Hochschule Niederrhein
University of Applied Sciences

Elektrotechnik
und Informatik
Faculty of Electrical Engineering
and Computer Science

Soft Thresholding for Visual Image Enhancement

Christoph Dalitz

Technischer Bericht Nr. 2014-01

IMPRESSUM

Technische Berichte des Fachbereichs Elektrotechnik und Informatik,
Hochschule Niederrhein

ISSN 2199-031X

HERAUSGEBER

Christoph Dalitz und Steffen Goebbels
Fachbereich Elektrotechnik und Informatik

ANSCHRIFT

Hochschule Niederrhein
Reinarzstr. 49
47805 Krefeld

http://www.hsnr.de/fb03/technische-berichte/

Soft Thresholding for Visual Image Enhancement

Christoph Dalitz
Institut für Mustererkennung
Hochschule Niederrhein
Reinarzstr. 49, 47805 Krefeld
christoph.dalitz@hsnr.de

Abstract

Thresholding converts a greyscale image into a binary image, and is thus often a necessary segmentation step in image processing. For a human viewer however, thresholding usually has a negative impact on the legibility of document images. This report describes a simple method for "smearing out" the threshold and transforming the greyscale image into a different greyscale image. The method is similar to fuzzy thresholding, but is discussed here in the simpler context of greyscale transformations and, unlike fuzzy thresholding, it is independent from the method for finding the threshold. A simple formula is presented for automatically determining the width of the threshold spread. The method can be used, e.g., for enhancing images for the presentation in online facsimile repositories.

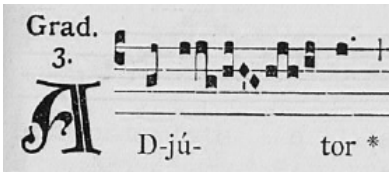
1 Introduction

Thresholding can be considered as a special case of image segmentation: it partitions the image pixels of a greyscale image into foreground (typically "black") and background ("white") pixels, thereby transforming the greyscale image into a binary image. As it is both an essential and a possibly difficult preprocessing step in many image processing systems, in particular for document image recognition, many different thresholding techniques have been proposed in the literature [1] [2]. The thresholding algorithm itself is very simple: let $f(x, y)$ be the grey value of the image at pixel position (x, y); then thresholding with threshold t transforms this image into a binary image $\tilde{f}(x, y)$ as follows:

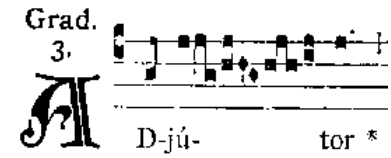
$$\tilde{f}(x,y) = \begin{cases} 1 & \text{if } f(x,y) \le t \\ 0 & \text{if } f(x,y) > t \end{cases} \qquad (1)$$

When the threshold is constant over the entire image, the thresholding is called *global*. When it depends on the position, i.e. $t = t(x, y)$, the thresholding is called *local*. The different thresholding algorithms vary in their rules for determining the threshold $t(x, y)$. An often deployed algorithm for global thresholding is Otsu's method [3].

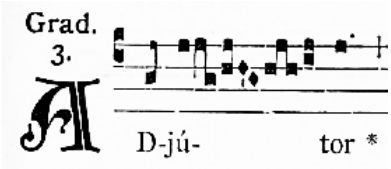
As can be seen in Fig. 1, converting a greyscale image to a binary image has the effect that object borders that look smooth in the greyscale image become ragged in the binary image. This negative effect on the legibility is remedied by replacing the binarization with a greyscale transformation that smears out the transition from black to white around the threshold value ("soft thresholding", see Fig. 1(c)).



(a) Greyscale image



(b) Otsu thresholding



(c) Soft thresholding

Figure 1: A Greyscale image binarized with Otsu's method and transformed with soft thresholding using the same threshold (image detail from "Graduale Romanum", Tournai, 1910).

Soft thresholding has some similarity with fuzzy thresholding [4], which assigns each grey value a "membership value" to foreground or background and sets pixels with a background membership value greater than 0.5 to white and the rest to black. Instead of utilizing the membership value for thresholding, it can be interpreted itself as a grey level, thereby defining a greyscale transformation. Fuzzy thresholding has two parameters, a membership function (typically Zadeh's "S function" [5]) and a band width. Based on these, different criteria can be postulated to be optimized, thereby yielding a threshold value t. Even though there have been proposals for automatically determining the band width [6], these do not work on all images and the band width must therefore in general be chosen manually to suit the grey-level histogram of the image [7].

