New Features for MinimumMaximumImageCalculator

Release 0.01

Robert Tamburo[1]

December 17, 2010

[1]robert.tamburo@gmail.com

Abstract

New functions are proposed for itkMinimumMaximumImageCalculator. One function allows for excluding pixels of a specified value from the computations. Another function stores all pixel indices of the maximum/minimum value. This paper is accompanied with source code for the filter and test, test images and parameters, and expected output images.

Contents

1 Implementation of Algorithm

The proposed features for `itkMinimumMaximumImageCalculator` are intended to be added to the current class while maintaining backwards compatibility. For this publication, the class is renamed `itkNewMinimumMaximumImageCalculator` to avoid confusion.

`itkMinimumMaximumImageCalculator` currently resides within ITK as a means to calculate the minimum/maximum intensity value within an image. The user can obtain these values and also the index of the pixel with this value. If multiple pixels have the minimum/maximum value, the last index traversed will be returned. This submission adds the ability to return all of the pixels with the computed minimum/maximum value. Indices are stored in an `std::vector` after the minimum/maximum value is identified within the Compute functions. The indices can be retrieved by `GetIndicesOfMinimum()` and `GetIndicesOfMaximum()`.

Another feature included with this submission is the ability to exclude a pixel value from the minimum/maximum computation. This functionality is useful, for example, when computing the minimum/maximum intensity values of an image with known values for the background. To maintain backwards compatibility three Compute functions were added with the pixel value to exclude as an argument: `Compute(PixelType excludeMin, PixelType excludeMax)`, `ComputeMaximum(PixelType excludeMax)`, and `ComputeMinimum(PixelType excludeMin)`

2 Example Usage

```
typedef itk::NewMinimumMaximumImageCalculator<ImageType> CalculatorType;
CalculatorType::Pointer calc = CalculatorType::New();
calc->SetInput(inputImage); // set the input image
calc->Compute(minToExclude, maxToExclude); // compute minimum/maximum intensity values
                                           // excluding pixels of intensity minToExclude
                                           // from the minimum computation and
                                           // maxToExclude from the maximum computation.
calc->GetIndicesOfMaximum(); // get indices of pixels with value equal to m_Maximum
calc->GetIndicesOfMinimum(); // get indices of pixels with value equal to m_Minimum
```

3 Testing

A test is included with this submission that exercises the new functions. The test generates a 200 × 200 image (Figure 1) containing a dark gray 25 × 25 square, light gray 49 × 49 square, and white 99 × 99 square and verifies that the resulting minimum/maximum values and minimum/maximum indices are as expected.

4 Software Used

This filter was developed on a Windows 7 64-bit computer. It has been successfully tested with ITK version 3.18.0, MinGW version 5.1.6, and CMake version 2.8.2 (Windows binary), and gcc version 4.3.4 20090804 release 1 under cygwin.

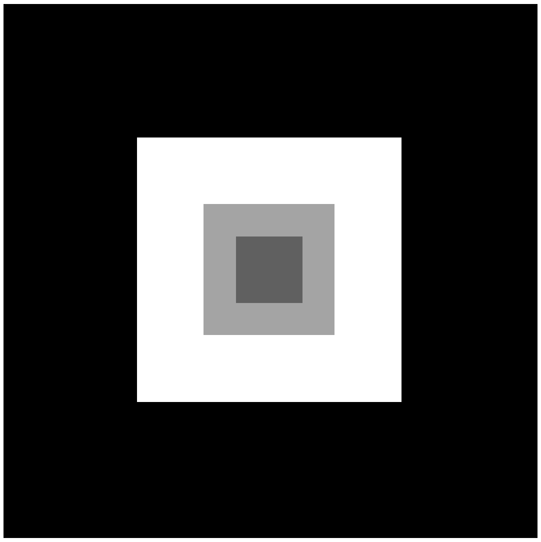

Figure 1: The generated image for the test.