenarienbeschreibung entsprechend des definierten Wellenlängenbereichs und der Sensorposition ein Radianzbild.

Dieses Rohbild kann in der Bildverarbeitungsphase entsprechend der Sensor-Charakteristiken (z.B. Point-Spread-Function, Rauschen) weiterverarbeitet werden.

### 2.1. Werkzeuge

EMIT besteht aus einer plattform-unabhängigen Software-Bibliothek mit C++-Programmierschnittstelle und einem Satz von Werkzeugen, die auf dieser Bibliothek aufbauen.

Für die geometrische Modellierung kommen Standardwerkzeuge, wie z.B. Multigen Creator oder 3D StudioMax zur Verwendung. Daneben wurde eine Reihe von Werkzeugen entwickelt, die es ermöglichen, die für eine Infrarot-Bildgenerierung notwendigen Attribute (erweiterte Materialeigenschaften, Temperaturen) aufzubringen und zu verwalten. Diese Werkzeuge benutzen dabei die Funktionalitäten der EMIT-Kernbibliothek, die im nächsten Kapitel beschrieben ist.

Die nächsten Abschnitte gehen auf eine Auswahl dieser Werkzeuge und deren Funktionalität ein.

#### 2.1.1. Szenarien-Editor

Der Szenarien-Editor ist die zentrale Benutzeroberfläche der EMIT Infrastruktur. Er ermöglicht es dem Benutzer, 3-dimensionale Szenarien interaktiv zu generieren und zu verändern. 3D-Modelle in Standard Dateiformaten können im Terrain platziert und animiert werden, sowie mit infrarot-spezifischen Attributen belegt werden. Die für die thermische und atmosphärische Modellierung notwendigen Parameter werden ebenfalls im Szenarien-Editor spezifiziert. Diese Szenarienbeschreibung wird im Collada-Dateiformat für weitere Bearbeitungsmöglichkeiten gespeichert.

Animationen von Objekten werden durch Bewegungspfade definiert, die der Benutzer durch interaktive Eingabe von Wegpunkten erzeugen kann.

Da der Szenarien-Editor auf die Funktionalitäten der EMIT-Kernbibliothek zurückgreift, ist es jederzeit möglich, zwischen der Infrarot-Sicht und einer TV-Sicht umzuschalten.

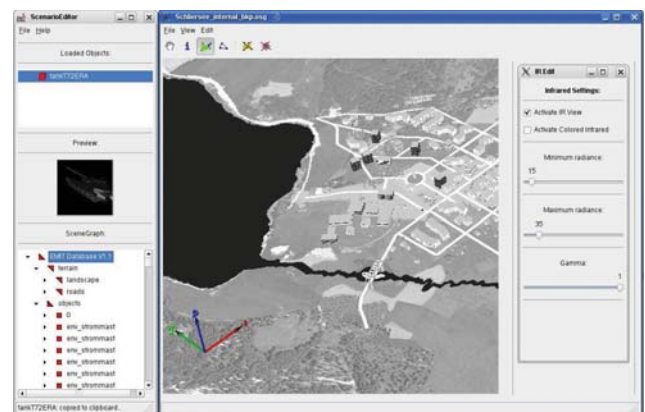


BILD 2. Szenarien-Editor

#### 2.1.2. Terrain-Builder

Mit Hilfe des Terrain-Builders kann man schnell texturierte 3-dimensionale Landschaften aus digitalen Höhendaten (Digital Elevation Data, DED) und georeferenzierten Bildern (Satelliten- / Luftbilder) erzeugen.

Das vom Benutzer ausgewählte Gebiet wird automatisch anhand der Vorgaben zur Detaillierungstiefe (Levels-of-Detail) in Kacheln unterteilt, die entsprechenden Dreiecksgitter erzeugt und mit Texturkoordinaten belegt.

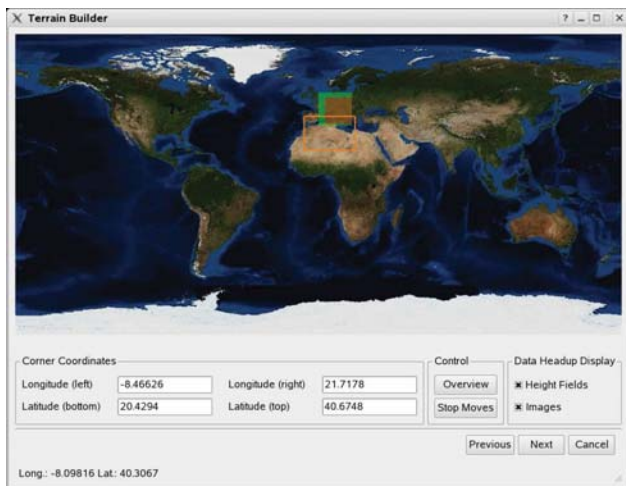


BILD 3. Terrain-Builder

### 2.1.3. Partikelsystem-Editor

Mit dem Partikelsystem-Editor können Gefechtsfeldeffekte wie Feuer, Rauch und Flares spezifiziert werden. Partikelsysteme sind texturierte, teiltransparente Polygone in grosser Anzahl, deren Bewegung und Temperatur durch physikalische Größen beeinflusst werden. Der Editor ermöglicht es dem Benutzer, interaktiv die verschiedenen Parameter eines Partikelsystems zu ändern, an Objekte zu binden, sowie die Auswirkungen in Echtzeit zu betrachten.

