Adaptive Thresholding Using Quadratic Cost Functions

Brian Whalen *ipadapt082@gmail.com*
Individual Contributor
Nesconset, 11767, USA

Abstract

This algorithm seeks to create a thresholding surface (also referred to as an active surface) derived from the initial image topology by means which minimize the image gradient, resulting in a smoothed image which can be used for adaptive thresholding. Unlike approaches which interpolate through points usually associated with high gradient values, each image point is uniquely characterized by a quadratic cost function determined by the gradient at that point along with a constraining potential determined by the image intensity. Minimization is achieved by allowing each point to deviate from its initial value so as to minimize the gradient, as balanced by a constraining potential which seeks to minimize the amount of deviation. The cost function also contains terms which cause the gradient of the thresholding surface to closely parallel those of the image in regions of near uniform intensity (where the absolute values of the gradients are small). This is done to reduce the effects of ghosting (or false segmenting) when thresholding, an important feature of this approach. Image binarization is achieved by comparing the values of the original image points with those of the thresholding surface; values above a given threshold are considered part of the foreground, while those below are considered part of the background.

Keywords: Image Segmentation, Adaptive Threshold, Active Surface, Gradient Minimization, Thresholding Surface.

1. INTRODUCTION

Image segmentation, the process by which an image is partitioned into regions deemed homogeneous based on some set of criteria (usually intensity), can be a challenging problem, particularly when lighting conditions are not optimal or are changing over time. These situations, where lighting cannot be completely controlled, arise in many machine vision applications, such as systems which need to operator outdoors for example, where ambient lighting is a factor and where the field of view is large enough to make even illumination difficult or impossible. In many of these instances a single threshold cannot be found which will adequately segment the image because of the changes in illumination over the scene, which can vary significantly over time, so attempting to remove these variations by modeling meets with limited success. This necessitates the need to threshold locally using adaptive thresholding methods. The most common methods for adaptive thresholding usually compare the intensity of an image point with that of the local mean or median calculated from a window of it' s neighboring points. These techniques work well when most of the pixels belong to the background, and the items of interest are small, such as printed text obscured by a shadow on some portion of the page. But this technique has difficult identifying large regions which cannot be completely covered by such a window because of their size. Alternatively, methods based on active surface modeling can be used to create a smoothed interpretation of the initial image whose intensities, when compared to those of the original image, can be used for thresholding in a manner similar to that of other adaptive thresholding methods. The method proposed herein was inspired by this idea, and proposes another active surface based model for adaptive thresholding.

2. THRESHOLDING APPROACHES

In summary, image segmentation partitions an image into regions, uniform with respect to some property of the image, such as intensity or texture, which has been determined as appropriate for separating the objects of interest from those that are not. This partitioning occurs by determining one or more thresholds for the property in question, calculated either globally or locally and possibly in conjugation with other properties (such as the image gradient) which can then be used to separate the image into regions of interest. A good review of image segmenting techniques is provided by A. M. Khan[1]. Situations where only a single threshold is sought for segmentation is referred to as binarization. Here pixels are either assigned to the foreground, meaning regions of interest, or to the background, indicating areas of the image which are of no interest. Being able to simplify a gray scale image in this manner often simplifies the task of understanding the image by subsequent processes, whether they be algorithmic or neural network in nature. This is particularly true in the field of medical imaging, and in the area of document image analysis, as discussed by Nagy[21].

2.1 Global Thresholding

Global thresholding indicates only a single threshold is used for segmenting the entire image, the thresholding value is not a function of position. When the pixel property of interest is intensity, the global threshold is usually obtained by examining the image's intensity histogram. For some images, particularly those with high contrast between the fore ground and the back ground (such as print on page), the clear bi-modal nature of the histogram simplifies the threshold selection. When this is not the case, automatic thresholding techniques such as Otsu's method [25] can be applied. Most automatic thresholding techniques examine the grayscale histogram in an attempt to identify it's two major modes or classes. In Otsu's method, the threshold is chosen such that the variance within each of the two classes is minimized. In another approach, proposed by Arifin, intra class variance is used to merge similar classes, a process which continues iteratively until only two classes remain[3]. Global thresholding techniques can also be applied over small regions, usually created from a rectangular partitioning of the image, from which local thresholds can be derived, as in the method of Chow and Kaneko, to be discussed later. In addition to segmentation based on thresholds, probabilistic, iterative-relaxation classification methods, which eliminate the need for threshold selection, have been proposed, see Bonnet[20]. For more complex images, banded thresholding or multi-thresholding techniques might be appropriate. A good discussion of such methods is provided by Sonka, HLavac , and Boyle[19] and by Sezgin and Sankur[18]. A comprehensive review of thresholding methods, including edge detection, fuzzy sets, and clustering techniques is provided by Pal[23]. Instead of indentifying thresholds for segmentation, regions can be created based on local uniformity using region growing techniques as well, as discussed by A. M. Khan[2], or by identifying distinct regions, initially determined by the zero crossing of the Laplacian, followed by an iterative smoothing and statistical means testing to determine stable regions from which thresholds can be extracted, see Tabbone[31].

2.2 Local Thresholding

For images were a single threshold is not sufficient local (or adaptive) thresholding techniques can be applied. With these methods thresholding becomes a function of position. Such methods usually use the mean, standard deviation, medium, or intensity range, calculated from the intensities within a window about each pixel, to create a threshold value. Any pixel whose intensity exceeds by some amount that of the function performed over the window are classified as part of fore ground, while all others are considered part of the back ground. Several common methods, listed below, will be considered when evaluating the performance of the proposed method:

Niblack's method - Here the threshold is determined from the mean μ and standard deviation σ computed within a window centered about the pixel to be threshold. If the center pixel's value is greater than ($\mu + k\sigma$), where k is user defined, the pixel is considered part of the foreground, otherwise it's considered part of the background.

Bernsen's method - Here the threshold is determined as the average of the minimum and maximum values found within a window centered about the pixel to be threshold, however if the contrast, defined as the difference between the maximum and minimum, is less than some value, the region is assumed uniform, and the pixel is assigned to either the foreground or background based on it's intensity value, see Bernsen[5].

Mean or C-Mean method - Here the threshold is determined as the mean of the gray scale values within the window. Pixels are assigned to the foreground if they are greater than the mean by a constant amount "C".

MidGray method - Here the threshold is determined as the average of the minimum and maximum values found within a window centered about the pixel to be threshold. Pixels are assigned to the foreground if they are greater than the mid-gray by a constant amount "C".

Sauvola's method - This approach partitions an image into equally sized rectangular regions and evaluates each partition for uniformity, based on this uniformity metric an approach is selected. In non-uniform regions a threshold is calculated using a criteria similar to Niblack's method. If the center pixel's value is greater than the threshold given by the formula $T = \mu\left(1 + k\left(\frac{\sigma}{R} - 1\right)\right)$, where k is user defined and R=128, the pixel is considered part of the foreground, otherwise it's considered part of the background, see Sauvola[13].

Phansalkar's method - Similar in approach to Sauvola's method only with an additional term based on the mean to help handle images with low contrast, the threshold value is determined by the formula $T = \mu\left(1 + pe^{-q\mu} + k\left(\frac{\sigma}{R} - 1\right)\right)$, with p and q typically set to 2 an 10 respectively, see Phansalkar[22].

A description for each of the algorithms applied can also be found in the ImageJ documentation web page[15]. Descriptions for Niblack's method and Brensen's method can be found in Blayvas[11], along with detailed descriptions found in Singh[32].

2.3 Edges and Contours

The creation of contours using edge detectors is a common segmentation method based not on intensity, but rather the gradient of the image. With such approaches, changes in intensity mark the boundaries between different regions. These boundaries are usually created by initially thresholding the image gradient, followed by non-maximal edge suppression and hysteresis to join separate edges into complete contours. A discussion of one such approach, Canny edge detection, is provided by Justin Liang[17]. One difficulty with edge based contouring is that many of the contours are not closed, and segmentation is not complete. This occurs because the gradient strength around the boundary of an object can vary considerably, perhaps from variations in lighting or texture which may be present, obscuring the boundary. To address this algorithms which use both the gradient, a local measure of intensity change, along with texture have been proposed by David Martin[9].

2.4 Active Contours

Active contours perform segmentation by means of a deformable contour whose final shape depends on the image and the energy function to be minimized. The process begins with an initial contour whose shape is successively changed over multiple iterations until a local minimum is found for the energy function. Forces for evolving the contour are usually determined from either the gradient information, or information based on image intensity. One of the motivations for using such approaches is to guarantee a closed boundary for the segmented region, unlike the boundaries generated from edge information alone, which often have discontinuities. The use of level set functions to generate active contours is discussed by Chan and Vese[33]. In this approach contour formation evolves over "time" from some initial level set until the energy functional is minimized, which is accomplished by setting up and solving the appropriate Euler-Lagrange partial differential equation.

2.5 Thresholding or Active Surfaces

Active surface modeling, much like dynamic thresholding, creates a smoothed interpretation of the initial image which, when compared to the original image, can be used for binary thresholding. These algorithms typically use gradient information to identify regions where the image is changing, and the gray values associated with these edges are used to define points (sometimes referred to as supporting points) through which the active surface interpolates. Discussions on the construction of such active surfaces and how they are used for adaptive thresholding are provided by Yanowitz and Bruckstien[30], I. Blayvas et al., where optimizations to the method of Yanowitz and Bruckstien are discussed [11] and by Fei Liu[10]. Once constructed, segmentation is performed by comparing the value of the active surface with that of the original image, points for which the values of the active surface are less than those of the original image are considered foreground points, and those for which the values of the active surface are greater than those of the original image are considered background points. In addition to problems with constructing interpolating surfaces (such as choosing the interpolating points, and functions to use for interpolation) problems also arise when the active surface intersects the image topology away from the interpolating points. Ideally the active surface should only intersect the image topology at the interpolating points, everywhere else the active surface is strictly greater than or strictly less than the image topology. This phenomena, sometimes referred to as "ghosting", results in a false segmentation of the image. The approach detailed by Fei Liu[10], for example, address this problem by using a repulsive force which moves the active surface away from the initial image everywhere, with the stable solution ideally interpolating between points where the gradient is highest. In addition to these approaches, the method of Chow and Kaneko may be applied. This method creates a set of thresholds from an overlapping rectangular partitioning of the image by applying Otsu's method (or some other means for automatic threshold detection) to each region. Interpolation is performed over the set of thresholds to obtain a local threshold at each point in the image[7].

2.6 Proposed Approach

The proposed algorithm creates an active surface, derived from the initial image topology, for the purposes of binary segmentation. As such, it can be grouped with local thresholding methods, or any method which creates as part of it's segmentation process a threshold map from which local thresholding values are derived. As such the proposed method does not seek to find a single threshold or set of thresholds to be used to segment the image globally. The proposed approach belongs with active surface methods. These methods regard the image as a topological map, with gray value expressed as height, and then seek to minimize an energy functional, balancing the minimization of the internal energy of the surface (usually expressed as the square of the gradient) with the effects of a restorative potential which seeks to limits this minimization.

In the proposed approach, each image point is characterized by a simple quadratic equation, derived from the image gradient in combination with a constraining potential, derived from the image intensity. In addition to these two terms, to prevent the active surface from intersecting the image topology in regions of uniformity (or ghosting, as described above) a term is introduced which forces the gradient of the active surface to follow that of the image topology in these regions. The effect of this term is weighted by the gradient of the image so that in regions where the gradient is large no contour following occurs, but it's influence increases to become the predominate term in uniform regions. These terms together create a set of quadratic equations which are simultaneously optimized, with the solution set being the desired active surface. Binary segmentation is achieved, as described above, by comparing the intensities of the original image points with those of the active surface; points above the surface are considered part of the foreground, while those below the surface are considered part of the background.

The proposed approach differs from the previously described methods in a number of ways:

1. Unlike many local thresholding methods, the image is not processed using a pre-determined window of fixed size.

2. Unlike the method of Chow and Kaneko, or that of Yanowitz and Bruckstien, the thresholding surface is not determined by interpolating over a set of thresholds.
3. Unlike active contour or level set method, there is no initial contour which must be described, and no knowledge of how long the final contour will be is needed in order to guide a solution.
4. No differential or non-linear equations are involved in determining the thresholding surface, the solution is determined by a set of linear equations, for which there is an exact solution.

With this approach the solution does not evolve over "time", like active contours or like the active surface proposed by Fei Liu, but is solved for directly and uniquely by linear means. Unlike the active surface approach of Yanowitz and Bruckstien, no interpolating surface is derived, nor is it necessary to find edges explicitly or define a set of supporting points and functions for interpolating between them. There is also no need to define the behavior of the thresholding surface at the boundary of the area of interest. And, unlike local adaptive thresholding methods, there's no need to define a window based on the feature size of interest to perform image smoothing.

