XML based scripting for Implicit Surfaces (xisl)

Július Parulek

Overview

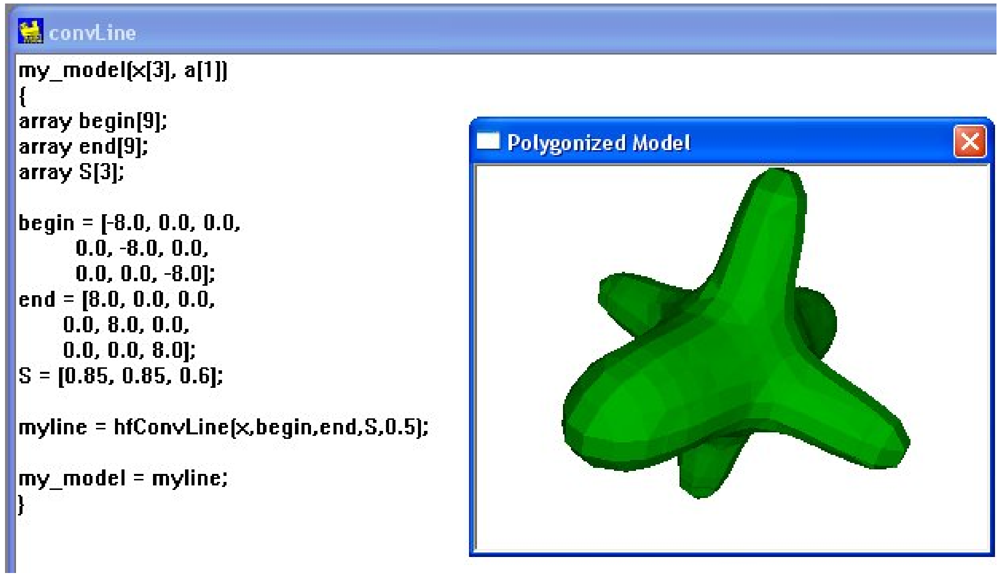
- Work description
- Related projects
- xisl library introduction
  - Implicit surface library
  - xisl language descriptions
- Examples and related work

Project description

- Well specified and easy interface between library and extern tools
- Solution for persistent storage of arbitrary complex implicit functions
- Easy extandability (add new implicits)
- Platform independcy

Hyperfun [Adzhiev et al.,99]

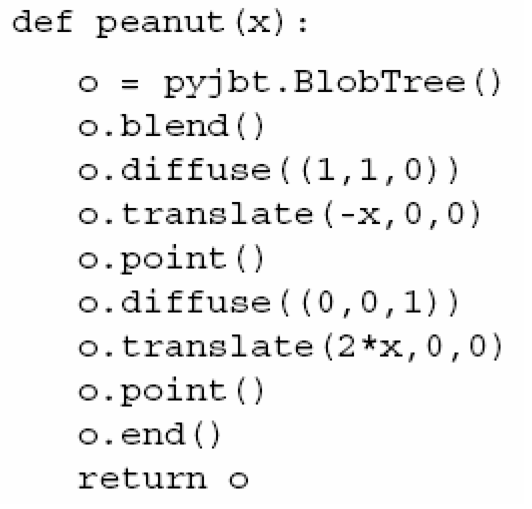
- Has its own scripting language
  - Conditional statements, loops
  - Suitable for defining arbitrary functions (math functions)
- Doesn't support multiobject scene



```
my_model(x[3], a[1])
{
array begin[9];
array end[9];
array S[3];

begin = [-8.0, 0.0, 0.0,
        0.0, -8.0, 0.0,
        0.0, 0.0, -8.0];
end = [8.0, 0.0, 0.0,
      0.0, 8.0, 0.0,
      0.0, 0.0, 8.0];
S = [0.85, 0.85, 0.6];

myline = hfConvLine(x,begin,end,S,0.5);

my_model = myline;
}
```

BlobTree [Wyvill et al.,99]

