

7.1.2015

Overlays in MITK

Bugsquashing Seminar
Christoph Kolb

dkfz.

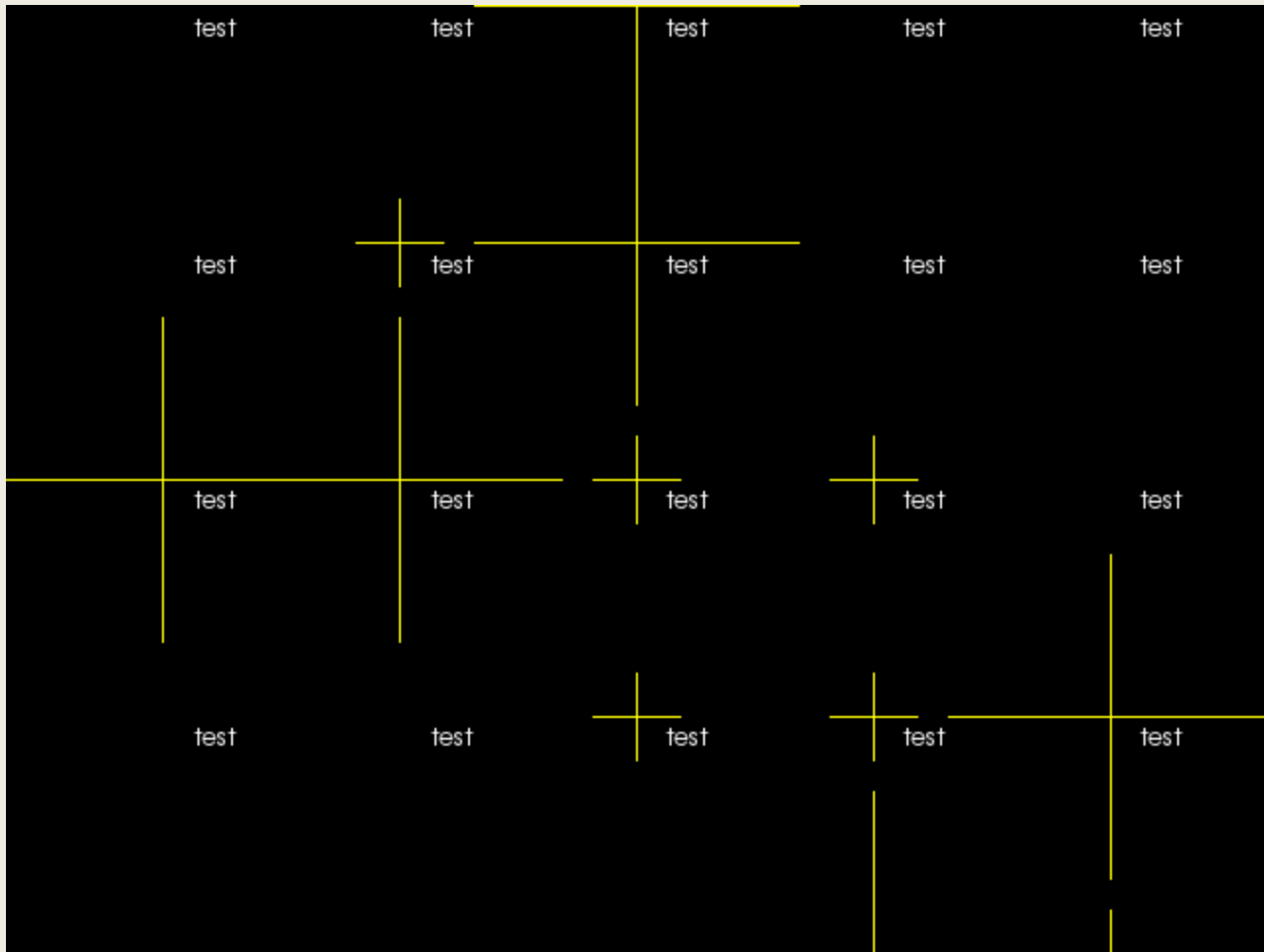
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

