Advanced Computer Graphics Proseminar

Univ.-Prof. Dr. Matthias Harders

Winter semester 2015



Goals of the Proseminar

- Introduction to photo-realistic image synthesis
- Rehearsal of theoretical concepts taught in lecture
- Practical global illumination (GI) programming
- Project work and presentation of results

Requirements

- Differential and integral calculus
- Linear algebra & trigonometry
- C/C++ knowledge
- Basic data structures
- OS Linux

Organisation

- Location: RR 21
- Mode: Continuously assessed (“Prüfungsimmanent”)
- Time: Mondays, 14:15–15:00
- Credit points: 2 ECTS-AP
- Cancellation: drop out possible until November 23rd

Accreditation

- Lecture and Proseminar officially listed under "Interdisciplinary Studies" (Interdisziplinäre Kompetenzen)
- Can be accredited as "Elective Course" (Wahlmodul)
- Submit form for accreditation to A.-M. Scheiring
- More details on http://informatik.uibk.ac.at/anrechnungen

Marking Scheme

Components

1) Two programming assignments
2) Final rendering project
3) Summary report and presentation
4) Theoretical exercises
5) Active participation

Programming Tasks

- Work in teams of 2-3 students (announce teams)
- Programming assignments (November 9th and 30th), focusing on Radiosity and Path Tracing
- Example code available through course webpage
- 3 weeks for solution
- Support and remarks provided in Proseminar

Programming Tasks

- Programs must compile and run in Linux in RR21
- Hand-in code electronically before deadline in OLAT (more details on this later)
- Provide brief information on program usage in README or in comments
- Presentation of solution in Proseminar

Programming Tasks

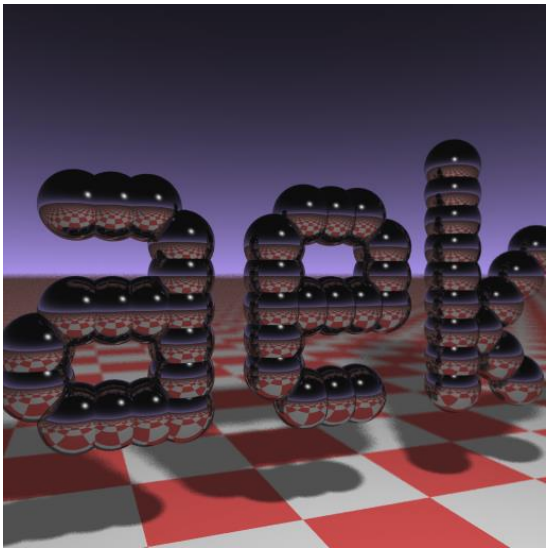
- Final rendering project for second half of semester
- Possible topics/extensions provided in proseminar
- Final project outline (until December 14th, 2015)
- Deadline for project (February 1st, 2016)
- Project presentation (February 1st, 2016)
- Summary report (February 8th, 2016)
- Coding style will be part of the grade

