
Some extra image logical operations

Release 0.00

Richard Beare

February 13, 2007

Richard.Beare@ieee.org

Abstract

Logic operations such as $I > J$, where I and J are images are useful when generating masks. This submission includes `==`, `>`, `>=`, `<` and `<=` filters based on the existing logic filters. This is a very minor submission - there may already be easy ways of performing the logic operations defined by these filters, but I couldn't find them.

Contents

1	Introduction	1
2	Using the new filters	1

1 Introduction

This submission is hardly worth describing. It consists of filters for pixelwise equality and comparison operations of images, which I couldn't see any other direct ways of achieving in ITK. Please indicate in your comments if I've missed something. The implementation is copied directly from the other logic filters, such as *itkAndImageFilter* and is therefore threaded.

2 Using the new filters

The new filters are: *itkEqualImageFilter*, *itkGreaterEqualImageFilter*, *itkGreaterImageFilter*, *itkLessEqualImageFilter*, *itkLessImageFilter*. Usage is identical to other logic filters and is illustrated in the test function - *check.cxx* included in the archive.

```
const int dim = 2;
```

```
typedef unsigned char PType;
```

```
typedef itk::Image< PType, dim > IType;

typedef itk::ImageFileReader< IType > ReaderType;
ReaderType::Pointer reader = ReaderType::New();
reader->SetFileName( argv[1] );

typedef itk::SmoothingRecursiveGaussianImageFilter<IType, IType> SmoothType;
SmoothType::Pointer smoother = SmoothType::New();

smoother->SetInput(reader->GetOutput());
smoother->SetSigma(1);

typedef itk::EqualImageFilter<IType, IType, IType> EqType;
typedef itk::GreaterImageFilter<IType, IType, IType> GTType;
typedef itk::GreaterEqualImageFilter<IType, IType, IType> GTEqType;
typedef itk::LessImageFilter<IType, IType, IType> LTType;
typedef itk::LessEqualImageFilter<IType, IType, IType> LTEqType;

EqType::Pointer eq = EqType::New();
GTType::Pointer gt = GTType::New();
GTEqType::Pointer gte = GTEqType::New();
LTType::Pointer lt = LTType::New();
LTEqType::Pointer lte = LTEqType::New();

eq->SetInput(reader->GetOutput());
gt->SetInput(reader->GetOutput());
gte->SetInput(reader->GetOutput());
lt->SetInput(reader->GetOutput());
lte->SetInput(reader->GetOutput());

eq->SetInput2(smoother->GetOutput());
gt->SetInput2(smoother->GetOutput());
gte->SetInput2(smoother->GetOutput());
lt->SetInput2(smoother->GetOutput());
lte->SetInput2(smoother->GetOutput());

typedef itk::ImageFileWriter< IType > WriterType;
WriterType::Pointer writer = WriterType::New();
writer->SetInput( eq->GetOutput() );
writer->SetFileName( argv[2] );
writer->Update();

writer->SetInput( gt->GetOutput() );
writer->SetFileName( argv[3] );
```

```
writer->Update();

writer->SetInput( gte->GetOutput() );
writer->SetFileName( argv[4] );
writer->Update();

writer->SetInput( lt->GetOutput() );
writer->SetFileName( argv[5] );
writer->Update();

writer->SetInput( lte->GetOutput() );
writer->SetFileName( argv[6] );
writer->Update();
```

References

- [1] L. Ibanez and W. Schroeder. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-10-6, <http://www.itk.org/ItkSoftwareGuide.pdf>, 2003.