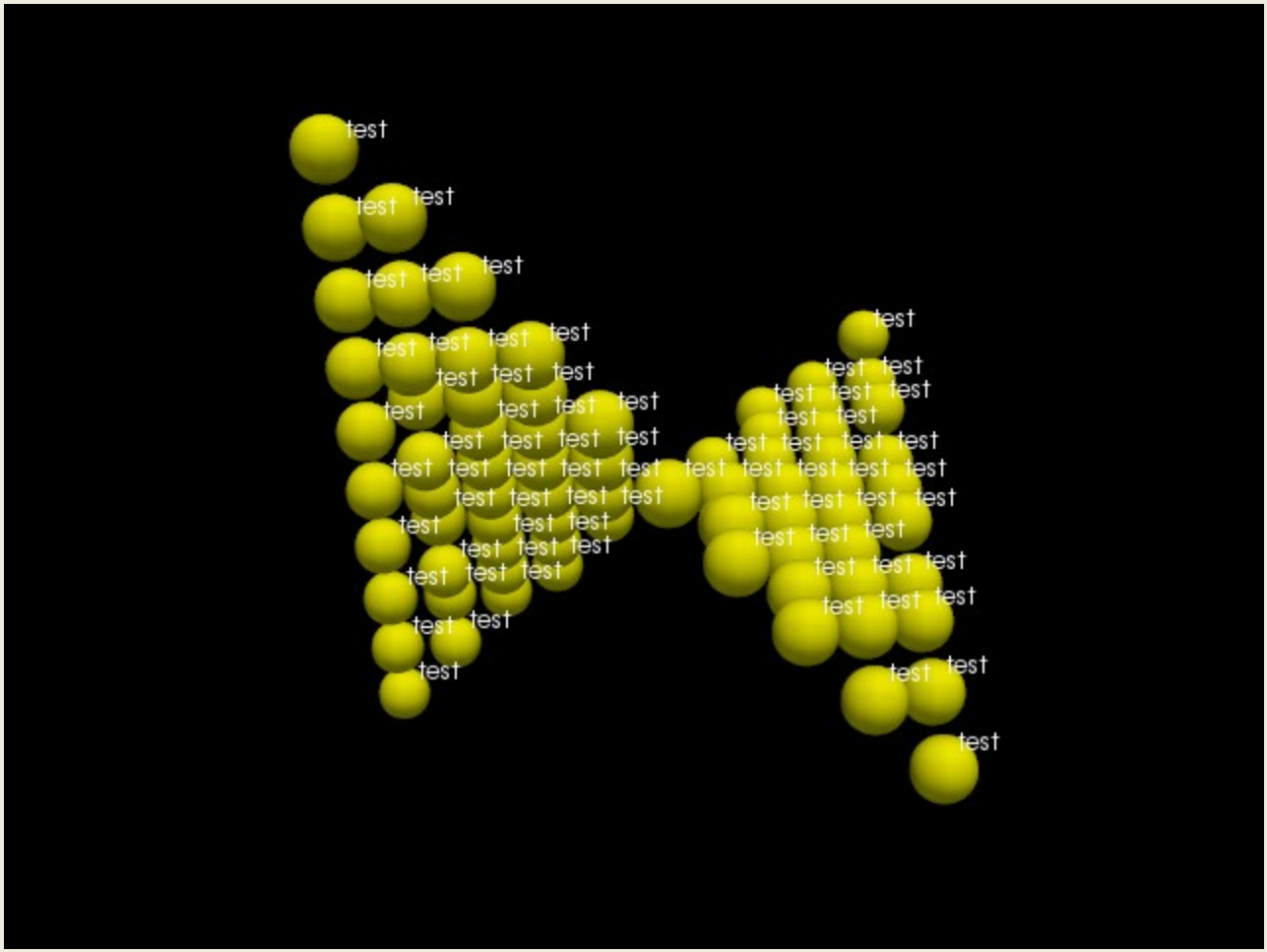


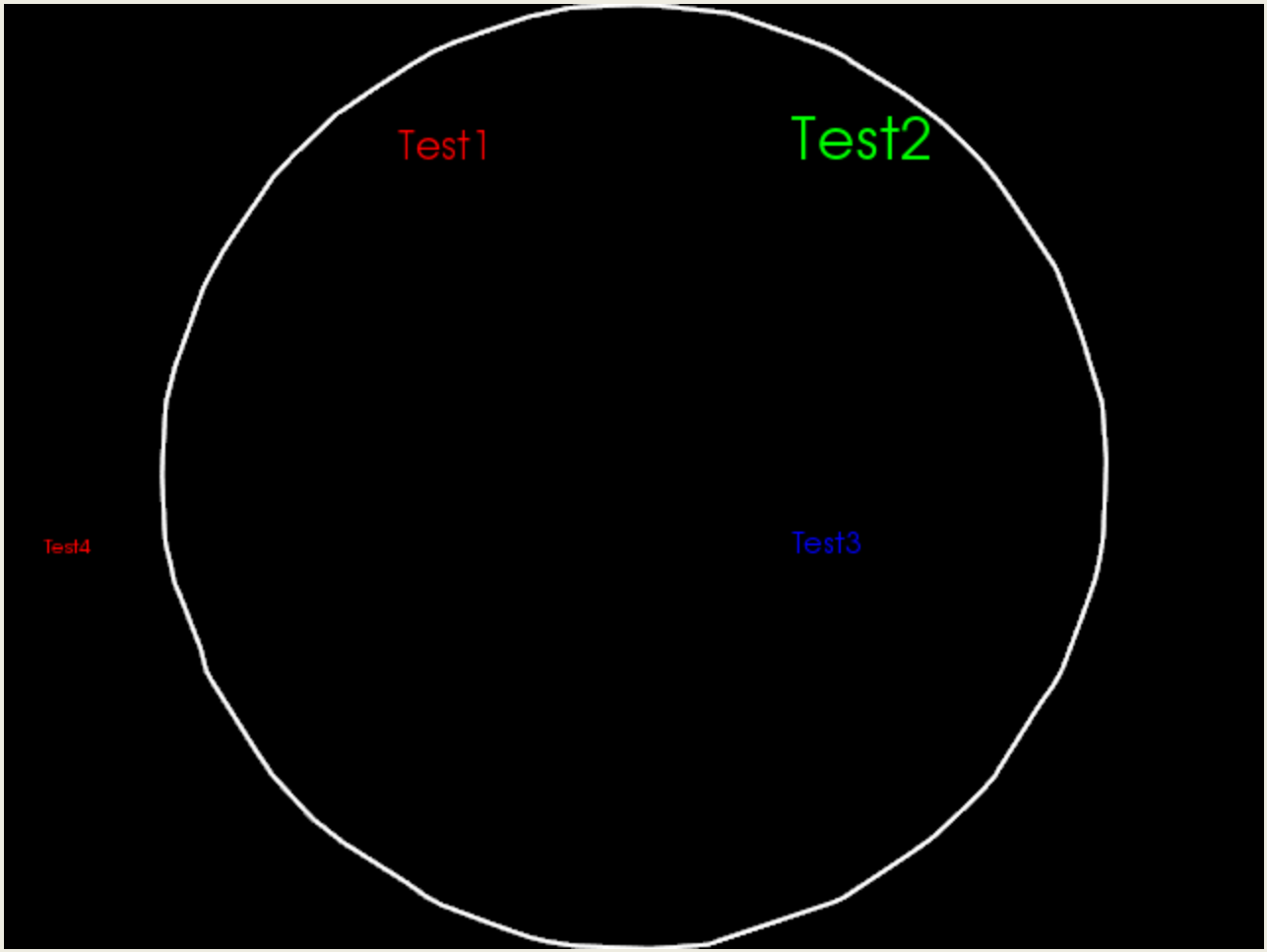
50 Years – Research for
A Life Without Cancer