- Support python scripting language
- Very huge library – only linux version
- Aimed to convolution surfaces
- Complex management of various packages and software deployment



```
def peanut(x):
    o = pyjbt.BlobTree()
    o.blend()
    o.diffuse((1,1,0))
    o.translate(-x,0,0)
    o.point()
    o.diffuse((0,0,1))
    o.translate(2*x,0,0)
    o.point()
    o.end()
    return o
```

xisl project

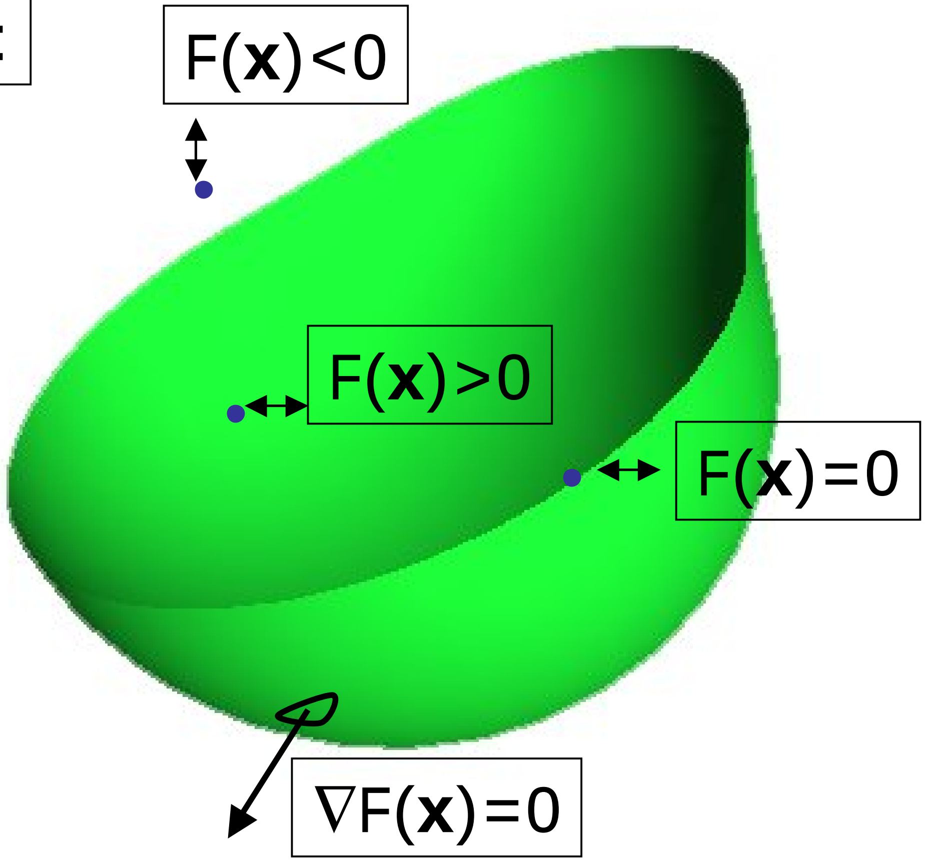
- Library of implicit surfaces written in C++
- Scripting is performed by XML tag language
    - XML is well defined widely used industrial standard
- Aimed to support multiobject scenes
    - Define relationships between objects

Implicit solid

Point classification according to the function value:

Example for sphere
$F(\mathbf{x})=r^2-x_1^2+x_2^2+x_3^2$

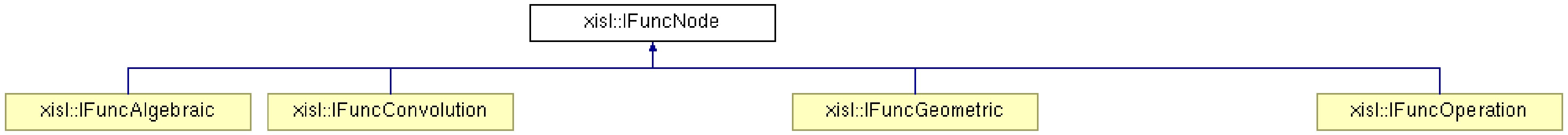
Distance to the surface is approximated as follows:
$d(x)=F(\mathbf{x})/|\nabla F(\mathbf{x})|$



