A Threshold with Hysteresis

Release 1.00

Sergio Vera[1], Frederic Pérez[1], Laura Lara[1] and Miguel Gonzalez[1]

December 19, 2011

[1]Alma IT Systems, Barcelona, Spain

Abstract

Thresholding is perhaps one of the most basic techniques in image processing. The Insight Toolkit (ITK `www.itk.org`) includes several threshold filters ranging from simple ones to Otsu based thresholding. Deep into the implementation of the Canny edge detection filter, ITK includes an implementation of hysteresis thresholding, but unfortunately this method can not be used as a generic filter. This paper presents an implementation of the hysteresis thresholding filter that can be used as a standalone ITK filter.

Contents

1 Introduction

In thresholding operations, individual pixels are marked as foreground/object or background pixels following a specific criterion. The simplest form of thresholding simply compares each pixel intensity with a reference (threshold) value. Any pixel above the reference value will be set as foreground. Other methods exist such as histogram based thresholding (Otsu, Isodata) or local thresholding (adaptive thresholding). A good collection of histogram-based thresholds for ITK can be found in [**?**].

Hysteresis thresholding uses a criterion that combines the actual pixel value and the value of neighbouring pixels to decide if a pixel belongs to background or foreground. The criterion used consists in two separated thresholds τ_1 and τ_2, $\tau_1 \leq \tau_2$.

A pixel x in an image I is called *strong* if $I(x) \geq \tau_2$. All pixels of the image below τ_1 are called *weak* pixels. Pixels in between the two thresholds are called *candidate* pixels. All strong pixels will be set to foreground.

All weak pixels will be set to background. All candidate pixels connected to a strong pixels with a n-path of candidate pixels will be set to foreground. A n-path(p_0, p_k) is a sequence $p_0, p_1, \ldots, p_i, \ldots, p_k$ such that p_i is a n-neighbour to p_{i-1} for $1 \leq i \leq k$. The parameter n defines the connectivity between neighbours: 4 or 8 in 2D images and 6, 18 or 27 in 3D volumes. More formally, we can define the Hystheresis threshold of the image I as:

$$\text{HystThr}(I) = \begin{cases} 1 & \text{if } I(x) \geq \tau_2 \\ 1 & \text{if } I(x) \geq \tau_1 \text{ and } \exists \text{ n-path}(x,y) \mid I(y) \geq \tau_2 \text{ and } \forall p \in n\text{-path}(x,y), p \geq \tau_1 \\ 0 & \textit{otherwise} \end{cases}$$

This paper presents a C++ implementation of a filter for ITK [**?**] that implements Hysteresis Thresholding.

2 Implementation

The filter has been implemented as a subclass of the `itk::ImageToImageFilter`. Getter/Setter methods exist to set the threshold values, as well as the values of foreground (`SetInsideValue()`) and background (`SetOutsideValue()`) pixels.

The algorithm uses a two pass strategy over the image. The first pass is responsible to set to foreground value all the strong pixels. Each pixel set to foreground value is stored in a queue so later, its neighbours can be processed in the second pass, where the connection between candidate pixels and strong pixels is checked. This second pass works as a flood fill from the strong pixels to any neighbouring candidate pixels using the default neighbourhood.

3 Usage

The usage is not different from any other filter of the ITK family. By default, the background value is set to zero and foreground value is set to the max value allowed by the pixel type of the input image. The following snippet of code illustrates the usage over a 2D image.

```
typedef itk::Image<unsigned char,2> uc2dImg;
typedef itk::BinaryHysteresisThresoldImageFilter<uc2dImg> HystThr;
HystThr::Pointer thr = HystThr::New();
thr->SetLowerThreshold(threshold1);
thr->SetUpperThreshold(Threshold2);
thr->SetInput(inputImg);
thr->Update();
```

4 Examples

Figure **??** shows examples of hysteresis thresholding on the well known Lena image. The hysteresis properties of this threshold introduces a sense of connected component to the method that allows to obtain a thresholded image with better connected regions, that often represent better objects of the original image.

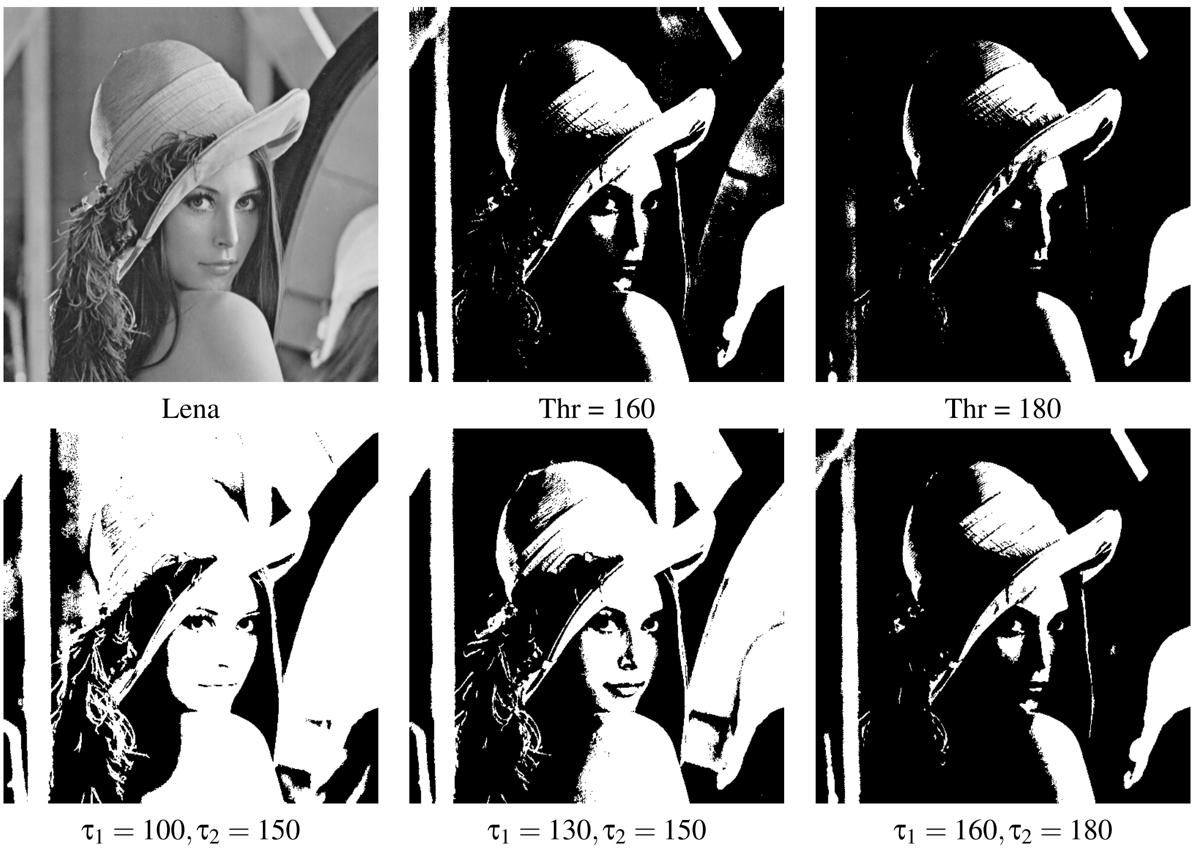

Figure 1: Example of hysteresis thresholding and ordinary thresholded using the classic Lena image. Upper row, original image and standard thresholding. Bottom row, thresholds with hysteresis.

A Acknowledgements

The first author would like to thank the *Generalitat de Catalunya* for the funding with grant 2009-TEM-00007.