matfyz

Computer Graphics Charles University

Ray-tracing in GrCis

© 2016 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**, 049distributedrt
    - switches for super-sampling, shadows, reflections, refrac-tions, multi-threading
- **animation**
  - 046cameranim
    - camera animation (going round the scene)
  - **062animation**
    - more general project, able to animate any scene part

Ray-tracing application

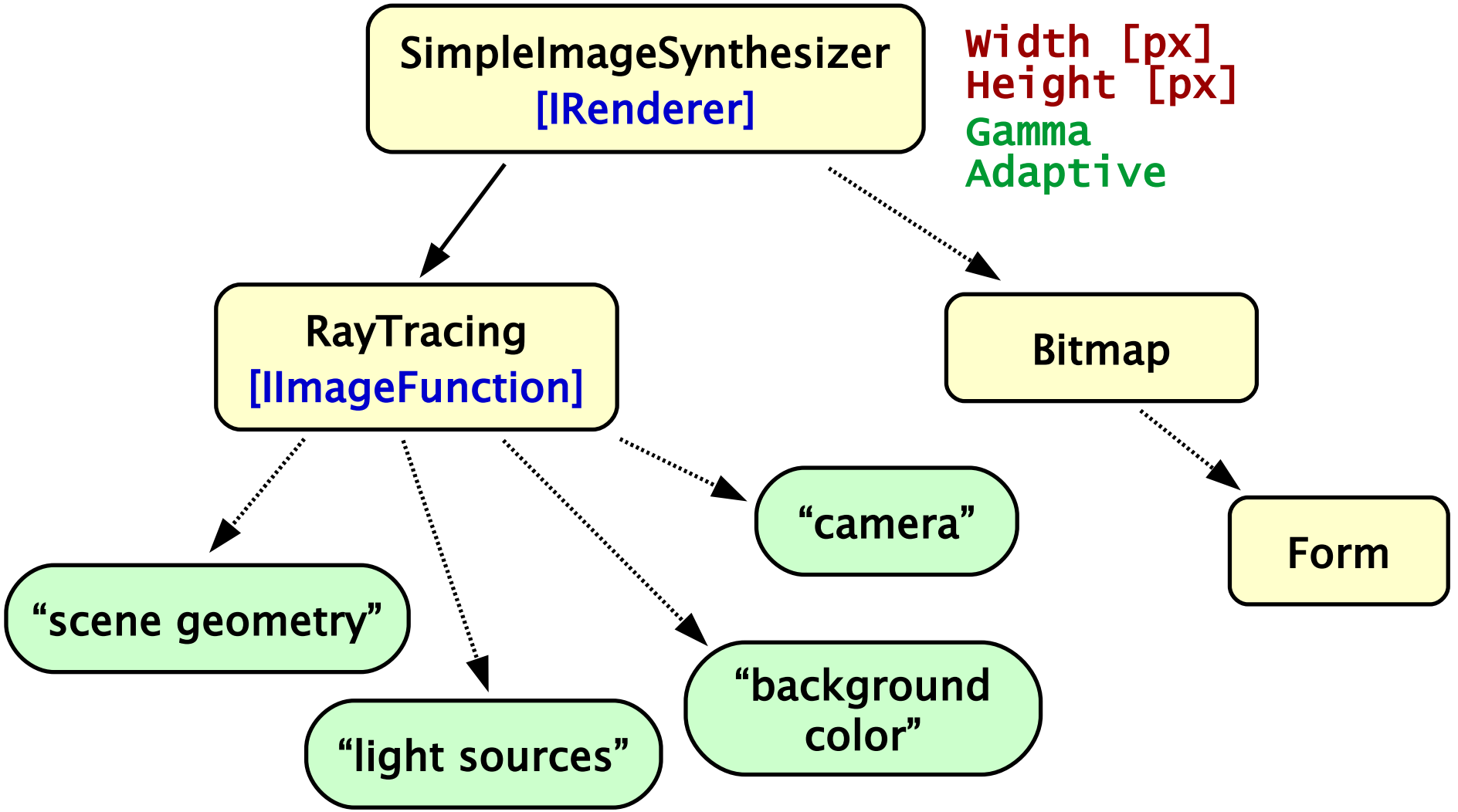
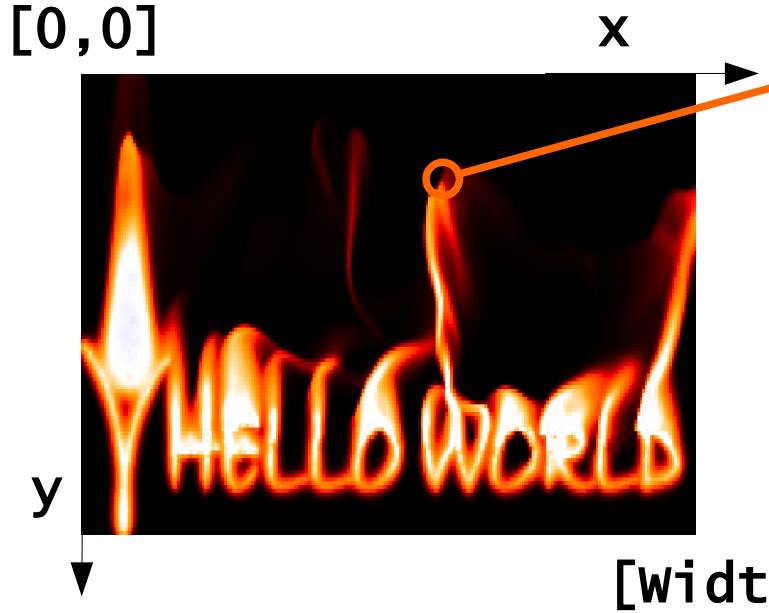