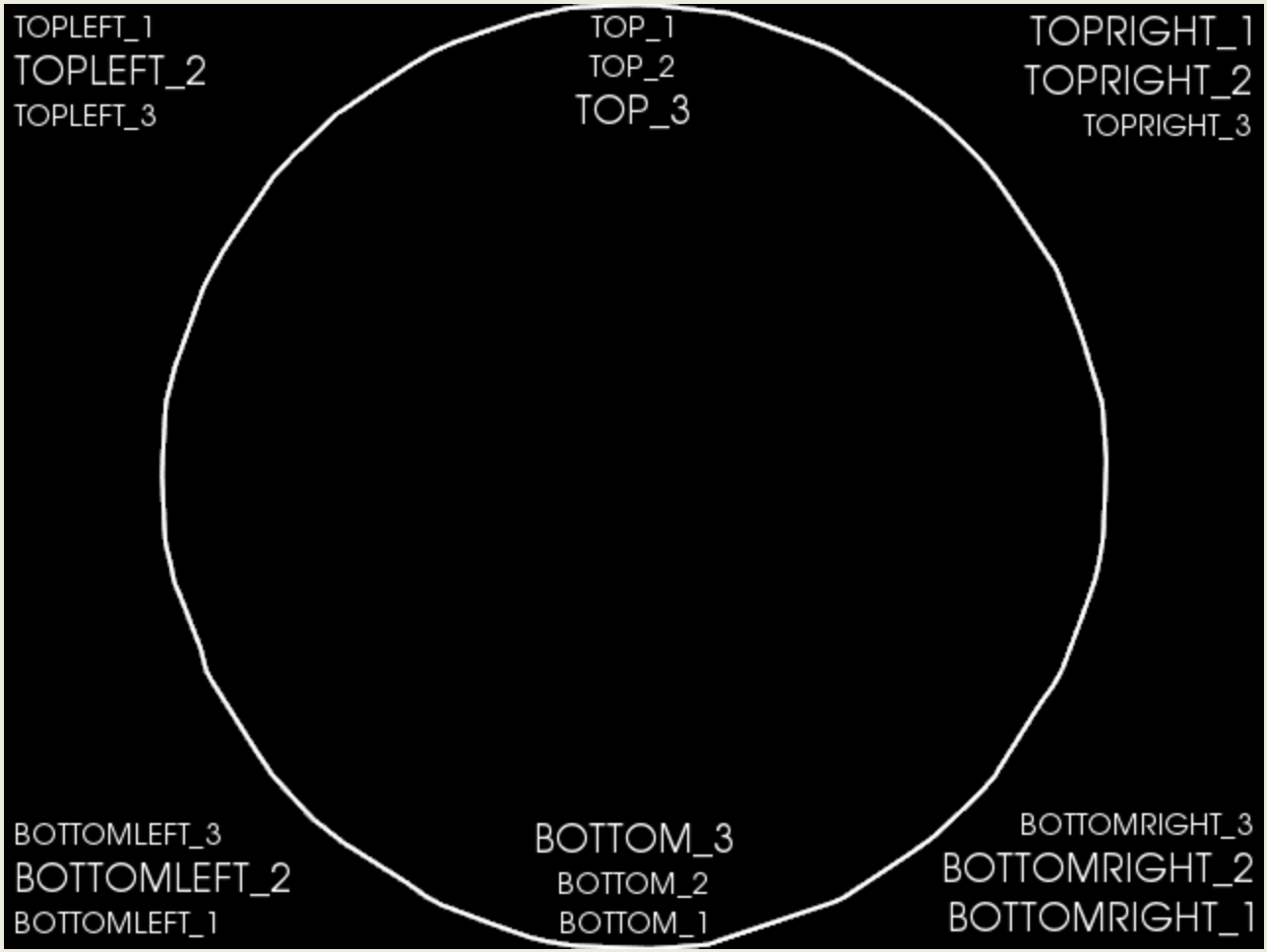
Use Cases

- Render additional information in 2D or 3D eg.:
 - Image metadata as text (patient name, dicom tags...)
 - 3D annotations for pointsets or surfaces
 - Legend (scale or lookup-table)
 - Anything...

