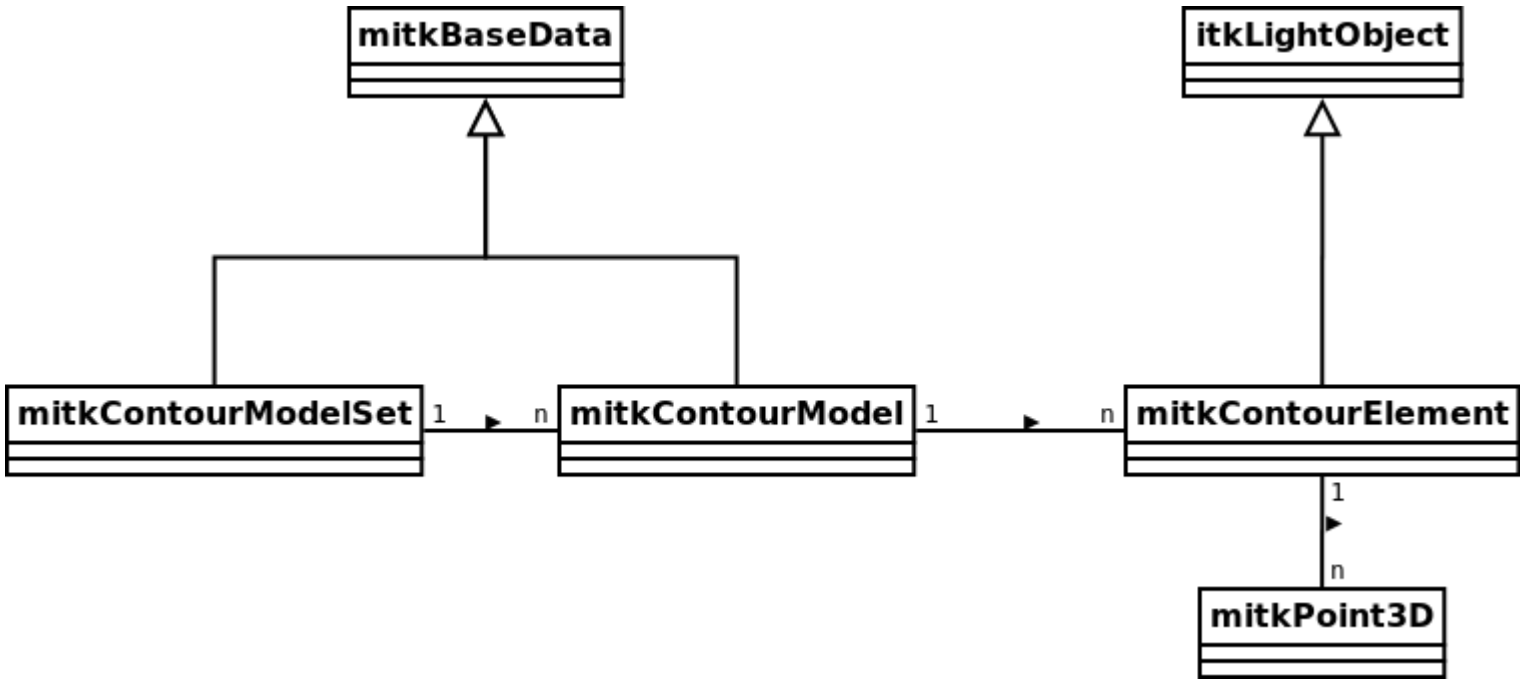
13/11/2013

ContourModel & ContourModelSet

Nico Riecker

dkfz.
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

Overview


mitkBaseData
itkLightObject
mitkContourModelSet
1
n
mitkContourModel
1
n
mitkContourElement
1
n
mitkPoint3D

mitkContourElement

- Uses mitk::Point3D
- Can take points as controlpoints

- Important methods:
  - AddVertex(mitk::Point3D, bool)
  - AddVertexAtFront(mitk::Point3D, bool)
  - InsertVertexAtIndex(mitk::Point3D, bool, int)
  - SetVertexAt(int, mitk::Point3D)

  - GetVertexAt(int)
  - GetVertexAt(mitk::Point3D, float)
  - GetVertexList()
  - GetControlVertices()
  - GetIndex(VertexType*)