Image function [IImageFunction]

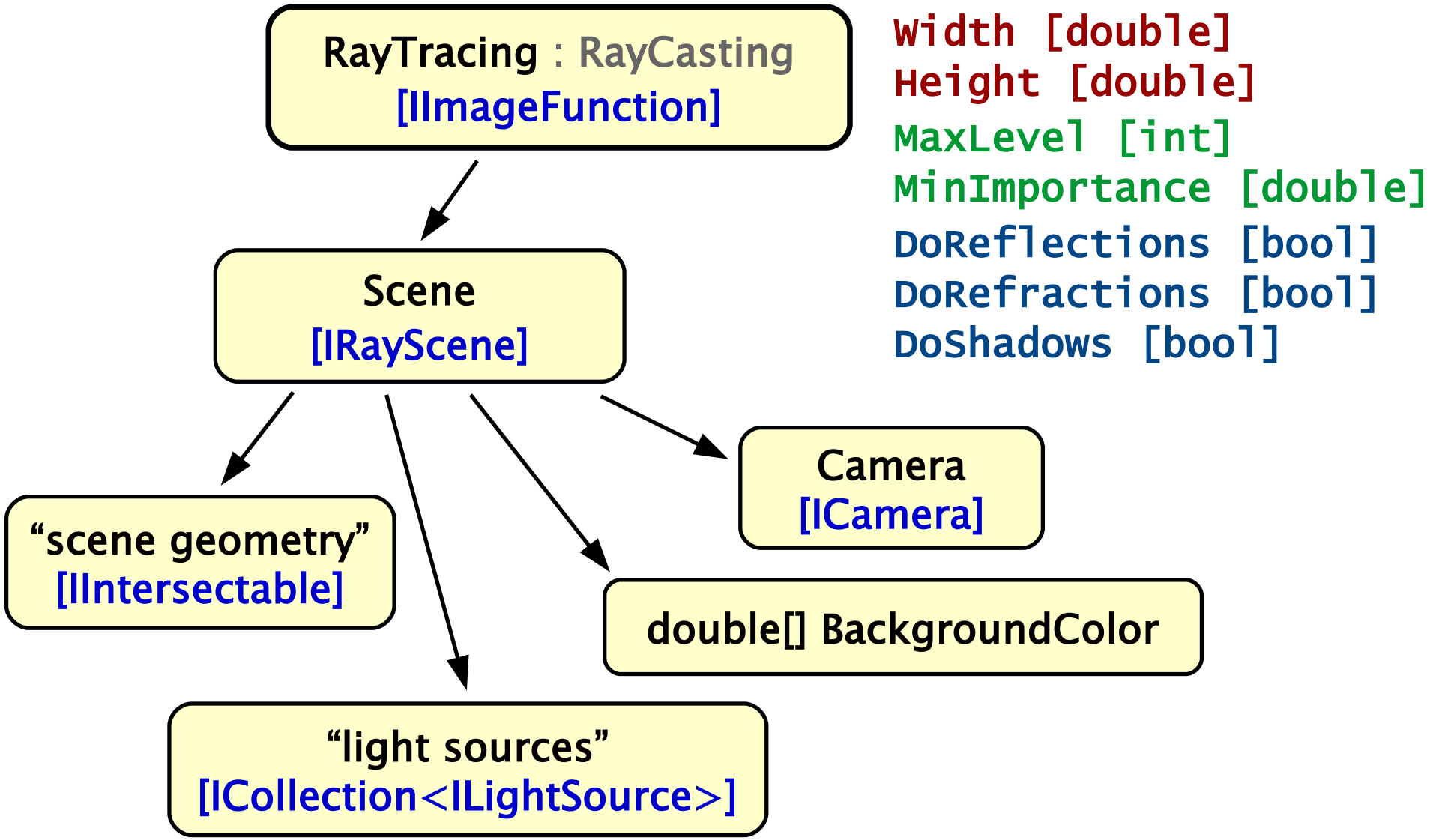
[interface IImageFunction]

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




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

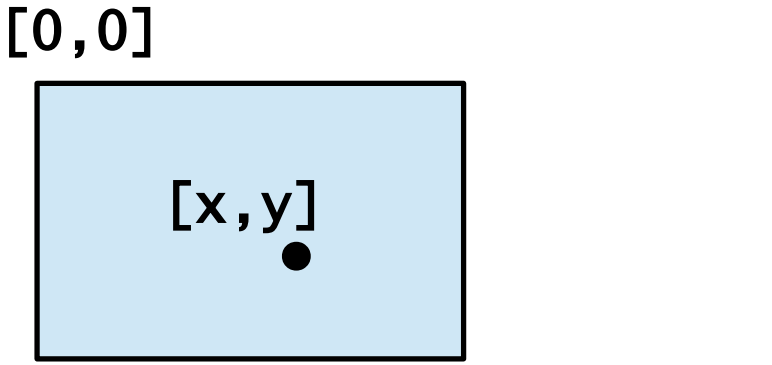
RayCasting, RayTracing

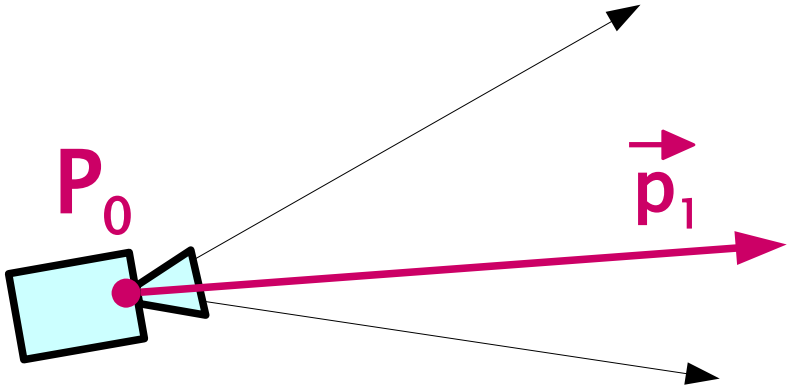


Camera [ICamera]