To create a set of equations implementing the above described approach, it is necessary to start by connecting neighboring points, then the square of the gradient between these points is calculated and it's sum computed. Next, a term is added which seeks to keep the gradient parallel to itself. These terms are both a function of the gradient, and are weighted in such a way that the first is weighted significantly higher than the second in the regions where the gradient is highest, and the second term weighted significantly higher than the first in the regions where the gradient is lowest. For simplicity, these weights are computed once and are based only on the gradient of the original image. Finally, a constraining potential which increases as points move away from their initial position is introduced. With these terms, during optimization the gradients which are initially high will be reduced, those which are initially low will remain approximately the same, and the amount each point moves will be limited by the constraining potential. Since only height is allowed to vary, the equation to be optimized can be written as a function of height only:

(1) $P(z) = \sum_i \left(h_i(z_i) + \sum_k \left(w_{1i,k}\left(m_{i,k}(z_i) - m_{0i,k}\right)^2 + w_{2i,k} m_{i,k}^2(z_i)\right)\right)$

Where $h_i(z_i)$ is the potential constraining the motion of z_i during optimization, $m_{i,k}(z_i)$ is the gradient between the ith point and it's kth connecting point, with $m_{0i,k}$ being it's initial value, and $w_{1i,k}$ and $w_{2i,k}$ are the weights associated with the two gradient dependent terms. These weights are strictly a function of $m_{0i,k}$, any weights which can enforce the behavior discussed above can be used but this would typically be some monotonically decreasing function such as:

$w_{1i,k} = w_{max} e^{-\rho|m_{0i,k}|}$ and correspondingly $w_{2i,k} = 1 - \left(w_{1i,k}/w_{max}\right)$ with $w_{max} > 1$

The gradient terms are a function of z_i only, and can be expressed as follows:

(2) $m_{i,k}(z_i) = \Delta z_{i,k}/r_{i,k}$ where $r_{i,k} = \left(\Delta x_{i,k}^2 + \Delta y_{i,k}^2\right)^{1/2}$

and $\Delta z_{i,k} = z_i - z_k$, $\Delta y_{i,k} = y_i - y_k$, and $\Delta x_{i,k} = x_i - x_k$

Many different forms can be used for the constraining potential $h_i(z_i)$, the only requirement being that as z_i moves away from it's initial value of z_{0i} the value of $h_i(z_i)$ increases monotonically, thus acting as a restoring force balancing the drive to minimize the remaining terms. Practically, a simple quadratic potential works best:

(3) $h_i(z_i) = \alpha_i(z_i - z_{0i})^2$ where $\alpha_i \geq 0$

Substituting for $h_i(z_i)$ and for $w_{2i,k}$, since it is a function of $w_{1i,k}$, yields:

(4) $P(z) =$

$$\sum_{i} \left(\alpha_i (z_i - z_{0i})^2 + \sum_{k} \left(w_{1i,k}\left(m_{i,k}(z_i) - m_{0i,k}\right)^2 + \left(1 - \left(w_{1i,k}/w_{max}\right)\right) m_{i,k}^2(z_i) \right) \right)$$

In order to find the values of z_i which optimize (1) it is necessary to solve the set of equations obtained from:

(5) $0 = \partial P(z)/\partial z_i$

This yields a set of linear equation with the form:

(6) $0 = \alpha_i(z_i - z_{0i}) + \sum_k \left(w_{1i,k}\left(m_{i,k}(z_i) - m_{0i,k}\right) + \left(1 - \left(w_{1i,k}/w_{max}\right)\right) m_{i,k}(z_i) \right)$

Grouping like terms yields:

(7) $\alpha_i z_{0i} + \sum_k w_{1i,k}\, m_{0i,k} = \alpha_i z_i + \sum_k \left((1 + M w_{1i,k}) m_{i,k}(z_i) \right)$ where $M = 1 - (1/w_{max})$

Expressing (7) in terms of z_i yields:

(8) $z_{0i}\left(\alpha_i + \sum_k w_{1i,k}\right) - \sum_k w_{1i,k}\, z_{0k} = z_i\left(\alpha_i + \mathrm{N_i} + \sum_k M w_{1i,k}\right) - \sum_k z_k\left(1 + M w_{1i,k}\right)$

Where $\mathrm{N_i}$ is the total number of points connected to the ith point. With (8) every point is now represented by a simple linear equation, with the solution set to these equations representing the desired thresholding surface.

2.7 Implementation

To solve for the set of z_i described by (8), linear solutions methods are directly applicable as the problem can be cast in the familiar form of $b = Ax$. Since each point will be expressed by an equation, and each equation only contains terms from the connecting points, the matrix A will be large and sparse so techniques best suited for this should be applied[29]. Iterative techniques such as the conjugate-gradient method [4] provide the means for solving such systems with greater speed than non-iterative techniques, and they were employed in the implementation of this strategy.

Next it is necessary to define the values for α_i. One solution would be to set $\alpha_i = c$ for all α_i where c is any constant such that $c \geq 0$. But the values for α_i need not all be the same; another solution would be to chose the values for α_i to preserve local uniformity, by which is meant any region where the curvature is sufficiently small and intensity is not varying rapidly. One way to achieve this would be to allow the individual points greater or lesser freedom to move based on local uniformity; points from uniform regions will be allowed more freedom to move than points from less uniform regions making it easier for the optimization to parallel the transformed regions. This can be accomplished simply by adjusting the values for α_i on a point by point basis, with points from uniform regions having smaller values for α_i than those from less uniform regions. Using the local curvature γ_l (as expressed by the Laplacian, for example) as a measure of uniformity, α_i can be calculated as follows:

(9) $\text{If } |\gamma_l| - \gamma_{min} \leq 0 \text{ then } \alpha_i = 0$ else

$$\alpha_i = \alpha(|\gamma_l| - \gamma_{min})/(\beta + |\gamma_l| - \gamma_{min})$$

$$\text{with } \alpha, \beta, \text{and } \gamma_{min} > 0$$

So when $|\gamma_l| \leq \gamma_{min}$where $\gamma_{min}$is the minimum curvature of a locally uniform region, then $\alpha_i = 0$, meaning no constraints are applied to z_i during optimization. The value for γ_{min} can be chosen by using a priori knowledge about the image, or automatically by creating a histogram of all values for $|\gamma_l|$ and deciding γ_{min} based on some percentage of the histogram values. The parameter β is chosen to determine the asymptotic behavior of α_i with respect to α. Once again, this can be obtained from knowledge about the image, but results are not very sensitive to values of β; for all test cases $\beta = 5$. Similarly the values for w_{max} and ρ are chosen to determine the amount of contour following. Large values for w_{max} cause terms associated with contour following to predominate the solution, while large values for ρ causes the effects of these terms to be quickly minimized. The effect of optimizing with this approach is to create a smoothed image, as will be evident from the examples provided below.

3. IMAGE SEGMENTATION

Once a smoothed image has been obtained through optimization, image segmentation can be performed by comparing the intensities of the original image with those of the smoothed image. The points $\{p_i\}$ which define the original image can be divided into background image points, I_b, and foreground image points, I_f, with the following formula:

(10) $p_i \in I_f \leftrightarrow (z_{fi} - z_{0i}) < \delta$ where δ can assume any value, although 0 would be typical, and z_{0i} and z_{fi} are the initial and final values for z_i.

As (10) implies, image points whose smoothed intensity is less than it's initial intensity by the amount δ shall be considered foreground image points, all other points belong to the background. Examples of applying this algorithm are provided below.

3.1 Experimental Results

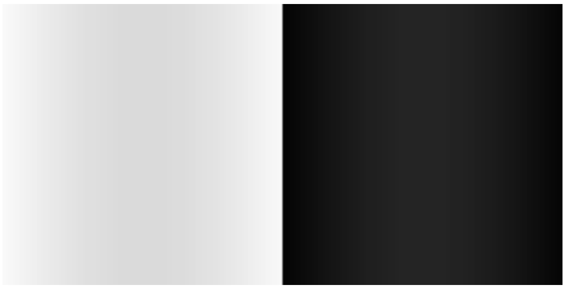
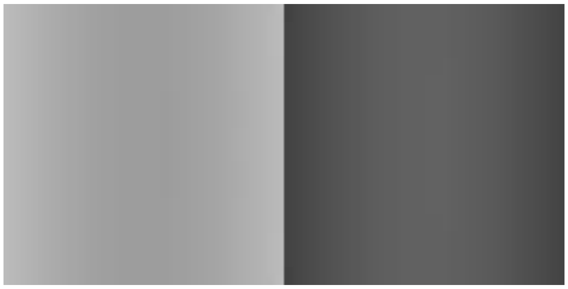
For the weight function $w_{1i,k}$, experimenting with different values of w_{max} showed that $w_{max} = 100$ works best, providing a good compromise between contour following and computational intensity, and this value is used for all examples. Also, ρ = 1.0 and $w_{1i,k} = w_{max}$ for all $|m_{0i,k}| \leq 1.0$ was used for all cases.

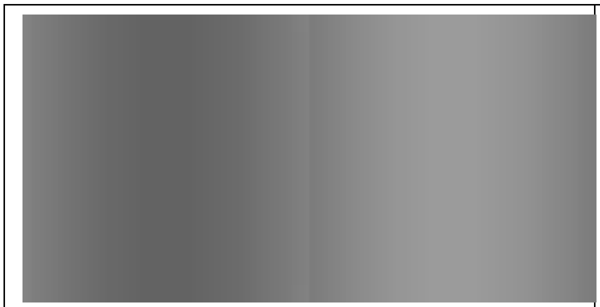
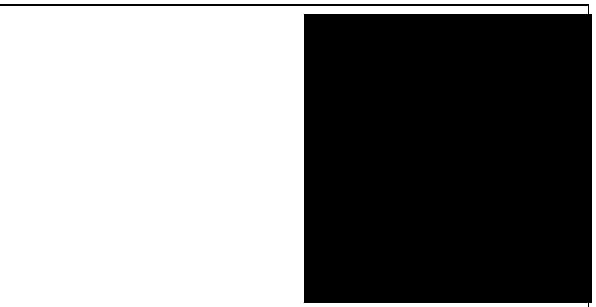
3.2 Synthetic Example

A synthetic example will help to illustrate how the algorithm works. Consider the discontinuous curve defined by the function:

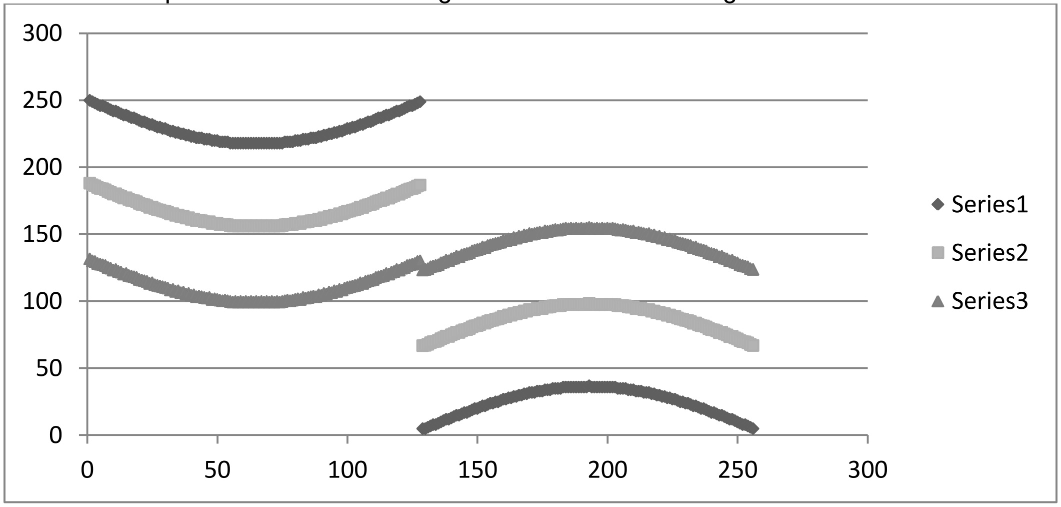
$$f(\varphi) = 205 + 32 \sin \varphi \ for - \pi < \varphi < 0 \ and \ f(\varphi) = 5 + 32 \sin \varphi \ for \ 0 \leq \varphi < \pi$$

For the purposes of testing, a 128x256 image was created with $\varphi = \pi(x - 128)/128$ and with $f(\varphi)$ mapped to grayscale values between 0 and 255. The original image and the results of optimizing with two different values for α and γ_{min} are shown:

 Original Test Image	 Smoothed Image with $\alpha = 4$, $\gamma_{min} = 3$

 Smoothed Image with $\alpha = 1/9$, $\gamma_{min} = 10$	 Segmented Image with $\delta = 0$

A cross-sectional plot of data taken through the middle of the images is shown below:



The data from "Series1" is from the original image, the data from "Series2" is from the optimized image with $\alpha = 4$, $\gamma_{min} = 3$ and the data from "Series3" is from the optimized image with $\alpha = 1/9$, $\gamma_{min} = 10$. Note that the smoothed data, except for being shifted laterally, follows the contour defined by the original data. As (9) suggests, decreasing values of α and increasing values of γ_{min} allow the data to shift further from it' s original position during optimization. In the above example, the largest slope values occur at x=128, where $f(\varphi)$ is discontinuous. From the above graph, it can be seen that optimization causes these slope values to be reduced, causing the two continuous regions to move towards each other, while the slopes within these continuous regions remain parallel to themselves.

