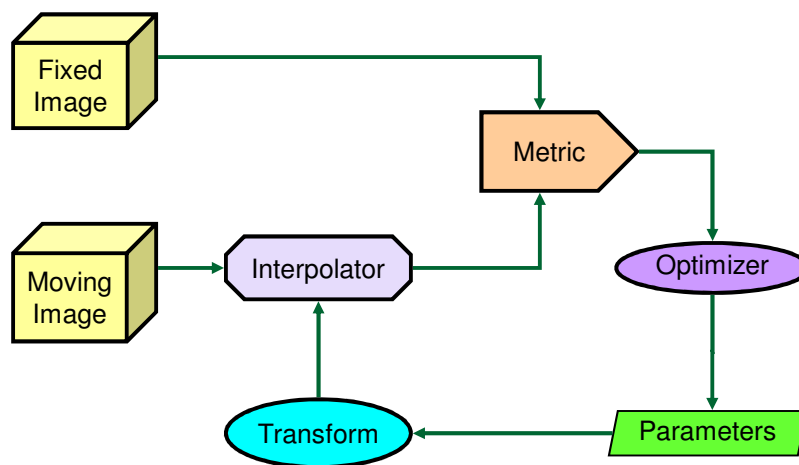


Image Registration

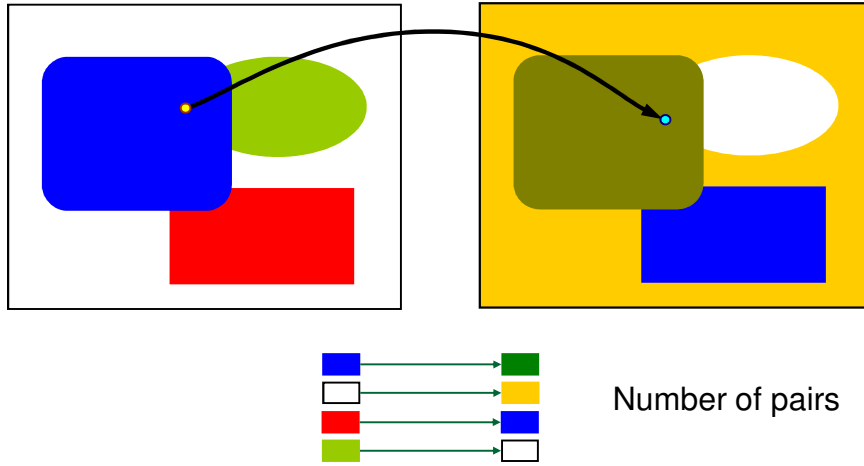
Lecture 10: Registration Components

Prof. Charlene Tsai

Basic Registration Framework

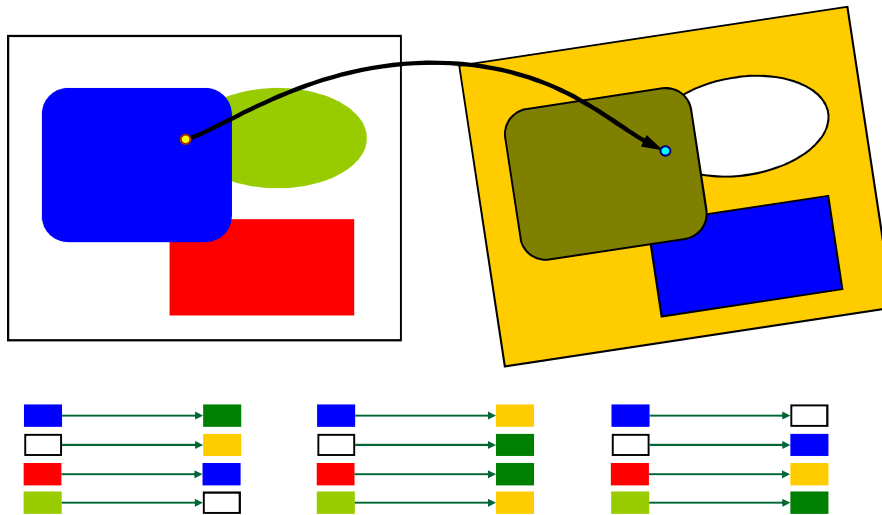


Multiple Image Modal



3

Multi-Modal Registration



More possible pairs

4

Intuitive Notion of Joint Entropy

The More Pairs Exist
The Larger the Joint Entropy

Reduction of Number of Pairs
Reduction of Joint Entropy

5

Mutual Information

Mutual Information =
Joint Entropy (Image A, Image B)
- Entropy Image A
- Entropy Image B

(will have one or two lectures on this topic)

6

Mattes Mutual Information

```
#include "itkImage.h"
#include "itkMattesMutualInformationImageToImageMetric.h"
#include "itkLinearInterpolateImageFunction.h"
#include "itkTranslationTransform.h"

typedef itk::Image< char, 2 > ImageType;

ImageType::ConstPointer fixedImage = GetFixedImage();
ImageType::ConstPointer movingImage = GetMovingImage();

typedef itk::LinearInterpolateImageFunction<
    ImageType,
    double >
    InterpolatorType;

InterpolatorType::Pointer interpolator = InterpolatorType::New();

typedef itk::TranslationTransform< double, 2 > TransformType;

TransformType::Pointer transform = TransformType::New();
```

