Computer-Graphik II - Blatt 5

Ray-Tracing – Framework

Prof. Dr. Gabriel Zachmann (zach@in.tu-clausthal.de)
David Mainzer (davidmainzer@gmx.net)

Institut für Informatik
11. Juni 2008

Gliederung

Allgemeine Informationen

- Wichtige Klassen

Details

- Source-Code – wichtige Funktionen
- Beispiel – Erzeugung eines Strahls

Ihre Aufgabe

Gliederung

Framework

GUI

- Basierend auf Qt
- 2 Fenster, links Szene in OpenGL und rechts Ergebnisbild des Raytracing
- Ansicht dient dem Debugging, Modifikation der Szene und Veranschaulichung des Raytracing-Prozesses

xml-Parser

- Einlesen Szene (xml-Format)
- Windows: fertige dll; Linux: Unterverzeichnis xmlParser inkl. Makefile
- Beispiel für xml-Datei liegen bei

Framework

Mathematische Hilfsklassen

- VectorT: Template für n-dimensionale Vektorarithmetik
- MatrixT: Template für quadratische n×n-Matrizen
- Matrix33T: Spezialisierung von MatrixT für 3×3-Matrizen
- ColorT: Template für RGB-Farbarithmetik

Raytracer I

- Grundfunktionen sind in Klasse Raytracer konzentriert
- Raytracer::render() erzeugt Strahlen für einzelne Pixel
- Raytracer::traceRay() verfolgt Strahl durch Szene

Framework

Raytracer II

- Raytracer::shade() wertet lokale Beleuchtungsmodell in einem Punkt der Szene aus; des Weiteren wir überprüft ob der Punkt im Schatten liegt (von Ihnen zu implementieren)
- Klasse PinholeCamera implementiert einfach Lochkamera
- PinholeCamera::generateRay() generiert für gegebenen Pixel (x,y) Strahl durch Augpunkt
- Klasse Ray dient der Repräsentation eines solchen Strahls – definiert durch Startpunkt und Richtung
- Bietet auch Funktionen zur Berechnung gebrochener und reflektierter Strahlen

Framework

Szenendefinition

- Durch virtuelle Basisklassen realisiert
- Surface virtuelle Basisklasse für geometrische Objekte; Funktion intersect() dient Berechnung Schnitt mit Strahl
- Shader virtuelle Basisklasse für Materialien
- Light virtuelle Basisklasse für Lichtquellen

Gliederung

Allgemeine Informationen
- Wichtige Klassen

Details
- Source-Code – wichtige Funktionen
- Beispiel – Erzeugung eines Strahls

Ihre Aufgabe

Wichtige Funktionen

- Hauptfunktion zur Erzeugung eines Strahls – Raytracer.cpp

```
void Raytracer::render( SurfaceList *pclScene,
        unsigned int uiMaxDepth, QImage *clQImage,
        RenderingMode mode )
```

- Funktion zur Verfolgung des Strahls – Raytracer.cpp

```
ColorType Raytracer::traceRay( const Surface *pclScene,
        const Ray &crclRay, unsigned int uiDepth,
        std::vector<GLRay> *glRays )
```

- PinholeCamera (x,y) - Koordinate der Loch Maske

```
Ray PinholeCamera::generateRay( double x, double y,
        double rWidth, double rHeight ) const
```

Beispiel

- Erzeugen eines Strahls

```
Ray ray = getCamera().generateRay( i, j, clQImage->width(),
      clQImage->height() );
```

- Verfolgung dieses Strahls durch die Szene; Rückgabe der Farbe

```
ColorType col = traceRay( pclScene, ray, uiMaxDepth, glRays );
```

Gliederung

Allgemeine Informationen
- Wichtige Klassen

Details
- Source-Code – wichtige Funktionen
- Beispiel – Erzeugung eines Strahls

Ihre Aufgabe

Ihre Aufgabe

Teil a)

- Implementieren Sie die Funktion Raytracer::shade() in Raytracer.cpp

- Funktion erzeugt Schattenstrahlen für Szene und (Phong-)Beleuchtung für entsprechende Pixel wird gesetzt (wenn nicht im Schatten)
- Funktion korrekt implementiert → Beleuchtung ungefähr so wie im OpenGL-Fenster (+zusätzlicher Schatten)

Ihre Aufgabe

Teil b)

- Implementieren Sie die Funktion Ray::reflectedRay() in Ray.cpp

- Erzeugt Reflektion
- Funktion soll aus Input-Parametern einen perfekt reflektierten Strahl berechnen
- Verwenden Sie diese Funktion um in Raytracer::traceRay() rekursiv den Farbanteil des gespiegelten Strahls zu berechnen und addieren Sie den erhaltenen Wert zum Farbwert hinzu

Ihre Aufgabe

Teil c)

- Implementieren Sie die Funktion Ray::refractedRay() in Ray.cpp

- Neben Reflexionen tragen gebrochene Strahlen zum realistischen Eindruck von Raytracing-Bildern bei
- Erzeugung eines durch Materialparameter bestimmten gebrochenen Strahls
- Verwenden Sie diese Funktion um in Raytracer::traceRay() rekursiv den Farbanteil des gespiegelten Strahls zu berechnen und addieren Sie den erhaltenen Wert zum Farbwert hinzu

Literatur I

- Andrew Glassner (ed.): *An Introduction to Ray Tracing*; Morgan Kaufman
- Peter Shirley: *Realistic Ray Tracing*; AK Peters.
- Matt Pharr, Greg Humpfreys: *Physically-Based Rendering*; Elsevier.