  - RemoveVertexAt(int)
  - RemoveVertexAt(mitk::Point3D, float)

  - IsEmpty()
  - Concatenate(mitk::ContourElement*, bool)

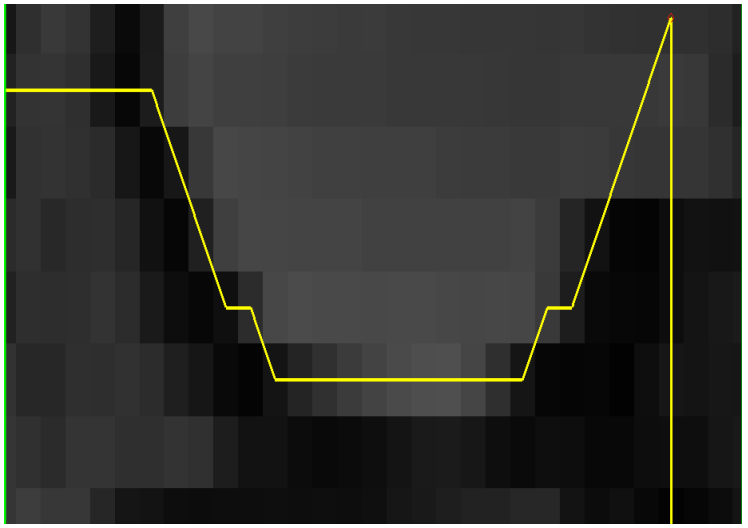
```
/** \brief Represents a single vertex of contour.
*/
struct ContourModelVertex
{
  ContourModelVertex(mitk::Point3D &point, bool active=false)
    : Coordinates(point), IsControlPoint(active)
  {

  };

  /** \brief Treat point special. */
  bool IsControlPoint;

  /** \brief Coordinates in 3D space. */
  mitk::Point3D Coordinates;
};
```

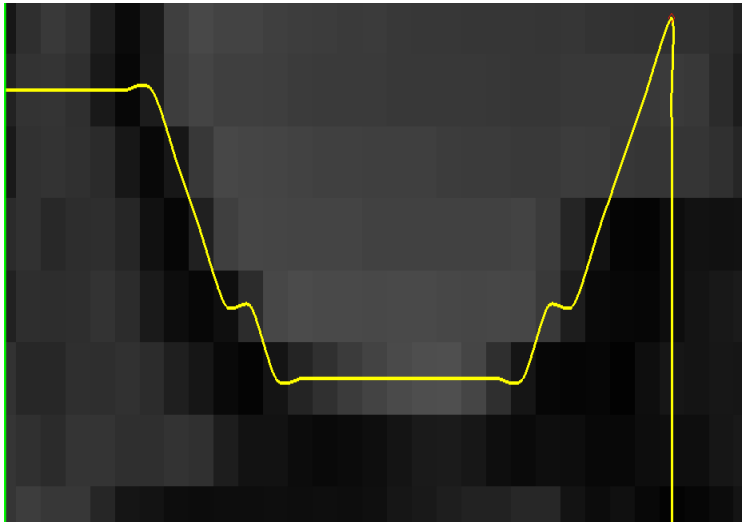
```
typedef ContourModelVertex VertexType;
typedef std::deque<VertexType*> VertexListType;
typedef VertexListType::iterator VertexIterator;
typedef VertexListType::const_iterator ConstVertexIterator;
```

mitkContourModel

```
/*+++++++++++++++ typedefs +++++++++++++++++++++++++++++++*/
typedef mitk::ContourElement::VertexType VertexType;
typedef mitk::ContourElement::VertexListType VertexListType;
typedef mitk::ContourElement::VertexIterator VertexIterator;
typedef mitk::ContourElement::ConstVertexIterator ConstVertexIterator;
typedef std::vector< mitk::ContourElement::Pointer > ContourModelSeries;
/*+++++++++++++++ END typedefs +++++++++++++++++++++++++++++*/
```

