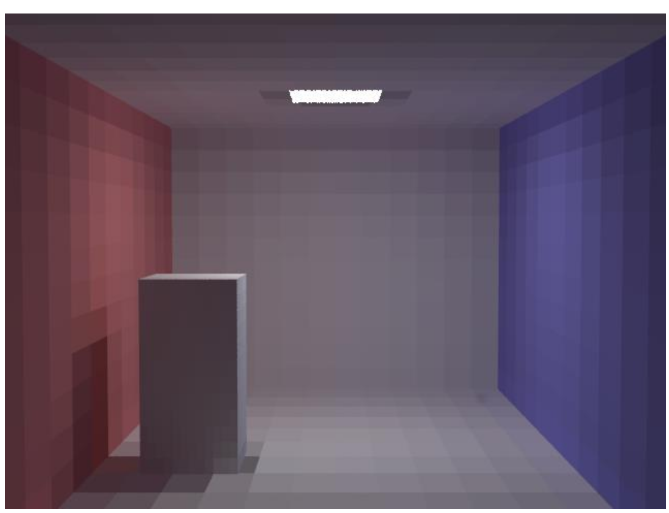
Advanced Computer Graphics Proseminar

Univ.-Prof. Dr. Matthias Harders

Winter semester 2015



Example Code Rendering



Programming Assignment 1

- Change geometric base element from rectangles (& rectangular patches) to triangles (& triangular patches)
- Change scene description to triangle meshes
- Implement ray-triangle intersection test
- Adapt Monte-Carlo integration

Ray-Triangle Intersection

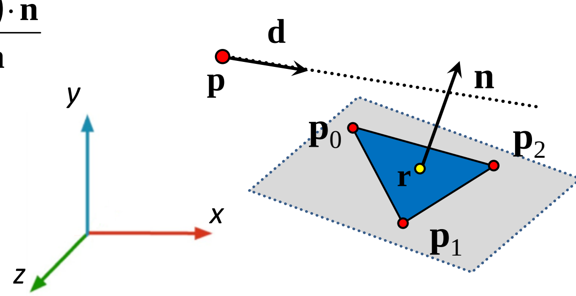
Option 1
- Intersection ray with triangle plane
- Test if (possible) intersection is inside triangle

Option 2 (Möller–Trumbore)
- Directly insert ray equation in parametric triangle form
- Solve using Cramer´s rule

Ray-Triangle Intersection – Option 1

- Ray-plane intersection test (plane given by triangle)

$$t = -\frac{(\mathbf{p}-\mathbf{r})\cdot\mathbf{n}}{\mathbf{d}\cdot\mathbf{n}}$$



$$\mathbf{n} = (\mathbf{p}_1 - \mathbf{p}_0)\times(\mathbf{p}_2 - \mathbf{p}_0)$$

$$\mathbf{r} = \mathbf{p}_i$$

Ray-Triangle Intersection – Option 1

- Test if (possible) intersection is inside triangle
- Point $\mathbf{q}_t$ at intersection of ray with plane

$$\mathbf{q}_t = \mathbf{p} + t\cdot\mathbf{d}$$

- Barycentric coordinates of arbitrary vertex $\mathbf{q}$ with respect to triangle (given by vertices $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$)

$$\begin{aligned}\mathbf{q} &= \lambda_0\mathbf{p}_0 + \lambda_1\mathbf{p}_1 + \lambda_2\mathbf{p}_2 \\ &= \lambda_0\mathbf{p}_0 + \lambda_1\mathbf{p}_1 + (1-\lambda_0-\lambda_1)\mathbf{p}_2\end{aligned} \qquad \lambda_0 + \lambda_1 + \lambda_2 = 1$$