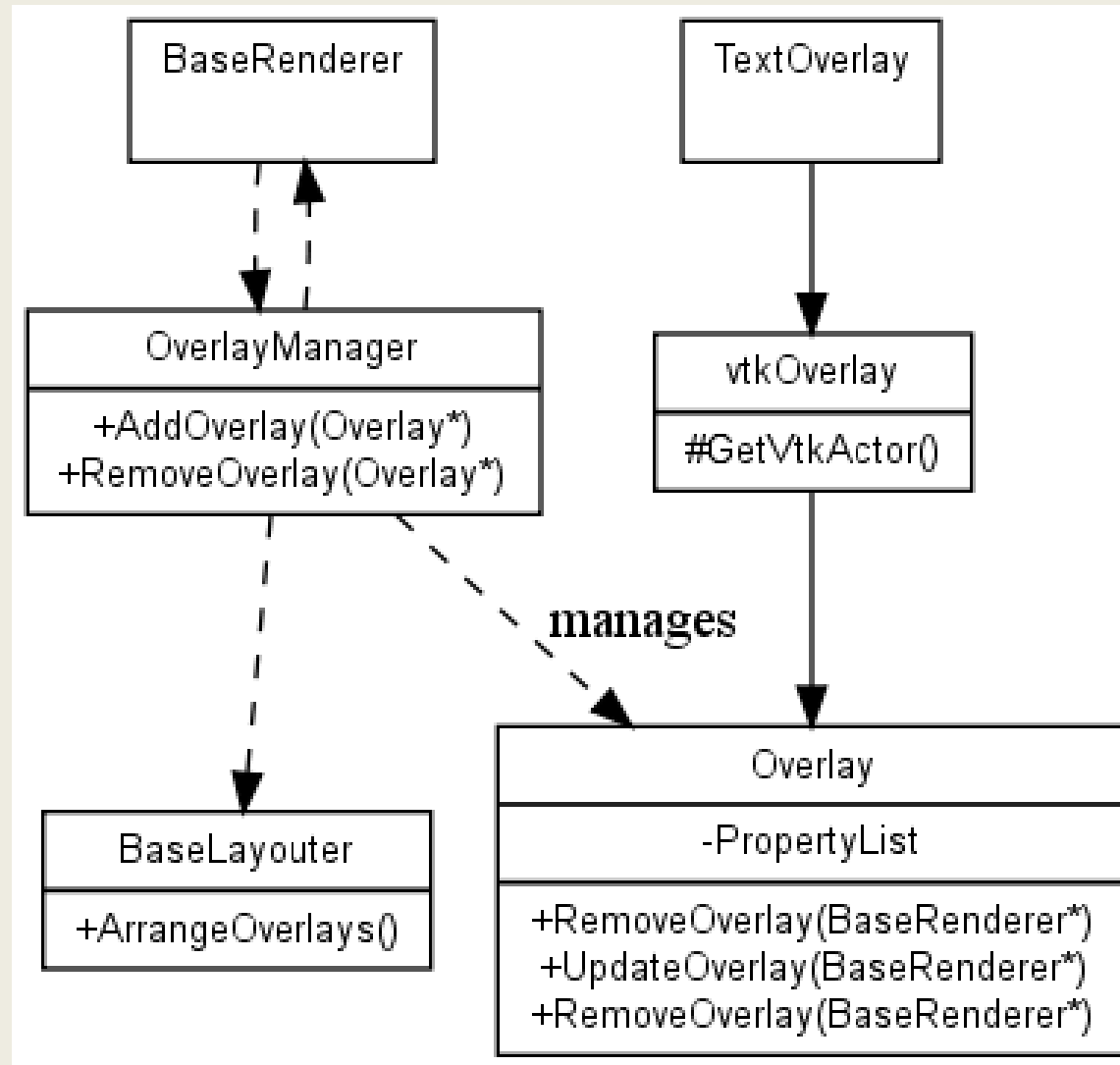








Overview



Add Overlay to Renderer

```
mitk::TextOverlay3D::Pointer to = mitk::TextOverlay3D::New();  
to->SetText("HALLO");  
mitk::Point3D p;  
p.Fill(50);  
to->SetFontSize(30);  