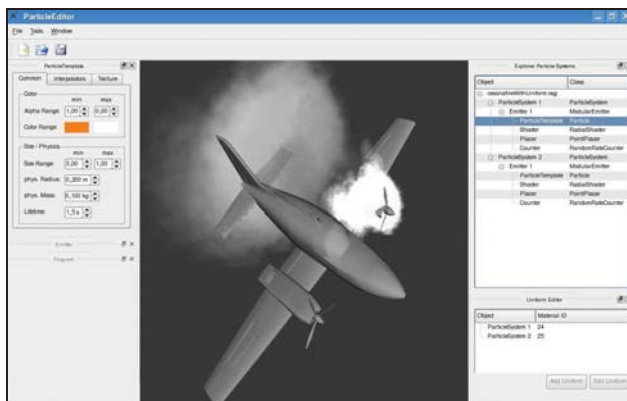


BILD 4. Partikelsystem-Editor

### 2.1.4. Interaktion

3-dimensionale Ansichten eines von EMIT berechneten Szenarios können einfach in graphische Benutzeroberflächen integriert werden. Es existieren Anbindungen an Benutzeroberflächen wie Qt oder wxWidgets.

Maus-, Tastatur- oder Joystick-Eingaben werden in EMIT eingespeist, um damit verschiedene Arten von Manipulatoren zu stimulieren. Mithilfe dieser Manipulatoren können z.B. Fahr- oder Flugbewegungen des Sensors generiert werden.

Es besteht auch die Möglichkeit, die erzeugten Bilder bildschirmfüllend oder im Stereo-Modus anzuzeigen.

### 2.1.5. Atmosphärische Modellierung

Die atmosphärischen Effekte werden durch die Integration des Standardwerkzeuges MODTRAN [1] berücksichtigt.

Zur einfacheren Bedienung wurde eine graphische Oberfläche entwickelt, mit der die zahlreichen Parameter von MODTRAN editiert werden können. Diese umfassen z.B. die Klimazone, Jahres- und Tageszeit, sowie diverse Wetterparameter.

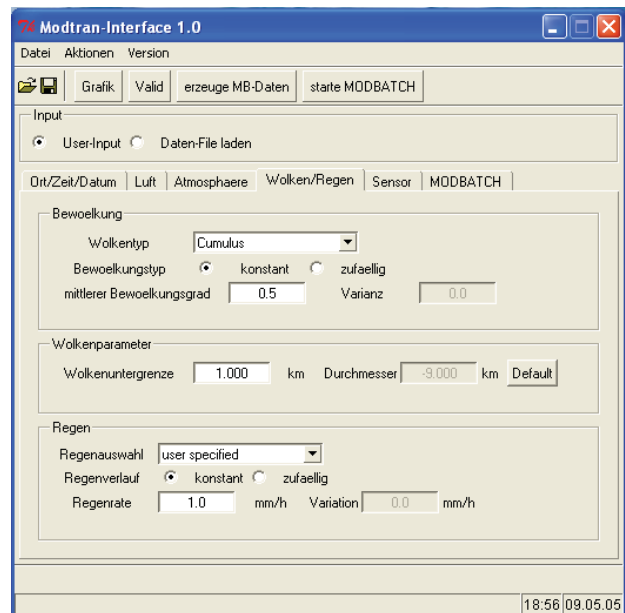


BILD 5. MODTRAN Benutzeroberfläche

Aus der Oberfläche heraus lassen sich Programme starten, die die benötigten Interpolationstabellen für das in EMIT verwendete Atmosphärenmodell erstellen.

### 2.1.6. Thermische Modellierung

Für die thermische Modellierung werden zwei verschiedene Module verwendet. Thermos [3] ist ein von LFK GmbH entwickeltes Modell zur Berechnung der Temperaturverteilung einer Vielzahl von Oberflächen, wie sie typischerweise in natürlichen Landschaften vorkommen. Es berücksichtigt die verschiedenen Beiträge zur Wärmebilanzgleichung wie z.B. Sonnen- und Himmelsstrahlung und Oberflächeneigenschaften. Es enthält ein Bodenfeuchtigkeits-, sowie ein einfaches Vegetationsmodell. Das Modell benutzt einen 1D-Code (Schichtenmodell) und wurde mit gemessenen Daten im Rahmen der Entwicklung des PARS3 LR Lenkflugkörpersystems validiert.