- Point inside triangle

$$0 \le \lambda_{0,1,2} \le 1$$

Ray-Triangle Intersection – Option 1

- Determine λ_0 and λ_1 (note: $\lambda_2 = 1 - \lambda_0 - \lambda_1$)

$$\mathbf{q}_t = \lambda_0 \mathbf{p}_0 + \lambda_1 \mathbf{p}_1 + (1 - \lambda_0 - \lambda_1)\mathbf{p}_2$$

$$\begin{bmatrix} q_{t,x} \\ q_{t,y} \\ q_{t,z} \end{bmatrix} = \lambda_0 \begin{bmatrix} p_{0,x} \\ p_{0,y} \\ p_{0,z} \end{bmatrix} + \lambda_1 \begin{bmatrix} p_{1,x} \\ p_{1,y} \\ p_{1,z} \end{bmatrix} + (1 - \lambda_0 - \lambda_1) \begin{bmatrix} p_{2,x} \\ p_{2,y} \\ p_{2,z} \end{bmatrix}$$

$$q_{t,x} = \lambda_0 p_{0,x} + \lambda_1 p_{1,x} + (1 - \lambda_0 - \lambda_1)\, p_{2,x}$$

$$q_{t,y} = \lambda_0 p_{0,y} + \lambda_1 p_{1,y} + (1 - \lambda_0 - \lambda_1)\, p_{2,y}$$

Ray-Triangle Intersection – Option 1

- Determine λ_0 and λ_1 (note: $\lambda_2 = 1 - \lambda_0 - \lambda_1$)

$$\lambda_0 (p_{0,x} - p_{2,x}) + \lambda_1 (p_{1,x} - p_{2,x}) = q_{t,x} - p_{2,x}$$

$$\lambda_0 (p_{0,y} - p_{2,y}) + \lambda_1 (p_{1,y} - p_{2,y}) = q_{t,y} - p_{2,y}$$

$$\begin{bmatrix} p_{0,x} - p_{2,x} & p_{1,x} - p_{2,x} \\ p_{0,y} - p_{2,y} & p_{1,y} - p_{2,y} \end{bmatrix} \cdot \begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} q_{t,x} - p_{2,x} \\ q_{t,y} - p_{2,y} \end{bmatrix}$$

$$\begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix} = \mathbf{T}^{-1} \cdot \begin{bmatrix} q_{t,x} - p_{2,x} \\ q_{t,y} - p_{2,y} \end{bmatrix}$$

Ray-Triangle Intersection – Option 1

- Analytic inverse of invertible 2×2 matrix $\mathbf{A}$

$$\mathbf{A}^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{\det \mathbf{A}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

$$\begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix} = \frac{1}{\det \mathbf{T}} \begin{bmatrix} p_{1,y} - p_{2,y} & p_{2,x} - p_{1,x} \\ p_{2,y} - p_{0,y} & p_{0,x} - p_{2,x} \end{bmatrix} \cdot \begin{bmatrix} q_{t,x} - p_{2,x} \\ q_{t,y} - p_{2,y} \end{bmatrix}$$

Ray-Triangle Intersection – Option 1

- Determine λ_0 and λ_1 (note: $\lambda_2 = 1 - \lambda_0 - \lambda_1$)

$$\lambda_0 = \frac{1}{\det \mathbf{T}} \left(\left(p_{1,y} - p_{2,y}\right) \cdot \left(q_{t,x} - p_{2,x}\right) + \left(p_{2,x} - p_{1,x}\right) \cdot \left(q_{t,y} - p_{2,y}\right) \right)$$

$$\lambda_0 = \frac{\left(p_{1,y} - p_{2,y}\right) \cdot \left(q_{t,x} - p_{2,x}\right) + \left(p_{2,x} - p_{1,x}\right) \cdot \left(q_{t,y} - p_{2,y}\right)}{\left(p_{0,x} - p_{2,x}\right) \cdot \left(p_{1,y} - p_{2,y}\right) - \left(p_{0,y} - p_{2,y}\right) \cdot \left(p_{1,x} - p_{2,x}\right)}$$

$$\lambda_1 = \frac{\left(p_{2,y} - p_{0,y}\right) \cdot \left(q_{t,x} - p_{2,x}\right) + \left(p_{0,x} - p_{2,x}\right) \cdot \left(q_{t,y} - p_{2,y}\right)}{\left(p_{0,x} - p_{2,x}\right) \cdot \left(p_{1,y} - p_{2,y}\right) - \left(p_{0,y} - p_{2,y}\right) \cdot \left(p_{1,x} - p_{2,x}\right)}$$