Mattes Mutual Information

```
typedef
    itk::MattesMutualInformationImageToImageMetric<
        ImageType,
        ImageType
    > MetricType;

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

metric->SetNumberOfHistogramBins( 20 );
metric->SetNumberOfSpatialSamples( 10000 );

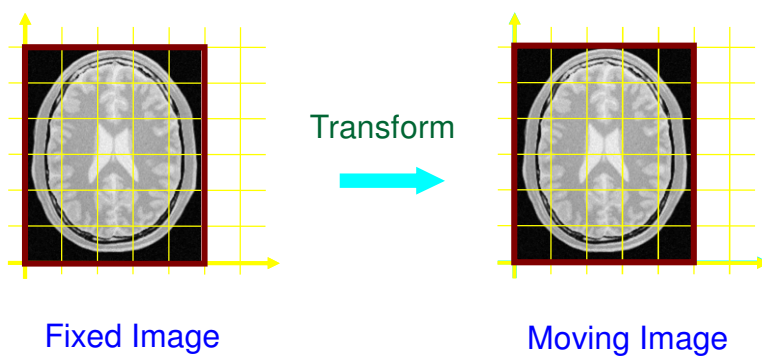
metric->SetInterpolator( interpolator );
metric->SetTransform( transform );

metric->SetFixedImage( fixedImage );
metric->SetMovingImage( movingImage );

MetricType::TransformParametersType translation( Dimension );
translation[0] = 12;
translation[1] = 27;

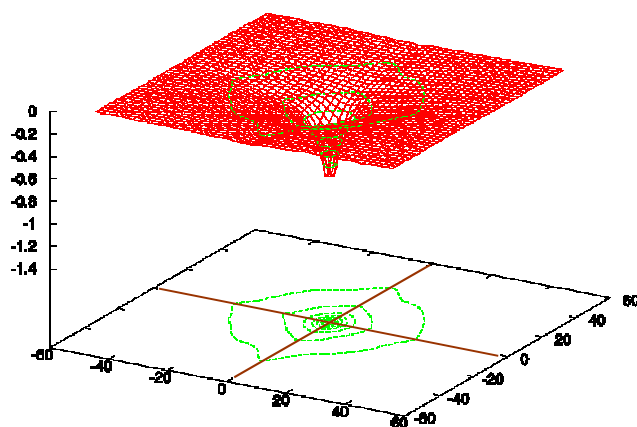
double value = metric->GetValue( translation );
```

Evaluating many matches



9

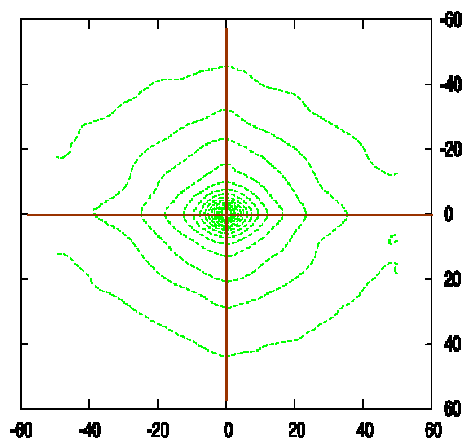
Plotting the Same Modality Metric Mattes Mutual Information



Transform Parametric Space

10

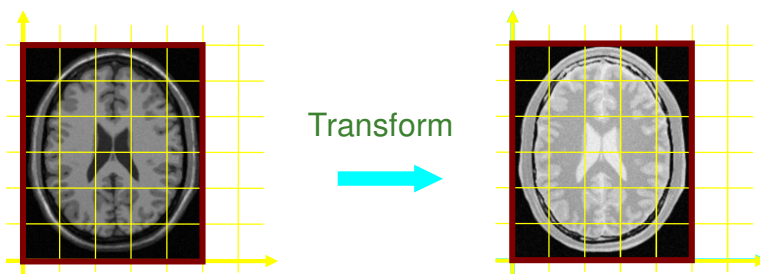
Plotting the Same Modality Metric Matters Mutual Information



Transform Parametric Space

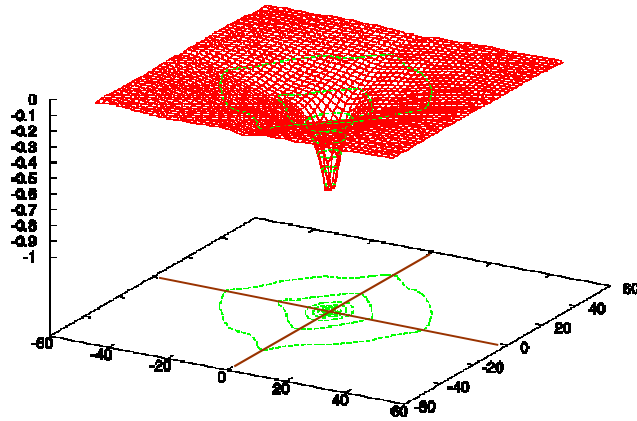
11

Evaluating many matches



12

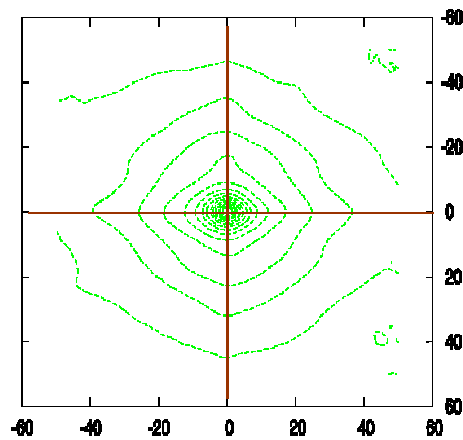
Plotting the Multi-Modality Metric Mattes Mutual Information