In Zusammenarbeit mit der DLR wird z.Z. ein verbessertes Modell (Thermal Surface Model - TSM) mit einem komplexeren und genaueren Vegetationsmodell entwickelt.

Um die Signaturen von Einzelmodellen zu berechnen, wird das COTS-Werkzeug RadThermIR der Firma ThermoAnalytics verwendet. RadThermIR benutzt einen 3D-Code und berücksichtigt aktive Wärmequellen, enthält aber kein Bodenfeuchtigkeitsmodell und nur ein einfaches Vegetationsmodell.

### 3. ARCHITEKTUR

Um Infrarot-Bilder in Echtzeit erzeugen zu können, müssen wichtige Anforderungen an die Hardware (insbesondere die Graphikkarte) und Software erfüllt sein:

- Durchgängige 32-bit Fließkommagenauigkeit.
- Programmierbare Graphikkarten (Shader).
- Render-to-Texture Fähigkeit
- Hohe Rechenleistung
- Großer Texturspeicher
- Synchronisierung von Bildgenerierung, IR-Projektor und Prüfling (Unit-Under-Test).

Echtzeitfähige Szenarien müssen in einer Szenegraph-Struktur aufgebaut sein, d.h. die Objekte des Szenarios sind hierarchisch sortiert. Für EMIT wurde das Produkt OpenSceneGraph [2] gewählt, da es eine große Verbreitung im Luftfahrt- und Verteidigungsbereich hat. Außerdem ist OpenSceneGraph eine OpenSource-Software, so dass Änderungen oder Erweiterungen bezüglich der Infrarotaspekte leicht realisiert werden können. Die Lizenzbedingungen der Software erlauben den Einsatz für kommerzielle Zwecke.

OpenSceneGraph unterstützt viele hochwertige Beschleunigungstechnologien, wie z.B. Database-Paging, Level-of-Detail, Billboards, Culling und viele weitere.

Um EMIT mit einer Vielzahl von Simulationen koppeln zu können, wurde besonderer Wert auf Plattformunabhängigkeit gelegt (derzeit werden Windows und Linux unterstützt), so dass die EMIT-Bibliothek durch Verwenden der C++-Schnittstelle direkt in eine Simulation eingebunden werden kann. Außerdem existiert eine Netzwerk-Schnittstelle auf Socket-Basis, mit der die EMIT-Funktionalität mit einer Vielzahl von Simulationen (z.B. auf Matlab/Simulink Basis) verwendet werden kann. Ein Beispiel für diese Art der Verwendung ist der bei LFK GmbH entwickelte Datenlink-Demonstrator [6].

Die Kernbibliothek von EMIT ist in plattformunabhängigem ISO-C++ implementiert und umfasst im Wesentlichen die Module Datenmanagement, Bildgenerierung und Bildverarbeitung.

Durch die offene Architektur und die Abstützung auf Standard-Datenformate ist es möglich, sehr einfach gemessene Daten oder Daten von externen Programmen in eine EMIT-Simulation einzuspeisen.

#### 3.1. Datenmanagement

Das Modul Datenmanagement umfasst das Speichern der Szenariodaten in einer Szenegraph-Struktur, sowie das Aktualisieren dieser Daten bei Bedarf, z.B. die Aktualisierung der Transformationsmatrix eines Fahrzeugmodells abhängig von der Simulationszeit.

Eine EMIT-Datenbasis besteht typischerweise aus einem Terrain, statischen und dynamischen Objekten, Spezialeffekten (Feuer, Explosionen, Staubwolken) sowie einem Atmosphärenmodell.

#### 3.1.1. Terrain

Das Terrain wird als Polygon-Gitter realisiert, auf das Texturen (Bilder aus dem sichtbaren Bereich, Temperaturbilder, Materialtexturen) aufgebracht werden. Die Geometrie des Terrains wird in einem eigenen Zweig der Szenegraph-Struktur gehalten, um Schnittpunktberechnungen zu beschleunigen, die z.B. für bodengebundene Objekte benötigt werden.

Für weitreichende Lenkflugkörper werden sehr große Datenbasen benötigt. Da diese Datenbasen nicht mehr komplett in den Hauptspeicher geladen werden können, wird hier die Technik des Database-Paging angewandt. Hierbei wird die Datenbasis bereits bei der Modellierung in Kacheln (Tiles) aufgeteilt. Die Kacheln, die in Zukunft benötigt werden, werden durch einen asynchronen Prozess geladen; nicht mehr benötigte Kacheln werden entladen.