Ray-Triangle Intersection – Option 2

- Proposed by Möller and Trumbore (1998)
- Avoids explicit computation of plane equation
- Faster computation and less memory consumption
- Directly insert ray equation in triangle equation

$$\mathbf{p} + t\mathbf{d} = \lambda_0 \mathbf{p}_0 + \lambda_1 \mathbf{p}_1 + (1 - \lambda_0 - \lambda_1)\mathbf{p}_2$$

$$\lambda_0 + \lambda_1 + \lambda_2 = 1$$

Ray-Triangle Intersection – Option 2

- Determine t, λ_0 and λ_1 (note: $\lambda_2 = 1 - \lambda_0 - \lambda_1$)

$$\mathbf{p} + t\mathbf{d} = \lambda_0 \mathbf{p}_0 + \lambda_1 \mathbf{p}_1 + (1 - \lambda_0 - \lambda_1)\mathbf{p}_2$$

$$\mathbf{p} - \mathbf{p}_2 = -t\mathbf{d} + \lambda_0(\mathbf{p}_0 - \mathbf{p}_2) + \lambda_1(\mathbf{p}_1 - \mathbf{p}_2)$$

$$\begin{bmatrix} -\mathbf{d} & \mathbf{e}_0 & \mathbf{e}_1 \end{bmatrix} \cdot \begin{bmatrix} t \\ \lambda_0 \\ \lambda_1 \end{bmatrix} = \mathbf{s}$$

Ray-Triangle Intersection – Option 2

- Cramer´s Rule for solving linear system of equations

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Ray-Triangle Intersection – Option 2

- Cramer´s Rule for solving linear system of equations

$$x_i = \frac{\det \mathbf{A}_i}{\det \mathbf{A}}$$

$$\mathbf{A}_i = \begin{bmatrix} a_{11} & \cdots & a_{1i-1} & b_1 & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2i-1} & b_2 & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{ni-1} & b_n & \cdots & a_{nn} \end{bmatrix}$$

(computationally inefficient, e.g. expansion $O(n \cdot n!)$)

Ray-Triangle Intersection – Option 2

- Determine t, λ_0 and λ_1 (note: $\lambda_2 = 1 - \lambda_0 - \lambda_1$)

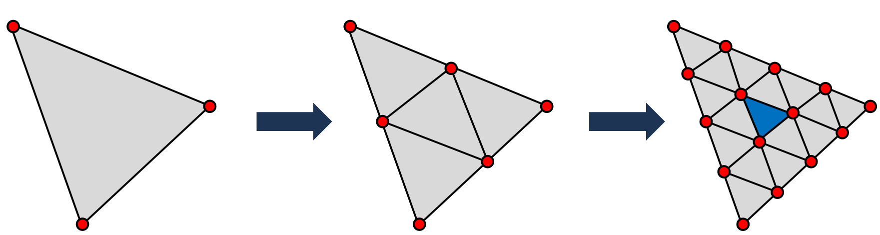
$$t = \frac{1}{\det \mathbf{T}} \det[\mathbf{s} \quad \mathbf{e}_0 \quad \mathbf{e}_1] \qquad \det[\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}] = -(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$$

$$\lambda_0 = \frac{1}{\det \mathbf{T}} \det[-\mathbf{d} \quad \mathbf{s} \quad \mathbf{e}_1]$$

$$\lambda_1 = \frac{1}{\det \mathbf{T}} \det[-\mathbf{d} \quad \mathbf{e}_0 \quad \mathbf{s}]$$