to->SetColor(1,0,0);  
to->SetPosition3D(p);
```

```
mitk::BaseRenderer* renderer =  
mitk::BaseRenderer::GetByName("stdmulti.widget1");
```

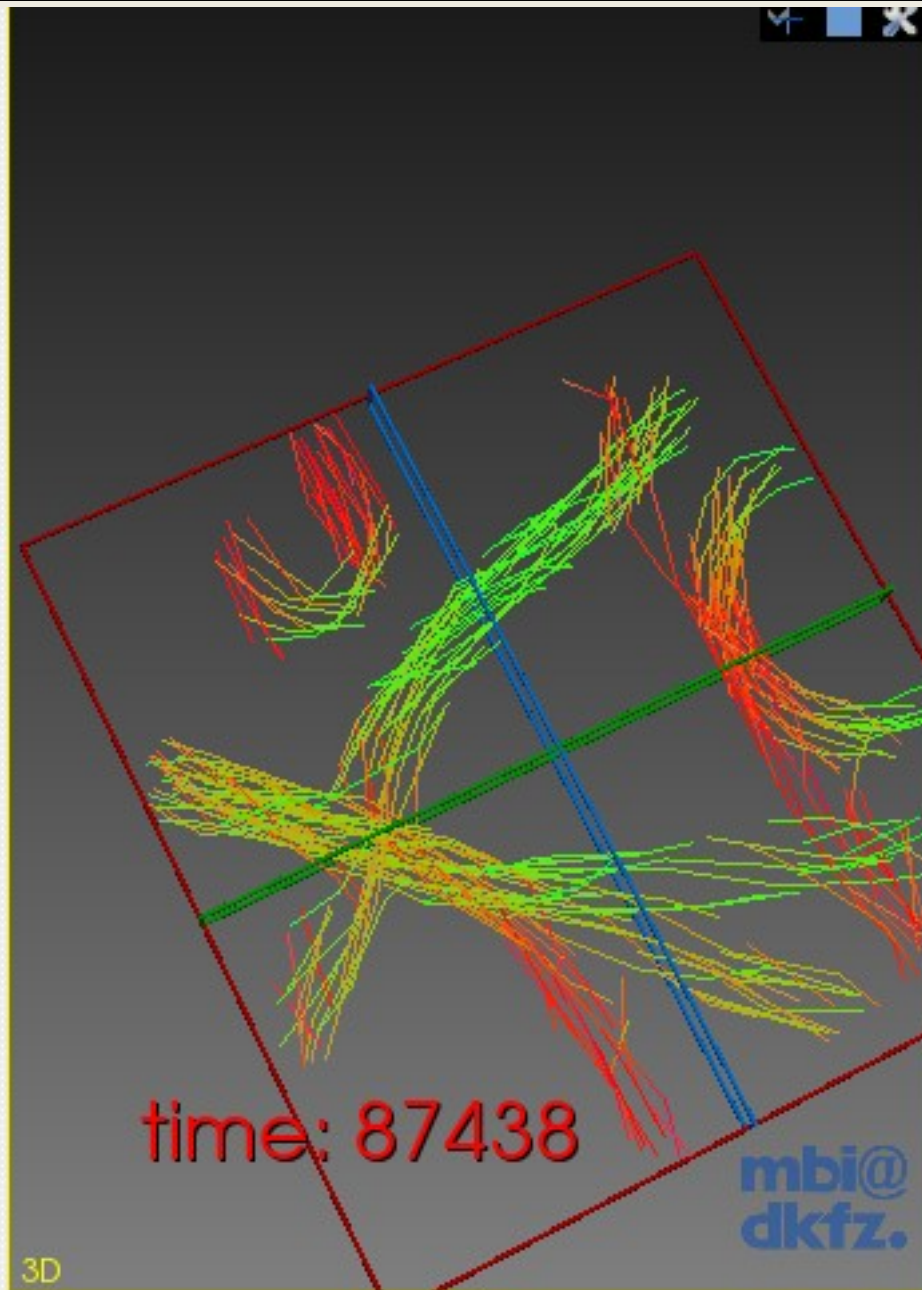
```
renderer->GetOverlayManager()->AddOverlay(to.GetPointer());
```