3.3 Image Metrics

This section discusses the metrics to be used to determine how well the proposed algorithm performs at segmenting a series of images. For some of the images considered a quantitative analysis, obtained by comparing various algorithms, including the proposed approach, against a ground truth segmentation, will be provided. To evaluate the results of the various segmentation algorithms applied two different segmentation metrics are calculated. The first metric is "Intersection over Union" or "IoU" and is described by Jeremy Jordan[14]. The IoU metric is defined as the ratio of the intersection of the results of segmentation with the ground truth segmentation over the union of the results of segmentation with the ground truth segmentation:

$$IoU = (S_a \cap S_{GT})/(S_a \cup S_{GT})$$

Where S_a is the segmentation produced by the algorithm and S_{GT} is the ground truth segmentation. The next set of metrics considered relate to pixel accuracy. With these approaches an analysis is performed pixel by pixel to determine if it's been classified correctly with respect to the ground truth image. Explanation for the various methods discussed below are given by Paul

Rosin[27]. Each pixel receives one of four classifications: True positive (TP), were the segmented pixel is properly identified as belonging to the object (as defined by the ground truth image); True negative (TN), were the segmented pixel is properly identified as belonging to the background; False positive (FP), were the segmented pixel is improperly identified as belonging to the object, instead of the background; False negative (FN), were the segmented pixel is improperly identified as belonging to the background, instead of the object. Pixel accuracy (PA) is defined as the ratio of the true positives and the true negatives to the total number of pixels (the previous sum plus false positives and false negatives):

$PA = (TP + TN)/(TP + TN + FP + FN)$,

Jaccard coefficient = $TP/(TP + FP + FN)$, and

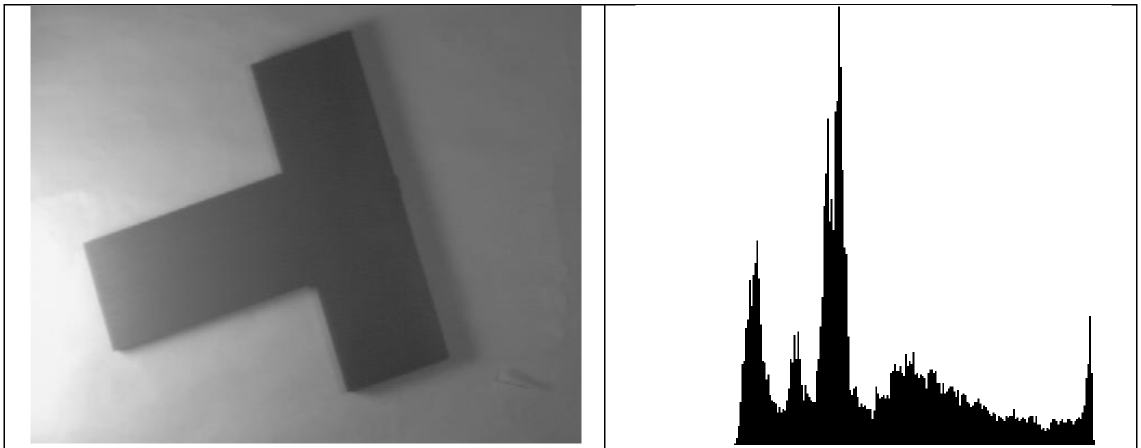
Yule coefficient = $|(TP/(TP + FP)) + (TN/(TN + FN)) - 1|$

With these metrics, it's not necessary to identify separate distinct objects, all pixels identified as belonging to the foreground are considered as one object, even if they are not joined. As a result, many of the segmentation approaches create multiple disjoint sets of foreground pixels, some of which belong to the object, others are artifacts of improper segmentation, which together create the one foreground object. These statistics will be reported as a percentage, with 100% a perfect match to the ground truth image.

3.4 Image Test Cases

This section contains the results of applying the proposed approach to a series of images. The images were chosen to high light different challenges associated with image segmentation and to demonstrated that the proposed approach has broad application. A qualitative analysis against ground truth sets is provided for four of the test images. For the images in figures 1 and 4 the ground truth sets were created by the author from the gradient information and hand editing to remove spurious artifacts, because none were available for these images. For figures 12 and 13, ground truth sets are provided along with the images. The proposed method was compared against methods readily available from free, open source applications such as OpenCV and ImageJ.

The image shown in Figure 1 below represents the kind of challenge this algorithm was meant to address, it's a large "T" shaped object which was been obliquely illuminated, making the selection of a single threshold for segmentation impossible:



Obliquely illuminated "T"[28]

Image Histogram

FIGURE 1

From the histogram for the image in Figure 1, it can be seen that the distribution is not bimodal in nature, illustrating the difficulty with choosing a single global threshold for performing segmentation.

To illustrate this, the results of a fixed thresholding using Otsu's method, as detailed in the ImageJ documentation for global auto thresholding[16], are shown below in Figure 3:

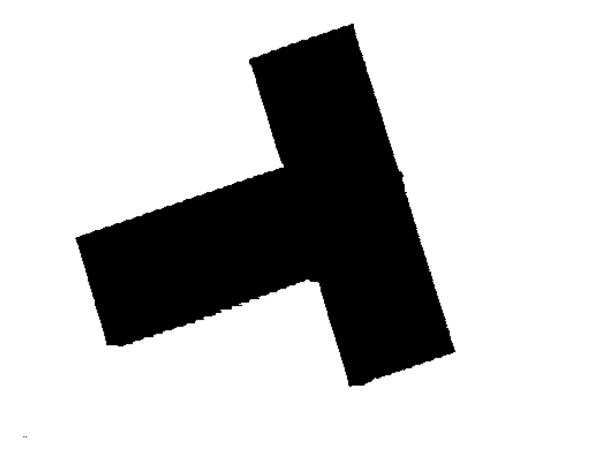


FIGURE 2: Ground Truth Image.

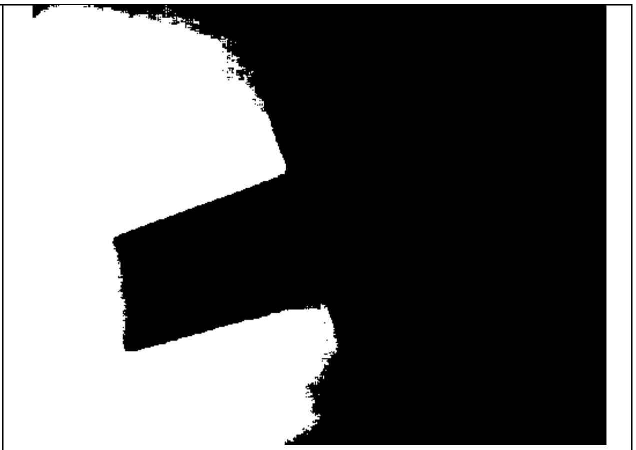
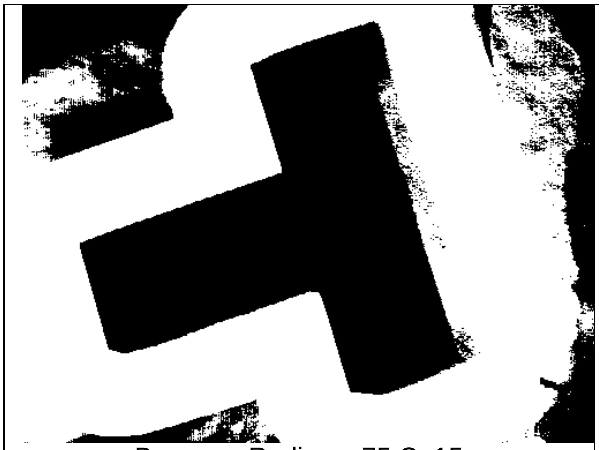


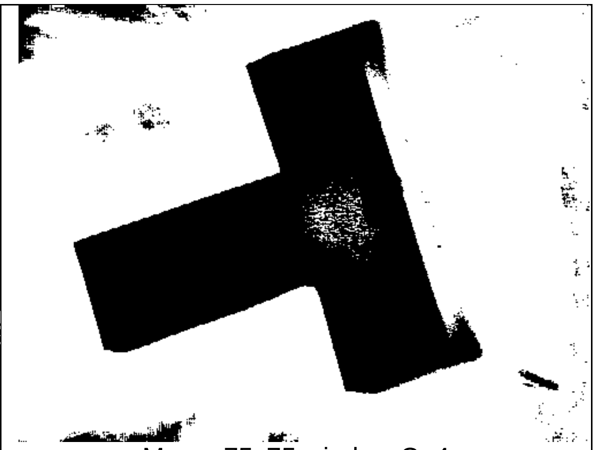
FIGURE 3: Otsu's method.

Because a single threshold will not adequately segment this image, local or adaptive thresholding techniques must be applied. For such a large object, adaptive thresholding using a neighborhood operator, even a large one, sometimes results in the interior of object missing, or unwanted artifacts are introduces.

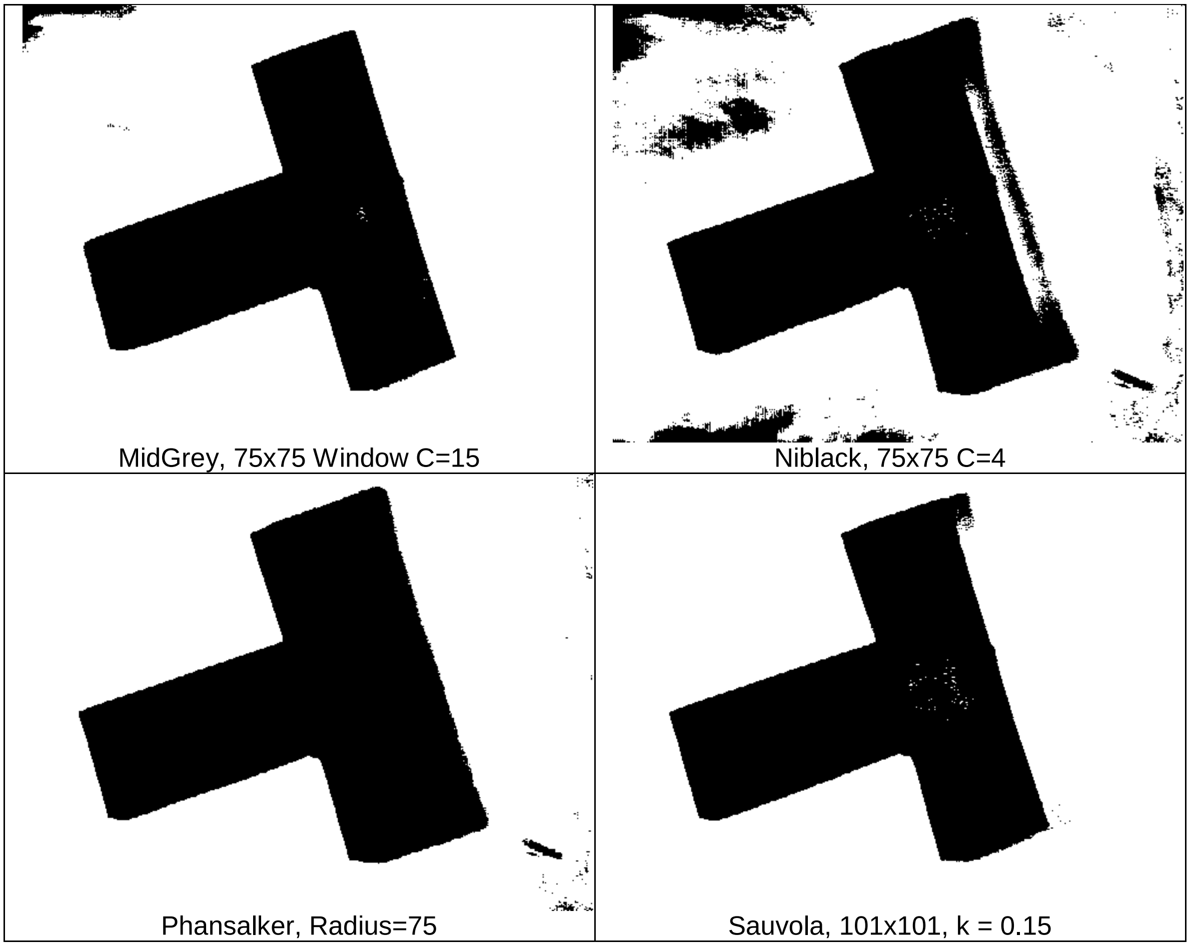
The results of several different adaptive thresholding methods, indicated beneath each picture, are presented below in Figure 4. All of the approaches illustrated below are based on applying statistical methods over a region local to each point in the image. Typically the region is either square or circular in shape, with it' s size being determined by the dimensions of the objects to be identified. As a result, a priori knowledge of the image is required for proper segmentation. Because the object here to be identified (the black "T") is quite large, a large window is required for detection, otherwise only the border of the object will be found and the interior will be missing.



