



# ITK Basic Filters

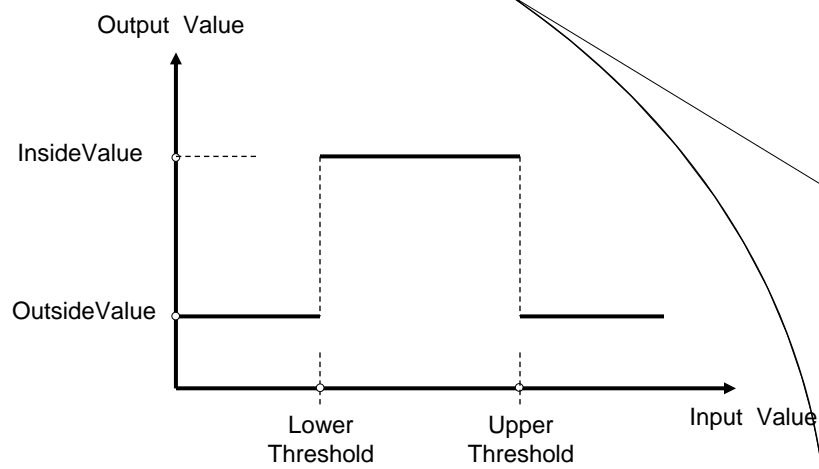
Luis Ibáñez  
William Schroeder  
Insight Software Consortium

## ITK Basic Filters

- Pixel-wise
  - Arithmetic, Casting, Thresholding
- Mathematical morphology
- Noise reduction
  - Gaussian, Anisotropic diffusion
- Derivatives

## Pixel-wise Intensity Filters

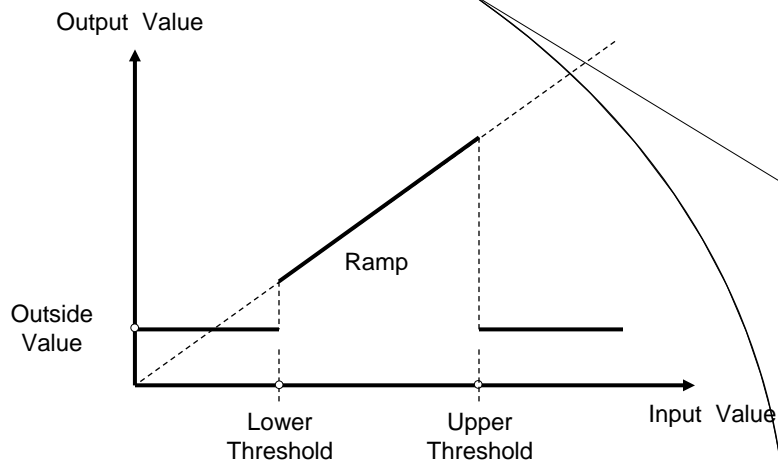
## Thresholding



# Thresholding

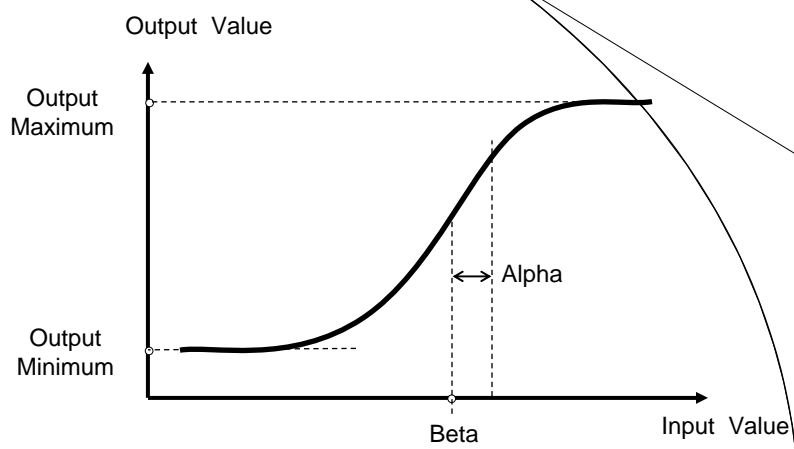
```
typedef itk::Image< unsigned char , 2 > ImageType;  
typedef itk::BinaryThresholdImageFilter< ImageType > FilterType;  
  
FilterType::Pointer filter = FilterType::New();  
  
filter->SetLowerThreshold( 50 );  
filter->SetUpperThreshold( 150 );  
  
filter->SetOutsideValue( 0 );  
filter->SetInsideValue( 255 );  
  
filter->SetInput( reader->GetOutput() );  
writer->SetInput( filter->GetOutput() );  
writer->Update();
```