Coding Style Remarks

[A. Kensler]

```
#include <stdlib.h>   // card > aek.ppm
#include <stdio.h>
#include <math.h>
typedef int i;typedef float f;struct v{
f x,y,z;v operator+(v r){return v(x+r.x
,y+r.y,z+r.z);}v operator*(f r){return
v(x*r,y*r,z*r);}f operator%(v r){return
x*r.x+y*r.y+z*r.z;}v(){}v operator^(v r
){return v(y*r.z-z*r.y,z*r.x-x*r.z,x*r.
y-y*r.x);}v(f a,f b,f c){x=a;y=b;z=c;}v
operator!(){return*this*(1/sqrt(*this%*
this));}};i G[]={247570,280596,280600,
249748,18578,18577,231184,16,16};f R(){
return(f)rand()/RAND_MAX;}i T(v o,v d,f
&t,v&n){t=1e9;i m=0;f p=-o.z/d.z;if(.01
<p)t=p,n=v(0,0,1),m=1;for(i k=19;k--;)
for(i j=9;j--;)if(G[j]&1<<k){v p=o+v(-k
,0,-j-4);f b=p%d,c=p%p-1,q=b*b-c;if(q>0
){f s=-b-sqrt(q);if(s<t&&s>.01)t=s,n=!(
p+d*t),m=2;}}return m;}v S(v o,v d){f t
;v n;i m=T(o,d,t,n);if(!m)return v(.7,
.6,1)*pow(1-d.z,4);v h=o+d*t,l=!(v(9+R(
),9+R(),16)+h*-1),r=d+n*(n%d*-2);f b=l%
n;if(b<0||T(h,l,t,n))b=0;f p=pow(l%r*(b
>0),99);if(m&1){h=h*.2;return((i)(ceil(
h.x)+ceil(h.y))&1?v(3,1,1):v(3,3,3))*(b
*.2+.1);}return v(p,p,p)+S(h,r)*.5;}i
main(){printf("P6 512 512 255 ");v g=!v
(-6,-16,0),a=!(v(0,0,1)^g)*.002,b=!(g^a
)*.002,c=(a+b)*-256+g;for(i y=512;y--;)
for(i x=512;x--;){v p(13,13,13);for(i r
=64;r--;){v t=a*(R()-.5)*99+b*(R()-.5)*
99;p=S(v(17,16,8)+t,!(t*-1+(a*(R()+x)+b
*(y+R())+c)*16))*3.5+p;}printf("%c%c%c"
,(i)p.x,(i)p.y,(i)p.z);}}
```

Coding Style Remarks



[A. Kensler]

Honor Code

- Feel free to discuss, but develop your own solution
- Clearly acknowledge external sources
- Plagiarism and collusion will result in 0 marks, and possibly further investigation
- Plagiarism is a serious offense in the academic (and any other) environment

Final Grade

- Marks
  - Two programming assignments: 40
  - Final project and report: 50
  - Active participation/presentation: 10
- Grading key (Sum of points : Final grade)

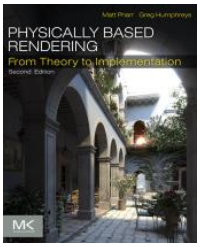
0 – 49	:	5
50 – 59	:	4
60 – 74	:	3
75 – 89	:	2
90 – 100	:	1

Material

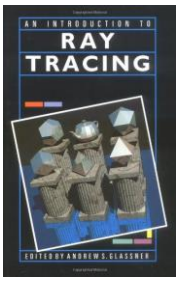
- Information, news, example code available on lecture webpage http://igs.uibk.ac.at/ (→ Teaching)
- General book recommendations
- Also, feel free to contact the instructor

Literature

- M. Pharr, G. Humphreys, “Physically-Based Rendering”, Morgan Kaufmann, 2nd edition, 978-0123750792, 2010. http://www.pbrt.org



- A. Glassner et al., “Introduction to Ray Tracing”, Academic Press, 1st edition, 978-0122861604, 1989.



Contact Details

Prof. Dr. Matthias Harders

- Email: matthias.harders@uibk.ac.at
- Phone: +43 (0)512 507 53338
- Office: ICT 3N09
- Office hours: by appointment

Content

- Organization
- Overview of global illumination packages
- Example rendering system: Mitsuba renderer

Usage of Global Illumination Renderers

- Animated films

[Dreamworks/PDI – *Renderman*, 2001]

[Pixar – *Renderman*, 2001]

Usage of Global Illumination Renderers

- Movie special effects

[Weta Digital – *Manuka*, 2014]

[Framestore – *Arnold*, 2013]

Usage of Global Illumination Renderers

- Commercials, virtual photographs

[Surreal Structures – *Arion*, 2011]

[Protograph – *Keyshot*, 2014]