Bernsen, Radius = 75 C=15



Mean, 75x75 window C=4



MidGrey, 75x75 Window C=15

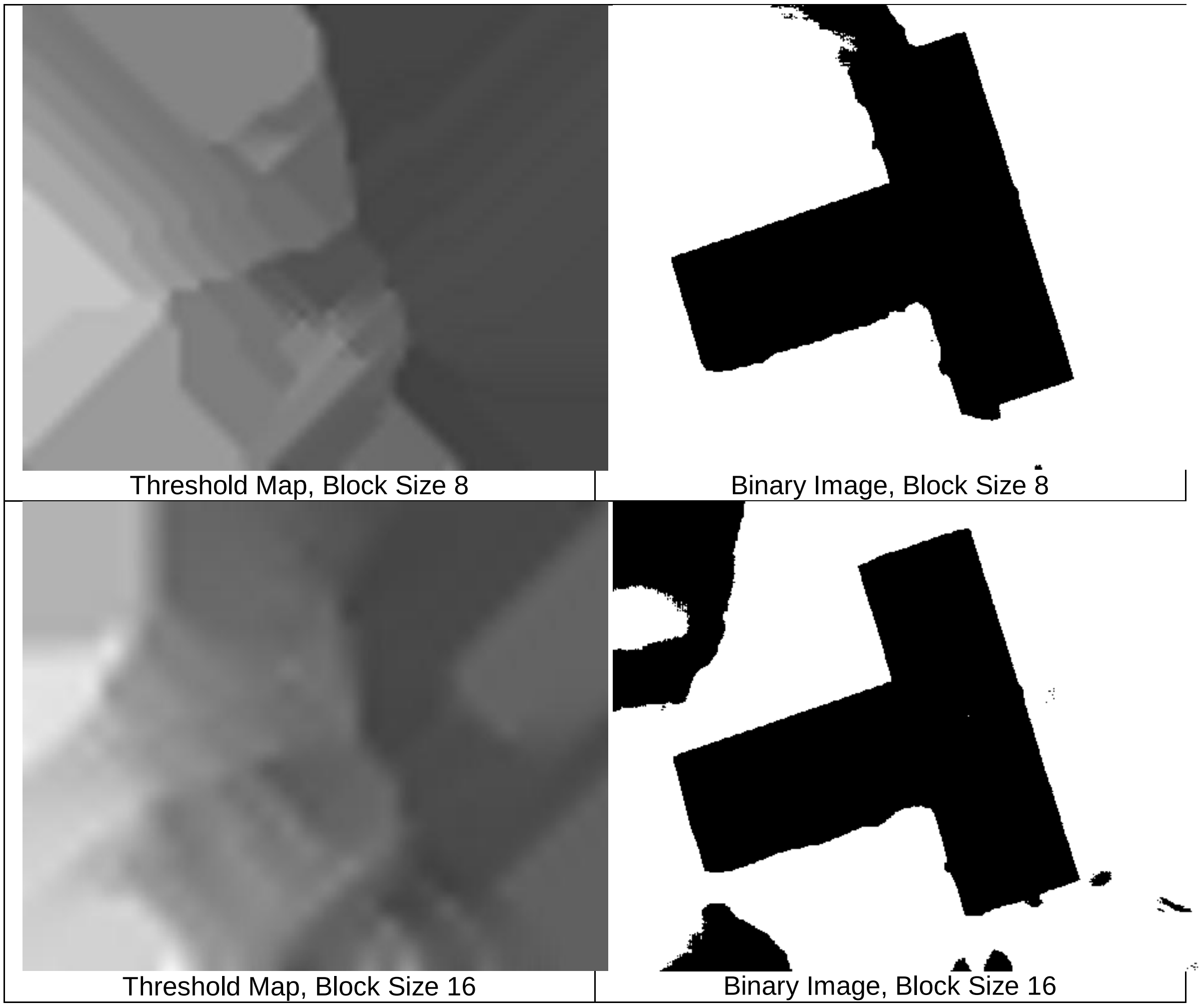
Niblack, 75x75 C=4

Phansalker, Radius=75

Sauvola, 101x101, k = 0.15

FIGURE 4

Note that except for Sauvola and the MidGrey methods, none of these techniques provides an adequate segmentation of the black "T" from the obliquely illuminated background, a detailed quantitative analysis will follow. In addition to local adaptive techniques shown above the method of Chow and Kaneko is also considered. To review, in this approach a grid of fixed size is defined and the local threshold is calculated for each grid using an automatic thresholding technique such as Otsu's method (for example). For any given point in the image a threshold is calculated by interpolating over the threshold values calculated in each grid. Figure 5 shows the results of segmentation, along with the threshold map, for grid sizes of 8 and 16 respectively:


Threshold Map, Block Size 8

Binary Image, Block Size 8

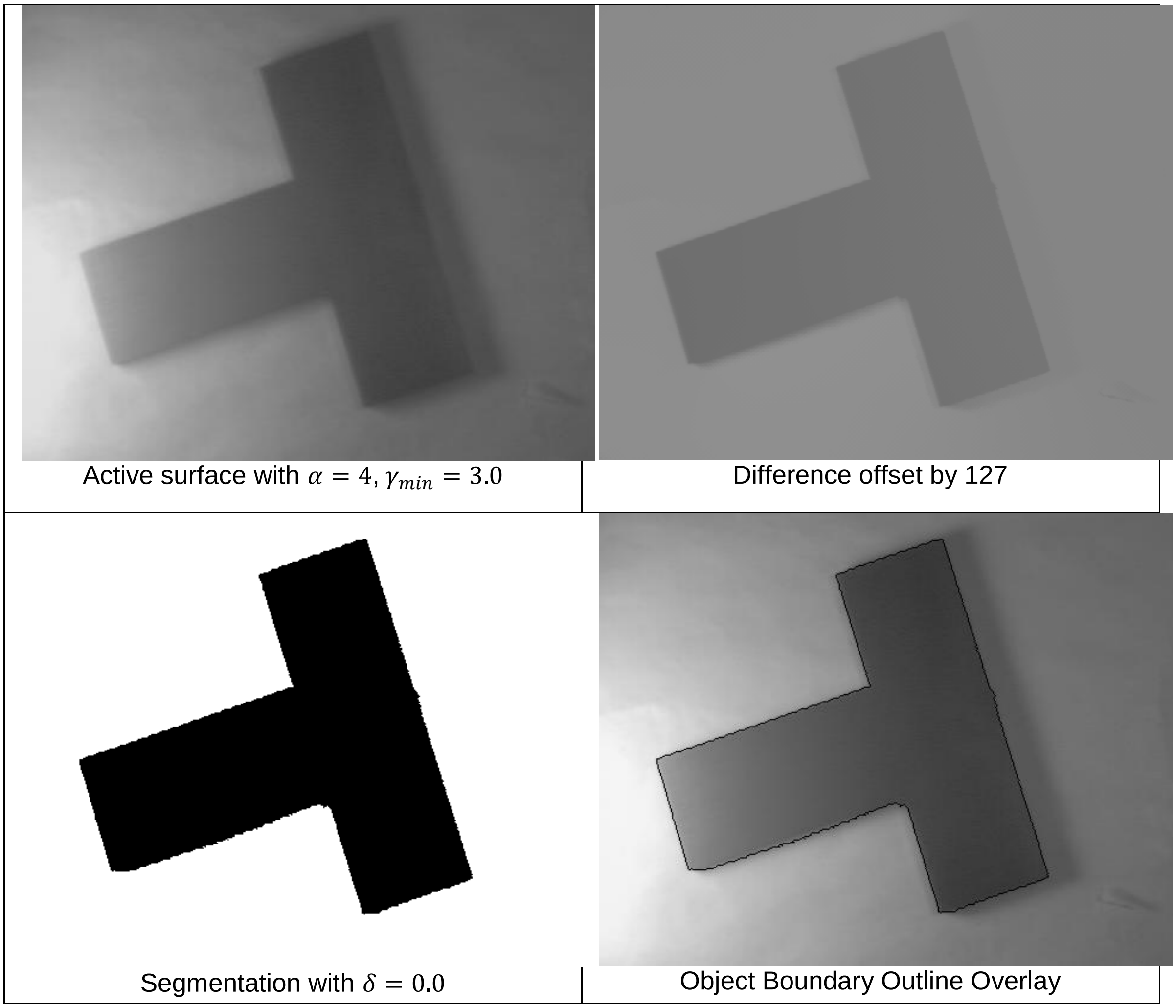
Threshold Map, Block Size 16

Binary Image, Block Size 16

FIGURE 5

With a block size of 16, the general outline of the object is captured. Note that just as with local adaptive methods, the success of segmentation is highly dependent upon the choice of block size, which needs to determined from information about the image before segmentation can be performed.

To applying the technique detailed in this paper it is necessary to first chose values for α, γ_{min}, and δ. For the first set $\alpha = 4$, $\gamma_{min} = 3.0$, and $\delta = 0.0$, for the second set $\alpha = 1/9$, $\gamma_{min} = 3.0$, and $\delta = 9.0$. The results of segmenting the image in Figure 1 with these two different sets of values are present in table form in Figure 6 and Figure 7. From first to last, the images are as follows: The first image shows the active surface resulting from optimization; The second image shows the difference between the original image and the optimized image offset by 127 to create a grayscale image. The third and fourth images show the results of segmentation along with an overlay on the original image of the detected object's boundary (outlined in black).