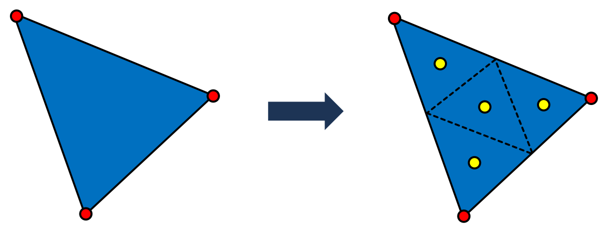
Triangular Patches

- Subdivide triangle by splitting each edge in half and connecting new vertices



Sampling Triangular Patches

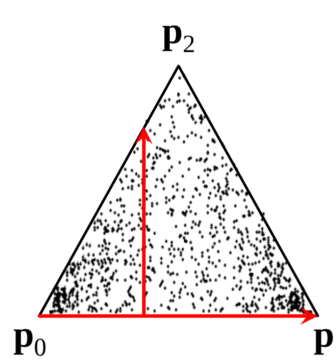
- Monte-Carlo integration requires (random) sampling
- Approximate by (suboptimal) regular sampling



Sampling Triangular Patches

- Random sampling of triangles using uniformly distributed random variable(s) $\xi \in [0,1]$
- Careful sampling strategy required for correct result
- Incorrect approach

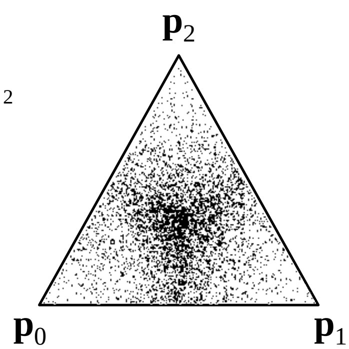
$$\mathbf{q} = \xi_0(\mathbf{p}_1 - \mathbf{p}_0) + \xi_1 \mathbf{h}(\xi_0)$$



Sampling Triangular Patches

- Random sampling of triangles using uniformly distributed random variable(s) $\xi \in [0,1]$
- Careful sampling strategy required for correct result
- Incorrect approach

$$\mathbf{q} = \xi_0 \mathbf{p}_0 + \xi_1 \mathbf{p}_1 + (1 - \xi_0 - \xi_1)\mathbf{p}_2$$



Sampling Triangular Patches

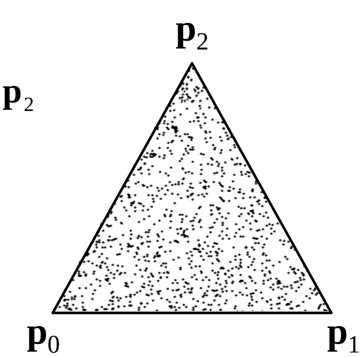
- Random sampling of triangles using uniformly distributed random variable(s) $\xi \in [0,1]$
- Careful sampling strategy required for correct result
- Correct approach

$$\mathbf{q} = \lambda_0 \mathbf{p}_0 + \lambda_1 \mathbf{p}_1 + (1 - \lambda_0 - \lambda_1)\mathbf{p}_2$$

$$\lambda_0 = 1 - \sqrt{\xi_0}$$

$$\lambda_1 = \xi_1 \sqrt{\xi_0}$$

$$\lambda_2 = 1 - \lambda_0 - \lambda_1 = \sqrt{\xi_0}(1 - \xi_1)$$



Proseminar Schedule

Date	Topic	Remark
12.10.	Introduction	
19.10.	Theory – Radiometry	Radiosity example code
26.10.	*(no proseminar - Nationalfeiertag)*	
2.11.	*(no proseminar - Allerseelen)*	
9.11.	Discussion of Radiosity code	Programming assignment 1
16.11.	Programming support and advice	
23.11.	Theory – Random sampling	Path Tracer example
30.11.	Discussion of Path Tracer code	Programming assignment 2, *Hand-in PA1*
7.12.	Presentation of solutions	
14.12.	Programming support and advice	*Project proposal (21.12. Hand-in PA2)*
	Christmas break	
14.1.	Presentation of solutions	
21.1.	Geometric Modelling	
28.1.	Programming support and advice	
4.2.	Project presentation	*Submission final project*