[interface ICamera]

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





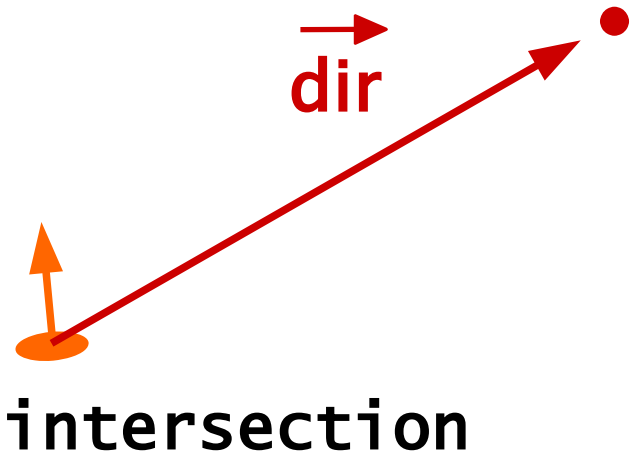
Ray: $P_0 + t \cdot \vec{p}_1$

$0 \leq t$

Light source [ILightSource]

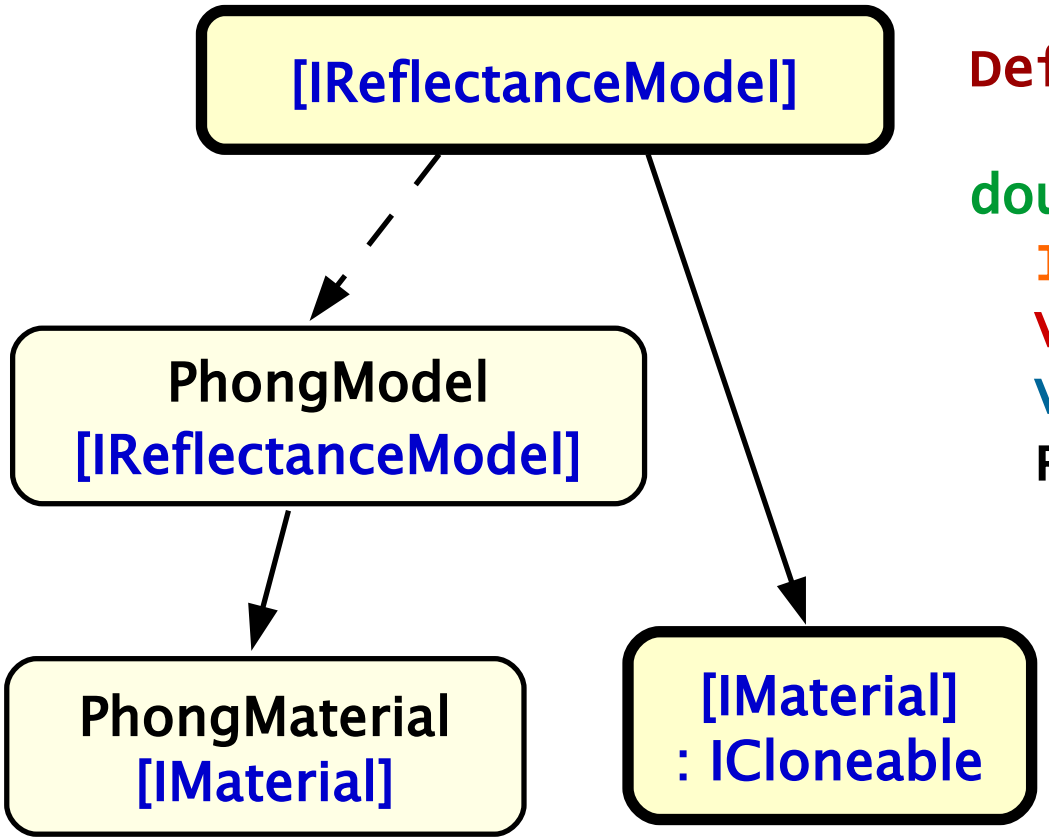
[interface ILightSource]

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



return: color (intensity)
dir: direction toward the light, zero for omnidirectional