Active surface with $\alpha = 4$, $\gamma_{min} = 3.0$

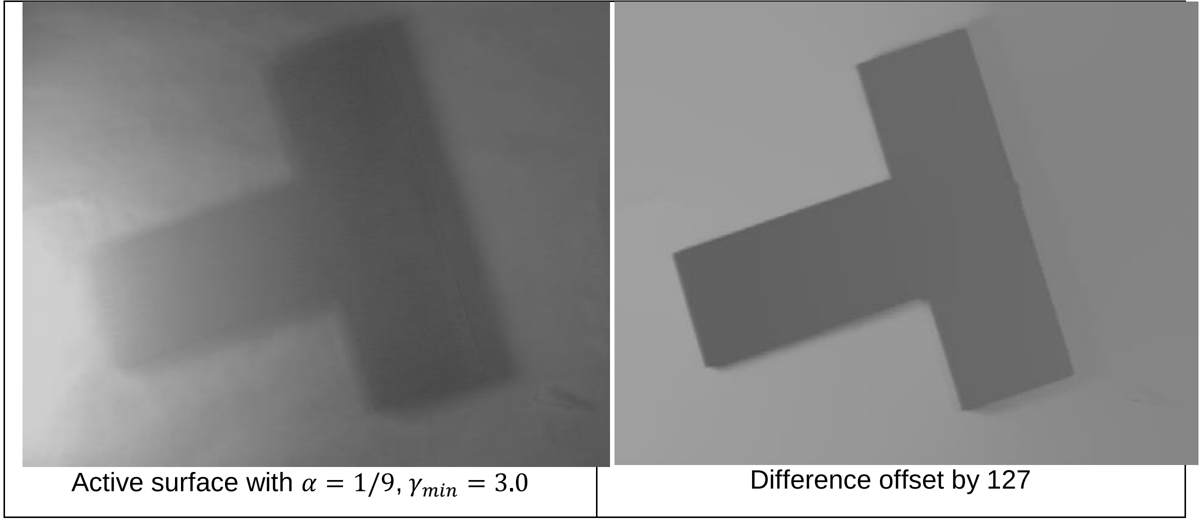
Difference offset by 127

Segmentation with $\delta = 0.0$

Object Boundary Outline Overlay

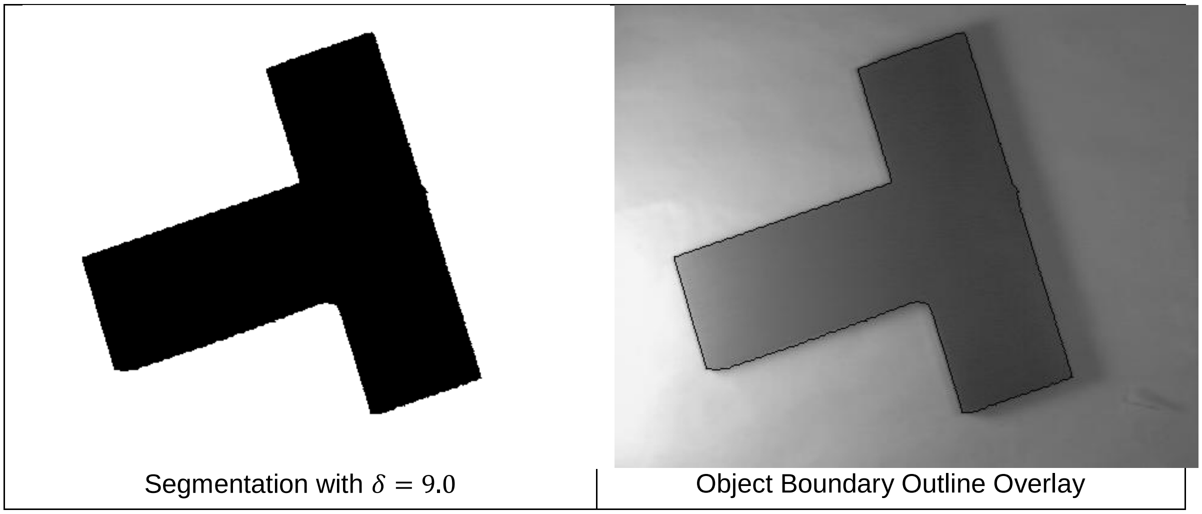
FIGURE 6: The results of segmenting with $\alpha = 4$, $\gamma_{min} = 3.0$, and $\delta = 0.0$.

Next, with different parameters:



Active surface with $\alpha = 1/9$, $\gamma_{min} = 3.0$

Difference offset by 127



Segmentation with $\delta = 9.0$

Object Boundary Outline Overlay

FIGURE 7: The results of segmenting with $\alpha = 1/9, \gamma_{min} = 3.0$, and $\delta = 9.0$.

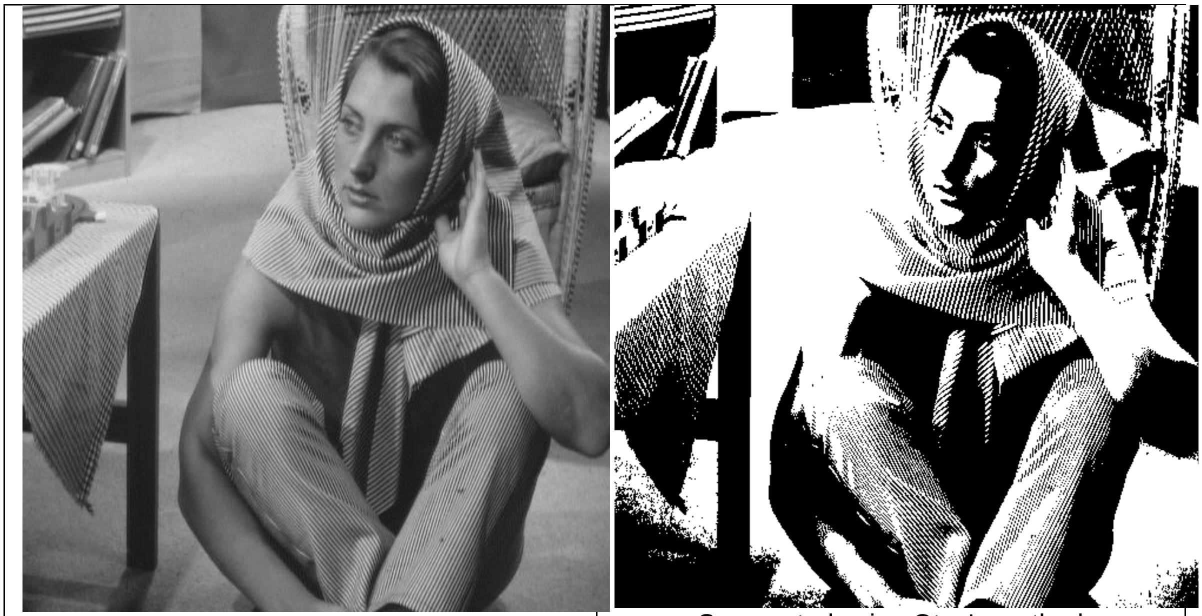
Note from the images in Figure 6 and Figure 7 that the method detailed herein properly segmented the object of interest (the black "T") from the obliquely illuminated background without the introduction of additional unwanted artifacts whereas most standard methods failed to do so. Compare the active surface of the proposed method with the threshold map created using the method of Chow and Kaneko and note that it much more closely follows the intensity profile of the original image, allowing for proper segmentation. As shown above, segmentation with the proposed method can be done with different values of α, with δ adjusted accordingly, demonstrating the algorithms relative insensitivity to starting parameters. The overall affect of decreasing α is to increase the smoothing of the optimized image, with sufficiently small values of α resulting in a nearly uniform image at a single gray value, and segmentation is then very close to the segmentation achieved with global thresholding methods. The accuracy results for all the methods surveyed, as illustrated above and compared against the ground truth image shown in Figure 2, are shown below in Table 1:

Algorithm	IoU	PA	Jaccard	Yule
Bernsen	0.6098	0.8379	0.6098	0.6097
Chow and Kaneko: BS=8	0.9225	0.9787	0.9226	0.9233
Chow and Kaneko: BS=16	0.728	0.9053	0.7281	0.7282
C-Mean	0.865	0.961	0.8651	0.8709
MidGrey	0.9578	0.9889	0.9579	0.9627
Niblack	0.7405	0.9111	0.7406	0.7407
Phansalker	0.8277	0.9473	0.8277	0.8284
Sauvola	0.9646	0.9908	0.9647	0.9686
Otsu	0.3692	0.5813	0.3692	0.3488
Proposed: α=4, γ=3, δ=0	0.9886	0.9971	0.9887	0.9944
Proposed: α=1/9, γ=3, δ=9	0.9912	0.9978	0.9913	0.9916

TABLE 1

Note that of the methods analyzed, the proposed approach performed best overall metrics, although good matching was also achieved with the MidGrey algorithm and Sauvola's method. Note that the IoU, Jaccard, and Yule metrics yield very similar results, with PA being higher because of the influence of the TN pixels, which typically correspond to large areas of the image which are of no interest.

Next, consider the highly textured image in Figure 8, along with it' s segmentation using Otsu's method:

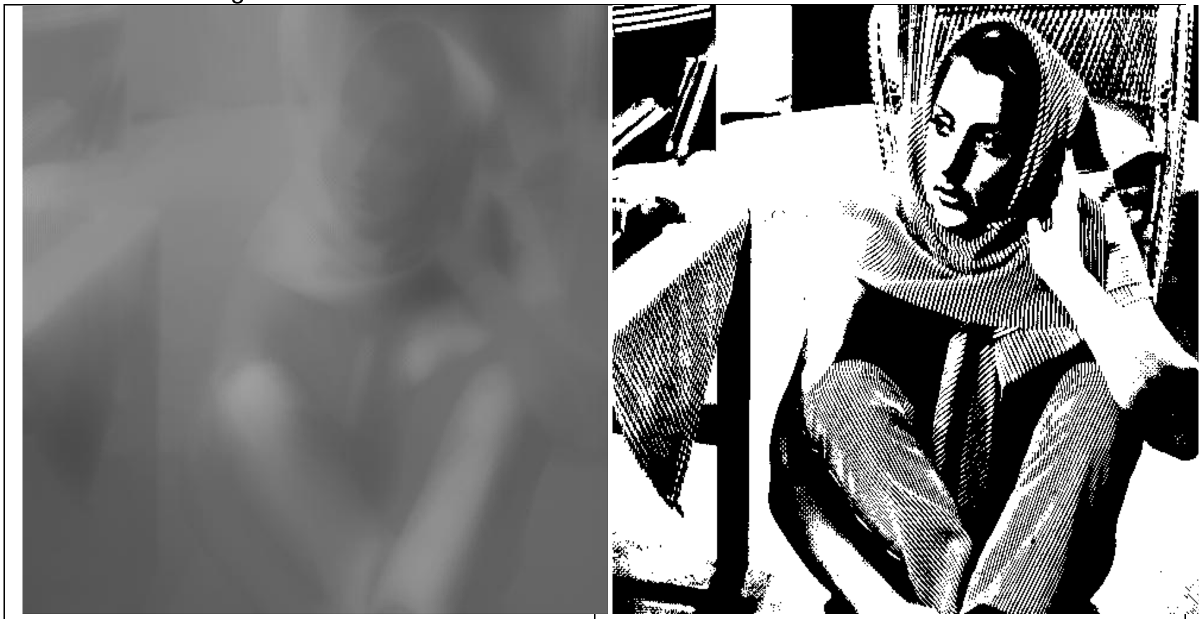


Original Image[34]

Segmented using Otsu's method

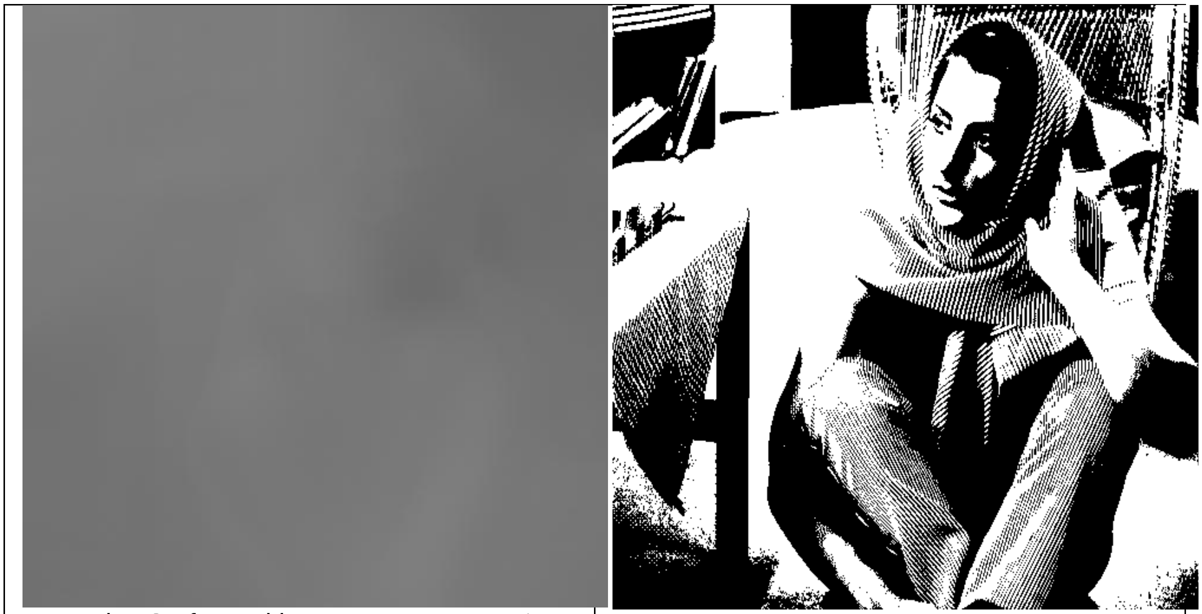
FIGURE 8: Image Segmented using Otsu's Method.