With fuzzy thresholding, the threshold is thus implicitly determined by the choice of the membership function and its band width and is not an independent variable. With soft thresholding, as presented in this report, the threshold is an independent variable that can be chosen to be optimal according to any other established method and the band width follows automatically from the threshold.

This report is organized as follows: in Sec. 2, the term "greyscale transformation" is explained and appropriate transfer functions for soft thresholding are presented, in Sec. 3, a formula is given for computing the free parameter of these greyscale transformations for a given image, and Sec. 4 gives examples how soft thresholding can be used with local thresholds.

A ready-to-run implementation of soft thresholding, as described in this report, has been implemented by the author within the free software *Gamera*[1], a python library for building document analysis systems [8].

2 Suitable greyscale transformations

A *greyscale transformation* is a point operation on a greyscale image that replaces each pixel value v by a new value $g(v)$ that depends only on the grey value of the pixel and not on its location or its neighborhood. Typical examples for greyscale transformations are contrast stretching and Gamma correction [9]. A

[1] `http://gamera.sf.net/`

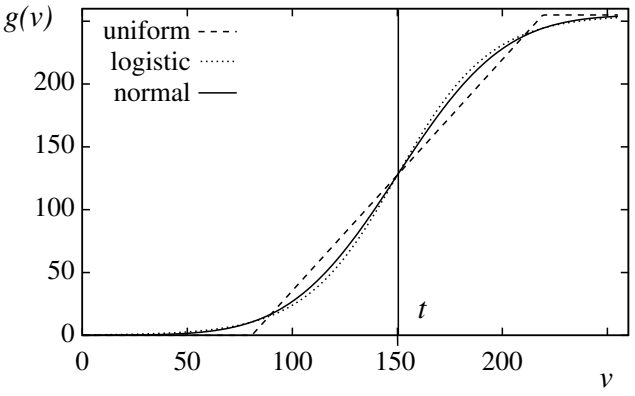


Figure 2: The transfer functions $g(v)$ defined in Eqs. (5-7). To make them comparable, the three functions are normalized to the same variance.

greyscale transformation is completely defined by its *transfer function* $g(v)$. Thresholding with threshold t can also be considered as a greyscale transformation with the transfer function

$$g(v) = \begin{cases} 0 & \text{for } v \le t \\ v_{\max} & \text{for } v > t \end{cases} \tag{2}$$

where v_{max} is the highest grey value (255 for 8bit greyscale images). We can rewrite this as

$$g(v) = v_{max} \cdot F(v-t) \quad \text{with} \tag{3a}$$

$$F(z) = \begin{cases} 0 & \text{for } z \le 0 \\ 1 & \text{for } z > 0 \end{cases} \tag{3b}$$

The ragged edges in Fig. 1(c) are due to the discontinuity of the function $F(z)$ at $z = 0$. For soft thresholding, it is thus a natural generalization of Eq. (3) to replace $F(z)$ by a continuous nondecreasing function with the three properties

$$\lim_{z\to-\infty} F(z) = 0 \tag{4a}$$

$$F(0) = 0.5 \tag{4b}$$

$$\lim_{z\to\infty} F(z) = 1 \tag{4c}$$

In other words, F is the cumulative distribution function $F(z) = P(Z < z)$ of a probability distribution with median zero. The choice of the underlying probability distribution then determines the transfer function, e.g.

Uniform distribution

$$g(v) = v_{max} \cdot \begin{cases} 0 & \text{for } v \le t - \frac{h}{2} \\ \dfrac{v-t}{h} + \dfrac{1}{2} & \text{for } |v-t| < \frac{h}{2} \\ 1 & \text{for } v \ge t + \frac{h}{2} \end{cases} \tag{5}$$

Logistic or Fermi-Dirac distribution

$$g(v) = \frac{v_{max}}{1 + \exp\left(-\dfrac{v-t}{\theta}\right)} \tag{6}$$

Normal distribution

$$g(v) = \frac{v_{max}}{2} \cdot \left(1 - \mathrm{erf}\left(\frac{v-t}{\sqrt{2\sigma^2}}\right)\right) \tag{7}$$

where h is the width of the uniform distribution, θ is the scale parameter ("temperature") of the logistic distribution, σ^2 is the variance of the normal distribution, and *erf* is the error function $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-x^2} dx$.