Transform Parametric Space

13

Plotting the Multi-Modality Metric Mattes Mutual Information



Transform Parametric Space

14

Mutual Information Variants in ITK

- Viola – Wells
- Mattes
- Mutual Information Histogram

15

Registration Framework with Resampling

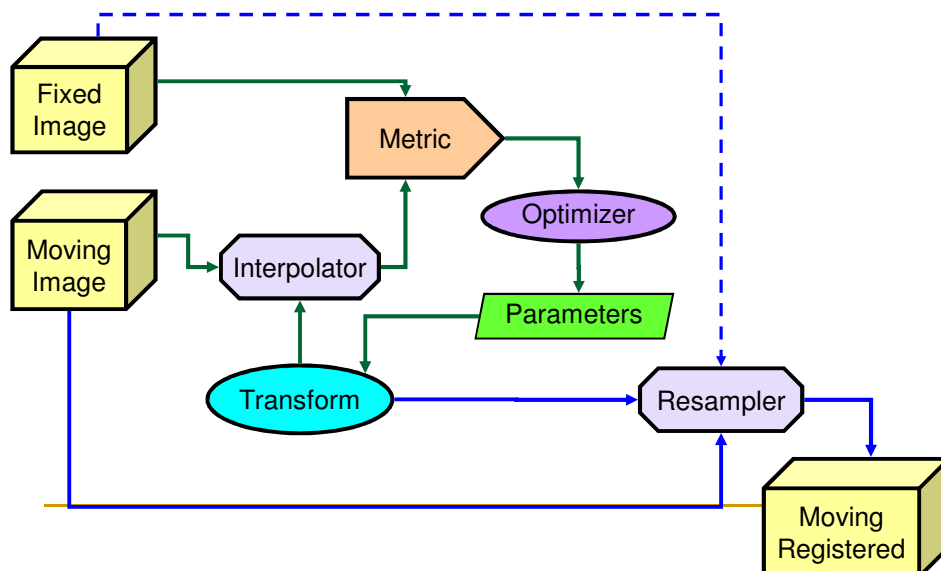


Image Registration

```
#include "itkImageRegistrationMethod.h"  
#include "itkTranslationTransform.h"  
#include "itkMeanSquaresImageToImageMetric.h"  
#include "itkLinearInterpolateImageFunction.h"  
#include "itkRegularStepGradientDescentOptimizer.h"  
#include "itkImage.h"  
#include "itkImageFileReader.h"  
#include "itkImageFileWriter.h"  
#include "itkResampleImageFilter.h"
```

17

Image Registration

```
const unsigned int Dimension = 2;  
typedef unsigned char PixelType;  
  
typedef itk::Image< PixelType , Dimension > FixedImageType;  
typedef itk::Image< PixelType , Dimension > MovingImageType;  
  
typedef itk::TranslationTransform< double, Dimension > TransformType;  
typedef itk::RegularStepGradientDescentOptimizer OptimizerType;  
typedef itk::LinearInterpolateImageFunction<  
    MovingImageType , double > InterpolatorType;  
  
typedef itk::MeanSquaresImageToImageMetric<  
    FixedImageType , MovingImageType > MetricType;  
  
typedef itk::ImageRegistrationMethod<  
    FixedImageType , MovingImageType > RegistrationType;
```

18

Image Registration

```
TransformType::Pointer transform = TransformType::New();
OptimizerType::Pointer optimizer = OptimizerType::New();
InterpolatorType::Pointer interpolator = InterpolatorType::New();
MetricType::Pointer metric = MetricType::New();
RegistrationType::Pointer registrar = RegistrationType::New();

registrar->SetTransform( transform );
registrar->SetOptimizer( optimizer );
registrar->SetInterpolator( interpolator );
registrar->SetMetric( metric );

registrar->SetFixedImage( fixedImageReader->GetOutput() );
registrar->SetMovingImage( movingImageReader->GetOutput() );
```

19

Image Registration

```
registrar->SetFixedImageRegion(
    fixedImageReader->GetOutput()->GetBufferedRegion() );

typedef RegistrationType::ParametersType ParametersType;

transform->SetIdentity();

registrar->SetInitialTransformParameters(
    transform->GetParameters() );

optimizer->SetMaximumStepLength( 4.00 );
optimizer->SetMinimumStepLength( 0.01 );
optimizer->SetNumberOfIterations( 100 );

optimizer->MaximizeOff();
```

20

Image Registration

```
try
{
    registrator->StartRegistration ();
}
catch( itk::ExceptionObject & excp )
{
    std::cerr << "Error in registration" << std::endl;
    std::cerr << excp << std::endl;
}

transform->SetParameters(
    registrator->GetLastTransformParameters() );
```