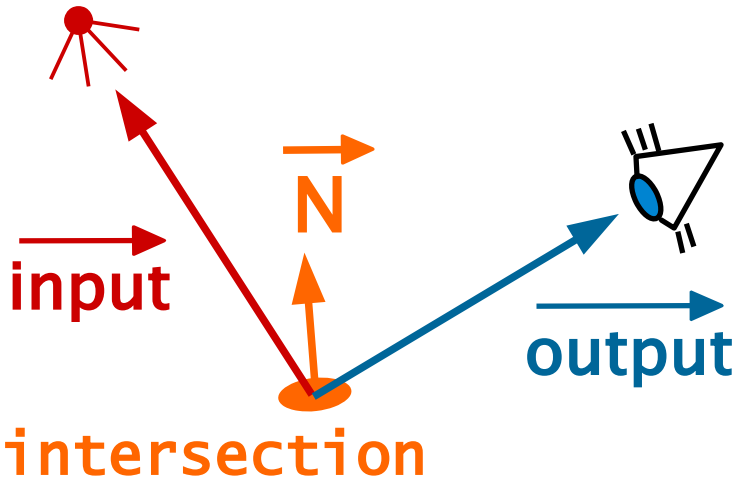
IReflectanceModel, IMaterial



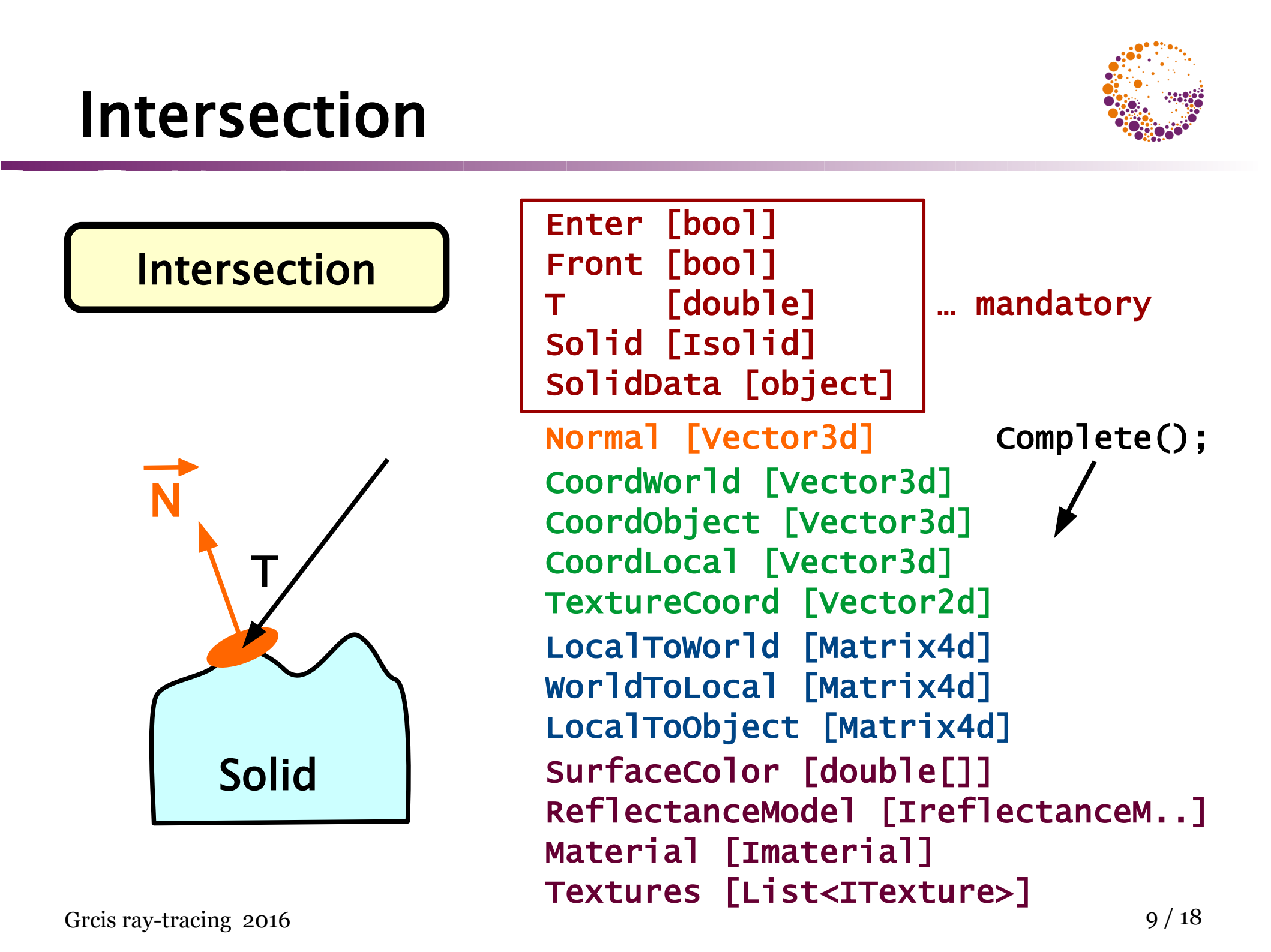

```
DefaultMaterial [IMaterial]

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