The parameters h, θ, and σ are all proportional to the square root of the variance of the underlying probability distribution, and can thus be interpreted as a *band width*. As can be seen in Fig. 2, these three transfer functions become somewhat similar when normalized to the same variance σ^2, i.e., with the choices

$$h = \sigma \cdot \sqrt{12} \quad \text{and} \quad \theta = \sigma \cdot \frac{\sqrt{3}}{\pi} \tag{8}$$

The transfer function based upon the normal distribution requires an implementation of the error function *erf*, which might not be available with all math libraries. As the transfer function based on the logistic distribution is quite similar, it can be used as a readily computable replacement.

3 Parameter determination

The threshold t in the transfer functions (5-7) can be determined with any threshold selection algorithm, for example with Otsu's method [3]. The important question still remains how to choose the other parameter σ, θ, or h (note that these three parameters can be related through Eq. (8)) in such a way that the result is visually an enhancement both compared to the original greyscale image and to the binary thresholded image.

Let $H(v)$ be the number of pixels in the image with grey value v. In other words, H is the *grey-level histogram* of the image. As a threshold t segments the pixels into the two classes "black" and "white", we can calculate the mean grey value v_w in the "white" class as

$$v_w = \frac{\sum_{v=t+1}^{v_{max}} v \cdot H(v)}{\sum_{v=t+1}^{v_{max}} H(v)} \tag{9}$$

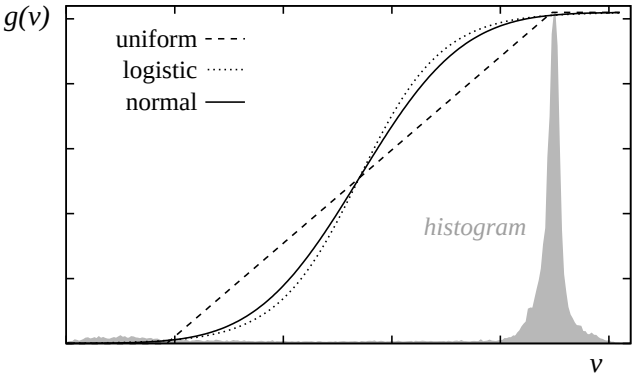


Figure 3: A grey-level histogram (grey) and the transfer function resulting from Eqs. (11-13). Note that the histogram has been scaled to fill out the range $[0, v_{max}]$. The Otsu threshold for this histogram was $t = 135$.

It is reasonable to expect from a "soft thresholding" algorithm that this value visually appears to be white, or is

$$g(v_w) = \alpha \cdot v_{max} \quad \text{with} \quad \alpha = 0.99 \tag{10}$$

Substituting (10) into the three transfer functions (5-7) and doing elementary calculations yields the following formulae for the parameter choice in the transfer functions:

Uniform distribution

$$h = \frac{v_w - t}{\alpha - 0.5} \;\approx\; 2(v_w - t) \tag{11}$$

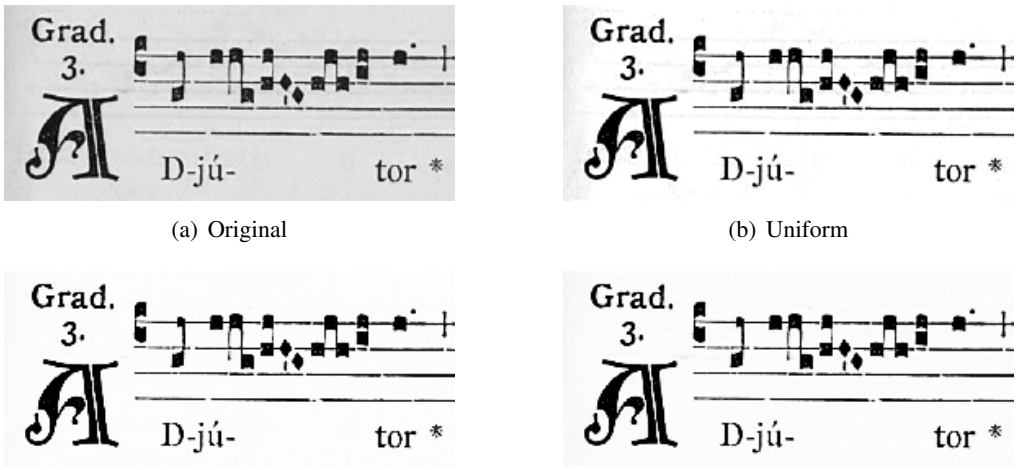
Logistic distribution

$$\theta = -\frac{v_w - t}{\ln(-1 + 1/\alpha)} \tag{12}$$

Normal distribution

$$\sigma = \frac{v_w - t}{z_\alpha} \tag{13}$$

where z_α is the α-quantile of the standard normal distribution, which is $z_\alpha = 2.3263$ for $\alpha = 0.99$. Fig. 3 shows the histogram of the image in Fig. 4(a) and the resulting transfer functions based on the Otsu threshold and Eqs. (11-13). The effect of the different transfer functions can be seen in Fig. 4. The linear transition from black to white of the uniform distribution actually makes the slight shading at the left border more visible and does not suppress the show through from



(a) Original (b) Uniform

(c) Logistic (d) Normal

Figure 4: The effect of the distribution, upon which the transfer function is based, on the result of soft thresholding.

the back of the scanned page. The results for the normal and the logistic distributions are better and both quite similar, with the logistic distribution slightly better with respect to suppressing show through.

The function *soft_threshold* in the Gamera framework therefore uses by default the transfer function based on the logistic distribution, and, when no threshold is provided by the user, Otsu's method is applied.

4 Local thresholding

When the threshold $t = t(x, y)$ is not constant over the entire image, but depends on the pixel position (x, y), it is no longer obvious how the parameters h, θ, or σ are to be determined. Depending on the local thresholding method, their values can be determined as follows:

a) When the local thresholding consists of a greyscale transformation followed by a global thresholding, the method from Sec. 3 can be used.

b) When the thresholding algorithm computes the threshold $t(x, y)$ from the neighborhood of the pixel (x, y), the same neighborhood can be used for computing a local $v_w(x, y)$ according to Eq. (9), which can then be inserted into Eqs. (11-13) to obtain local parameters.

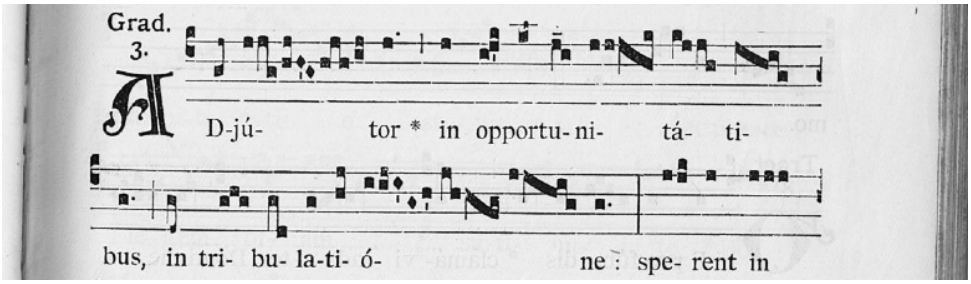
c) Global parameters can be obtained from a v_w that is the mean grey value of all pixels assigned to class "white" by the thresholding algorithm.

An example for a) is the shading subtraction described in [10], which subtracts from each pixel value the maximum value of its $k \times k$ neighborhood and then performs a global thresholding on the resulting image. Note that the size k must be chosen so large that a window always contains background pixels. As this cannot be guessed automatically, the filter size needs to be chosen manually by the user.

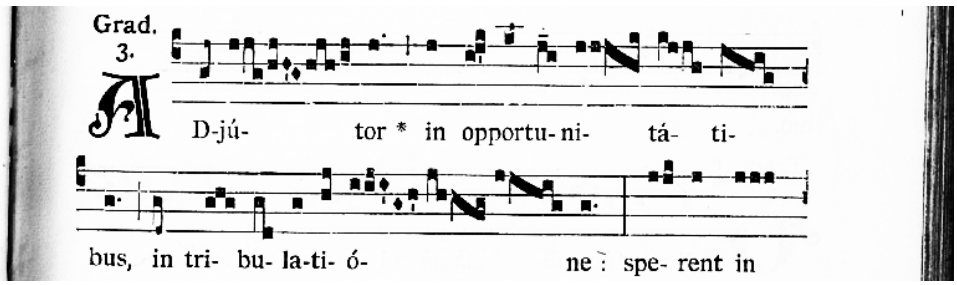
As a binarization method, this local thresholding is implemented in the Gamera function *shading_subtraction* with the use of a fast maximum filter implementation based on Ref. [11]. An adaption of this method for soft thresholding is given in Listing 1, and the result can be seen in Fig. 5. Compared to Fig. 5(b), the shadow from the book binding is absent in Fig. 5(c).

```
# input  = grey image, filter size k
# output = soft thresholded image
def soft_shading_subtraction(image, k)
    shade = image.min_max_filter(k,1)
    shade = shade.to_float()
    imagef = image.to_float()
    diff = imagef.subtract_images(shade)
    diff = diff.to_greyscale()
    return diff.soft_threshold()
```