21

Image Registration

```
typedef itk::ResampleImageFilter<
    FixedImageType , MovingImageType > ResamplerType;

ResamplerType ::Pointer resampler = ResamplerType::New();

resampler->SetTransform ( transform );
resampler->SetInput( movingImageReader->GetOutput() );

FixedImageType ::Pointer fixedImage = fixedImageReader->GetOutput();
resampler->SetOrigin( fixedImage->GetOrigin() );
resampler->SetSpacing( fixedImage->GetSpacing() );
resampler->SetSize(
    fixedImage->GetLargestPossibleRegion()->GetSize() );

resampler->Update();
```

22

Tracking Progress (Observing Registration)

```
#include "itkCommand.h"

class CommandIteration : public itk::Command {
public:
    typedef CommandIteration Self;
    typedef itk::Command SuperClass;
    typedef itk::SmartPointer< Self > Pointer;
    itkNewMacro( Self );

protected:
    CommandIteration() {};

public:
    typedef itk::RegularStepGradientDescentOptimizer OptimizerType;
    typedef const OptimizerType * OptimizerPointer;
```

23

Observing Registration

```
void Execute( itk::Object * caller, const itk::EventObject & event )
{
    this->Execute( (const itk::Object *) caller, event );
}

void Execute( const itk::Object * caller, const itk::EventObject & event )
{
    OptimizerPointer optimizer =
        dynamic_cast< OptimizerPointer >( caller );

    if( typeid( event ) == typeid( itk::IterationEvent ) )
    {
        std::cout << optimizer->GetCurrentIteration() << " : ";
        std::cout << optimizer->GetValue() << " : ";
        std::cout << optimizer->GetCurrentPosition() << std::endl;
    }
}
```

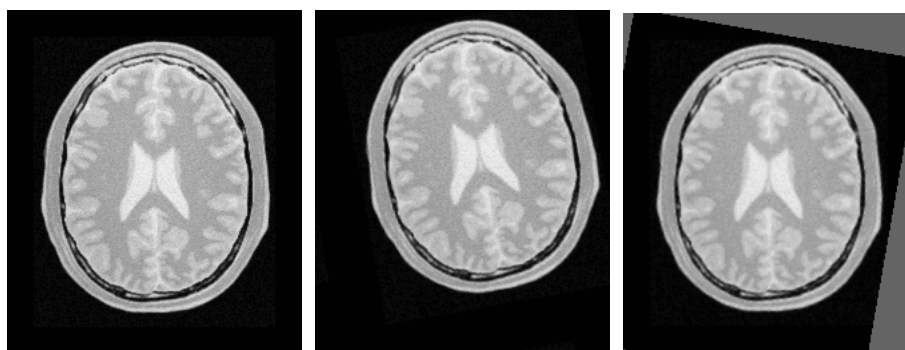
24

Observing Registration

```
CommandIteration::Pointer observer = CommandIteration::New();  
optimizer->AddObserver(itk::IterationEvent(), observer )  
  
try  
{  
    registrator->StartRegistration ();  
}  
catch( itk::ExceptionObject & excp )  
{  
    std::cerr << "Error in registration" << std::endl;  
    std::cerr << excp << std::endl;  
}
```

25

Registration dirty little secrets (parameter scaling)



