

---

# Exponentiation filters

Gaëtan Lehmann

November 2, 2010

INRA, UMR 1198; ENVA; CNRS, FRE 2857, Biologie du Développement et Reproduction, Jouy en Josas, F-78350, France.

## Abstract

Two simple filters are contributed to compute the exponential of an image: the first one to raise the image to the power of a constant, the other one to raise the image to the power of the values provided in another image.

Latest version available at the [Insight Journal](http://hdl.handle.net/10380/3227) [ <http://hdl.handle.net/10380/3227> ]  
Distributed under [Creative Commons Attribution License](#)

## Contents

|          |                                               |          |
|----------|-----------------------------------------------|----------|
| <b>1</b> | <b><a href="#">ConstantPowImageFilter</a></b> | <b>1</b> |
| <b>2</b> | <b><a href="#">PowImageFilter</a></b>         | <b>2</b> |
| <b>3</b> | <b><a href="#">Wrapping</a></b>               | <b>3</b> |

---

## 1 ConstantPowImageFilter

ConstantPowImageFilter raises the input image to the power of a constant provided by the user. Here is an example of usage:

```
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"
#include "itkSimpleFilterWatcher.h"

#include "itkConstantPowImageFilter.h"
```

```
int main(int argc, char * argv[])
```

```

{
    if( argc != 4 )
    {
        std::cerr << "usage: " << argv[0] << " input output constant" << std::endl;
        std::cerr << " input: the input image" << std::endl;
        std::cerr << " output: the output image" << std::endl;
        // std::cerr << " : " << std::endl;
        exit(1);
    }

    const int dim = 2;

    typedef unsigned char PType;
    typedef itk::Image< PType, dim > IType;
    typedef itk::Image< unsigned short, dim > OType;

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

    typedef itk::ConstantPowImageFilter< IType, double, OType > FilterType;
    FilterType::Pointer filter = FilterType::New();
    filter->SetInput( reader->GetOutput() );
    filter->SetConstant( atof(argv[3]) );
    itk::SimpleFilterWatcher watcher(filter, "filter");

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

    return 0;
}

```

## 2 PowImageFilter

PowImageFilter raises the first input image to the power of the values of the second image. Here is an example of usage:

```

#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"
#include "itkSimpleFilterWatcher.h"

#include "itkPowImageFilter.h"

int main(int argc, char * argv[])
{
    if( argc != 4 )

```

```

{
    std::cerr << "usage: " << argv[0] << " input pow_input output" << std::endl;
    std::cerr << " input: the input image" << std::endl;
    std::cerr << " output: the output image" << std::endl;
    // std::cerr << " : " << std::endl;
    exit(1);
}

const int dim = 2;

typedef unsigned char PType;
typedef itk::Image< PType, dim > IType;
typedef itk::Image< unsigned short, dim > OType;

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

ReaderType::Pointer reader2 = ReaderType::New();
reader2->SetFileName( argv[2] );

typedef itk::PowImageFilter< IType, IType, OType > FilterType;
FilterType::Pointer filter = FilterType::New();
filter->SetInput( reader->GetOutput() );
filter->SetInput( 1, reader2->GetOutput() );
itk::SimpleFilterWatcher watcher(filter, "filter");

typedef itk::ImageFileWriter< OType > WriterType;
WriterType::Pointer writer = WriterType::New();
writer->SetInput( filter->GetOutput() );
writer->SetFileName( argv[3] );
writer->Update();

return 0;
}

```

### 3 Wrapping

The wrappers for WrapITK for the two new filters are provided in the Wrapping directory.

### 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.