```
double[] Color
double Kt
double n
```



Intersection



Intersection

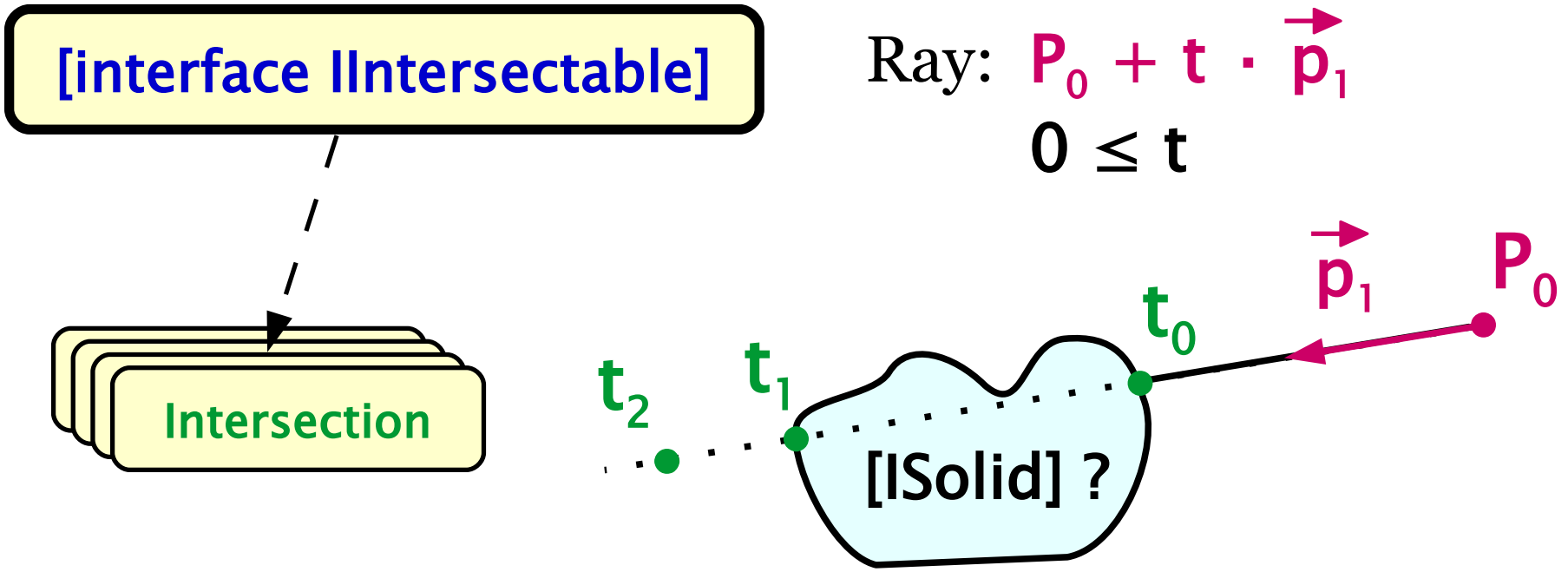

```
Enter [bool]
Front [bool]
T     [double]
Solid [Isolid]
SolidData [object]
```

… mandatory

```
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();

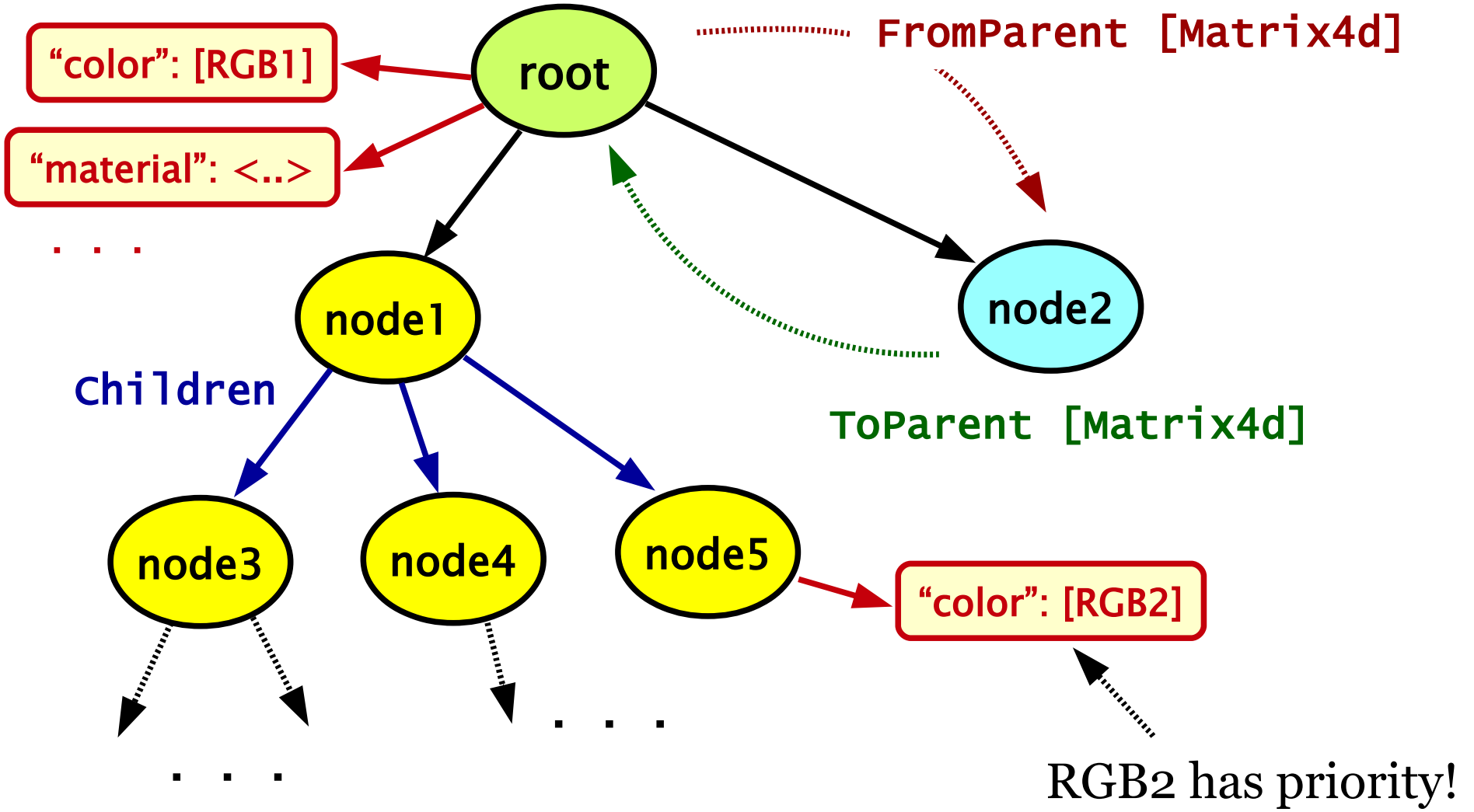
Intersectable object [IIntersectable]