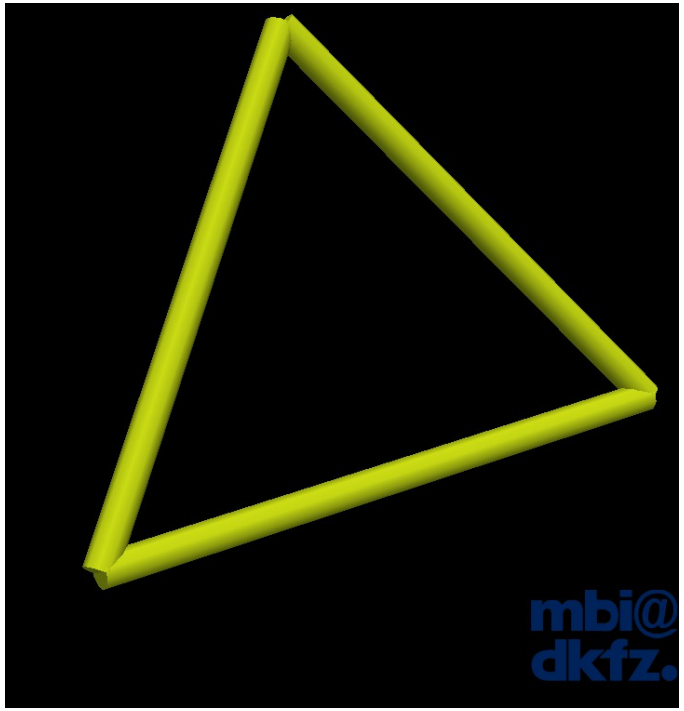
- Can manage timesteps
- You can select a vertex and hold the selection
- Inherit from mitkBaseData
  - Geometry information
- Subdivision curve property (interpolation)
- Some additional (to mitkContourElement) methods:
  - Close() and Open()
  - GetNumberOfVertices()
  - SelectVertexAt(int)





mitkContourModelExample

```
mitk::ContourModel::Pointer contourModel = mitk::ContourModel::New();

mitk::Point3D point0; point0[0] = 0; point0[1] = 0; point0[2] = 0;
mitk::Point3D point1; point1[0] = 10; point1[1] = 10; point1[2] = 10;
mitk::Point3D point2; point2[0] = 20; point2[1] = 0; point2[2] = 0;

contourModel->AddVertex(point0);
contourModel->AddVertex(point1);
contourModel->AddVertex(point2);
contourModel->Close();
```