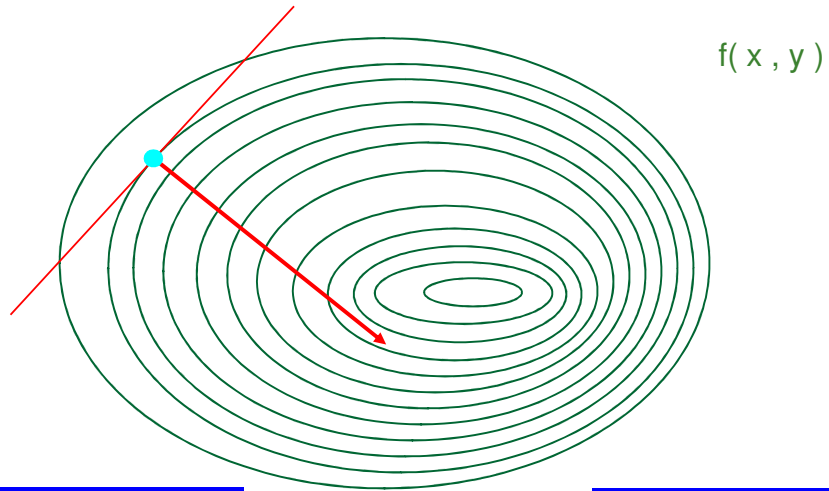
Fixed Image

Moving Image

Registered
Moving Image

26

Gradient Descent Optimizer

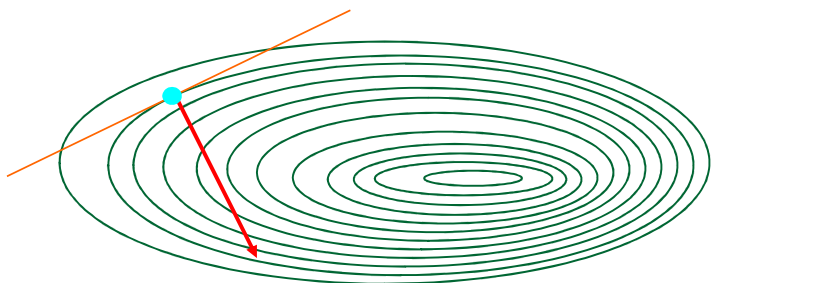


$$\mathbf{G}(x, y) = \nabla f(x, y)$$

$$\mathbf{S} = \mathbf{L} \cdot \mathbf{G}(x, y)$$

27

Gradient Descent Optimizer



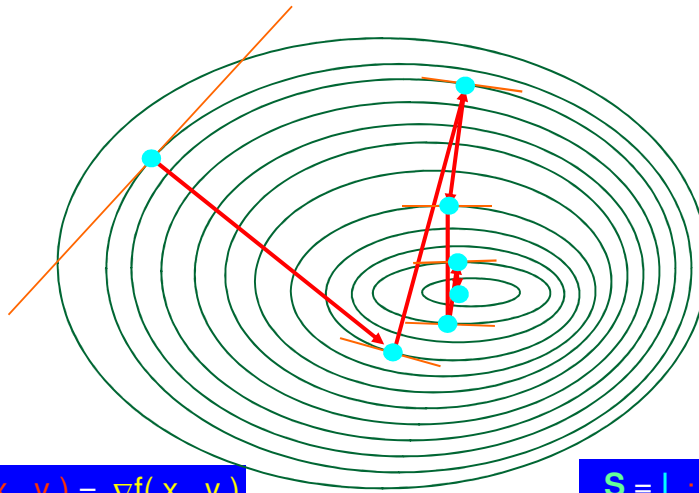
$$\mathbf{G}(x, y) = \nabla f(x, y)$$

$$\mathbf{S} = \mathbf{L} \cdot \mathbf{G}(x, y)$$

28

Gradient Descent Optimizer

$f(x, y)$



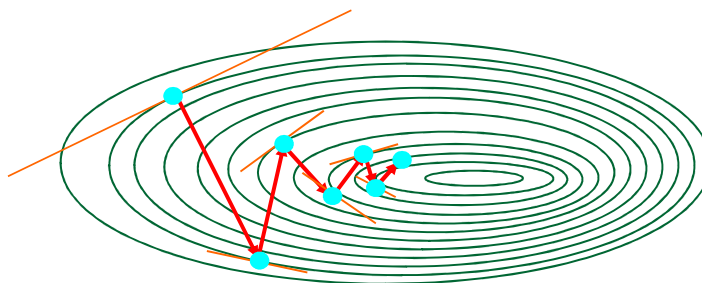
$$\mathbf{G}(x, y) = -\nabla f(x, y)$$

$$\mathbf{S} = L \cdot \mathbf{G}(x, y)$$

29

Gradient Descent Optimizer

$f(x, y)$



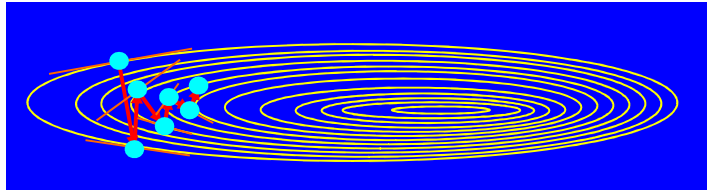
$$\mathbf{G}(x, y) = -\nabla f(x, y)$$

$$\mathbf{S} = L \cdot \mathbf{G}(x, y)$$

30

Gradient Descent Optimizer

$f(x, y)$



$$\mathbf{G}(x, y) = \nabla f(x, y)$$

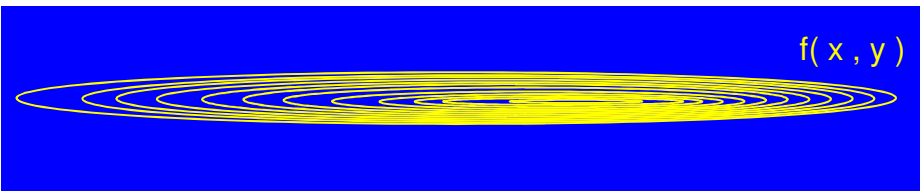
$$\mathbf{S} = \mathbf{L} \cdot \mathbf{G}(x, y)$$

31

Gradient Descent Optimizer

How about a factor 1:100 ?

