itkMultipleUnlabeledImagesToLabeledImageFilter

Release 0.01

Robert Tamburo

October 18, 2006

University of Pittsburgh
Department of Psychiatry

Abstract

This document describes a filter called `itkMultipleUnlabeledImagesToLabeledImageFilter`, which takes as input a vector of images and generates a single image of labeled pixels. Submitted with this document is the source code for the filter and source code for demonstrating filter usage via an image input as an argument. Also included is source code for testing the functionality of the filter.

Contents

1 Filter Description and Implementation

A filter called `itkMultipleUnlabeledImagesToLabeledImageFilter` has been developed for the Insight Toolkit. This filter takes a vector of images as input, as well as a vector of intensity thresholds associated with each image. The intensity thresholds are used in conjunction with the `itkThresholdImageFilter` to assign a label to pixels in each image. The individually labeled images are combined into a single image via the `itkAddImageFilter`.

There are a couple of caveats to be aware of when using this filter:

1. If a pixel has multiple labels, the final label will be the sum of the individual labels.

2. This filter derives from `ImageToImageFilter`, but no input images are required, i.e., `this->SetNumberOfRequiredInputs(0)`.

2 Filter Usage

Usage of this filter is now described. The reader is referred to main.cxx for a working demonstration. The filter is instantiated much like other filters found in the Insight Toolkit, i.e.,

```
typedef itkMultipleUnlabeledImagesToLabeledImageFilter <ImageType> FilterType;

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

The list of input images is set by passing an `std::vector` of itk::Image smart pointers to the filter via `SetInputImageList()`. An exception is thrown if there are no images are in the input image list. Individual image thresholds are set via passing a `std::vector` of values (of image pixel type) to the filter via `SetIntensityThresholds()`. An exception is thrown if the number of thresholds does not equal the number of input images. The thresholds can be lower intensity thresholds by setting `SetUseLowerThreshold()` to true or upper intensity thresholds by setting `SetUseLowerThreshold()` to false.

3 Example

The code in `MultipleUnlabeledImagesToLabeledImageFilterTest` reads in the 4 input images (test1.bmp, test2.bmp, test3.bmp, test4.bmp) and utilizes this filter to produce a single labeled image. The usage for this test is shown below:

```
MultipleUnlabeledImagesToLabeledImageFilterTest testfiles testoutput.png 1 lower
```

The first argument is a text file containing the names of input images. The second argument is the desired named of the final output image. The third argument is the intensity threshold to be used by the filter. The last argument is a flag to use the thresholds as a lower intensity threshold. Note that for this test the same lower intensity threshold is used for each image. The input images and final output image are shown in Fig. 1.

Figure 1: A. through D. show the input test images and E. shows the labeled output image with assigned labels overlayed. Note that the output image intensities were scaled for visualization purposes.

4 Software Requirements

This filter was written with the following software installed:

1. Insight Toolkit 2.8.1.
2. CMake 2.4 patch 3.

This filter was tested with the included main.cxx on a MacBook pro with gcc 4.0.1 with 0 errors and 0 warnings.

itkMultipleUnlabeledImagesToLabeledImageFilter

Release 0.02

Robert Tamburo

November 1, 2006

University of Pittsburgh
Department of Psychiatry

Abstract

This document describes a filter called `itkMultipleUnlabeledImagesToLabeledImageFilter`, which takes as input a vector of images and generates a single image of labeled pixels. Submitted with this document is the source code for the filter and source code for demonstrating filter usage via an image input as an argument. Also included is source code for testing the functionality of the filter.

Contents

1 Filter Description and Implementation

A filter called `itkMultipleUnlabeledImagesToLabeledImageFilter` has been developed for the Insight Toolkit. This filter takes a vector of images as input, as well as a vector of intensity thresholds associated with each image. The intensity thresholds are used in conjunction with the `itkThresholdImageFilter` to assign a label to pixels in each image. The individually labeled images are combined into a single image via the `itkAddImageFilter`.

There are a couple of caveats to be aware of when using this filter:

1. If a pixel has multiple labels, the final label will be the sum of the individual labels.

2. This filter derives from `ImageToImageFilter`, but no input images are required, i.e., `this->SetNumberOfRequiredInputs(0)`.

2 Filter Usage

Usage of this filter is now described. The reader is referred to main.cxx for a working demonstration. The filter is instantiated much like other filters found in the Insight Toolkit, i.e.,

```
typedef itkMultipleUnlabeledImagesToLabeledImageFilter <ImageType> FilterType;

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

The list of input images is set by passing an `std::vector` of itk::Image smart pointers to the filter via `SetInputImageList()`. An exception is thrown if there are no images are in the input image list. Individual image thresholds are set via passing a `std::vector` of values (of image pixel type) to the filter via `SetIntensityThresholds()`. An exception is thrown if the number of thresholds does not equal the number of input images. The thresholds can be lower intensity thresholds by setting `SetUseLowerThreshold()` to true or upper intensity thresholds by setting `SetUseLowerThreshold()` to false.