Implicit function interface

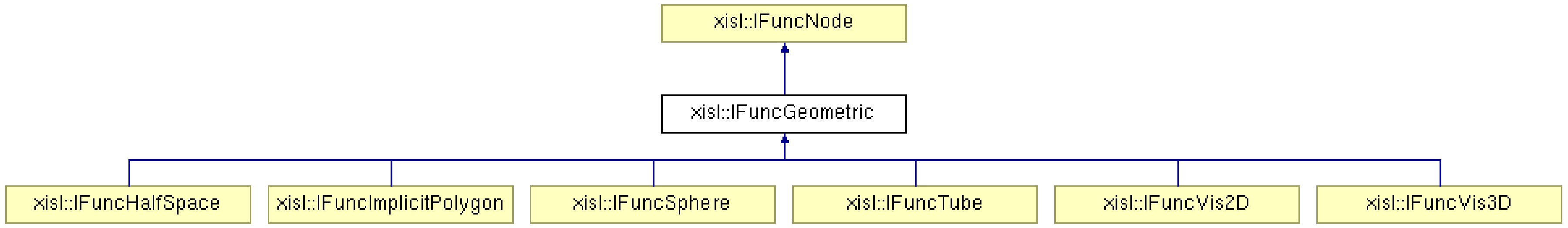
- xisl::IFuncNode = top level abstract class
- Virtual methods
  - Eval(x,y,z) – function evaluation
  - Gradient(x,y,z) – gradient evaluation
  - BoundingBox() – domain computation

xisl - implicit tree overview



- Each class is inherited from the IFuncNode that encapsulate all functional interface
- 4 main subcontainers
  - IFuncAlgebraic (quadric, superquadric)
  - IFuncConvolution (point, line convolution)
  - IFuncGeometric (all other objects)
  - IFuncOperation (union, intersection, subtraction,...)

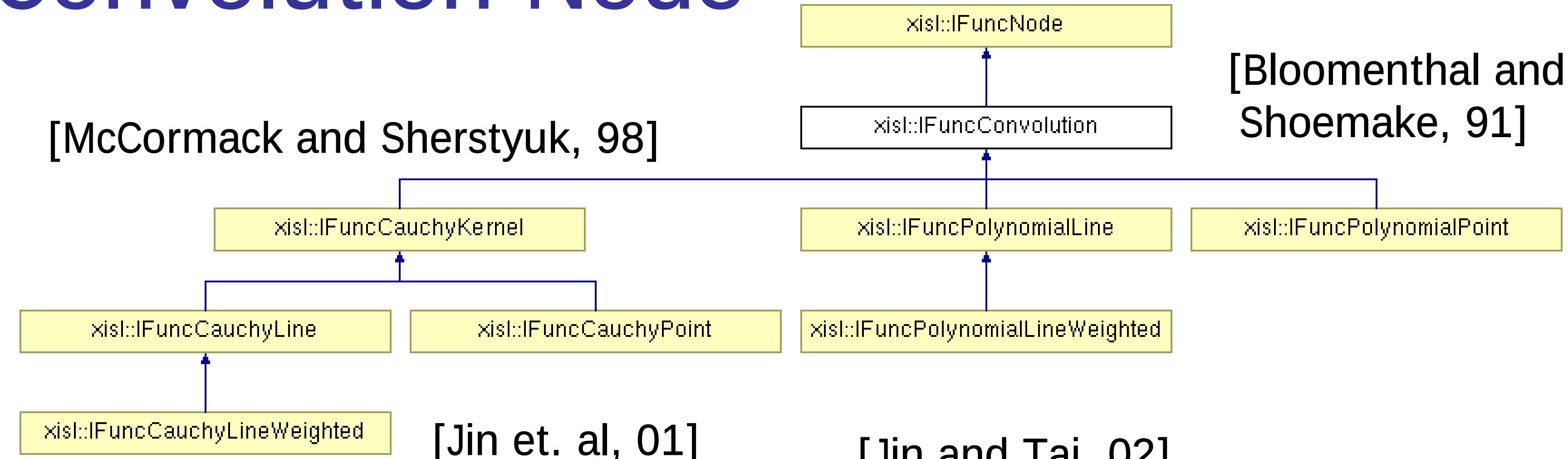
Geometric Node



[Pasko et. al, 96]