$f(x, y)$

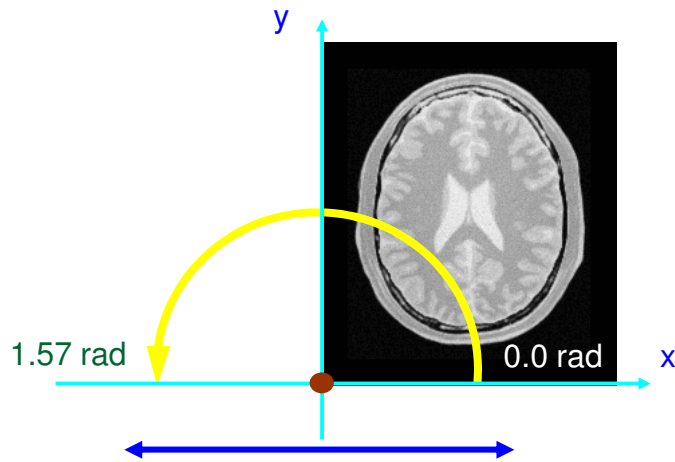


$$\mathbf{G}(x, y) = \nabla f(x, y)$$

$$\mathbf{S} = \mathbf{L} \cdot \mathbf{G}(x, y)$$

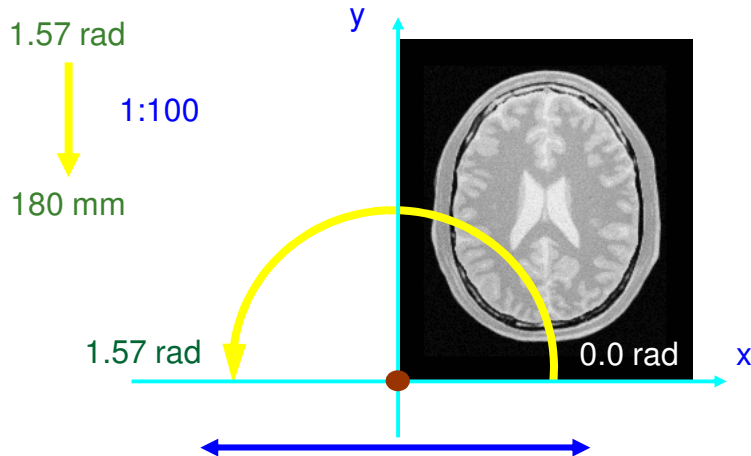
32

Radians & Millimeters



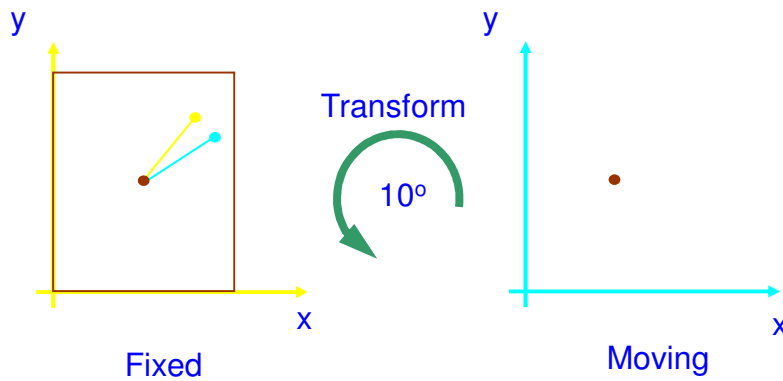
33

Radians & Millimeters



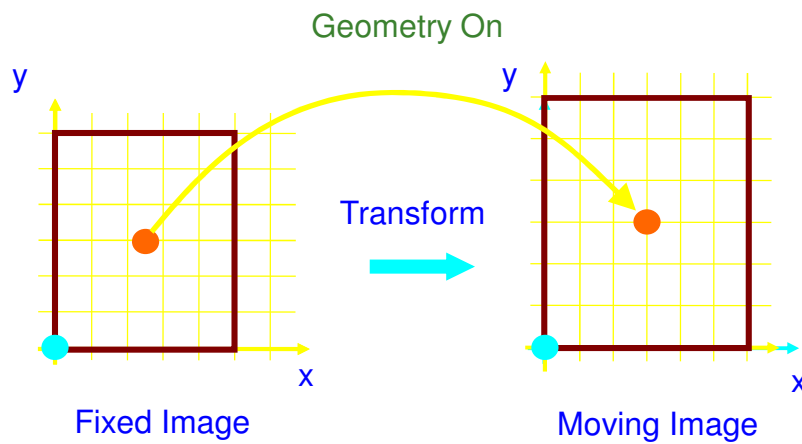
34

Other Transforms: Centered Rigid 2D



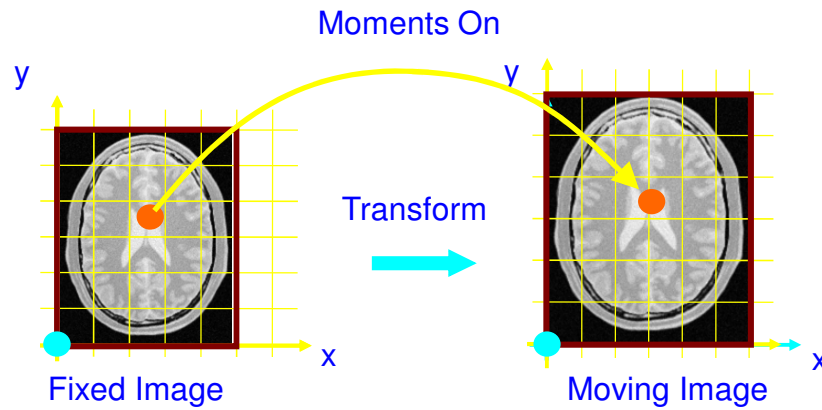
35

Centered Transform initializer



36

Centered Transform Initializer



$$X_m = \text{Sum}(x \cdot I) / \text{Sum}(I)$$
$$Y_m = \text{Sum}(y \cdot I) / \text{Sum}(I)$$

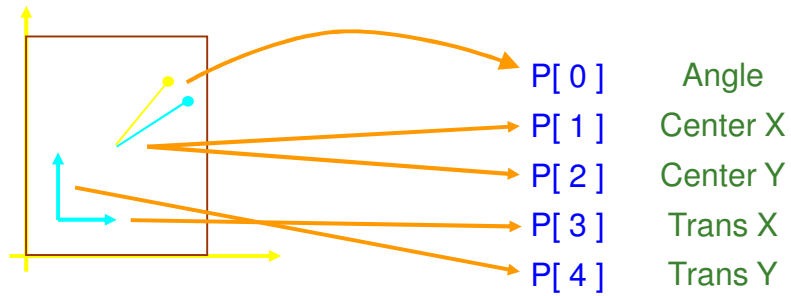
37

Centered Rigid 2D Transform

```
registrator->SetFixedImageRegion(  
    fixedImageReader->GetOutput()->GetBufferedRegion() );  
  
initializer->SetTransform ( transform );  
initializer->SetFixedImage( fixedImageReader->GetOutput() );  
initializer->SetMovingImage( movingImageReader->GetOutput() );  
  
initializer->MomentsOn();  
  
initializer->InitializeTransform();  
  
registrator->SetInitialTransformParameters(  
    transform->GetParameters() );
```