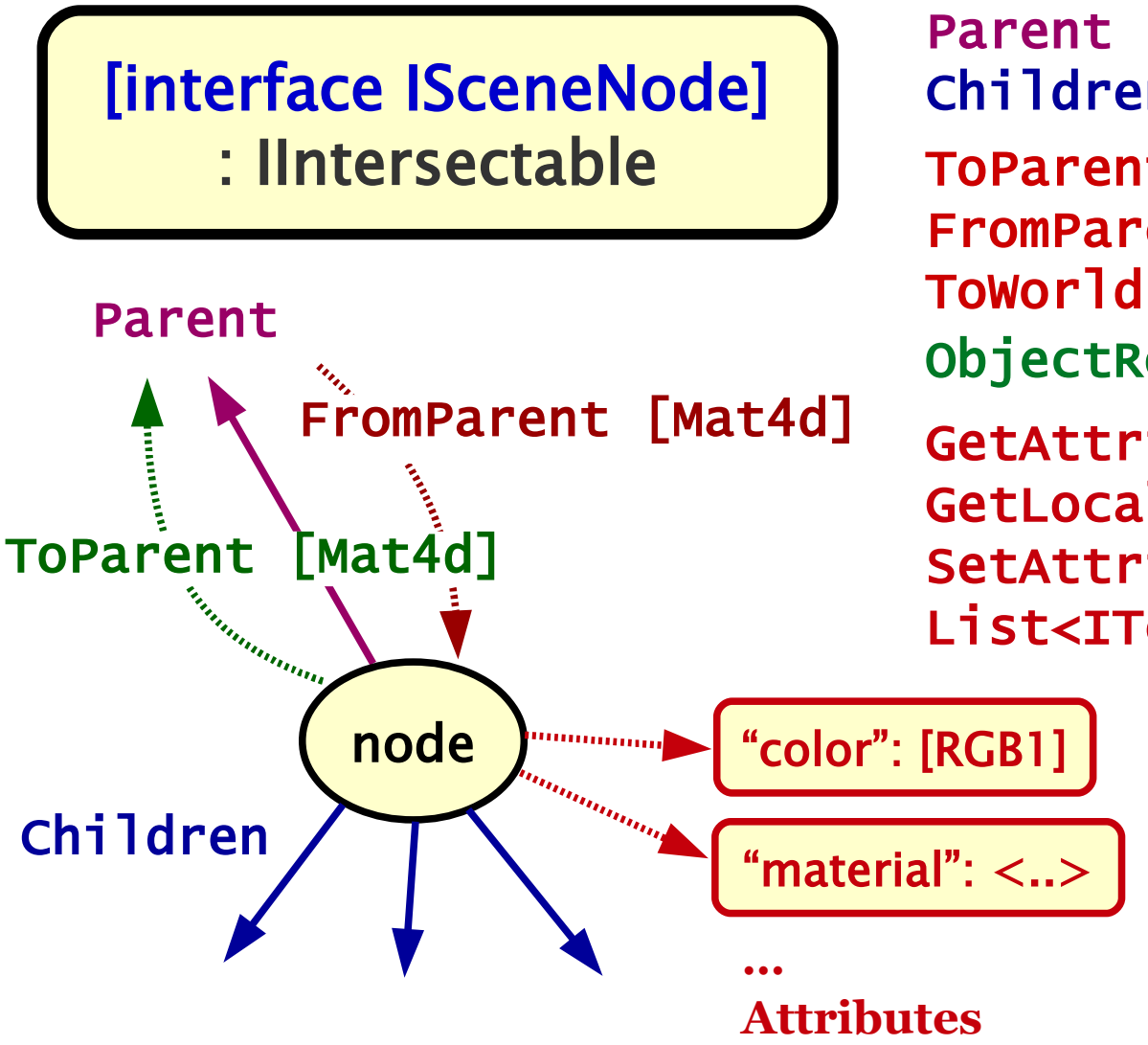

```
LinkedList<Intersection> Intersect ( Vector3d p0, p1 )

void CompleteIntersection ( Intersection inter )
```

Scene hierarchy



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]