[Turk and O’Brien, 99]

Convolution Node


xisl::IFuncNode
xisl::IFuncConvolution
xisl::IFuncCauchyKernel
xisl::IFuncPolynomialLine
xisl::IFuncPolynomialPoint
xisl::IFuncCauchyLine
xisl::IFuncCauchyPoint
xisl::IFuncPolynomialLineWeighted
xisl::IFuncCauchyLineWeighted

[Bloomenthal and Shoemake, 91]

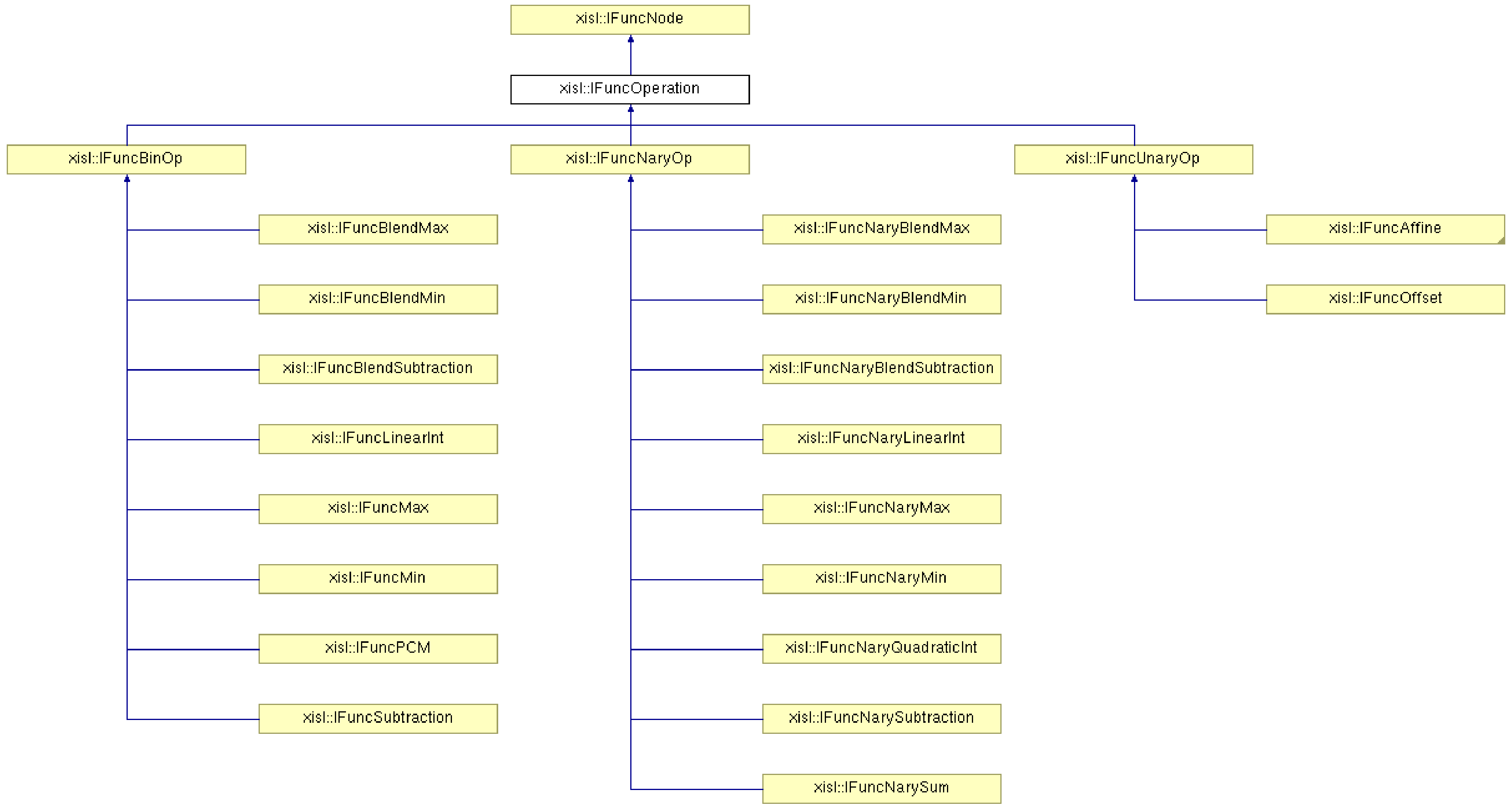
[McCormack and Sherstyuk, 98]

