



ITK Spatial Objects

Kitware Inc.

Spatial Objects - Motivation

- Image are a limited representation of physical reality
- Images are not good shape descriptors
- Many of the medical imaging problems relate to defining regions in space

Spatial Objects - Applications

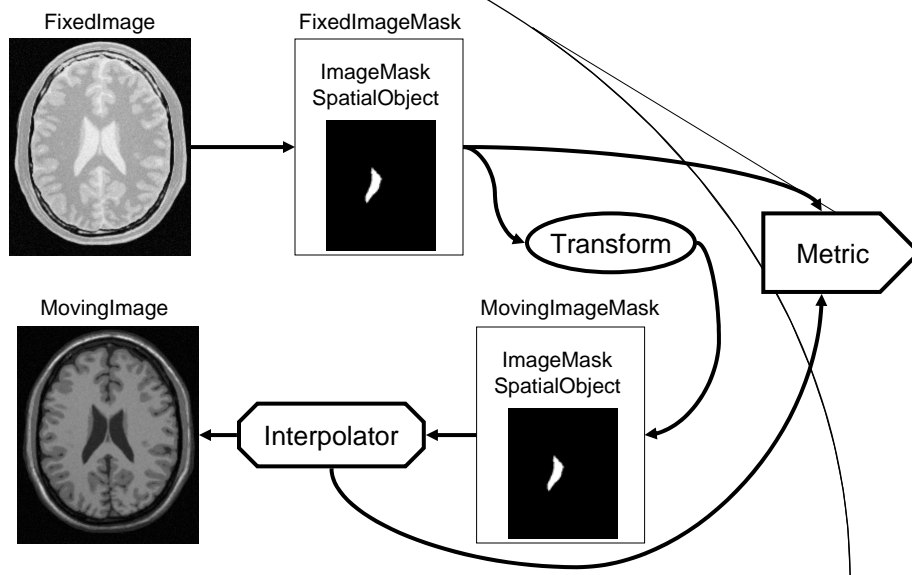
- Model to Image Registration (model-based segmentation)
- Model to Model Registration
- Atlas Formation
- Describing Relationships between objects

Spatial Objects - Features

- Form Hierarchies in Space (scene-graphs)
- Parent-Child Relationships
- Spatial Transform to Parent
- Spatial Transform to World

Spatial Objects Registration Applications

Application in Registration



Masking Registration

```
#include "itkMaskImageSpatialObject.h"

typedef itk:: ImageMaskSpatialObject< Dimension > MaskType;

MaskType::Pointer maskSpatialObject = MaskType ::New();

maskSpatialObject->SetImage( maskReader->GetOutput() );

metric->SetFixedImageMask( maskSpatialObject );

registration->StartRegistration();
```

Masked Registration

Run Command:

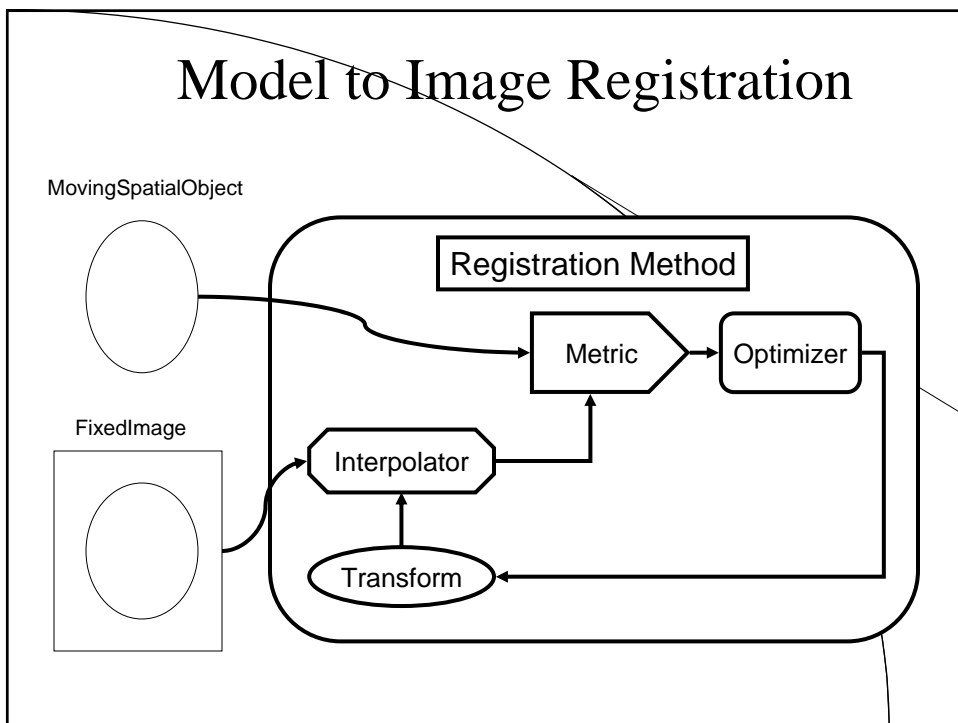
```
ImageRegistrationWithMask
BrainProtonDensitySliceBorder20.png
BrainProtonDensitySliceR10X13Y17.png
BrainProtonDensitySliceBorder20Mask.png
registeredOutput.png
differenceAfter.png
differenceBefore.png
```

Output:

```
Registered image
Difference images before and after
```

Exercise 37a

Model to Image Registration



Model to Image Registration

```
#include "itkImage.h"
#include "itkImageFileReader.h"
#include "itkEllipsoidSpatialObject.h"

#include "itkImageToSpatialObjectMetric.h"
#include "itkImageToSpatialObjectRegistrationMethod.h"
#include "itkLinearInterpolateImageFunction.h"
#include "itkTranslationTransform.h"
#include "itkAmoebaOptimizer.h"

int main( int argc, char * argv[] )
{

typedef itk::Image< float, 3 > ImageType;

typedef itk::EllipseSpatialObject< 3 > EllipseType;
```

Model to Image Registration

```
EllipseType::Pointer ellipse = EllipseType::New();

ellipse->SetRadius( 50.0 );

EllipseType::TransformType::OffsetType offset;

offset[0] = 70.0;
offset[1] = 50.0;
offset[2] = 50.0;

ellipse->GetObjectToParentTransform()->SetOffset( offset );
ellipse->ComputeObjectToWorldTransform();
```

Model to Image Registration

```
typedef itk::ImageToSpatialObjectRegistrationMethod<
    ImageType, EllipseType > RegistrationType;

RegistrationType::Pointer registration = RegistrationType::New();

typedef itk::SimpleImageToSpatialObjectMetric<
    ImageType, EllipseType > MetricType;

MetricType::Pointer metric = MetricType::New();

registration->SetFixedImage( reader->GetOutput() );
registration->SetMovingSpatialObject ( ellipse );

registration->StartRegistration();
```

Model to Image Registration

Run Command:

```
ModelToImageRegistration sphere.mhd 5
```

Output:

```
Translation that overlaps the Ellipse to the image
```