The affects of segmenting the image from Figure 8 with different values of α, γ_{min}, and δ are shown below in Figure 9:



Active Surface with $\alpha = 1/9, \gamma_{min} = 10$

Image segmentation with $\delta = 0$



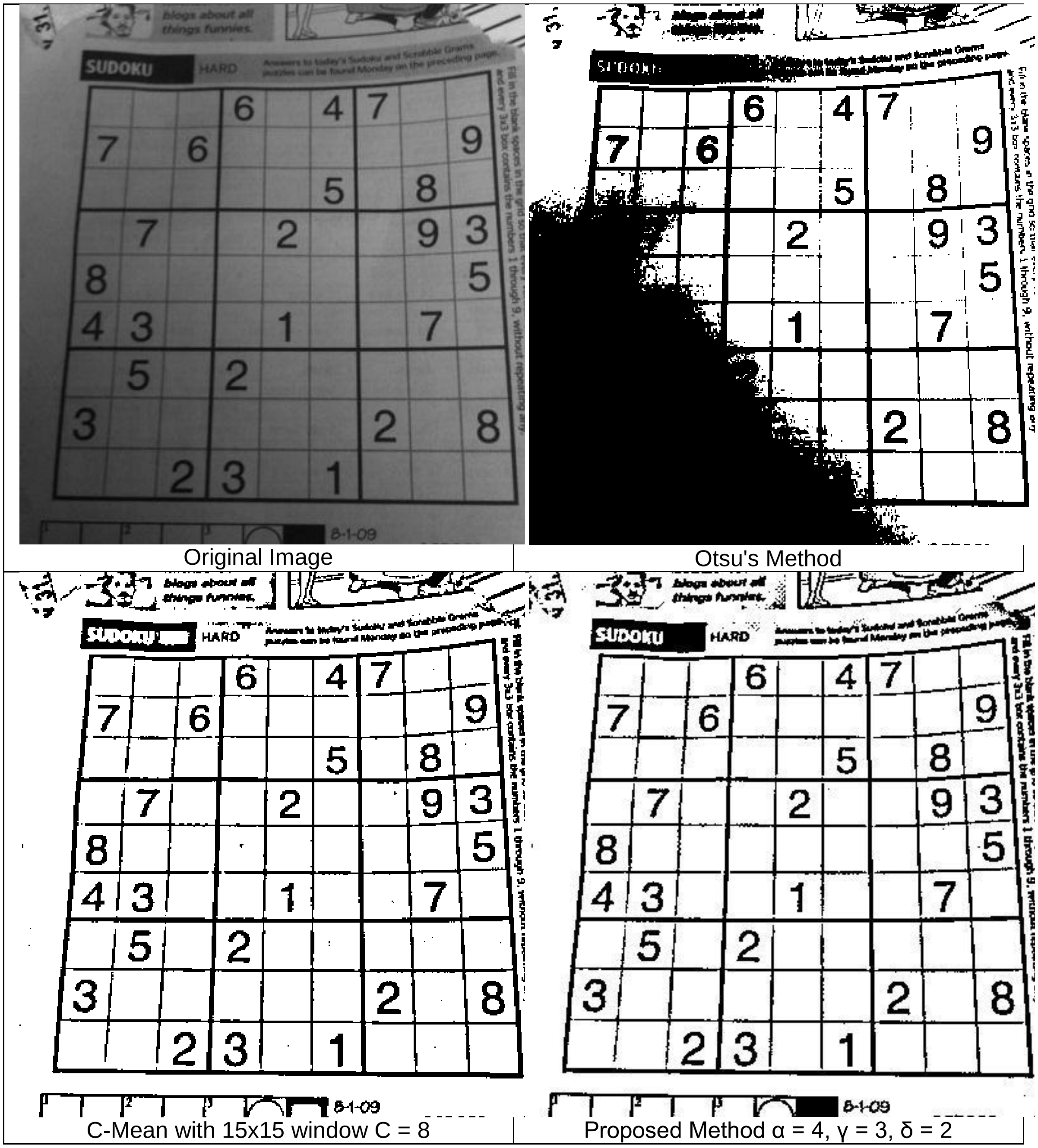
Active Surface with $\alpha = 0.01, \gamma_{min} = 25$

Image segmentation with $\delta = 5$

FIGURE 9: Active surface and segmentation with different values of α, γ_{min}, and δ.

As mentioned earlier, decreasing values of α and increasing values of γ_{min}result in greater smoothing in the optimized image. Note that smoothing with $\alpha = 0.01, \gamma_{min} = 25$ results in a nearly uniform active surface, and segmenting with $\delta = 5$ results in an output very similar to that obtained using Otsu's method.

Figure 10 presents the results of segmenting a document, a Sudoku puzzle[24], using several different methods, including the proposed approach. From first to last, the images are as follows: The original image is first, the second shows segmentation using Otsu's method. The third image shows segmentation using C-Mean adaptive thresholding with a 15x15 window and C=8, and the final image shows the results of segmentation with the proposed method using with $\alpha = 4$, $\gamma_{min} = 3.0$ and $\delta = 2$, note that these are the similar to the settings used for the large "T", illustrating the flexibility of this approach and it's ability to segment very different objects using the similar parameters The performance of the proposed approach compares well with the C-Mean method, but in addition to properly segmenting the text, it preserves the black box surrounding "SUDOKU" near the top and the small black box at the bottom of the image.



Original Image

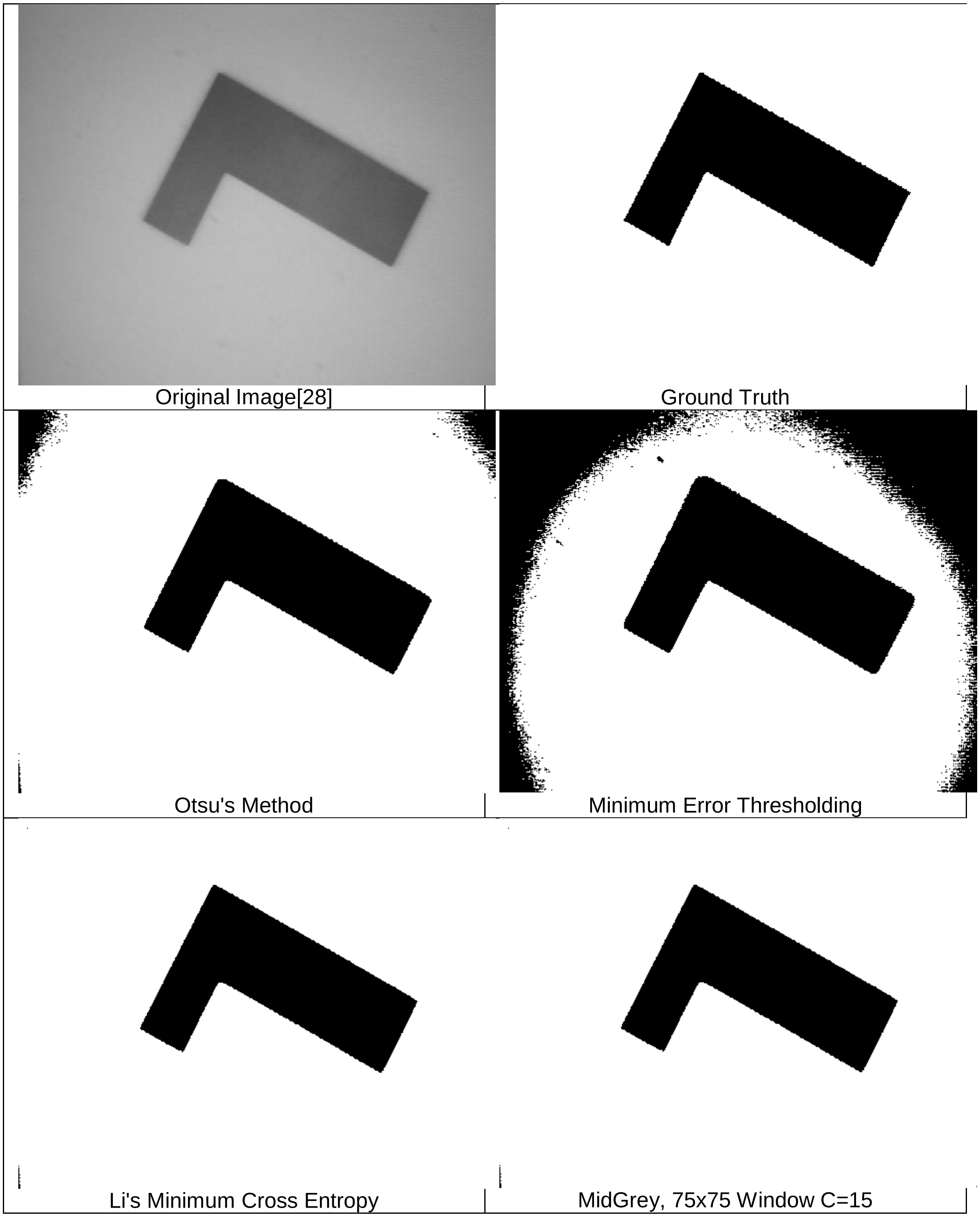
Otsu's Method

C-Mean with 15x15 window C = 8

Proposed Method α = 4, γ = 3, δ = 2

FIGURE 10: Segmentation of Sudoku Puzzle.

Figure 11 presents the results of segmenting an image of a non-uniformly illuminated object [28] using several different methods. The original image is first, the second image shows the ground truth image, and the remaining images shows segmentation using various other methods. Note that while in all cases the object is properly segmented, additional artifacts are also identified because of the uneven illumination (a situation significantly mitigated using Li's minimum cross entropy, as shown in the fifth image of this sequence, see Li[8] and also N. Pal[26] for discussions of this method). Results are also presented using Minimum error thresholding [12], and the MidGrey approach along with the method of Chow and Kaneko.



Original Image[28]