[Jin et. al, 01]

[Jin and Tai, 02]

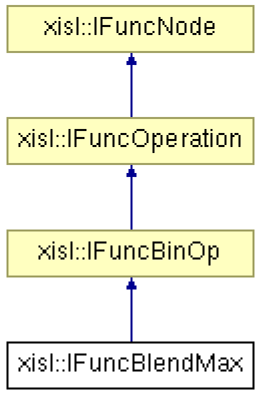
Operation node

[Pasko et. al, 95]
[Ricci, 72]
[Dekkers et. al, 04]



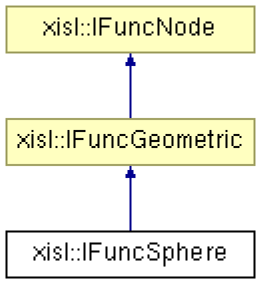
xisl language

- TinyXML (http://sourceforge.net/projects/tinyxml)
  - Reads XML and creates C++ objects representing the XML document
- Each IFuncNode can be defined through its xisl tags



```
  <binBlendedUnion blend="2">
               definition of obj1
               definition of obj2
 </binBlendedUnion>
```

```
  <gSphere>
               <wPoint x="0" y="0" z="0" w="1"/>
 </gSphere>
```



xisl file blocks

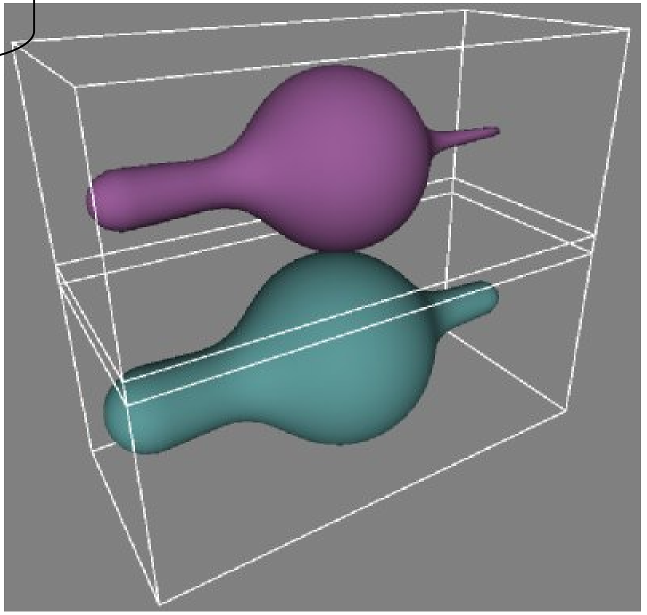
- Definition block – declaration of objects attached to names
  - <defObject name="obj1">
    - Combination of tags for declaring an implicit object
  - </defObject>
  - <includeXISL fileName="test.xisl"/>
- Main block – objects to process
  - <main>
    - Add objects previously declared
      <getObject name="obj1/>
    - Direct declaration of new objects
  - </main>

xisl - example

```
<?xml version="1.0"?>
<defObject name="A">
        <blendedUnion blend="2">
                <gTube>
                        <wPoint x="-2" y="0" z="0" w="0.4"/>
                        <wPoint x="2" y="0" z="0" w="0.2"/>
                </gTube>
                <gSphere>
                        <wPoint x="0" y="0" z="0" w="1"/>
                </gSphere>
        </blendedUnion>
</defObject>
<main>
        <getObject name="A"/>
        <translation x="0" y="2.0" z="0">
                <offset value="-0.15">
                        <getObject name="A"/>
                </offset>
        </translation>
</main>
```

