782 Schedule & Notes

Tentative schedule - subject to change at a moment's notice. This is only a guide and not meant to be a strict schedule of how fast the material will be taught. The order of material is fairly set for the early part of the course, but becomes more fluid as the quarter progresses.

Lectures	Dates	Topic	Chapters
1	Sept 21	Introduction	1
1	Sept 23	Basic ray tracing review	1
2	Sept. 26, 28	PBRT	1
3	Sept. 30, Oct. 3, 5	Shapes	3-4
2	Oct. 7, 10	Color and radiometry	5
3	Oct. 12, 14, 17	Tone reproduction	8
1	Oct. 19	Reflection models	9
1	Oct. 21	Materials	10
1	Oct. 24	Textures	11
1	Oct. 26	Sampling	7.4-7.6
2	Oct. 28, 31	Monte Carlo Fundamentals	13-14
2	Nov. 2, 4	Monte Carlo Integration	15.1-15.6
5	Nov. 7, 9, 11, 14, 16	Light Transport	16-17
1	Nov. 18	Sub-surface scattering	18.2.3
3	Nov. 21, 23, 28	Volume rendering	12, 15.7
4	Nov. 30, Dec. 2, 5, 7	Radiosity	Cohen & Wallace

Introduction

```
Reading: Syllabus; Section 1.1, pp. 1-4; Section 1.5, pp. 38-39;
         Lab  0 Assignment; pbrt setup;
Lectures: 1
```

- Introduction to class
  - instructor, grader
  - overview of course
- Roster
- Syllabus
  - course description
  - prereqs
  - Book
  - grade components
  - Course software - pbrt
  - grading
  - academic misconduct
- Course software - pbrt
  - Read 1.1: Literate programming
  - look at installation notes: pbrtsetup.html
- Labs
  - overview
  - Lab 0 - specifics
  - getting started with course software

Basic Ray Tracing

```
Reading: Section 1.2, pp. 4-16
Lectures: 1
```

Basic idea of ray tracing

- perspective by projecting rays through virtual pixels
- bounce rays around environment
- recursive ray tracing
    - initial ray spawns other rays
    - SHADE(origin,ray)
- photorealistic rendering

Steps of basic ray tracing

- camera model
    - set up camera coordinate system
    - pinhole camera: plane in-front or behind camera
    - view volume
    - establish virtual frame buffer: position pixel plane on view plane
    - sample frame buffer: generate rays
- intersection objects
    - parametric form of ray: origin + t*direction
    - spheres: algebraic v. geometric
    - polygon intersection
- illuminate point of intersection
    - point lights
    - area lights
    - light energy
    - distance falloff
    - tilted surface
    - shadows: shadow rays
- surface scattering
    - Bidirectional Reflectance Distribution Function (BRDF)
    - Bidirectional Transmission Distribution Function (BTDF)
    - Bidirectional Scattering Distribution Function (BSDF)
- Recursive Ray Tracing
    - Light Transport Equation (aka rendering equation
    - Whitted’s algorithm
    - Reflection ray
    - Refraction ray
- Ray propagation
    - participating medium
    - volume light transport equation

Refine ray tracing

- speed up intersection testing
    - bounding volumes
    - partition space
- sample pixel: pixel as an area of screen
- sample area light source: how much is visible from light source?
- surface scattering: interaction of ray and material

PBRT

```
Reading: Section 1.3, pp. 16-36
Lectures: 2
```

NOTE: refer to the book for actual code segments; the ones here are condensed versions

The PBRT implementation of ray tracing

- Phases of execution
  - parse -> instance of the Scene class
  - rendering: Scene::Render() determines light at film plane using rays
  - postprocessing
- Scene representation

```
int main() {
  pbrtInit()
  process scene description
  pbrtCleanup()
}
```

- Process scene file
  - geometric objects, Scene::aggregate
  - lights
  - camera
  - participating media: Volume region
  - Surface integrator
  - volume integrator
  - sampler: choose points on image planes to trace through
- Main rendering loop

```
Render() {
  allocate and initialize sample
  allow integrators to pre-process scene
  trace rays: main loop
  clean-up
}
<
```

- Scene Methods
  - Render - holds variables that represent the scene; forwards requests to methods of Scene's member variables
  - Intersect - traces the given ray; return boolean of whether intersection and fills in Intersection structure with info
  - IntersectP - checks for existence of intersection only
  - WorldBound - returns 3D bounding box of Scene:aggregate
  - Li - compute radiance along a ray
  - Transmittance - returns attenuation due to participating media
- Integrator for Whitted-style ray tracing (1.3.5, pp. 30-36)
  See Figure 1.16,, p. 32

Shapes

```
Reading: Chapter 2 as needed to review geometry and transformations
         Sections 3.0-3.6; pp. 89-133
         Sections 4.0-4.2 pp. 169-180
Lectures: 3
```

- Primitives:
  - Basic Shape Interface
    - ObjectBound, WorldBound
    - CanIntersect()
    - Intersect()
    - IntersectP()
    - Refine()
    - GetShadingGeometry()
    - Area()
  - Spheres
  - Cylinders
  - Disks
  - Other Quadrics
  - Triangles and Meshes
- GeometricPrimitive class: represents a single shape and combines shape with material properties
  - also holds AreaLight if it emits light
  - Intersect ()- pass on to shape:intersect
  - GetBSDF(): forwarded to Material
- InstancePrimitive provides instances; each has its own transform
  - Intersect(): forwarded on to the shared primitive
  - WorldBound(): transformed shared primitive's WorldBound
- Aggregate class - container for acceleration structures (grid & k-d tree)

Color and Radiometry

```
Reading: Chapter 5
Lectures: 2
```

- radiometry - propagation of EM radiation; wavelengths 370-730 nm visible
  SPD - spectral power distribution - amount of light at each wavelength
- Spectral Representation
  - Spectrum class
  - XYZ Color
  - tristimulus theory of color perception
- Basic Radiometry

- radiation transfer
- geometric optics
- Basic Quantities

  - Flux - power
  - Irradiance - area density of flux
  - Solid Angle & Intensity: steradians, flux energy / solid angle
  - Radiance flux per unit area per solid angle
  - incident & exident radiance functions

- Radiance Integrals

  - projected solid angle
  - integrate over area

- Surface Reflectance & BDRF >

Tone Reproduction

```
Reading: Chapter 8
Lectures: 3
```

- Film interface
- Image film
- Image pipeline
- Perceptual issues and tone mapping
- Final imaging pipeline states

Reflection Models

```
Reading: Chapter 9
Lectures: 1
```

- Basic interface
- Specular reflection and transmission
- Lambertian reflection
- Microfacet models
- Lafortune model
- Fresnel incidence effects

Materials

```
Reading: Chapter 10
Lectures: 1
```

- BSDFs
- Material interface and implementations
- Bump mapping

Texture

```
Reading: Chapter 11
Lectures: 1
```

- Sampling and aliasing
- Texture coordinate generation
- Texture interface and basic textures
- Image texture
- Solid and procedural texturing
- Noise

Sampling

Reading: Chapter 7.4-7.6
Lectures: 1

Monte Carlo Fundamentals

Reading: Chapter 13-14
Lectures: 2

Monte Carlo Integration

Reading: Chapter 15.1-15.6
Lectures: 2

Light

Reading: Chapter 16-17
Lectures: 5

Sub-Surface Scattering

Reading: Chapter 18.2.3
Lectures: 1

Volume Rendering

Reading: Chapter 12, 15.7
Lectures: 3

Radiosity

Reading: Chapter 17, Cohen & Wallace
Lectures: 2