Zur Steigerung der Bildrate trägt die Kombination von Database-Paging mit verschiedenen Detaillierungsstufen bei. Hierbei werden weit entfernte Kacheln, die aber noch im Sichtbereich liegen, in einer niedrigen Auflösung geladen, und sukzessive bei Annäherung durch detailliertere Stufen ersetzt.

#### 3.1.2. Objekte

Alle weiteren geometrische Bestandteile des Szenarios werden als Objekte behandelt. Diese können unterschiedliche Attribute besitzen, wie z.B. statisch oder dynamisch, Terrain-Following, Billboard.

Objekte können bezüglich ihrer Position, Ausrichtung oder Größe in Abhängigkeit von der Simulationszeit animiert werden. Zu diesem Zweck kann ein Bewegungspfad einem Objekt zugewiesen werden, der aus einer Liste von Wegpunkten besteht. Ein Wegpunkt enthält einen Zeitstempel, sowie die Position, Ausrichtung und Skalierung.

Die Art der Interpolation zwischen diesen Wegpunkten kann als lineare oder spline-basierte Interpolation gewählt werden.

#### 3.1.3. Spezialeffekte

Um Spezialeffekte wie Feuer, Explosionen und Staubwolken zu simulieren, werden Partikelsysteme verwendet. Ein einzelnes Partikel wird durch ein texturiertes, teiltransparentes Viereck abgebildet. Ein Partikelsystem besteht aus einem Partikel-Emitter und physikalisch basierten Algorithmen, welche abhängig von der aktuellen Simulationszeit die Position, Geschwindigkeit, Transparenz und Lebensdauer der einzelnen Partikel berechnen.

Spezialeffekte, die nicht sinnvoll modelliert werden können, oder bei denen reale Daten verwendet werden müssen, können durch Abspielen einer Bildsequenz (z.B. aufgenommenes Video einer Wärmebildkamera) dargestellt werden. Diese Bildsequenzen werden dabei auf einer virtuellen Leinwand in Echtzeit abgespielt, die – wie alle anderen Objekte auch – beliebig im Szenario platziert werden kann.



### 3.1.4. Umgebung

Das Atmosphärenmodell von EMIT muss die Dämpfung und Pfad-Radianz für unterschiedlichste Pfade elektromagnetischer Ausbreitung berechnen können.

Da diese Berechnungen sehr zeitaufwändig sind, werden sie mit MODTRAN vorab im Offline-Modus berechnet und als Interpolationstabellen abgelegt. Diese Tabellen werden für die spätere Verarbeitung durch die Graphikkarte beim Einlesen des Szenarios in Texturen konvertiert und auf die Graphik-Karte geladen.

Zur Simulation des Himmels wird automatisch eine Halbkugel (sky sphere) generiert, die entsprechend der Ausdehnung des Terrains skaliert wird. Texturen können auf diese Halbkugel projiziert werden, um Wolken darstellen zu können.

## 3.2. Bildgenerierung

EMIT unterstützt zwei Arten der Bildgenerierung:

- Direkte Bildgenerierung
- Indirekte Bildgenerierung

### 3.2.1. Direkte Bildgenerierung

Falls Temperaturen oder Radianz-Werte der Geometrie (auf Polygon- oder Vertexebe) oder Texturen zugewiesen sind, können diese direkt von der Bildgenerierung (dem Shader-Programm) verwendet werden.

Temperaturen müssen vor der weiteren Verwendung im Shader-Programm entsprechend dem verwendeten Wellenlängenbereich in Radianzen umgerechnet werden. Dies geschieht unter Verwendung des Planck'schen Gesetzes:

$$(1) \quad R_0 = 2hc^2 \cdot \int_{\lambda_1}^{\lambda_2} \frac{1}{\lambda^5 \cdot e^{\frac{hc}{\lambda kT}}} \cdot d\lambda \quad [W\mu m^4 m^{-2} sr^{-1}]$$

wobei

$$h \approx 6.626176 \cdot 10^{-34} [Js] \quad \text{Planck Konstante}$$

$$c \approx 2.997922 \cdot 10^8 [ms^{-1}] \quad \text{Lichtgeschwindigkeit}$$

$$k \approx 1.380662 \cdot 10^{-23} [JK^{-1}] \quad \text{Boltzmann Konstante}$$

Für die Berechnung werden die wellenlängenabhängigen Radianzen über den betrachteten Wellenlängenbereich aufintegriert. Da dies sehr rechenaufwändige Operationen sind, werden für bestimmte Temperaturbereiche Interpolationstabellen vorberechnet. Diese werden nicht äquidistant verteilt, sondern in Temperaturbereichen, die von besonderer Bedeutung sind, feiner unterteilt.

Die Konvertierung von Temperaturen in Radianzen vereinfacht sich dadurch in zwei Suchoperationen, sowie einer Interpolation zwischen den gefundenen Bereichsgrenzen. Dies kann auf aktuellen Graphikkarten sehr schnell ausgeführt werden.