Definition block

Main block



API for loading xisl file

include xislParser and IFuncLib

```
#include <IFuncLib.h>
#include <xislParser.h>
```

Declarations

```
  xisl::xislParser     parser;
  xisl::IFuncNameMap   funcNameMap;
  xisl::IFuncVector    funcVector;
```

Load xisl file; returns true if file is a xml file and all tags are well defined

```
if (!parser.loadFile("test.xisl"))
                return false;
```

parse xisl definition block into IFuncNameMap structure

```
parser.parseDefBlock(funcNameMap);
```

parse xisl main block into IFuncVector structure using IFuncNameMap structure

```
parser.parseMainBlock(funcNameMap,funcVector);
```

funcVector contains a list of created functions

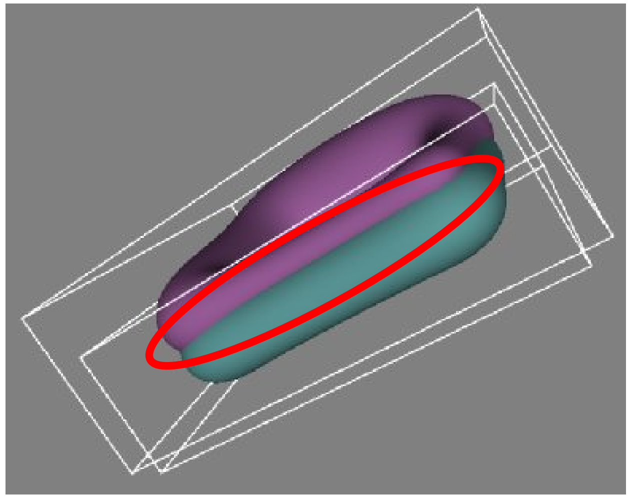
```
typedef std::map<std::string,IFuncNode*> xisl::IFuncNameMap
```

```
typedef std::vector<IFuncNode*> xisl::IFuncVector
```

Multiobject scene support

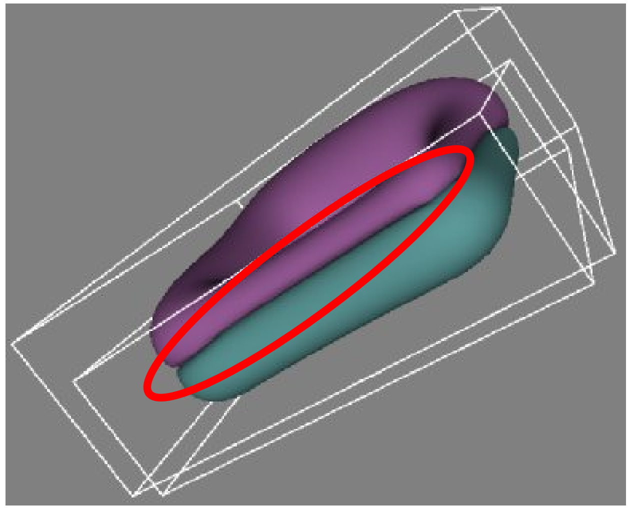
Scene consists of two objects that intersect

```
<main>
                <getObject name="A"/>
                <getObject name="B"/>
</main>
```



Modification of implicit functions not to intersect: Precisse Contact Modeling [Gascuel]