38

Centered Rigid 2D Transform



39

Centered Rigid 2D Transform

```
typedef OptimizerType::ScaleType    OptimizerScalesType;  
  
OptimizerScalesType optimizerScales(  
    optimizer->SetMaximumStepLength() );  
  
const double translationScale = 1.0 / 1000.0 ;  
  
optimizerScales[ 0 ] = 1.0;  
optimizerScales[ 1 ] = translationScale;  
optimizerScales[ 2 ] = translationScale;  
optimizerScales[ 3 ] = translationScale;  
optimizerScales[ 4 ] = translationScale;  
  
optimizer->SetScales( optimizerScales );
```

40

Centered Rigid 2D Transform

```
try
{
    registrar->StartRegistration ();
}
catch( itk::ExceptionObject & excp )
{
    std::cerr << "Error in registration" << std::endl;
    std::cerr << excp << std::endl;
}

transform->SetParameters(
    registrar->GetLastTransformParameters() );

// Skip resampling here
```

41

Other Transforms: Affine

$$\begin{vmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{vmatrix} \quad \begin{vmatrix} T_1 \\ T_2 \end{vmatrix}$$

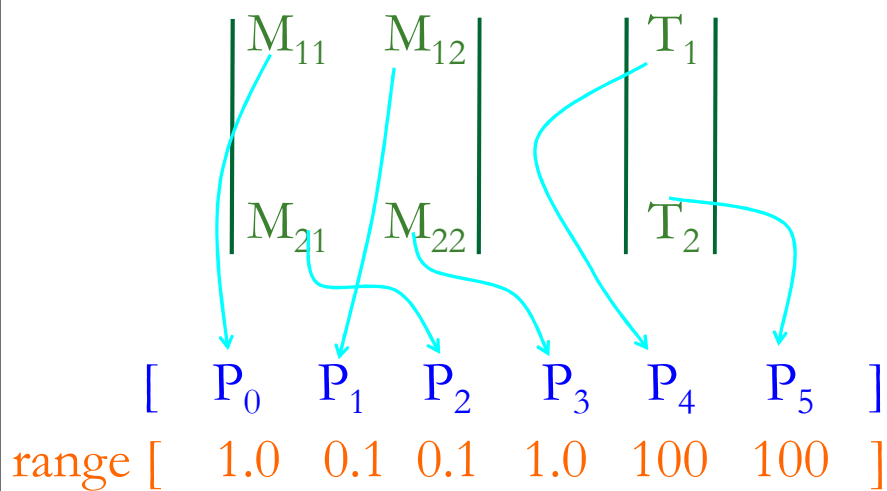
42

Coefficients - Orders of Magnitude

$$\begin{bmatrix} Q_x \\ Q_y \end{bmatrix} = \begin{bmatrix} 1.0 & 0.1 \\ 0.1 & 1.0 \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \end{bmatrix} + \begin{bmatrix} 100 \\ 100 \end{bmatrix}$$

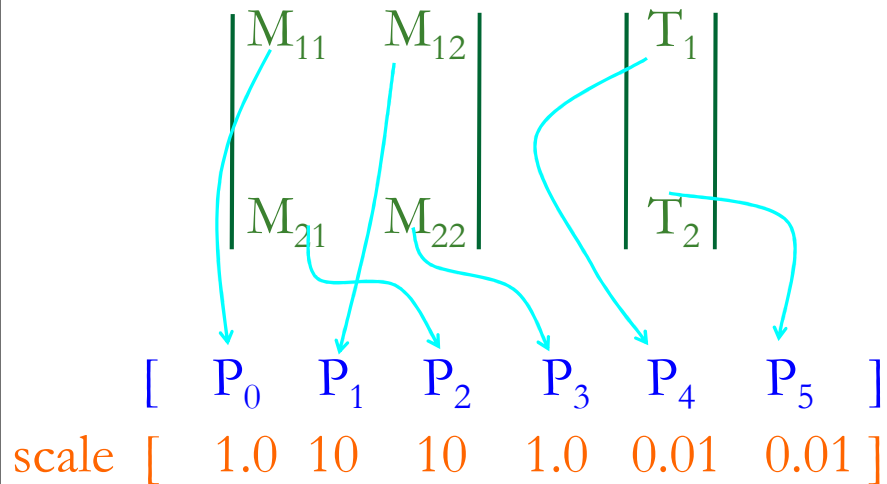
43

Affine Transform - Parameterization



44

Affine Transform – Parameter Scaling



45

Centered Affine Transform

```
#include "itkImageRegistrationMethod.h"
#include "itkCenteredAffineTransform.h"
#include "itkMeanSquaresImageToImageMetric.h"
#include "itkLinearInterpolateImageFunction.h"
#include "itkRegularStepGradientDescentOptimizer.h"
#include "itkCenteredTransformInitializer.h"
#include "itkImage.h"
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"
#include "itkResampleImageFilter.h"
```