Temperatur-Texturen können auf zwei Arten verwendet

werden: Direkte Texturen, sowie Modulations-Texturen. Die direkten Temperatur-Texturen überschreiben evtl. vorhandene Temperaturen auf Geometrie-Ebene. Modulations-Texturen vermischen die Geometrie-Temperaturen mit den Werten in der Textur.

### 3.2.2. Indirekte Bildgenerierung

Bei der indirekten Bildgenerierung werden nur Materialnummern der Geometrie bzw. den Texturen zugewiesen. Zusätzlich werden Interpolationstabellen für Temperaturen angelegt, die abhängig vom Material und der Oberflächenausrichtung sind.

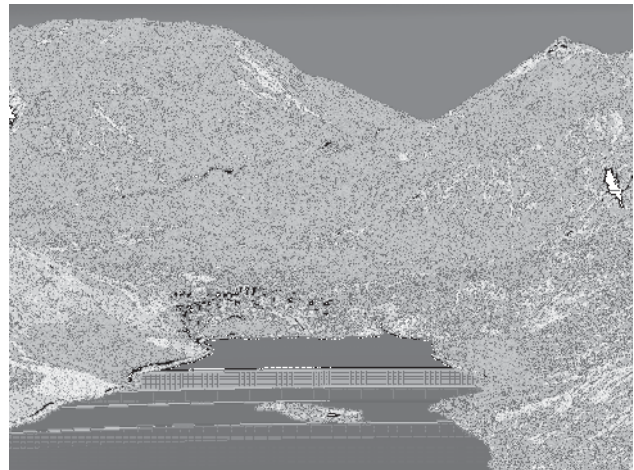


BILD 6. Beispiel für indirekte Bildgenerierung

Im Shader-Programm wird in einem ersten Schritt das Material und der Normalenvektor des vom Sehstrahl getroffenen Polygons bestimmt. Der resultierende Radianzwert wird durch Auslesen des entsprechenden Temperaturwertes aus der Interpolationstabelle bestimmt.

Indirekte Bildgenerierung hat gegenüber der direkten Methode den Vorteil, dass nur die Temperatur-Interpolationstabellen von den aktuellen Wetterbedingungen abhängig sind, d.h. die Terrain-Datenbasis muss beim Wechsel der Tageszeit oder der Wetterbedingungen nicht geändert werden.

## 3.3. Bildverarbeitung

Nach der Erzeugung der Rohbilder können diese direkt auf der Graphikkarte weiterverarbeitet werden. Dies wird benutzt, um Aliasing-Effekte zu reduzieren oder um Sensor-Effekte zu modellieren.

### 3.3.1. Prinzip

Aktuelle Graphikkarten unterstützen die Technik "render-to-texture". Dabei wird das erzeugte Bild nicht in den Bildschirmspeicher geschrieben, sondern in eine Textur im Graphikkartenspeicher. Diese Textur kann dann als Eingabe für weitere Durchläufe der Bildverarbeitungs-Pipeline verwendet werden.

Wird diese Textur auf ein Viereck aufgebracht, können im Fragment-Shader implementierte Filteralgorithmen auf das Bild angewendet werden. Da aktuelle Graphikkarten mehr als 100 Shader-Prozessoren besitzen, können

selbst aufwändige Algorithmen in Echtzeit realisiert werden.

### 3.3.2. Anwendungsbeispiele

EMIT bietet eine einfache Schnittstelle zur Bildverarbeitung an, die alle notwendigen Schritte, um die Render-to-Texture-Funktionalität anzusprechen, kapselt. Außerdem erlaubt dieses Framework das beliebige Kombinieren der Filter in eine Bildverarbeitungs-Pipeline. Nur die eigentlichen Algorithmen für den gewünschten Filter müssen als Fragment-Shader in GLSL implementiert werden.

#### 3.3.2.1. Antialiasing

Um Aliasing-Effekte zu verhindern, wird ein Bild in einer höheren Auflösung berechnet, und vor der Anzeige oder Ausgabe auf die gewünschte Auflösung mittels bilinearer Interpolation reduziert. Moderne Graphikkarten bieten diese Funktionalität bereits in Hardware an, jedoch nicht für die von EMIT benötigte Genauigkeit in 32-bit Fließkommazahlen. Deshalb wurde dieser Downsampling-Algorithmus als Bestandteil der Bildverarbeitungskette als Shader-Programm realisiert.