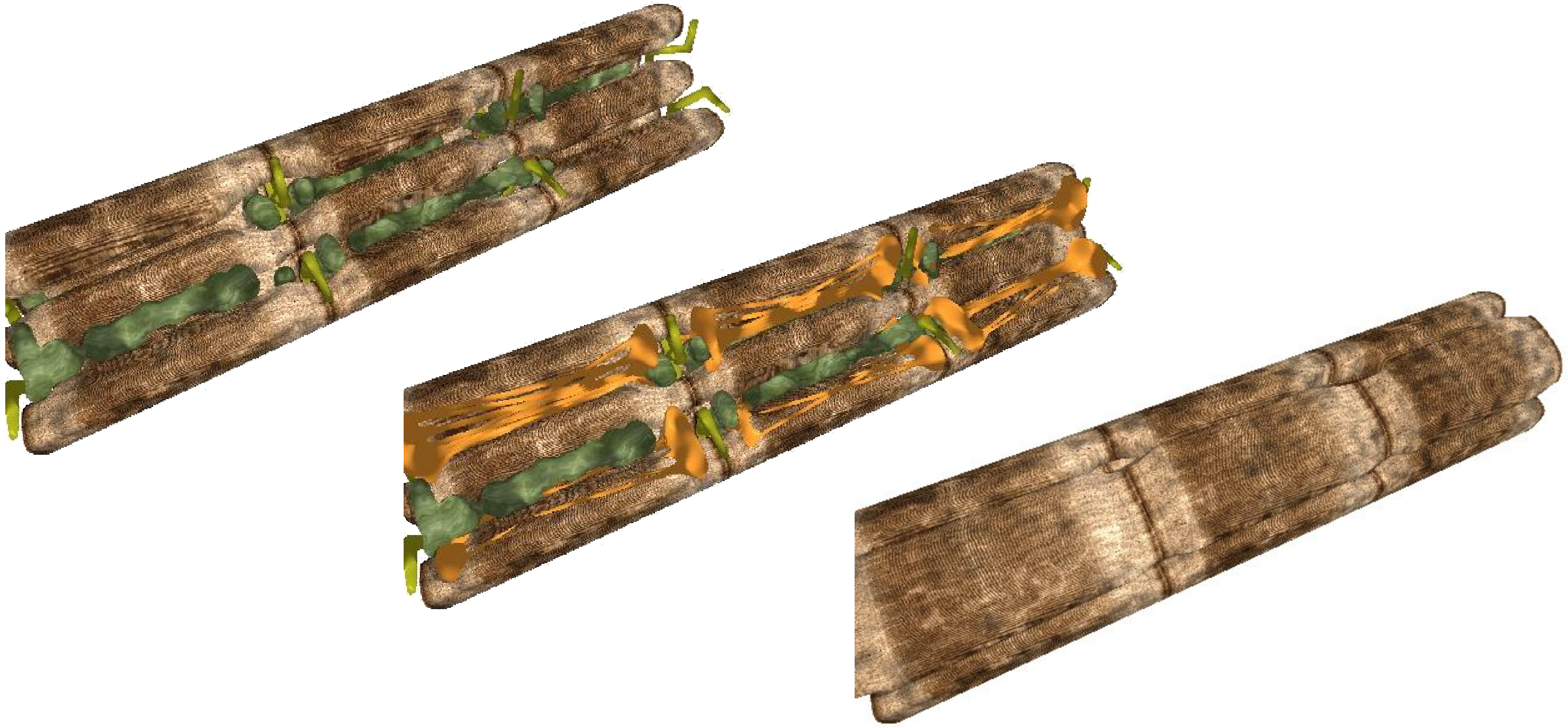
```
<main>
        <relationship type="BASIC">
                <getObject name="A"/>
                <getObject name="B"/>
        </relationship>
</main>
```



Related and future work

- GUI editor
- Various tools
  - Polygonization (MC,Bloomenthal)
  - Exporters (Hyperfun, Povray)
- Add implicit nodes for other objects and operations
- Extend library for implicit nodes that represents the biological objects
  - Definition of tags that provide growth of the objects

