Ray-tracing in GrCis

© 2016-2018 Josef Pelikán
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz
http://cgg.mff.cuni.cz/~pepca/
Ray-tracing–related projects

- old, simple
  - 018raycasting, 019raytracing

- best demo
  - 048rtmontecarlo, 048rtmontecarlo-script, 049distributedrt
    - switches for super-sampling, shadows, reflections, refractions, multi-threading, CS-script scene definition

- animation
  - 046camerananim
    - camera animation (going round the scene)
  - 062animation, 062animation-script
    - more general project, able to animate any scene part
Ray-tracing application

SimpleImageSynthesizer

RayTracing

Bitmap

Form

“scene geometry”

“light sources”

“camera”

“background color”

width [px]

Height [px]

Gamma

Adaptive
Image function [IImageFunction]

```c
interface II mageFunction

double Width
double Height
long GetSample ( double x, double y, double[] color )
```

```c
[0,0] x

double[] color ..
double[3]  // RGB
double[len]  // spectral color
```

```
[Width,Height]
```

```
Y H E L L O W O R L D
```

RayCasting, RayTracing

RayTracing : RayCasting
    [IImageFunction]

Scene
    [IRayScene]

"scene geometry"
    [INtersectable]

"light sources"
    [ICollection<ILightSource>]

Width [double]
Height [double]
MaxLevel [int]
MinImportance [double]
DoReflections [bool]
DoRefractions [bool]
DoShadows [bool]

BackgroundColor double[]
Camera [ICamera]

```
[interface ICamera]

double AspectRatio
double Width
double Height

bool GetRay (double x, double y,
out Vector3D p0, out Vector3D p1)
```

Ray: \( P_0 + t \cdot p_1 \)  
\( 0 \leq t \)
Light source [ILightSource]

```
[interface ILightSource]

double[] GetIntensity ( Intersection intersection, 
    out Vector3D dir )
```

- **dir**: direction toward the light, zero for omnidirectional
- **intersection**: point of intersection
- **return**: color (intensity)
IReflectanceModel, IMaterial

DefaultMaterial [IMaterial]

double [] ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

double[] ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

double[] ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

double[] ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

PhongModel [IReflectanceModel]

PhongMaterial [IMaterial]

[IMaterial] : ICloneable

double[] ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

PhongModel [IReflectanceModel]

PhongMaterial [IMaterial]

[IMaterial] : ICloneable

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )

void ColorReflection ( Intersection intersection, Vector3d input, Vector3d output, ReflectionComponent comp )
Intersection

Enter [bool]
Front [bool]
T [double] ... mandatory
Solid [Isolid]
SolidData [object]

Normal [Vector3d]
CoordWorld [Vector3d]
CoordObject [Vector3d]
CoordLocal [Vector3d]
TextureCoord [Vector2d]
LocalToWorld [Matrix4d]
WorldToLocal [Matrix4d]
LocalToObject [Matrix4d]
SurfaceColor [double[]]
ReflectanceModel [IreflectanceM..]
Material [Imaterial]
Textures [List<ITexture>]

Complete();
Intersectable object [IIntersectable]

[interface IIntersectable]

LinkedList<Intersection> Intersect ( Vector3d p0, p1 )

void CompleteIntersection ( Intersection inter )
Scene hierarchy

- **root**
  - **node1**
    - **node3**
    - **node4**
    - **node5**
  - **node2**
    - **FromParent [Matrix4d]**
    - **ToParent [Matrix4d]**
  - **color**: [RGB1]
  - **material**: <..>

- **color**: [RGB2]

RGB2 has priority!
Scene node [ISceneNode]

- Parent [ISceneNode]
- Children [ISceneNodes[]]
- ToParent [Matrix4d]
- FromParent [Matrix3d]
-ToWorld, ToObject [Matrix4d]
- ObjectRoot [bool]
- GetAttribute (name)
- GetLocalAttribute (name)
- SetAttribute (name, value)
- List<ITexture> GetTextures()
Solid [ISolid]

[interface ISolid] : ISceneNode

Ray: \( \mathbf{P}_0 + t \cdot \mathbf{p}_1 \)
\( 0 \leq t \)

LinkedList<Intersection> Intersect ( Vector3d \( \mathbf{p}_0, \mathbf{p}_1 \) )
void CompleteIntersection ( Intersection inter )

Sphere
Cube
Plane
Cylinder
Torus …
Texture [ITexture]

[interface ITexture]

long Apply (Intersection inter)

Texture order matters!

Intersection

Normal [Vector3d]
TextureCoord [Vector2d]
SurfaceColor [double[]]
ReflectanceModel [IreflectanceM..]
Material [Imaterial]

[ISolid]?
Animation [ITimeDependent]

[interface ITimeDependent]
  : ICloneable

double Start
double End
double Time

“Clone-on-write”
• for multi-threaded rendering
• cloning a copy for each thread

cyclic

Start  End  ...

one-time

Start  End
Independent stratified sampling

- **multi-dimensional** open sampling: \([0,1]^D\)
  - D is not known in advance
  - *any internal component* of a ray-tracer might be sampled (integral averaging)

- **hidden sampling** mechanism
  - *any component* can use additional global values stored in the static *class MT*:
    
    ```
    [ThreadStatic] ... TLS (automatic data instance for each thread)
    ```

    ```
    int rank ... order of the current sample (in the current pixel)
    int total ... total number of samples in the current pixel
    RandomJames rnd ... random number number generator
    ```
Independent stratified sampling

```
int rank
int total
RandomJames rnd

sample/ray #2 from total 64 samples

MT.rank = 2;
MT.total = 64;
```
References

- Subversion repository:
  `svn://cgg.mff.cuni.cz/grcis/trunk`

- Ray-tracing in GrCis:

- GrCis library:
  `http://cgg.mff.cuni.cz/~pepca/grcis/`

- Image gallery:
  `http://cgg.mff.cuni.cz/~pepca/gr/grcis/`