# Histogram-based thresholding - some missing methods

Richard Beare, Department of Medicine, Monash University, Melbourne, Australia.

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

## Abstract

Using intensity histograms to estimate image thresholds is a long established practice in image processing and image analysis and a wide variety of techniques have been developed. Different techniques are appropriate for different intensity distributions. This article implements a number of standard techniques not currently available in ITK.

## Introduction

This contribution includes classes for threshold estimation using the

following methods: Huang(Huang and Wang, 1995)

Intermodes and Minimum (Prewitt and Mendelsohn), IsoData (Ridler and Calvard),

Li (Li and Lee, 1993; Li and Tam, 1998), MaxEntropy (Kapur et al.),

KittlerIllingworth (Kittler and Illingworth), Moments (Tsai), Yen (Yen et al.), RenyiEntropy (Kapur et al.), Shanbhag (Shanbhag) and Triangle (Zack et al.).

All classes are largely derived from the AutoThresh (Landini) package for ImageJ. Parts of the brief outline below are taken from the presentation associated with the HistThresh (Niemistö) matlab toolbox, which was also a source of information for the AutoThresh package. The exception is the triangle method, which was written before discovery of the AutoThresh packge and before this project got slightly out of hand.

These classes have been included in ITK 4 and are implented using the histogram framework. Methods for setting number of histogram bins and

Thresholding algorithms

Huang

*itkHuangThresholdImageFilter* implements Huang's fuzzy thresholding using Shannon's entropy function (Huang and Wang). The measure of fuzziness represents the difference between the original image and its binary version. For a given threshold level the fuzzy membership function for a pixel is defined by the absolute difference between the pixel gray level and the average gray level of the region to which it belongs, with a larger difference leading to a smaller membership value. The optimal threshold is the value that minimizes the fuzziness, as defined by Shannon's entropy function, applied to the fuzzy membership functions.

Intermodes

*itkIntermodesThresholdImageFilter* implements the methods described in (Prewitt and Mendelsohn). The histogram is iteratively smoothed until only 2 peaks remain. In one variant the threshold is the midpoint of the two peaks, while in the other it is the minimum point between the peaks. The two variants are selected using the *UseIntermodeOff* method. Not good for histograms with very unequal peaks.

IsoData

*itkIsoDataThresholdImageFilter* implements Ridler and Calvard's (Ridler and Calvard) isodata method. Computes average of voxels below initial threshold and above initial threshold. Threshold is set to the average of the two. Repeat until the threshold is larger than the average of the brightness of the two regions.

Li

*itkLiThresholdImageFilter* implements Li's minimum cross entropy method. This method selects a threshold that minimizes the cross entropy between original and thresholded images.

MaxEntropy

*itkMaxEntropyThresholdImageFilter* implements the method described in (Kapur et al.), which chooses a threshold such that the entropies of distributions above and below threshold are maximised. This is one of several entropy-based approaches.

RenyiEntropy

*itkRenyiEntropyThresholdImageFilter* Similar to MaximumEntropy, but using a different entropy measure (Kapur et al.).

KittlerIllingworth

*itkKittlerIllingworthThresholdImageFilter* implements the minimum error thresholding method (Kittler and Illingworth). A threshold is selected that minimizes the number of misclassifications between the two normal distributions with the given means, variances and proportions. This method assumes a Gaussian mixture model, similar to the Otsu method.

Moments

*itkMomentsThresholdImageFilter* implements Tsai's moment preserving approach (Tsai) that chooses a threshold such that the binary image has the same first three moments as the grey level image.

Yen

*itkYenThresholdImageFilter* implements thresholding based on a maximum correlation criterion (Yen et al.) as a more computationally efficient alternative to entropy measures.

Shanbhag

*itkShanbhagThresholdImageFilter* implements Shanbhag's extension of the Kapur method (Shanbhag) that includes a distance from the threshold in the entropy measure.

Triangle

*itkTriangleThresholdImageFilter* implements a variant of (Zack et al.)

The triangle method constructs a line between the histogram peak and the farthest end of the histogram. The threshold is the point of maximum distance between the line and the histogram. This implementation uses robust (default is 1% and 99%) estimation of histogram ends.

Results

A cropped and smoothed version of the BrainProtonDensitySlice image (Figure 1) is used to demonstrate the results. Smoothing has been applied to avoid the gaps in the histogram and cropping has been applied to remove a peak at 0 caused by padding in the original. This is a slightly challenging image to threshold as there are 3 classes, but the two brighter ones overlap. The mode-based approaches, results in Figure 2, select thresholds between the darkest (background) class and the foreground, thus merging the two foreground classes. Some of the entropy measures, shown in Figure 3, attempt to split the two foreground classes, with varying degrees of success. The thresholds are shown on the histogram in Figure 4.