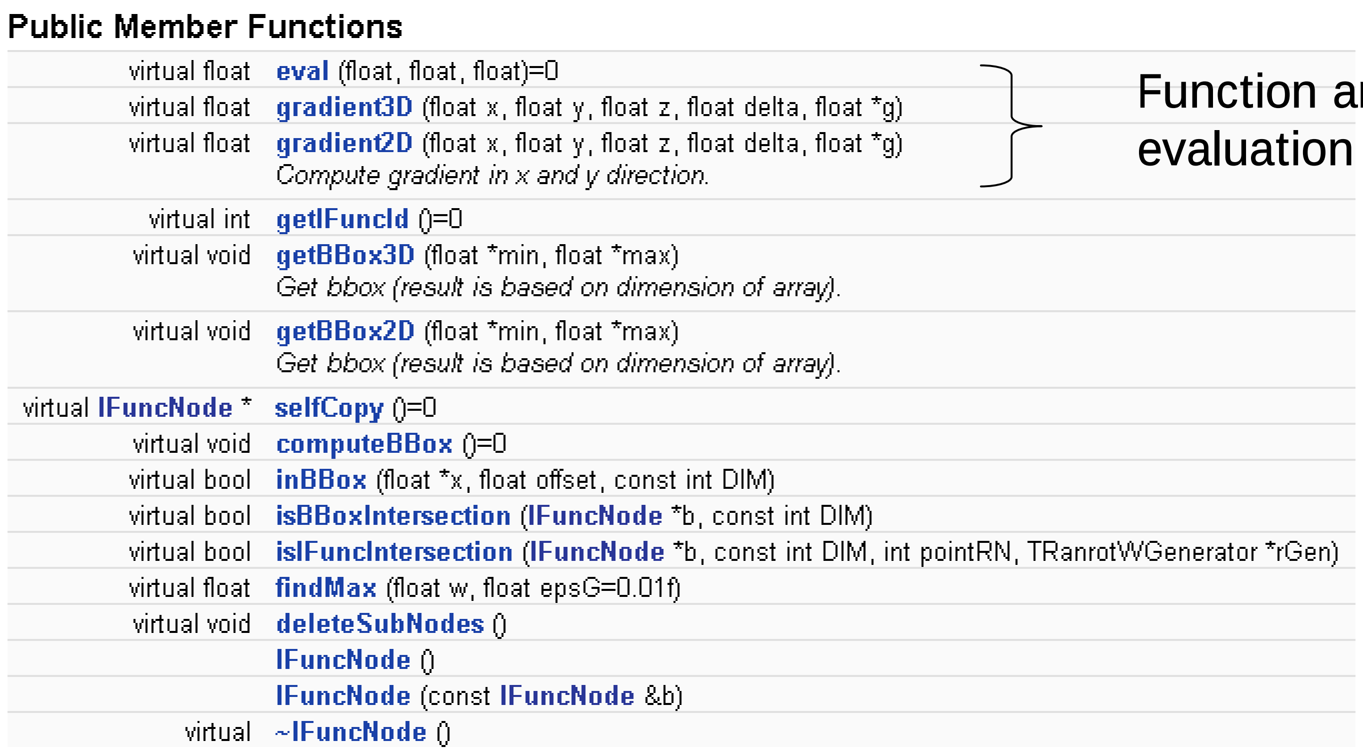
Povray examples



Thank you for your attention

Implicit function class interface

class xisl::IFuncNode



Public Member Functions

virtual float	**eval** (float, float, float)=0
virtual float	**gradient3D** (float x, float y, float z, float delta, float *g)
virtual float	**gradient2D** (float x, float y, float z, float delta, float *g) *Compute gradient in x and y direction.*
virtual int	**getIFuncId** ()=0
virtual void	**getBBox3D** (float *min, float *max) *Get bbox (result is based on dimension of array).*
virtual void	**getBBox2D** (float *min, float *max) *Get bbox (result is based on dimension of array).*
virtual **IFuncNode** *	**selfCopy** ()=0
virtual void	**computeBBox** ()=0
virtual bool	**inBBox** (float *x, float offset, const int DIM)
virtual bool	**isBBoxIntersection** (**IFuncNode** *b, const int DIM)
virtual bool	**isIFuncIntersection** (**IFuncNode** *b, const int DIM, int pointRN, TRanrotWGenerator *rGen)
virtual float	**findMax** (float w, float epsG=0.01f)
virtual void	**deleteSubNodes** ()
	IFuncNode ()
	IFuncNode (const **IFuncNode** &b)
virtual	**~IFuncNode** ()

Function and gradient evaluation

Collision detection