#### 3.3.2.2. Atmosphärische Effekte

Das Atmosphärenmodell wird auf das Rohbild als Bildverarbeitungsprozess angewendet. Die Länge des Pfades durch die Atmosphäre wird für jedes Pixel bestimmt, sowie der Azimut und Elevationswinkel der Sichtlinie des Sensors durch dieses Pixel und die Höhe des Sensors über Grund. Abhängig von diesen Werten, wird die Pfadradianz und Dämpfung durch Auslesen der Atmosphären-Interpolationstabellen bestimmt und auf die Radianzwerte des Rohbildes („Zero-Range-Radiance“) wie folgt angerechnet:

$$(2) \quad R = R_0(1 - \alpha) + R_p$$

wobei  $\alpha$  die atmosphärische Dämpfung (0.0 .. 1.0),  $R_0$  die Zero-Range-Radiance des Rohbildes und  $R_p$  die Pfadradianz bezeichnet.

#### 3.3.2.3. Visualisierung

Um das berechnete 32-bit Radianzbild auf einem Bildschirm anzeigen zu können, muß es in ein 8-bit Graustufenbild oder in ein falschfarbenkodierte Bild umgewandelt werden. Hierfür kann neben den Bereichsgrenzen auch eine Gammakorrektur für eine nicht-lineare Interpolation spezifiziert werden.

Um für Hardware-in-the-Loop Simulationen die Radianzbilder auf einem Infrarot-Projektor ausgeben zu können, müssen die 32-bit Radianzbilder auf 16-bit Genauigkeit anhand einer kalibrierten Look-Up-Tabelle reduziert werden, und in den Rot- und Blau-Kanälen der DVI-Schnittstelle übertragen werden.

#### 3.3.2.4. Sensor Effekte

Mit der EMIT Bildverarbeitungskette ist es möglich, typische Sensoreffekte wie Rauschen, Point-Spread-

Funktion oder Verzerrungen auf der Graphikkarte als Shader-Programm zu realisieren.

## 4. AUSBLICK

EMIT wird ständig weiterentwickelt, um den Anforderungen der Projekte zu genügen. Im Folgenden werden drei aktuelle Forschungs- bzw. Entwicklungsgebiete vorgestellt, die eine quantitative und qualitative Verbesserung der Infrarot-Bildgenerierung zum Ziel haben.

### 4.1. Integration von RadThermIR

Zur detaillierten Signatur-Berechnung von Ziel-Modellen wird das kommerzielle Werkzeug RadThermIR verwendet. Hierbei ergeben sich im Hinblick auf Echtzeit-Bildgenerierung bei der Integration von detaillierten RadThermIR-Modellen mit dynamischen Signaturen folgende Problemstellungen:

- Zeitschritte von RadThermIR liegen typischerweise im Sekunden- bis Minutenbereich, während für Echtzeit-Bildgenerierung Generierraten von 50 Hz und mehr benötigt werden.
- Um die Echtzeit-Bedingungen einhalten zu können, müssen alle benötigten Daten auf der Graphikkarte verfügbar sein.
- Die Daten einer RadThermIR Berechnung liegen in einem proprietären Dateiformat (TDF) vor.

Die Lösung dieser Problematik kann mit 3-dimensionalen Texturen wie folgt erreicht werden:

#### Schritt 1: Extrahieren der Geometrie

Die Geometrie wird aus der TDF-Datei extrahiert und in einen hierarchischen Szenegraph umgewandelt. Dabei werden TDF-Assemblies als DOF (Degree of Freedom) Knoten, TDF-Parts werden als Gruppen-Knoten abgebildet. TDF-Elements werden in Dreiecke und Vierecke konvertiert.

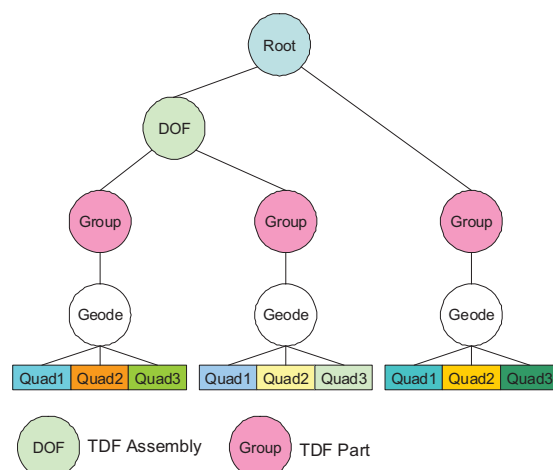


BILD 7. Abbilden der RadTherm-Struktur auf eine Szenegraph-Hierarchie

## Schritt 2: Extrahieren der Temperaturen

Für jeden mit RadThermIR berechneten Zeitschritt wird eine Textur mit 32-bit Fließkommagenauigkeit erzeugt. Jedes Texel enthält dabei die Temperatur eines TDF-Elements, d.h. eines Polygons der Geometrie. Die erzeugten Texturen werden in einem 3-dimensionalen Textur-Stack gespeichert.

## Schritt 3: Erzeugen der Texturkoordinaten

Für jeden Eckpunkt eines Polygons werden Texturkoordinaten (u,v) erzeugt. Diese geben das Texel an, welches die Temperatur für das Polygon enthält.

