itkImageFunctionImageFilter: A New Filter To Apply An itkImageFunction To Every Pixel In An itkImage

Release 0.00

David Doria

July 19, 2011

Rensselaer Polytechnic Institute, Troy, NY

Abstract

This document presents a new class to apply an itkImageFunction to every pixel in an itkImage. This functionality is almost identical to itkUnaryFunctorImageFilter, but it uses an itkImageFunction rather than an itk::Functor. As some functionalities in ITK seem to have been implemented as image functors (itk::Functor::AND, itk::Functor::Atan, etc.) while other seems to have been implemented as itkImageFunction's (itk::BinaryThresholdImageFunction, itk::CentralDifferenceImageFunction, etc.), it seems reasonable to be able to apply any of these operations to an entire image.

The code is available here: `http://review.source.kitware.com/#change,2008`

Contents

1 Introduction

This document presents a new class which applies an itkImageFunction to every pixel in an itkImage. This functionality is almost identical to itkUnaryFunctorImageFilter, but it uses an itkImageFunction rather than an itk::Functor. As some functionalities in ITK seem to have been implemented as image functors (itk::Functor::AND, itk::Functor::Atan, etc.) while other seems to have been implemented as itkImageFunction's (itk::BinaryThresholdImageFunction, itk::CentralDifferenceImageFunction, etc.), it seems reasonable to be able to apply any of these operations to an entire image.

2 Template Parameters

The itkImageFunctionImageFilter is templated over the input image type, the output image type, and the type of the ImageFunction. It is necessary to template this filter over the ImageFunction type because it is impossible to create a pointer to the superclass itkImageFunction. This is because at compile time, the 3rd required template parameter of itkImageFunction, the TCoordRep, is not known:

```
typedef typename itk::ImageFunction< TInputImage,
                                     TOutputImage, [?]>::Pointer ImageFunctionPointer;
```

The correct type would be TImageFunction::CoordRepType, which cannot be known at compile time unless you specify the ImageFunction type as a template parameter.

3 Code Snippet

The interface to the filter is very simple. The user must simply setup an itkImageFunction and pass it to the itkImageFunctionImageFilter using the SetFilter function. The image on which to operate is passed as usual via the SetInput function.

```
  typedef itk::GaussianBlurImageFunction< UnsignedCharImageType >
          GaussianBlurImageFunctionType;
  GaussianBlurImageFunctionType::Pointer gaussianFunction =
          GaussianBlurImageFunctionType::New();
  GaussianBlurImageFunctionType::ErrorArrayType setError;
  setError.Fill( 0.01 );
  gaussianFunction->SetMaximumError( setError );
  gaussianFunction->SetSigma( 1.0);
  gaussianFunction->SetMaximumKernelWidth( 3 );

  typedef itk::ImageFunctionImageFilter<UnsignedCharImageType,
               FloatImageType,GaussianBlurImageFunctionType>
               ImageFunctionImageFilterType;
  ImageFunctionImageFilterType::Pointer imageFunctionImageFilter =
               ImageFunctionImageFilterType::New();
  imageFunctionImageFilter->SetInput(image);
  imageFunctionImageFilter->SetFunction(gaussianFunction);
```

```
imageFunctionImageFilter->Update();
```