<http://docs.mitk.org/nightly-qt4/OverlaysPage.html>

Use Overlays in VtkMapper

```
mitk::FiberBundleXMapper3D::GenerateDataForRenderer(  
  
    localStorage->m_Overlay->SetVisibility(true);  
  
    renderer->GetOverlayManager()->AddOverlay(  
        localStorage->m_Overlay.GetPointer(), renderer);  
  
}
```

```
void mitk::FiberBundleXMapper3D::ReleaseGraphicsResources(mitk::BaseRenderer *renderer)  
{  
    FBXLocalStorage3D* localStorage = m_LocalStorageHandler.GetLocalStorage(renderer);  
    renderer->GetOverlayManager()->RemoveOverlay(localStorage->m_Overlay.GetPointer());  
}
```



Use Layouter

```
mitk::Overlay2DLayouter::Pointer topleftLayouter =  
    mitk::Overlay2DLayouter::CreateLayouter(  
        mitk::Overlay2DLayouter::STANDARD_2D_TOPLEFT(), renderer);  
  
overlayManager->AddLayouter(topleftLayouter.GetPointer());  
  
overlayManager->SetLayouter(  
    textOverlay.GetPointer(),  
    mitk::Overlay2DLayouter::STANDARD_2D_TOPLEFT(), renderer);
```

<http://docs.mitk.org/nightly-qt4/OverlaysPage.html>

Implement Overlay

```
class TextOverlay3D : public mitk::VtkOverlay3D
```

<http://docs.mitk.org/nightly-qt4/OverlaysPage.html>

Implement Overlay

```
mitk::TextOverlay3D::LocalStorage::LocalStorage()
{
    // Create a subclass of vtkActor:
    m_follower = vtkSmartPointer<vtkFollower>::New();
}
```

<http://docs.mitk.org/nightly-qt4/OverlaysPage.html>

Implement Overlay

```
void mitk::TextOverlay3D::UpdateVtkOverlay(mitk::BaseRenderer *renderer)
{
  LocalStorage* ls = this->m_LSH.GetLocalStorage(renderer);
  if(ls->IsGenerateDataRequired(renderer,this)) // only if things have changed
  {
    ls->m_follower->SetPosition(
      GetPosition3D(renderer)[0]+GetOffsetVector(renderer)[0],
      GetPosition3D(renderer)[1]+GetOffsetVector(renderer)[1],
      GetPosition3D(renderer)[2]+GetOffsetVector(renderer)[2]);
    ls->m_textSource->SetText(GetText().c_str());
    //...
    ls->UpdateGenerateDataTime();
  }
}
```

<http://docs.mitk.org/nightly-qt4/OverlaysPage.html>

Implement Overlay

```
vtkProp* mitk::TextOverlay3D::GetVtkProp(BaseRenderer *renderer) const
{
    LocalStorage* ls = this->m_LSH.GetLocalStorage(renderer);
    return ls->m_follower;
}
```

<http://docs.mitk.org/nightly-qt4/OverlaysPage.html>