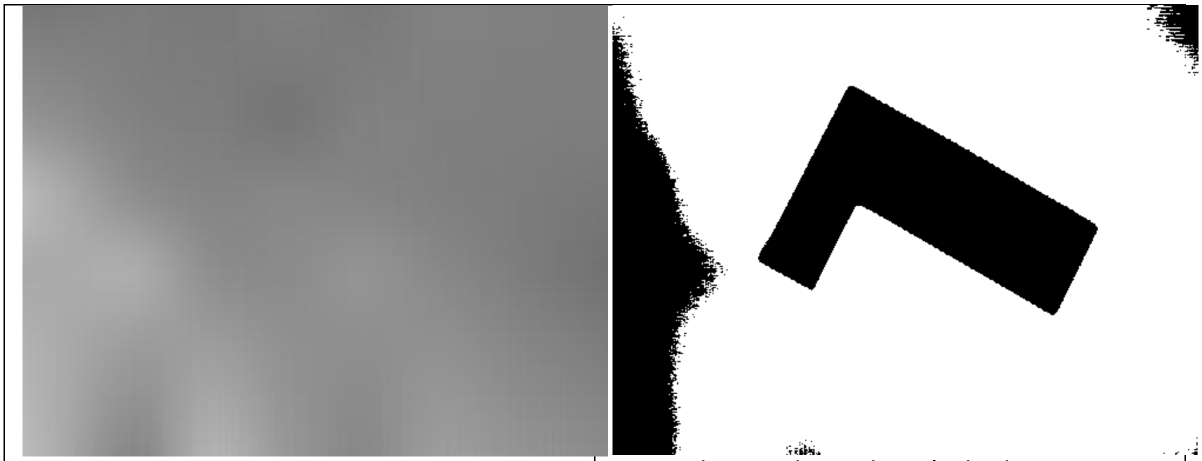
Ground Truth

Otsu's Method

Minimum Error Thresholding

Li's Minimum Cross Entropy

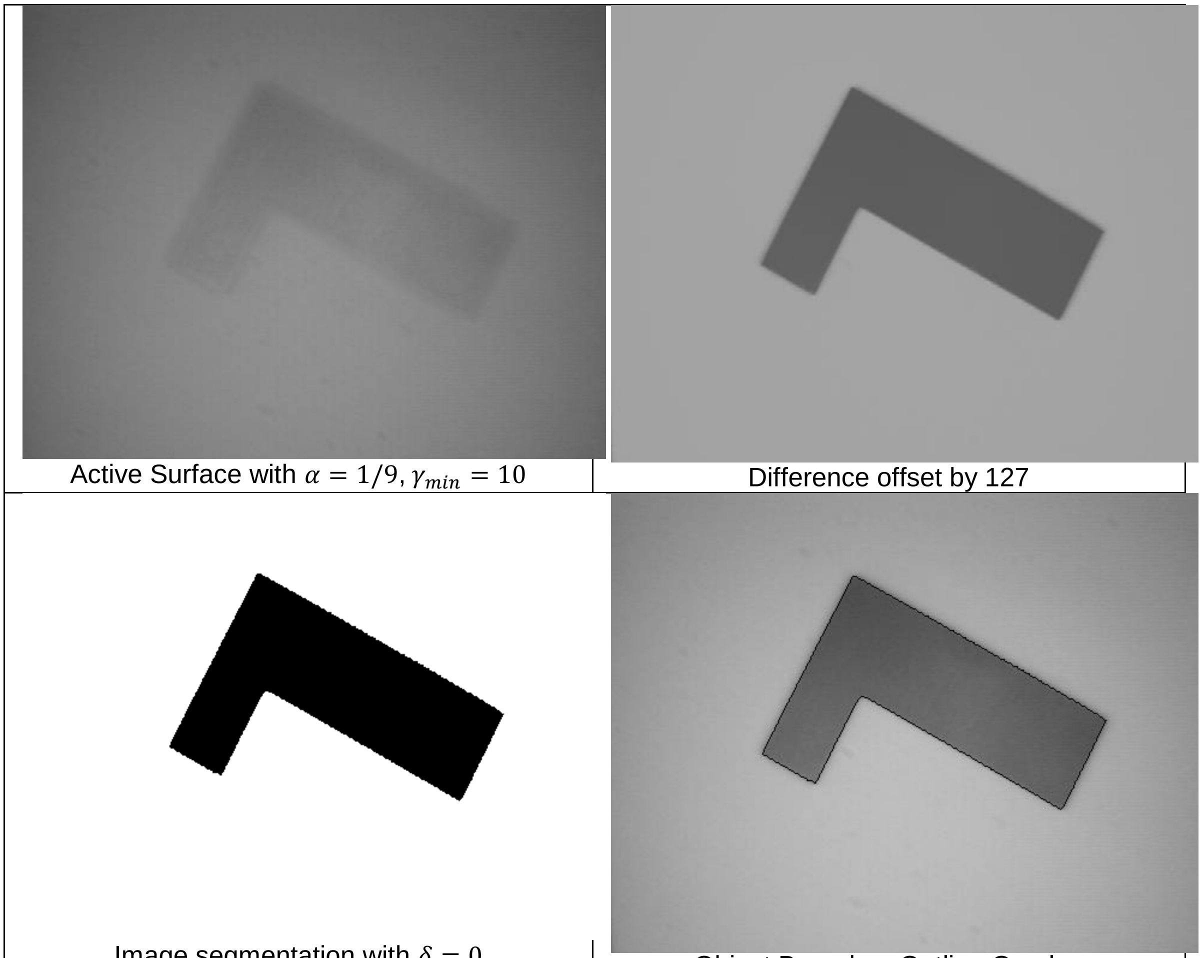
MidGrey, 75x75 Window C=15



Chow and Kaneko, Map with Block Size = 64

Chow and Kaneko, Block Size = 64

The next four images show the results of processing the image using the proposed approach:



Active Surface with $\alpha = 1/9, \gamma_{min} = 10$

Difference offset by 127

Image segmentation with $\delta = 0$

Object Boundary Outline Overlay

FIGURE 11

The first image is of the active surface created by smoothing the original image with $\alpha = 1/9, \gamma_{min} = 10$. The second image (proceeding form left to right, top to bottom) shows the difference between the original image and the active surface, offset by 127 so the results can be displayed. Note that the non-uniformity has been removed, allowing the object of interest to be properly segmented from the background without any additional artifacts being created, as shown in the third image. The forth image shows the outline of the segmented object over-laid on the original image. Metrics for all approaches are shown below in Table 2:

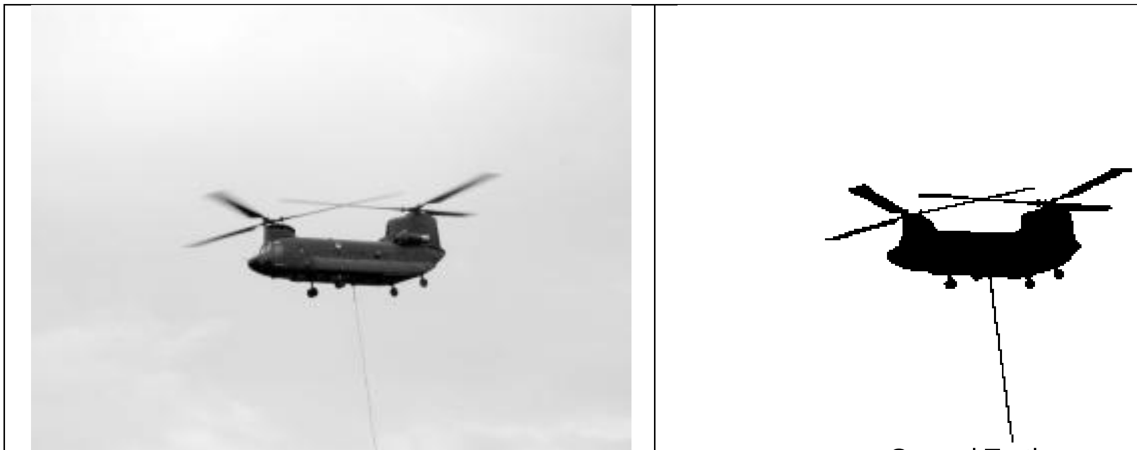
Algorithm	IoU	PA	Jaccard	Yule
Otsu	0.8024	0.9684	0.8024	0.803
Li's Minimum Cross Entropy	0.9561	0.9941	0.9561	0.9614
Minimum Error	0.3328	0.7425	0.3328	0.3328
MidGrey	0.9887	0.9985	0.9887	0.9948
Chow and Kaneko: BS=64	0.5564	0.8976	0.5564	0.5564
Proposed: α=1/9, γ=10, δ=0	0.9939	0.9992	0.9939	0.9952

TABLE 2

Note that the proposed method performed best amongst all the methods considered.

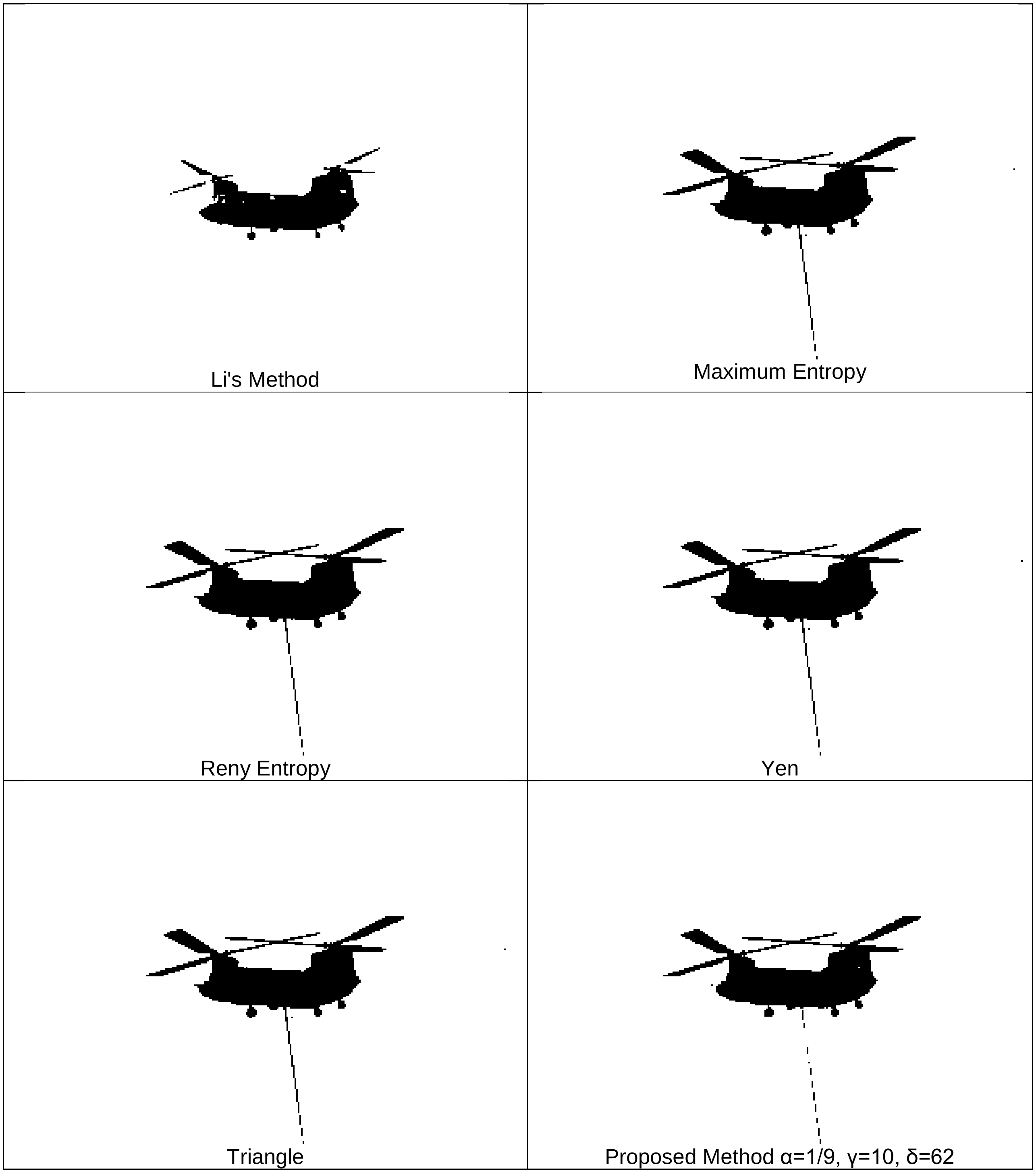
The following two images are taken from a public single object database made available online by the Computer Science and Applied Mathematics department of the Weizmann institute of Science[6]. Ground truth segmentations are provided with each image in the database.

Figure 12 shows the results of using several algorithms, in addition to the proposed approach, to segment the image of a helicopter:



Original Image[6]

Ground Truth



Li's Method

Maximum Entropy

Reny Entropy

Yen

Triangle

Proposed Method α=1/9, γ=10, δ=62

FIGURE 12

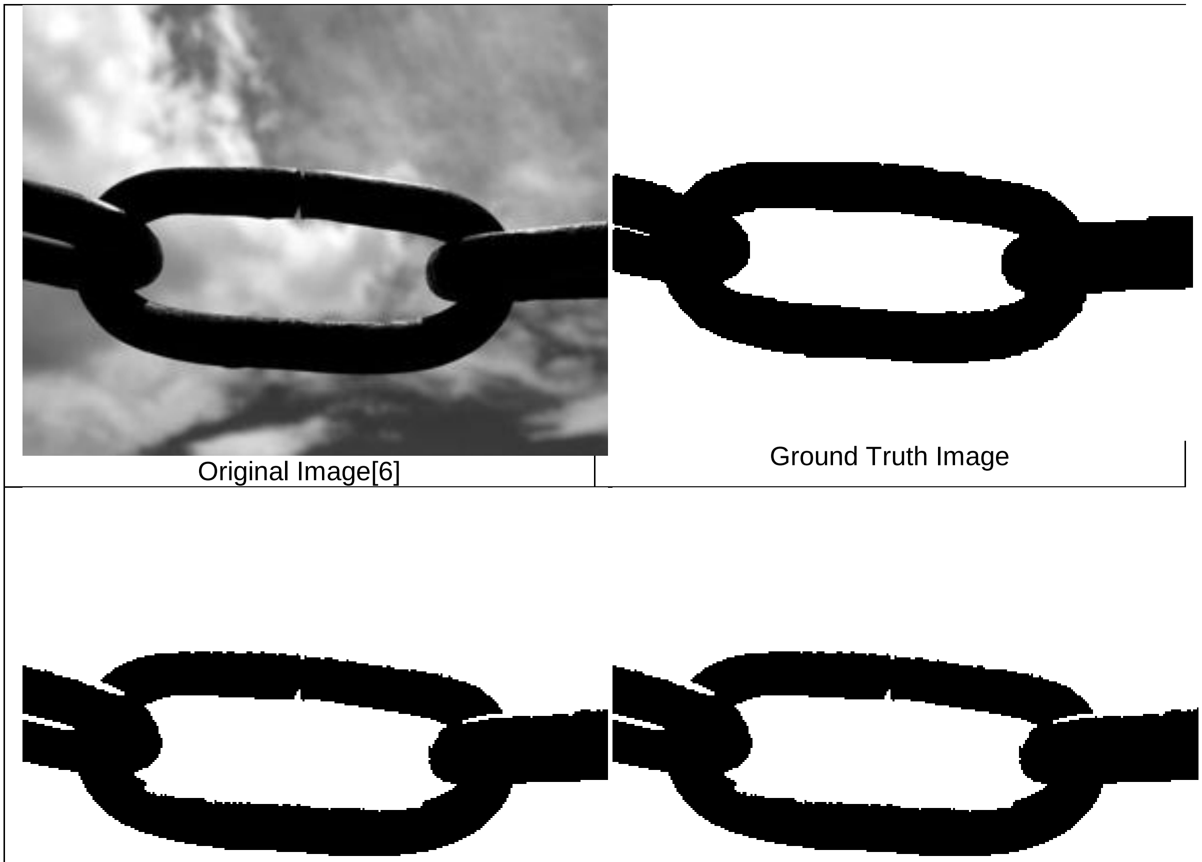
Metrics for the tested approaches are shown below in Table 3:

Algorithm	IoU	PA	Jaccard	Yule
Li's Minimum Cross Entropy	0.7533	0.9878	0.7533	0.9874
Reny Entropy	0.9381	0.9968	0.9381	0.9485
Maximum Entropy	0.9321	0.9964	0.9321	0.9406
Yen	0.9321	0.9964	0.9321	0.9406
Triangle	0.9321	0.9964	0.9321	0.9406
Proposed: α=1/9, γ=10, δ=62	0.9347	0.9967	0.9347	0.9596

TABLE 3

All methods listed, including the proposed approach, provide a good segmentation of the original image, as defined by the ground truth set.