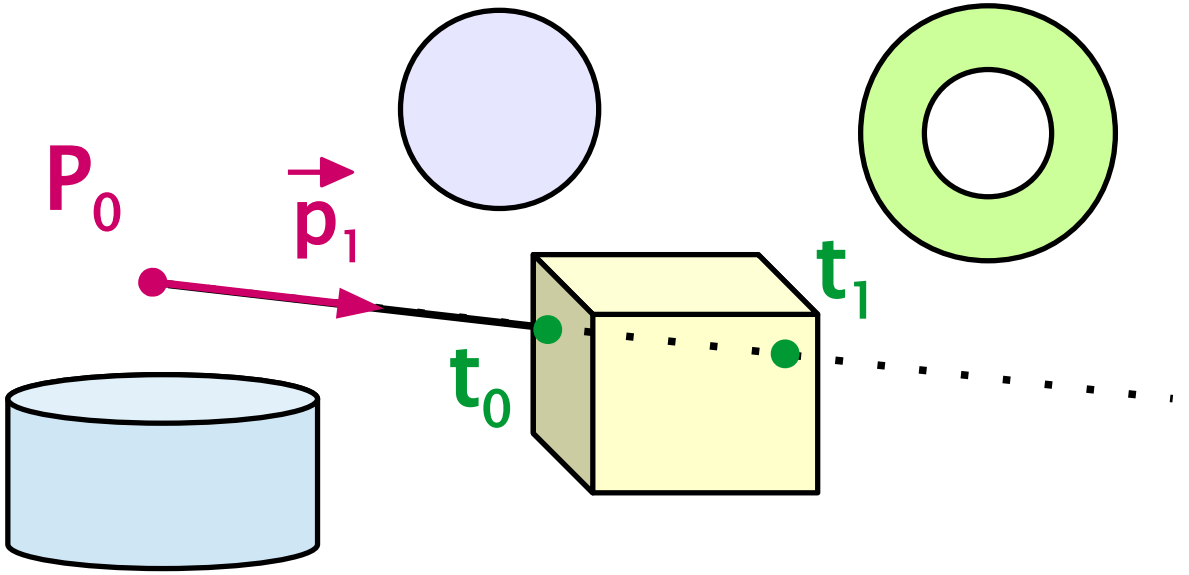
[interface ISolid]
: ISceneNode

Ray: $P_0 + t \cdot \vec{p}_1$

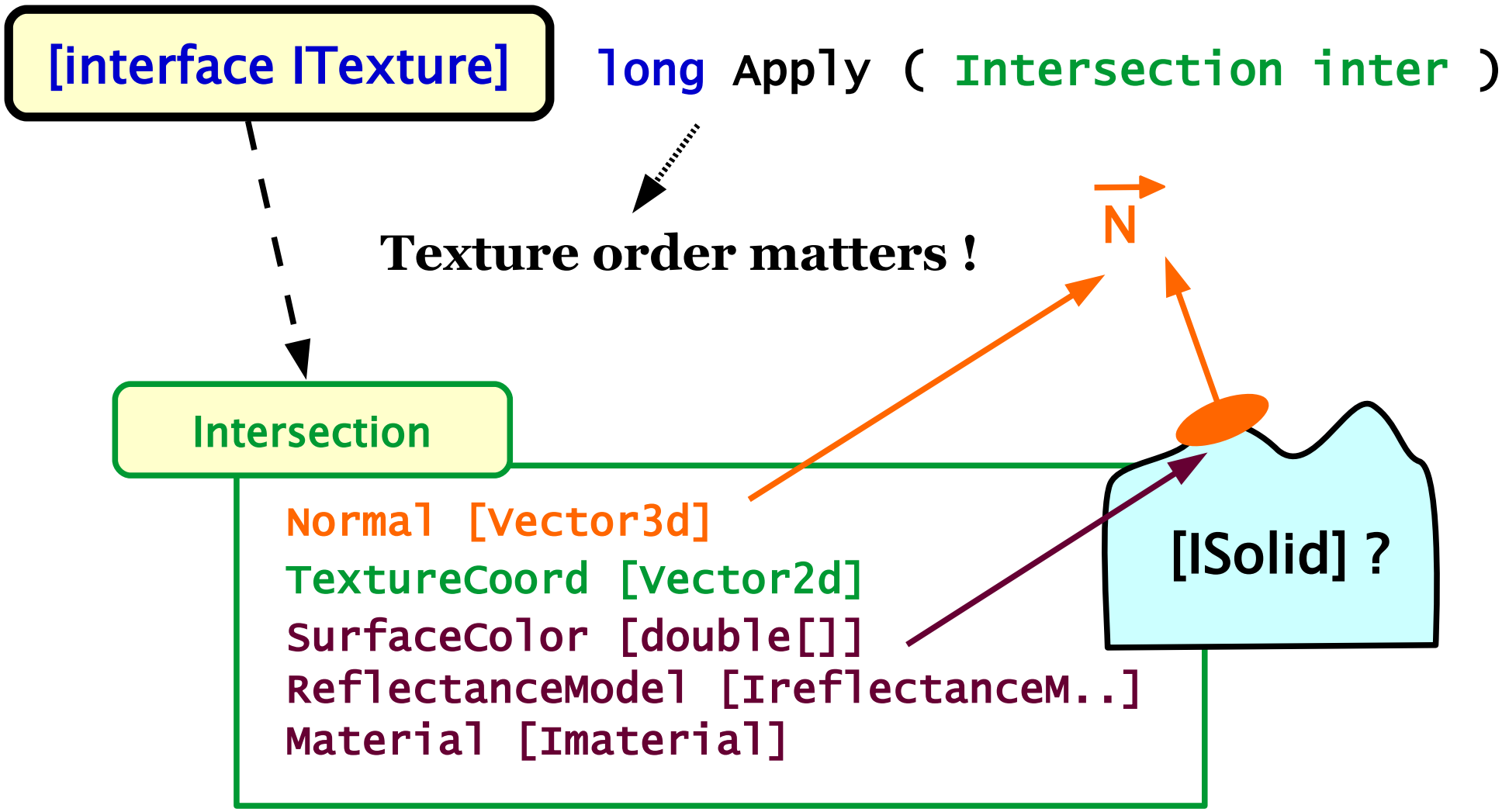
$0 \leq t$

```
LinkedList<Intersection> Intersect ( Vector3d p0, p1 )
void CompleteIntersection ( Intersection inter )
```

- Sphere
- Cube
- Plane
- Cylinder
- Torus ...