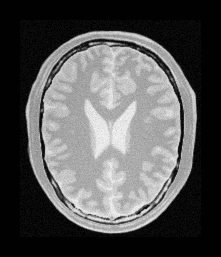


Figure Input image

|  |  |  |
| --- | --- | --- |
| outKittlerIllingworth.png | outIntermodes.png | outMinimum.png |
| Kittler Illingworth | Intermodes | Intermodes (Minimum) |
| outIsoData.png | outMoments.png | outTriangle.png |
| IsoData | Moments | Triangle |

Figure Thresholding results (A)

|  |  |  |
| --- | --- | --- |
| outHuang.png | outLi.png | outMaxEntropy.png |
| Huang | Li | Maximum entropy |
| outRenyiEntropy.png | outYen.png | outShanbhag.png |
| Renyi entropy | Yen | Shanbhag |

Figure Thresholding results (B)

|  |  |
| --- | --- |
| hist_resultsA.pdf | hist_resultsB.pdf |

Figure Image histogram and thresholds selected by various methods

Usage

These classes can be used in a similar way to the well established

*OtsuThresholdImageFilter*} Classes are templated over input and output image types and have methods to set the output *inside* and *outside* pixel values, the number of histogram bins - *SetNumberOfHistogramBins*. Intermodes has a method to select the

intermodes or minimum selection option - *SetUseIntermodes* – and the maximum number of smoothing iterations - *SetMaximumSmoothingIterations*.

All thresholding classes have an associated calculator class that operates on the histogram to estimate the threshold.

Comments, conclusions and further work

Histogram-based approaches to estimating thresholds are very useful, but also can be surprisingly sensitive to changes in image characteristics. Padding images, for example, can easily add a large spike to a histogram that can cause unexpected outputs from many methods. In the example illustrated here, changing the framing of the scene could easily change the significance of the background class leading to very different behaviour. Underlying assumptions built into some methods, such as Gaussian mixture models, can become unrealistic if image acquisition conditions (e.g. lighting) change, and such changes are not necessarily obvious to the naked eye. Always use with care and always attempt to constrain the areas to which such estimation is applied in order to minimize the chances of changes in image characteristics affecting application performance.

This contribution is a set of standard thresholding methods available in other packages. The implementation is derived from the ImageJ java plugin. These classes have been included recently in the ITK 4 distribution via Gerrit and include a variety of tests.

Several of the articles cited describe extensions to the basic methods to support multi-level thresholding. Additional control of the histogramming parameters can also be included. Only the most basic are in the current version. Finally, masked versions of these classes are also likely to be useful. This should be relatively simple to implement using the masked histogram infrastructure. Hopefully these extensions can be implemented in the future.

## References

Huang, L.K., Wang, M.J.J., 1995. Image thresholding by minimizing the measures of fuzziness. Pattern Recognition 28, 41-51.

Kapur, J.N., Sahoo, P.K., Wong, A.K.C., 1985. A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics, and Image Processing 29, 273-285.

Kittler, J., Illingworth, J., 1986. Minimum error thresholding. Pattern Recognition 19, 41-47.

Landini, G.e.a., Auto Threshold.

Li, C.H., Lee, C.K., 1993. Minimum cross entropy thresholding. Pattern Recognition 26, 617-625.

Li, C.H., Tam, P.K.S., 1998. An iterative algorithm for minimum cross entropy thresholding. Pattern recognition letters 19, 771-776.

Niemistö, A., HistThresh toolbox for Matlab.

Prewitt, J., Mendelsohn, M.L., 1965. The Analysis of Cell Images. Annals of the New York Academy of Sciences 128, 1035-1053.

Ridler, T.W., Calvard, S., 1978. Picture thresholding using an iterative selection method. IEEE transactions on Systems, Man and Cybernetics 8, 630-632.

Shanbhag, A.G., 1994. Utilization of information measure as a means of image thresholding. CVGIP: Graphical Models and Image Processing 56, 414-419.

Tsai, W.H., 1985. Moment-preserving thresholding: A new approach. Computer Vision, Graphics, and Image Processing 29, 377-393.

Yen, J.C., Chang, F.J., Chang, S., 1995. A new criterion for automatic multilevel thresholding. Image Processing, IEEE Transactions on 4, 370-378.

Zack, G.W., Rogers, W.E., Latt, S.A., 1977. Automatic measurement of sister chromatid exchange frequency. Journal of Histochemistry & Cytochemistry 25, 741.