# Thresholding



## Exercise 6

## Sigmoid



# Sigmoid

```
typedef itk::Image< unsigned char , 2 > ImageType;
typedef itk::SigmoidImageFilter< ImageType, ImageType > FilterType;

FilterType::Pointer filter = FilterType::New();

filter->SetOutputMinimum( 0 );
filter->SetOutputMaximum( 255 );

filter->SetAlpha( 40 ); // it could be negative
filter->SetBeta( 128 );

filter->SetInput( reader->GetOutput() );
writer->SetInput( filter->GetOutput() );
writer->Update()
```

# Exercise 7

# Mathematical Morphology

```
typedef unsigned char PixelType;

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

typedef itk::BinaryBallStructuringElement< PixelType , 2 >
                                             StructuringElementType;

typedef itk::BinaryDilateImageFilter<
    ImageType,
    ImageType,
    StructuringElementType > FilterType;

FilterType::Pointer filter = FilterType::New();
```

# Mathematical Morphology

```
StructuringElementType element;

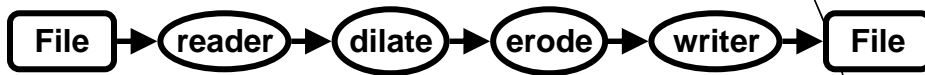
element.SetRadius( 1 );
element.CreateStructuringElement();

filter->SetKernel( element );
filter->SetDilateValue( 255 );

thresholder->SetInput( reader ->GetOutput( ) );
filter->SetInput( thresholder ->GetOutput( ) );
writer->SetInput( filter ->GetOutput( ) );

writer->Update();
```

# Mathematical Morphology



# Exercise 8

## Noise Reduction

## Gaussian Smoothing

```
typedef unsigned char PixelType;
typedef itk::Image< PixelType , 2 > ImageType;
typedef itk::SmoothingRecursiveGaussianImageFilter<
    ImageType,
    ImageType,
    > FilterType;

FilterType::Pointer filter = FilterType::New();

filter->SetSigma( 3.0 ); // sigma is given in millimeters

filter->SetNormalizeAcrossScale( true );
```

# Gaussian Smoothing

```
smoother->SetInput( reader->GetOutput() );  
writer->SetInput( smoother->GetOutput() );
```

```
writer->Update();
```