3 Example

The code in `MultipleUnlabeledImagesToLabeledImageFilterTest` reads in the 4 input images (image1.bmp, image2.bmp, image3.bmp, image4.bmp) and utilizes this filter to produce a single labeled image. The usage for this test is shown below:

```
MultipleUnlabeledImagesToLabeledImageFilterTest output.png 1 lower /images/image1.bmp
                                                /images/image2.bmp /images/image3.bmp
                                                /images/image4.bmp
```

The first argument is the desired named of the final output image. The second argument is the intensity threshold to be used by the filter. The third argument is a flag to use the thresholds as a lower intensity threshold. The final outputs are the names of the input images. Note that for this test the same lower intensity threshold is used for each image. The input images and final output image are shown in Fig. 1. A label of 1 is assigned to the pixels from image1.bmp, a label of 2 is assigned to the pixels from image2.bmp, etc. To visualize the final labeled image, the intensities was scaled in graphic editing software.

Figure 1: A. through D. show the input test images and E. shows the labeled output image with assigned labels overlayed. Note that the output image intensities were scaled for visualization purposes.

4 Software Requirements

This filter was written with the following software installed:

1. Insight Toolkit 2.8.1.
2. CMake 2.4 patch 3.

This filter was tested with the included main.cxx on a MacBook pro with gcc 4.0.1 with 0 errors and 0 warnings.

itkMultipleUnlabeledImagesToLabeledImageFilter

Release 0.03

Robert Tamburo

December 15, 2006

University of Pittsburgh
Department of Psychiatry

Abstract

This document describes a filter called `itkMultipleUnlabeledImagesToLabeledImageFilter`, which takes multiple images as input and generates a single image of labeled pixels. Pixels to be labeled for each image are determined by an intensity threshold. Submitted with this document is the source code for the filter and source code for demonstrating filter usage via an image input as an argument. Also included is source code for testing the functionality of the filter.

Contents

1 Filter Description and Implementation

A filter called `itkMultipleUnlabeledImagesToLabeledImageFilter` has been developed for the Insight Toolkit. This filter takes multiple images as input, as well as a vector of intensity thresholds associated with each image. Binary input images are not required. The intensity thresholds determine those pixels that are to be labeled. The intensity thresholds are used in conjunction with the `itkThresholdImageFilter` to assign a label to pixels in each image. The individually labeled images are combined into a single image via the `itkAddImageFilter`. This filter has the caveat that if corresponding input pixels are to be labeled, the final label will be the sum of the individual labels.

2 Filter Usage

Usage of this filter is now described. The reader is referred to main.cxx for a working demonstration. The filter is instantiated much like other filters found in the Insight Toolkit, i.e.,

```
typedef itkMultipleUnlabeledImagesToLabeledImageFilter <ImageType> FilterType;

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

The input images are set by passing a pointer to each input image and index via `SetInput()`. An exception is thrown if there are no images are in the input image list. Individual image thresholds are set via passing a `std::vector` of values (of image pixel type) to the filter via `SetIntensityThresholds()`. An exception is thrown if the number of thresholds does not equal the number of input images. The thresholds can be lower intensity thresholds by setting `SetUseLowerThreshold()` to true or upper intensity thresholds by setting `SetUseLowerThreshold()` to false.

3 Example

The code in `MultipleUnlabeledImagesToLabeledImageFilterTest` reads in the 4 input images (image1.bmp, image2.bmp, image3.bmp, image4.bmp) and utilizes this filter to produce a single labeled image. The usage for this test is shown below:

```
MultipleUnlabeledImagesToLabeledImageFilterTest output.png 1 lower /images/image1.bmp
                                                /images/image2.bmp /images/image3.bmp
                                                /images/image4.bmp
```

The first argument is the desired named of the final output image. The second argument is the intensity threshold to be used by the filter. The third argument is a flag to use the thresholds as a lower intensity threshold. The final outputs are the names of the input images. Note that for this test the same lower intensity threshold is used for each image. The input images and final output image are shown in Fig. 1. A label of 1 is assigned to the pixels from image1.bmp, a label of 2 is assigned to the pixels from image2.bmp, etc. To visualize the final labeled image, the intensities were scaled in a graphics editing software suite.

Figure 1: A. through D. show the input test images and E. shows the labeled output image with assigned labels overlayed.

4 Software Requirements

This filter was written with the following software installed:

1. Insight Toolkit 2.8.1.
2. CMake 2.4 patch 3.

This filter was tested with the included main.cxx on a MacBook pro with gcc 4.0.1 with 0 errors and 0 warnings.