Robust Automatic Threshold Selection

Gaëtan Lehmann

November 17, 2006

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

Abstract

Robust Automatic Threshold Selection (RATS) [1] is a fast and noise robust automatic thresholding method based on gradients in the image. The basic idea is to choose the threshold at the intensity where the gradient are the highest.

1 Description

RATS is a method to automatically compute a threshold without using the histogram of the image. It compute a weighted average of the image intensity, using the gradient of the image, as shown in the following equation:

$$T = \frac{\sum\limits_{p \ in \ D} G(p) \ . \ I(p)}{\sum\limits_{p \ in \ D} G(p)}$$

where I is the input image, and G is the gradient of I. Several methods may be used to compute G. Both I and G are defined on the domain D.

As detailed in [2], using the same method with a pow of the gradient gives better results with noisy images, by giving a higher weight to the high gradients:

$$T = \frac{\sum\limits_{p \ in \ D} G(p)^m \ . \ I(p)}{\sum\limits_{p \ in \ D} G(p)^m}$$

with m greater than 1, typically 2.

2 Implementation

The filter provided in this article use a pow of 1 by default, to fit the historical algorithm. Because there is many ways to compute the gradient, the user is in charge to compute the gradient magnitude image, and to pass it to the filter. A mask can also be used to limit the computation to a given zone in the image.

3 Code example

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

#include "itkRobustAutomaticThresholdImageFilter.h"
#include "itkGradientMagnitudeRecursiveGaussianImageFilter.h"

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

  if( argc != 4 )
    {
    std::cerr << "usage: " << argv[0] << " inputImage outputImage pow" << std::endl;
    exit(1);
    }

  const int dim = 2;

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

  typedef float RPType;
  typedef itk::Image< RPType, dim > RIType;

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

  typedef itk::GradientMagnitudeRecursiveGaussianImageFilter< IType, RIType > GradientType;
  GradientType::Pointer gradient = GradientType::New();
  gradient->SetInput( reader->GetOutput() );
  gradient->SetSigma( 10 );
  gradient->Update();

  typedef itk::RobustAutomaticThresholdImageFilter< IType, RIType > FilterType;
  FilterType::Pointer filter = FilterType::New();
  filter->SetInput( reader->GetOutput() );
  filter->SetGradientImage( gradient->GetOutput() );
  filter->SetPow( atof(argv[3]) );

  itk::SimpleFilterWatcher watcher(filter, "filter");

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

  return 0;
}
```

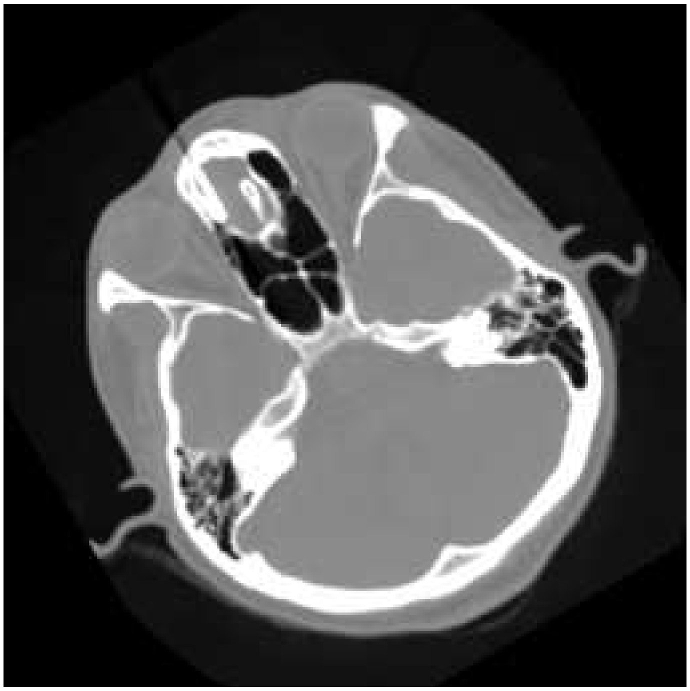

Figure 1: The input image.

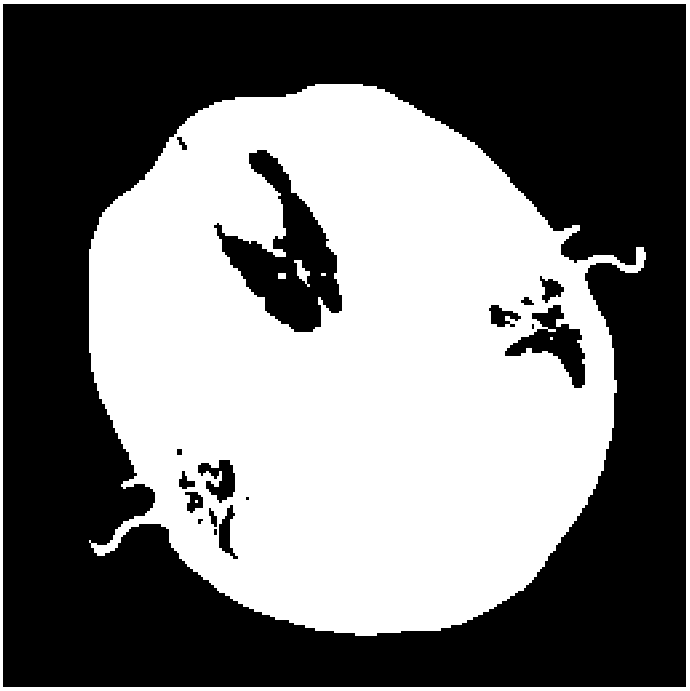

Figure 2: The thresholded image.

4 Note on access to the cited articles

None of the two articles cited are available easily (for free), and the content of this contribution is based on other sources which are using those article. You may found some difference in the algorithm compared to the original ones.

A day will come when there will be nothing else than open access...

References

[1] J. Kitler, J. Illingworth, and J. Foglein. Threshold selection based on a simple image statistic. *Computer vision, graphics, and image processing*, 30:125–147, 1985. (document)

[2] M.H.F. Wilkinson. Optimizing edge detectors for robust automatic threshold selection: coping with edge curvature and noise. *Graph. Models Image Proc.*, 60:385–401, 1998. 1