## Schritt 4: Zeitabhängige Animation

Während der Simulation wird zur Laufzeit eine dritte Texturcoordinate (w) berechnet. Diese wird aus der aktuellen Simulationszeit berechnet. Die Interpolation (Blending) zwischen benachbarten Texturen wird von der zugrundeliegenden OpenGL-Implementierung automatisch und hardwarebeschleunigt ausgeführt.

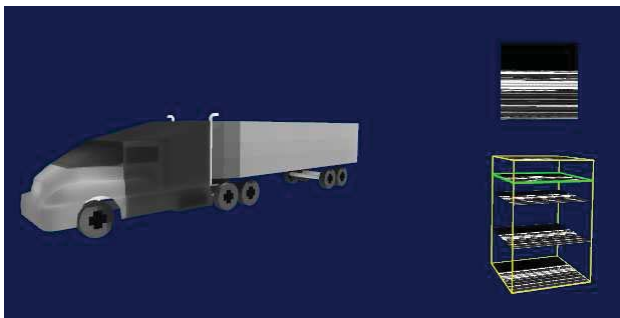


BILD 8. Temperaturverteilung auf Beispielobjekt mit zugehörigem Textur-Stack

## 4.2. Echtzeit Ray-Tracing

In Zusammenarbeit mit der Hochschule München wurde letztes Jahr eine auf der Ray-Tracing-Technologie basierende Bildgenerierung entwickelt, die auf die Daten des Szenegraphen zugreift und den Ray-Tracing-Algorithmus direkt auf der Graphikkarte unter Ausnutzung der großen Anzahl von Prozessoren ausführt. Der Ray-Tracing Algorithmus ist dabei in der Shader-Sprache GLSL realisiert.

Insbesondere wurde Wert auf die performante Bildgenerierung von realistischen Szenarien mit einer typischen Mischung aus statischen und dynamischen Anteilen gelegt. Als optimal erwies sich eine Kombination aus verschiedenen Datenstrukturen:

Für die statischen Anteile werden kd-Bäume mit SAH (Surface Area Heuristic) basierter Unterteilung als Beschleunigungsstruktur verwendet. Diese Bäume bieten ein Optimum an Traversierungsgeschwindigkeit, benötigen jedoch viel Zeit zum Aufbau.

Die dynamischen Szenarioanteile werden durch lokale, binäre kd-Bäume abgebildet, die in einer Bounding-Volume-Hierarchy (BVH) organisiert sind. Diese Anordnung bietet keine optimale Traversierungsge-

schwindigkeit, kann jedoch sehr schnell neu aufgebaut werden.

Der Ray-Tracer bietet eine frei wählbare Rekursionstiefe und beliebig zuschaltbare Schatten- und Reflexionsstrahlen. Die Primärstrahlen werden durch den Rasterizer der Graphikkarte beschleunigt.



BILD 9. Beispielszene Echtzeit-Ray-Tracing

Realistische und für die LFK GmbH relevante Szenarien laufen auf einem PC mit einer NVIDIA GeForce 8800 GTX mit interaktiver Bildgenerierrate.

Derzeit wird an der Erweiterung für den infraroten Spektralbereich gearbeitet.

## 4.3. Integration von SpeedTree

SpeedTree ist eine im Computerspiele-Sektor weit verbreitete C++-Bibliothek zur realitätsnahen Darstellung von Vegetation. Diese wird derzeit adaptiert und erweitert, um für IR-Szenarien einsetzbar zu sein.

## 4.4. Thermal Surface Model

Das thermische Modell TSM besteht aus einer Software, die in Zusammenarbeit mit der DLR Oberpfaffenhofen entwickelt wurde. Das zugrunde liegende physikalische Modell ist ein 1D-Schichtenmodell, für das eine Wärmebilanz unter Berücksichtigung des Strahlungsaustauschs und des turbulenten Wärmeaustauschs mit der Atmosphäre, sowie die Wärmeleitung im Boden berechnet wird. Das Modell beinhaltet ein Feuchtigkeitsmodell, das in Abhängigkeit von Niederschlag, Verdunstung und Oberflächenabfluß des Niederschlags den Feuchtigkeitsgehalt der Schichten berechnet und diesen dann bei der Wärmebilanzrechnung berücksichtigt. Hierbei werden auch Gefrier- und Auftauvorgänge eingerechnet. Auch die Simulation von Schneeschichten ist möglich.

Ein weiterer Bestandteil des TSM ist ein Vegetations-

modell basierend auf dem in [7] beschriebenen Modell, das die Bestimmung der Temperatur von Vegetationsschichten unter Berücksichtigung des Energieaustauschs mit der Umgebung erlaubt. Das Modell berücksichtigt auch die physiologischen Einflüsse auf die Temperatur der Vegetationsschicht (Stomatawiderstand, Photosynthese).