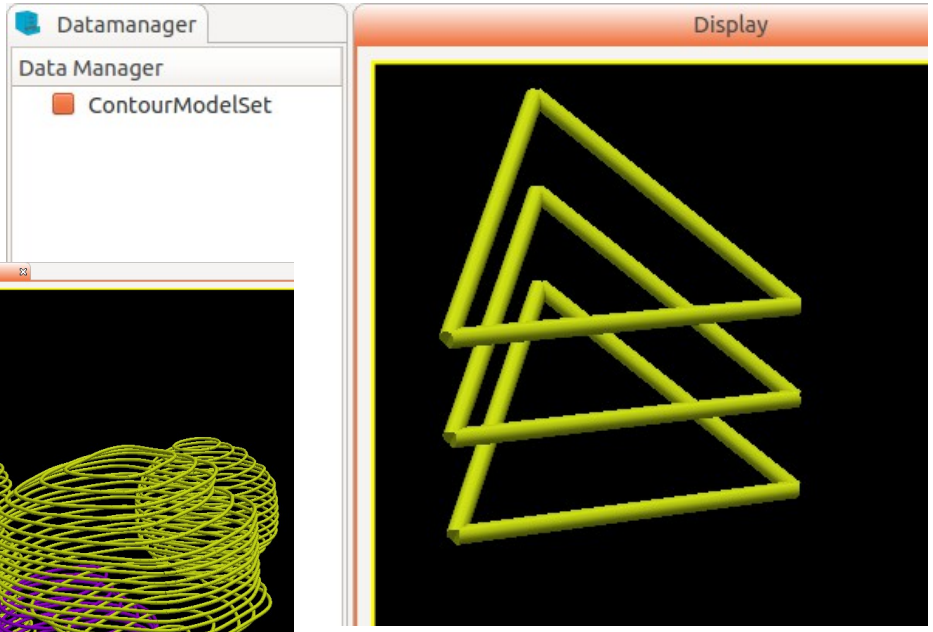
mitkContourModelSet

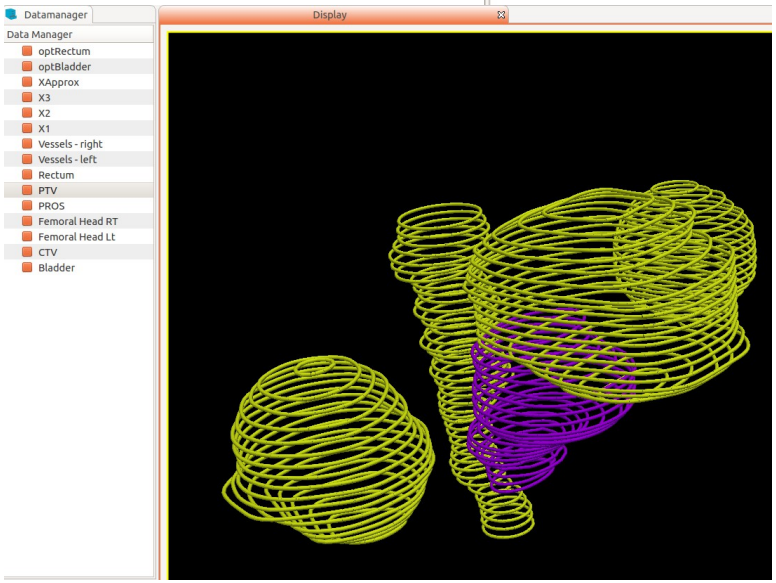
- Containerclass for mitkContourModel
- Reader/Writer
- Special mappers:
  - mitkContourModelSetGLMapper2D
  - mitkContourModelSetMapper3D

```
typedef std::deque<mitk::ContourModel::Pointer> ContourModelListType;
typedef ContourModelListType::iterator ContourModelSetIterator;
```

mitkContourModelSetExample

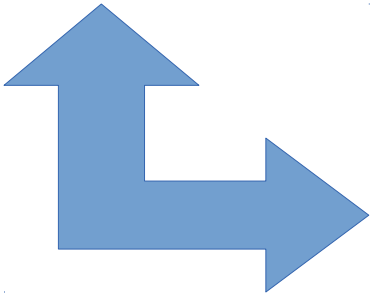
```
mitk::ContourModel::Pointer contourModel0 = mitk::ContourModel::New();
mitk::ContourModel::Pointer contourModel1 = mitk::ContourModel::New();
mitk::ContourModel::Pointer contourModel2 = mitk::ContourModel::New();

...

mitk::ContourModelSet::Pointer contourModelSet = mitk::ContourModelSet::New();

contourModelSet->AddContourModel(contourModel0);
contourModelSet->AddContourModel(contourModel1);
contourModelSet->AddContourModel(contourModel2);


mitk::DataNode::Pointer node = mitk::DataNode::New();
node->SetData(contourModelSet);
node->SetName("ContourModelSet");
GetDataStorage()->Add(node);
```





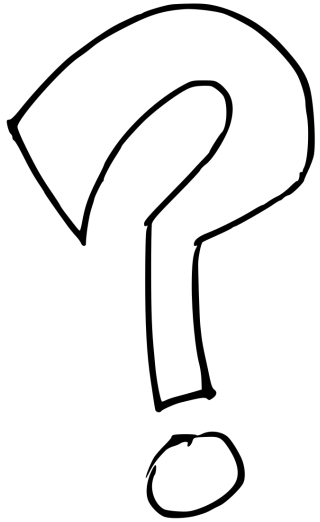
MitkContourModelSet Reading Example

```
for(mitk::ContourModelSet::ContourModelSetIterator iterator = contourModelSet->Begin(); iterator != contourModelSet->End(); iterator++)
{
  mitk::DataNode::Pointer node = mitk::DataNode::New();
  node->SetData(iterator->GetPointer());
  node->SetName("ContourModel");
  GetDataStorage()->Add(node);
}
```



```
for(int i=0; i<contourModelSet->GetSize(); i++)
{
  mitk::DataNode::Pointer node = mitk::DataNode::New();
  node->SetData(contourModelSet->GetContourModelAt(i));
  node->SetName("ContourModel");
  GetDataStorage()->Add(node);
}
```

Thank you for your attention!



Any questions?