Texture [ITexture]

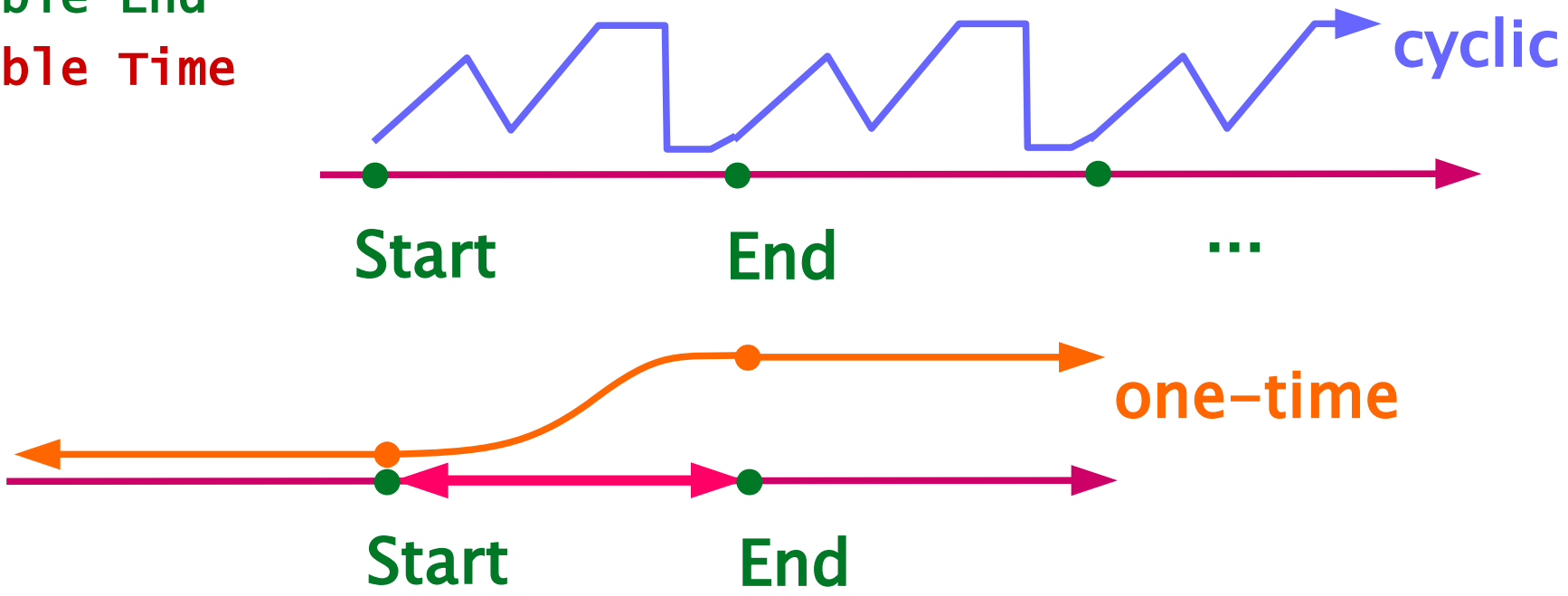


Animation [ITimeDependent]

[interface ITimeDependent]
: ICloneable

```
double Start
double End
double Time
```

"Clone-on-write"
- for multi-threaded rendering
- cloning a copy for each thread



Independent stratified sampling

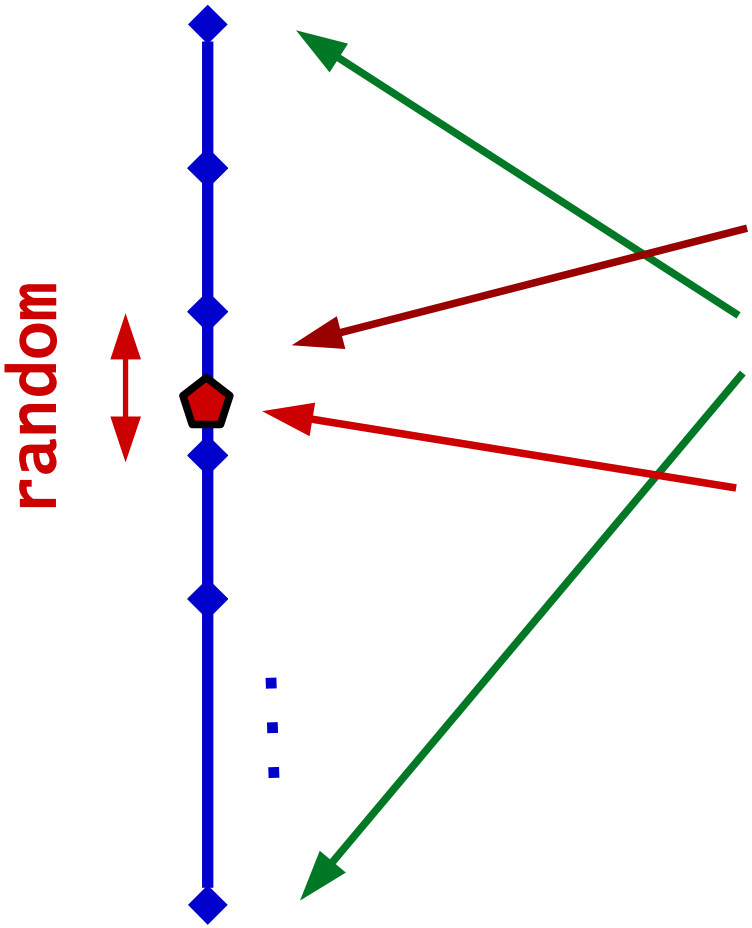
- **multi-dimensional** open sampling: $[0,1]^D$
  - D is not known in advance
  - any internal component of a ray-tracer might be sampled (integral averaging)
- **hidden sampling** mechanism
  - every component implements 3 additional arguments:

`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 generator specific to the current thread

Independent stratified sampling




```
int rank
int total
RandomJames rnd
```

```
f ( …, 2, 64, rnd, … )
```

sample/ray **#2** from total **64** samples

References

- Subversion repository:
  `svn://cgg.mff.cuni.cz/grcis/trunk`
- Ray-tracing in GrCis:
  `http://cgg.mff.cuni.cz/~pepca/grcis/rt.php`
- GrCis library:
  `http://cgg.mff.cuni.cz/~pepca/grcis/`
- Image gallery:
  `http://cgg.mff.cuni.cz/~pepca/gr/grcis/`