Vom Anwender können die Schichten sowohl bezüglich ihrer Dicken als auch ihrer Materialien und deren physikalischen Eigenschaften eingestellt werden. Damit sind die Temperaturen von beliebigen natürlichen Oberflächen, auch unter Berücksichtigung ihrer Orientierung zur Sonne, für vorgegebene Zeiträume berechenbar und können in der EMIT-Datenbasis für IR-Simulationen gespeichert werden.

Das TSM wurde mit Hilfe von Messdaten validiert. Bild 10 zeigt einen Vergleich der mit TSM berechneten Temperaturen für eine Schotteroberfläche mit den durchgeführten Messungen:

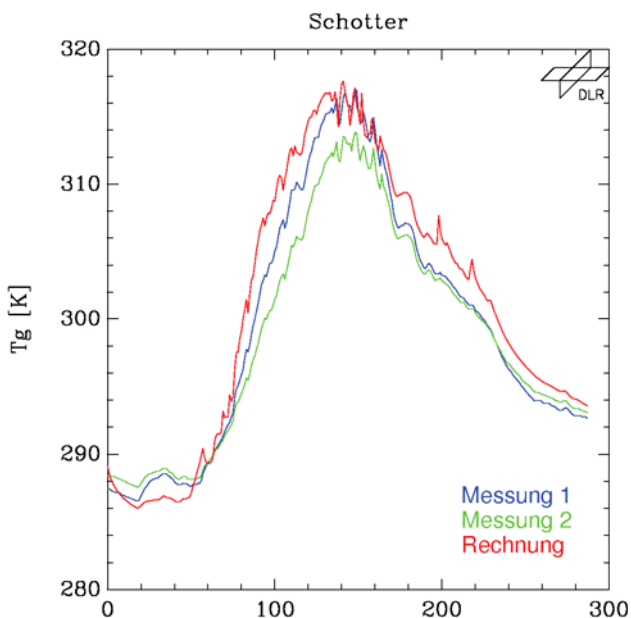


BILD 10. TSM-Temperaturen für Schotter im Vergleich zu gemessenen Temperaturen

Die Validierung und Entwicklung des TSM wird derzeit fortgeführt.

## 5. SUMMARY

LFK-Lenkflugkörpersysteme GmbH hat langjährige Erfahrungen mit kommerziellen Infrarot-Bildgeneriersystemen in verschiedenen Phasen der Lenkflugkörperentwicklung gesammelt. Im Laufe der Jahre tauchte die Notwendigkeit auf, volle Kontrolle über den Bildgenerierprozess zu gewinnen, unabhängig von Exportrestriktionen zu werden, sowie neue Anforderungen in kurzer Zeit unabhängig von Entwicklungszyklen eines Herstellers zu implementieren. Dies führte im Jahr 2003 zum Start der Entwicklung von EMIT, dem **Extensible Multi-spectral Image Generation Tool**. Es wurde bisher in diversen Lenkflugkörper-Projekten (z.B. PARS3 LR, Taurus KEPD350, Milan) eingesetzt. In Kooperation mit

Hochschulen und Instituten wird EMIT ständig verbessert und um neue Funktionalitäten erweitert.

## 6. SCHRIFTTUM

- [1] MODTRAN - MODerate spectral resolution atmospheric TRANsmittance algorithm and computer model, developed by US Airforce Research Laboratories/VSBT in collaboration with Spectral Sciences, Inc.  
The code is distributed by Ontar Corporation.
- [2] R. Osfield, M. Weiblen, OpenSceneGraph - Past, Present and Future, SIGGRAPH, 2006
- [3] P. Schaetz, Thermos - A Thermal Model for the Use of Synthetic Scenarios in IR Simulations, proceedings of IR Target and Background Modelling and Simulation (ITBM&S), FGAN-FOM, Ettlingen, Germany, 2005
- [4] P. Obermeier, EMIT - An Infrastructure for Infrared Image Generation, Proceedings of IR Target and Background Modelling and Simulation (ITBM&S), FGAN-FOM, Ettlingen, Germany, 2005.
- [5] R. Wiedemann, Generation of IR Images from Synthetical 3D-Scenarios Using Programmable Graphics Cards, proceedings of IR Target and Background Modelling and Simulation (ITBM&S), FGAN-FOM, Ettlingen, Germany, 2006.
- [6] S. Lohmann, Man-In-the-Loop-Simulator: Aufbau und erste Ergebnisse eines Simulators zur Online-Führung eines Lenkflugkörpers unter Verwendung von Suchkopfbildern. DGLR Kongress 2009, Aachen, Germany, 2009.
- [7] The Common Land Model, Y. Dai et al., American Meteorological Society, August 2003