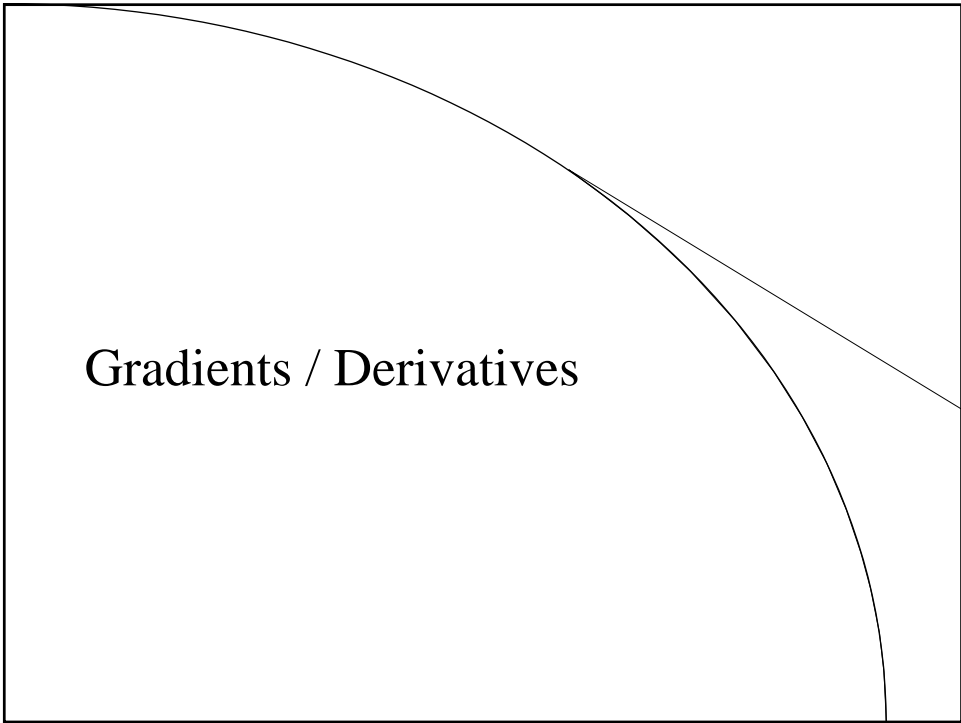
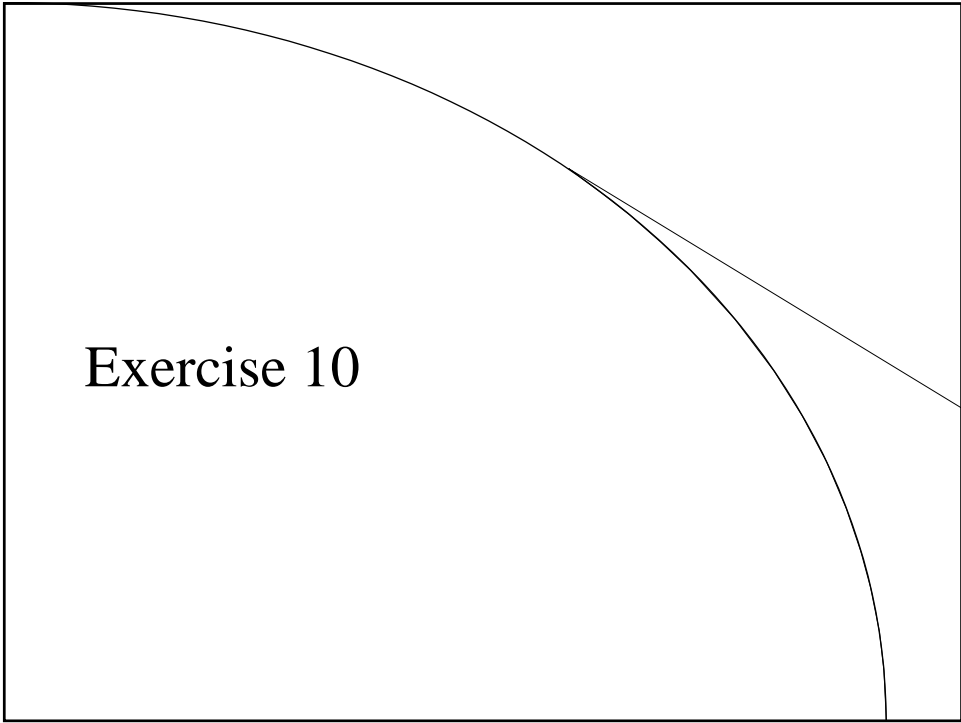
## Exercise 9

# Anisotropic Diffusion

```
typedef float PixelType;  
  
typedef itk::Image< PixelType , 2 > ImageType;  
  
typedef itk::GradientAnisotropicDiffusionImageFilter<  
    ImageType,  
    ImageType,  
    > FilterType;  
  
FilterType::Pointer filter = FilterType::New();  
  
filter->SetTimeStep( 0.05 );  
filter->SetNumberOfIterations( 10 );  
filter->SetConductanceParameter( 3.0 );
```

# Anisotropic Diffusion

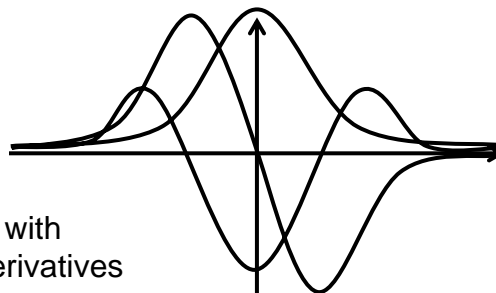
- GradientAnisotropicDiffusionImageFilter
- CurvatureAnisotropicDiffusionImageFilter
- CurvatureFlowImageFilter



## Gaussian Derivative

```
typedef unsigned char PixelType;  
typedef itk::Image< PixelType , 2 > ImageType;  
typedef itk::GradientRecursiveGaussianImageFilter<  
    ImageType > FilterType;  
  
FilterType::Pointer filter = FilterType::New();  
  
filter->SetSigma( 3.0 ); // sigma is given in millimeters  
  
filter->SetNormalizeAcrossScale( true );
```

## Gaussian Gradient



Convolution with  
Gaussian derivatives

## Gradient Magnitude

```
typedef unsigned char PixelType;  
typedef itk::Image< PixelType , 2 > ImageType;  
typedef itk::GradientMagnitudeRecursiveGaussianImageFilter<  
    ImageType > FilterType;  
  
FilterType::Pointer filter = FilterType::New();  
  
filter->SetSigma( 3.0 ); // sigma is given in millimeters  
  
filter->SetNormalizeAcrossScale( true );
```

## Exercise 11

