Kappa Sigma Clipping

Gaëtan Lehmann

November 13, 2006

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

Abstract

When an image is mostly composed of background pixels, most of the automatic thresholding methods are failing to produce a relevant threshold. This is mainly caused because one mode is over represented compared to the other in the histogram of the image. The Kappa-Sigma clipping, a method largely used in astronomy, don't try to separate 2 modes in the histogram, but rather try to find the properties of the only usable mode, the one of the background, and compute a threshold to select the values significantly different of the background.

1 Description

The images mostly formed of background pixels are not so common in biological 2D images. However, the third dimension make this case a lot more common, and it's not rare to have some images produced by a laser scanning confocal microscope which contain more than 90% of background pixels. The histogram of those images often appear to be mono-modal, as the one shown in Figure 1. The classical automatic thresholding methods are trying to find a threshold which maximize the separation of (at least) two modes in the histogram, and so are often performing badely when one mode is largely over represented compared to the other.

Kappa-Sigma clipping use an other approach: it found the mean and sigma of the background, and use this two properties to select the pixels significantly different of the background. Mean and sigma are first computed on the entire image, and a threshold is computed as `mean + f * sigma`. This threshold is then used to select the background, and recompute a new threshold with only pixels in the background. This algorithm shouldn't converge to a value, so the number of iterations must be provided. In general, two iterations are used.

2 Code example

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

#include "itkKappaSigmaThresholdImageFilter.h"
```

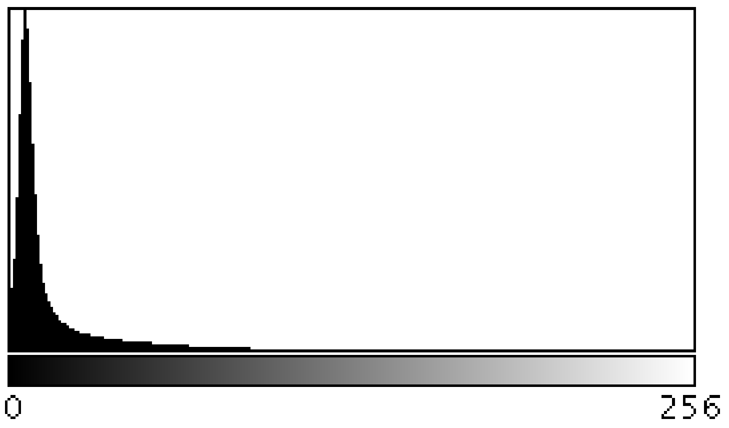

Figure 1: Histogram of an image mainly composed of background pixels.

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

  if( argc != 5 )
    {
    std::cerr << "usage: " << argv[0]
              << " input output sigmaFactor numberOfIterations" << std::endl;
    exit(1);
    }

  const int dim = 3;

  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::KappaSigmaThresholdImageFilter< IType > FilterType;
  FilterType::Pointer filter = FilterType::New();
  filter->SetInput( reader->GetOutput() );
  filter->SetSigmaFactor( atof(argv[3]) );
  filter->SetNumberOfIterations( atoi(argv[4]) );

  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;
}
```

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.