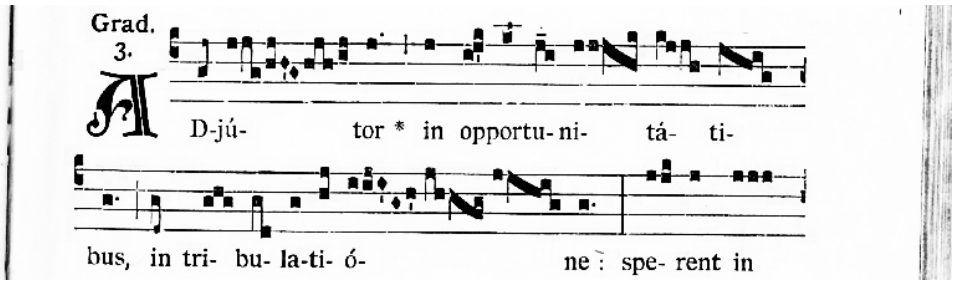
Listing 1: Python implementation for soft thresholding with shading subtraction utilizing the image processing functions provided by the Gamera framework.



(a) Greyscale image



(b) Global soft thresholding



(c) Soft thresholding after shading subtraction

Figure 5: A greyscale image with shading on the left edge globally soft thresholded with Otsu's threshold and after shading subtraction ($k = 17$) according to Listing 1 (image detail from "Graduale Romanum", Tournai, 1910).

5 Conclusions

The soft thresholding algorithm presented in this report is a greyscale transformation that can be used to visually enhance scanned document images. When the scans have regions with varying illumination (shading), as typically occurs with thick books due to the book binding, the combination of soft thresholding with shading subtraction yields decent results.

The author has made a freely available implementation of soft thresholding within the Gamera framework for document analysis and recognition. As this is a python library function, and as it determines the parameters for soft thresholding automatically when no parameters are provided by the user, the method is usable out-of-the-box to automatically process large repositories of online facsimiles. It is also useful as a superior alternative to binarization for preparing images for printed facsimile editions.

References

[1] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, pp. 1277–1294, 1993.

[2] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal on Electronic Imaging*, vol. 13, pp. 146–168, 2004.

[3] N. Otsu, "A threshold selection method from grey-level histograms," *IEEE Transactions on*

Systems, Man, and Cybernetics, vol. 9, pp. 62–66, 1979.

[4] D. Sen and S. Pal, "Histogram thresholding using fuzzy and rough measures of association error," *IEEE Transactions on Image Processing*, vol. 18, pp. 879–888, 2009.

[5] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.

[6] H. Cheng and Y. Lui, "Automatic bandwidth selection of fuzzy membership functions," *Information Sciences*, vol. 103, pp. 1–21, 1997.

[7] C. A. Murthy and S. K. Pal, "Fuzzy thresholding: mathematical framework, bound functions and weighted moving average technique," *Pattern Recognition Letters*, vol. 11, pp. 197–206, 1990.

[8] M. Droettboom, K. MacMillan, and I. Fujinaga, "The Gamera framework for building custom recognition systems," in *Symposium on Document Image Understanding Technologies*, pp. 275–286, 2003.

[9] Y. Shi, J. Yang, and R. Wu, "Reducing illumination based on nonlinear gamma correction," in *IEEE International Conference on Image Processing (ICIP 2007)*, pp. 529–532, 2007.

[10] K. Tönnies, *Grundlagen der Bildverarbeitung*. München: Pearson Studium, 2005.

[11] J. Gil and M. Werman, "Computing 2-d min, median, and max filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 504–507, 1993.