Selected Commercial Renderers

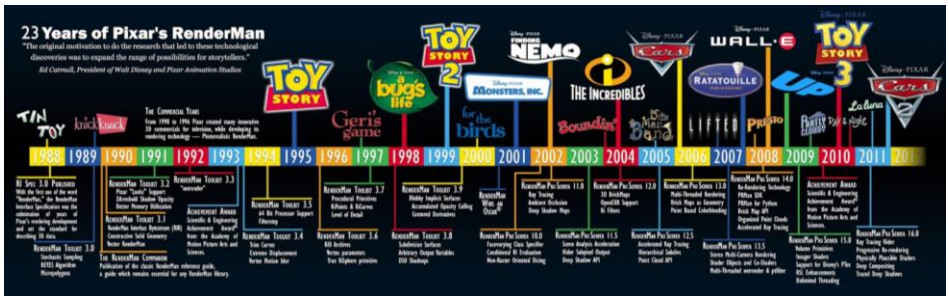
- *Arnold* – Solid Angle (www.solidangle.com)
- *V-Ray* – Chaos Group (www.chaosgroup.com)
- *Maxwell Render* – Next Limit (www.maxwellrender.com)
- *Mental Ray* – Nvidia (www.nvidia-arc.com/mentalray)
- *Octane* – Otoy (home.otoy.com)
- *KeyShot* – Luxion (www.keyshot.com)
- …

Selected Free/Open Source Renderers

- *Mistuba Render* (www.mitsuba-renderer.org)
- *pbrt* (www.pbrt.org)
- *LuxRender* (www.luxrender.net)
- *POVRay* (www.povray.org)
- …

- *RenderMan* – Pixar (renderman.pixar.com) (non-commercial use)

RenderMan



Mitsuba Renderer

- Developed by W. Jakob (now at ETH Zurich) https://www.mitsuba-renderer.org/
- Research-oriented system implementing different global illumination approaches
- Written in C++, provided as free software (GNU GPL)
- Executables for Windows, Linux, Mac OS
- Command-line and graphical user interface (`mtsgui`)

Mitsuba
Physically Based Renderer

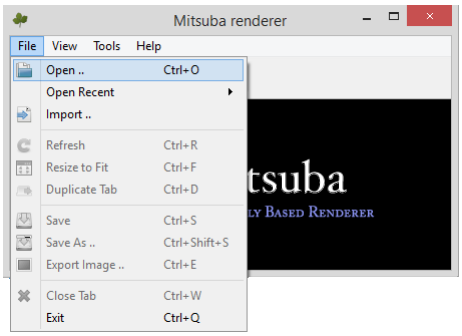
Mitsuba Features

- Offline physically-based rendering
- Scene descriptions in XML-based format
- Can also import COLLADA 1.4 & Wavefront OBJ scenes
- Real-time scene preview (using virtual point light), on appropriate GPU hardware
- Left mouse button & arrows keys control camera
- Documentation available online
  https://www.mitsuba-renderer.org/docs.html

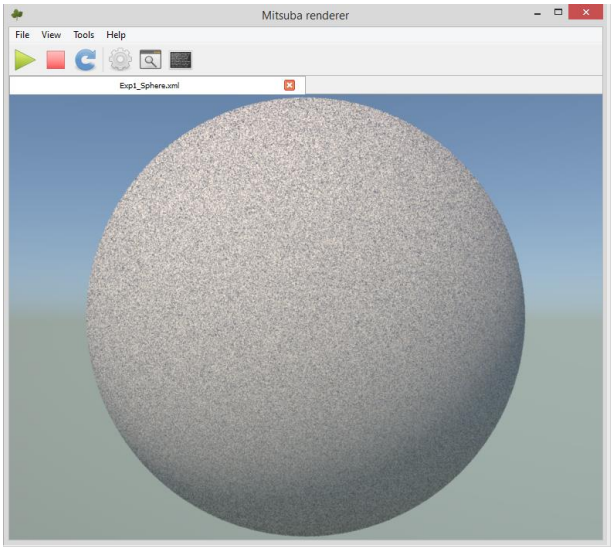
Getting Started

- Download example ZIP file from OLAT
- Set path: `source MitsubaPaths.source`
- Start GUI: `mtsgui`
- Open first example:

  `Exp1_Sphere.xml`



First Basic Example



First Basic Example

```
<?xml version="1.0" encoding="utf-8"?>
<scene version="0.5.0">
    <shape type="sphere">
        <float name="radius" value="1"/>
    </shape>
</scene>
```

Note: perspective camera, integrator, etc. automatically set to default

Loading Meshes

```
<?xml version="1.0" encoding="utf-8"?>
<scene version="0.5.0">
    <shape type="obj">
        <string name="filename" value="teapot.obj"/>
    </shape>
</scene>
```


Defining Material

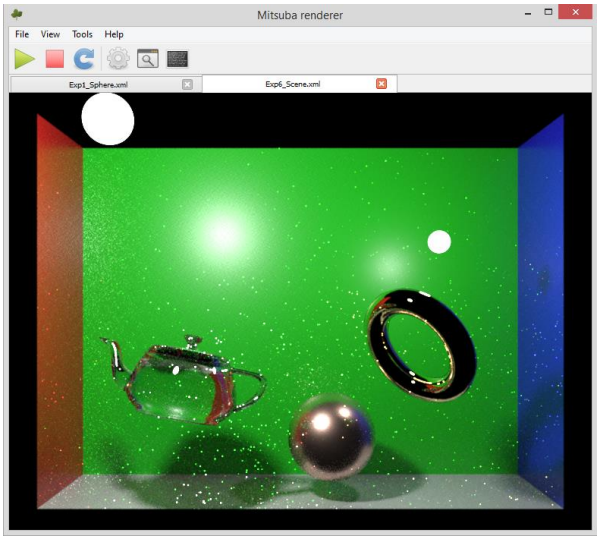
```
<?xml version="1.0" encoding="utf-8"?>
<scene version="0.5.0">
    <shape type="obj">
        <string name="filename" value="teapot.obj"/>
        <bsdf type="roughplastic">
            <float name="alpha" value="0.1"/>
        </bsdf>
    </shape>
</scene>
```

Bidirectional scattering distribution function

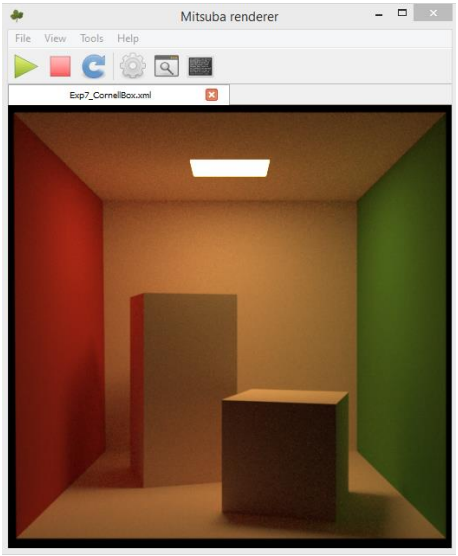
Setting Integrator, Camera, and Sampling

```
…
    <integrator type="path"/>
    <sensor type="perspective">
        <transform name="toWorld">
            <lookat origin="0.217,1.575,-9.76653"
                    target="0.217,1.575,0"
                    up="0,1,0"/>
        </transform>

        <float name="fov" value="45"/>

        <sampler type="halton">
            <integer name="sampleCount" value="32"/>
        </sampler>
    </sensor>
…
```


Complex Scene



Cornell Box



Tasks for Next Week

- Form teams of 2-3 students, email names
- Create an interesting rendering with Mitsuba (send image, short explanation via email)
- Online article on state of CGI rendering: http://www.fxguide.com/featured/the-art-of-rendering/
- After next lecture: download theoretical exercise sheet and think about possible solutions

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.	Presentation of solutions	Path Tracer example
30.11.	Discussion of Path Tracer code	Programming assignment 2, *Hand-in PA1*
7.12.	Programming support and advice	
14.12.	Presentation of solutions	*Project proposal* *(21.12. Hand-in PA2)*
Christmas break		
14.1.	Geometric Modelling	
21.1.	Procedural Modelling	
28.1.	Programming support and advice	
4.2.	Project presentation	*Submission final project*