46

Centered Affine Transform

```
const unsigned int Dimension = 2;
typedef unsigned char PixelType;

typedef itk::Image< PixelType , Dimension >      FixedImageType;
typedef itk::Image< PixelType , Dimension >      MovingImageType;

typedef itk::CenteredAffineTransform< double >    TransformType;

typedef itk::CenteredTransformInitializer<
    TransformType ,
    FixedImageType ,
    MovingImageType
    >      InitializerType;
```

47

Centered Affine Transform

```
TransformType::Pointer transform = TransformType::New();
InitializerType::Pointer initializer = InitializerType::New();
OptimizerType::Pointer optimizer = OptimizerType::New();
InterpolatorType::Pointer interpolator = InterpolatorType::New();
MetricType::Pointer metric = MetricType::New();
RegistrationType::Pointer registrator = RegistrationType::New();

registrator->SetTransform( transform );
registrator->SetOptimizer( optimizer );
registrator->SetInterpolator( interpolator );
registrator->SetMetric( metric );

registrator->SetFixedImage( fixedImageReader->GetOutput() );
registrator->SetMovingImage( movingImageReader->GetOutput() );
```

48

Centered Affine Transform

```
registrator->SetFixedImageRegion(  
    fixedImageReader->GetOutput()->GetBufferedRegion() );  
  
initializer->SetTransform ( transform );  
initializer->SetFixedImage( fixedImageReader->GetOutput() );  
initializer->SetMovingImage( movingImageReader->GetOutput() );  
  
initializer->MomentsOn(); // Same as for centered rigid 2D  
  
initializer->InitializeTransform();  
  
registrator->SetInitialTransformParameters(  
    transform->GetParameters() );
```

49

Centered Affine Transform

```
typedef OptimizerType::ScaleType OptimizerScalesType;  
  
OptimizerScalesType optimizerScales(  
    optimizer->GetNumberOfParameters() );  
  
optimizerScales[ 0 ] = 1.0;  
optimizerScales[ 1 ] = 10.0;  
optimizerScales[ 2 ] = 10.0;  
optimizerScales[ 3 ] = 1.0;  
optimizerScales[ 4 ] = 0.01;  
optimizerScales[ 5 ] = 0.01;  
  
optimizer->SetScales( optimizerScales );
```

50

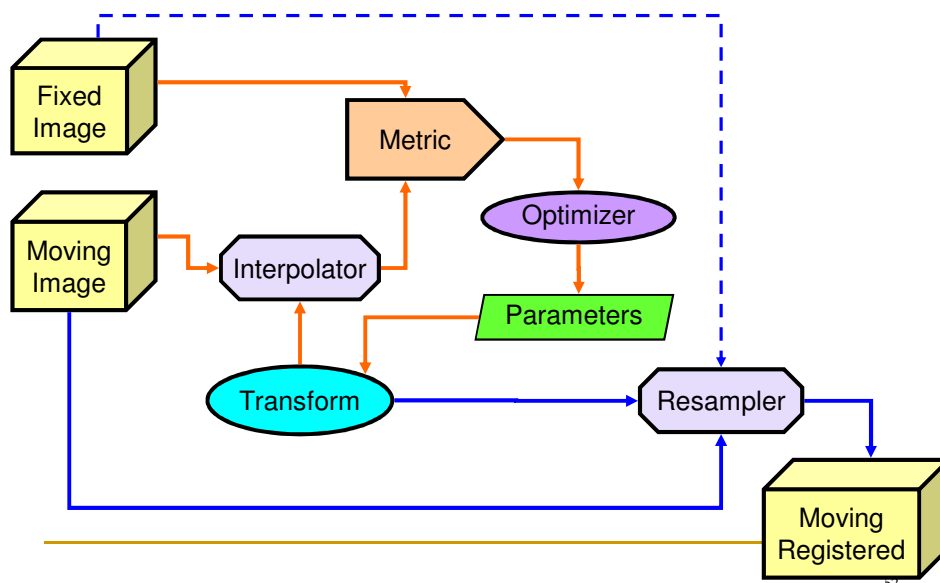
Centered Affine Transform

```
try
{
    registrar->StartRegistration ();
}
catch( itk::ExceptionObject & excp )
{
    std::cerr << "Error in registration" << std::endl;
    std::cerr << excp << std::endl;
}

transform->SetParameters(
    registrar->GetLastTransformParameters() );
```

51

Image Registration Framework



52

Final Resampling

```
typedef itk::ResampleImageFilter<
    FixedImageType , MovingImageType > ResamplerType;

ResamplerType::Pointer resampler = ResamplerType::New();

resampler->SetTransform ( transform );
resampler->SetInput( movingImageReader->GetOutput() );

FixedImageType::Pointer fixedImage = fixedImageReader->GetOutput();
resampler->SetOrigin( fixedImage->GetOrigin() );
resampler->SetSpacing( fixedImage->GetSpacing() );
resampler->SetSize(
    fixedImage->GetLargestPossibleRegion()->GetSize() );

resampler->Update();
```

53

Summary

- Multi-modality registration
 - Mutual information
- Tracking registration process
- Dirty little secret
 - Parameter scaling
- Discussed few more transformation types
 - Centered rigid transform
 - Affine
 - Centered affine

54