Figure 13 shows a silhouetted image of a chain against a background of sky. The results of using several algorithms, including the proposed approach, to segment the image are shown below:

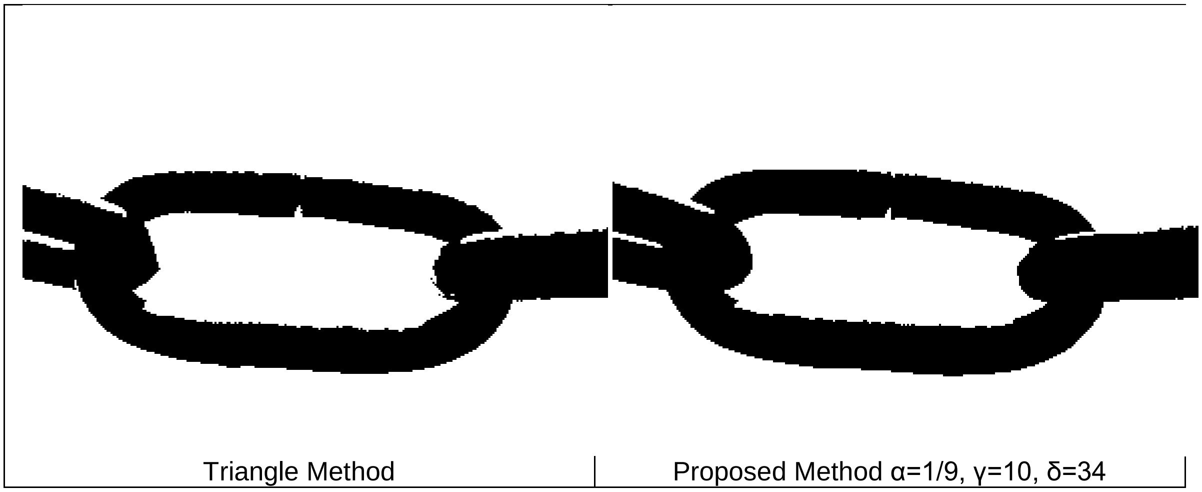


Original Image[6]

Ground Truth Image

Huang's Method

Minimum Cross Entropy


Triangle Method

Proposed Method α=1/9, γ=10, δ=34

FIGURE 13

Metrics for the tested approaches are shown below in Table 4:

Algorithm	IoU	PA	Jaccard	Yule
Huang	0.8983	0.9753	0.8983	0.9685
Li's Minimum Cross Entropy	0.8992	0.9756	0.8992	0.9688
Triangle	0.8502	0.9637	0.8502	0.9543
Proposed: α=1/9, γ=10, δ=34	0.9195	0.9805	0.9195	0.9749

TABLE 4

The proposed approached performed better overall than the other approaches. In both of the above examples more algorithms were considered than presented, as those which did poorly by either failing to segment the object from the background or by including other artifacts than the object, were excluded altogether. Upon reviewing the metrics, it can be seen that only the Triangle method and Li's Minimum Cross Entropy, in addition to the proposed approach, were able to provide adequate segmentation on the two sets of images, with the proposed approach performing better than each in both cases. Below in Table 5 is a summary, the top three rows are for Figure 12, the bottom three for Figure 13:

Algorithm	IoU	PA	Jaccard	Yule
Li's Minimum Cross Entropy	0.7533	0.9878	0.7533	0.9874
Triangle	0.9321	0.9964	0.9321	0.9406
Proposed: α=1/9, γ=10, δ=62	0.9347	0.9967	0.9347	0.9596
Li's Minimum Cross Entropy	0.8992	0.9756	0.8992	0.9688
Triangle	0.8502	0.9637	0.8502	0.9543
Proposed: α=1/9, γ=10, δ=34	0.9195	0.9805	0.9195	0.9749

TABLE 5

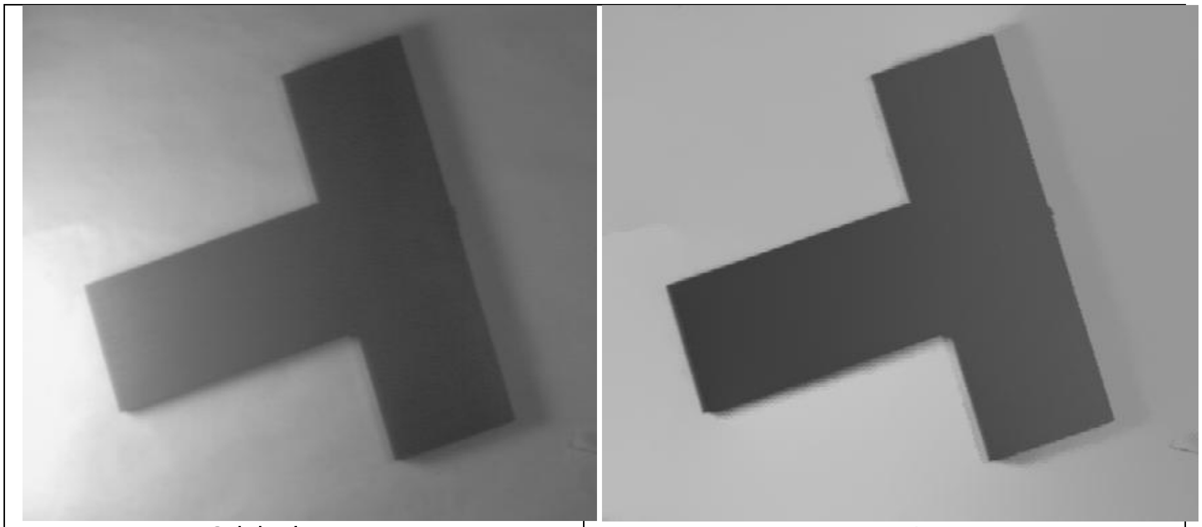
4. ADVANTAGES AND DISADVANTAGES

The principle advantage of the proposed approach is it's ability to segment large objects, normally a challenging task for adaptive thresholding algorithms because of the need to select a window size large enough to cover the object but small enough to allow it to be separated properly from the background without crearting artifacts or holes within the object itself. The proposed approach handles large uniform areas (even areas were illumination is uneven, as shown in particular with regard to the image in Figure 1) without the creation of ghost artifacts because of it's contour following properties. These help the algorithm create a thresholding surface which parallels that of the image in nearly uniform regions, with optimization only being performed in transitional areas were the gradient is high. Another advantage is that no a-prior knowledge about the object to be segmented is required. For local neighborhood based techniques optimal segmentation requires that the selection of the window size be determined by the size of the object to be detected, thus requiring some understanding of the scene. The same holds true for thresholding methods which partition the image into rectangular regions in order to find local thresholds over which to interpolate. Note that for the object in Fig.1, using different block sizes with the method of Chow and Kaneko resulted in noticeably different segmentations. With the proposed approach knowledge about the image, or the objects to be segmented, is not necessary; most of the algorithm parameters are determined from the gradient information alone, the entire process being determined by several parameters, values which typically generalize well. Only γ_{min} is best chosen on an image by image basis and this can be chosen directly from a histogram of the gradient. This approach also offers the advantage of simplicity over techniques which require the identification of edges, processing windows and associated parameters, or interpolating functions along with sets of interpolating points. Problems associated with the behavior of interpolating functions in areas away from the interpolating points, along with problems near the edges of the image, are also avoided resulting in an algorithm which is simple to use.

The primary disadvantage of the proposed approach is processing speed. The proposed approach is computationally intensive, since every point in the image is represented by an equation, all of which need to be optimized simultaneously. Even for a modest sized image of 512x512 a total of 262144 equations are required, and most cameras have much larger array sizes. For example, the image in Figure 1, with a size of 496 by 380, took a total of 8.825 seconds to segment. Another disadvantage is the need to specify three parameters for segmentation, most algorithms require only one or two.

5. FUTURE DEVELOPMENT

Future efforts will focus on speed improvements and automated parameter generation. A conjugate gradient method is currently used to iteratively determine the solution, investigation into other iterative approaches, such as Guass-Seidel, should be conducted to determine if processing can be speed up. In addition to speed improvements, methods for automatically generating the parameters needed by the proposed approach should be developed. In particular, as mentioned above, the value for γ_{min} can possibly be determined from a histogram of the gradient of the image by noting, as is done with other approaches, that high gradient values are typically associated with the edges of objects to be detected. The value for δ can also be determined automatically by applying global thresholding techniques to the histogram obtained from the difference between the original image and the image obtained from smoothing using the proposed method. Examples of this image transformation are shown in Figures 6, 7 and 9. The results of this image transformation for the object shown in Figure 1 are shown below, they are similar to those shown in Figure 6, except the transformed image has histogram equalization applied to increase it's dynamic range:



Original Image

Transformed Image

FIGURE 14

Note that the transformed image retains the subtleties and nuances of the original, including the shadow cast by the part. At this point, it should be possible to segment the transformed image using a variety of global segmentation techniques. This approach would eliminate the need to determine the value for δ explicitly. These efforts should result in an improved version of the proposed algorithm which is faster and easier to use.

6. CONCLUSION

In conclusion, a method has been presented for performing image segmentation based upon a gradient minimization strategy which successfully creates an smooth surface that can be used for adaptive thresholding and image segmentation. Comparisons with commonly used approaches against test cases with ground truth segmentations indicated comparable or superior results, while retaining the advantage of simplicity to use.

7. REFERENCES

[1] A. M. Khan, R. S. (September 2013). Image Segmentation Methods: A Comparative Study. *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-4,*, 1-9.

[2] A.M. Khan, R. S. (September 2013). Image Segmentation Methods: A Comparative Study. *International Journal of Soft Computing and Engineering (IJSCE), 3*(4), 4.

[3] Agus Zainal Arifin, A. A. (2006). Image segmentation by histogram thresholding using hierarchical cluster analysis. *Pattern Recognition Letters, 27*, 1515–1521.

[4] Atkinson, K. E. (1978). *An Introduction to Numerical Analysis.* New York: John Wiley & Sons.

[5] Bernsen, J. (1986). Dynamic thresholding of gray level images. *Proc. of the 8th Int. Conf. on Pattern Recognition*, 1251–1255.

[6] Brandt, S. A. (2007). Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration. *Proceedings of the IEEE Conference on Computer Vision and Pattern*

Recognition(June). Retrieved from http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/index.html

[7] C.K. CHow, T. K. (February 15, 1972). Automatic Boundray Detection of the Left Ventricle from Cineangiograms. *Computers and Biomedical Research, 5*, 388-410.

[8] Chun-hung Li, K. L. (1993). Minimum cross entropy tresholding. *Pattern Recognition, 26*, 617-625.

[9] David R. Martin, C. C. (2004). Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 26*(1), 1-20.

[10] Fei Liu, Y. L. (April 2003). Active surface model-based adaptive thresholding algorithm by repulsive external force. *Journal of Electronic Imaging 12(2)*, 299-306.

[11] Ilya Blayvas, A. B. (2006). Efficient computation of adaptive threshold surfaces for image binarization. *Pattern Recognition, The journal of the Pattern Recognition Society, 39*, 89-101.

[12] J. Kittler, J. I. (1986). Minimum error thresholding. *Pattern Recognition, 19*(1), 41-47.

[13] J. Sauvola, M. P. (1999). Adaptive document image binarization. *Pattern Recognition, 33*, 225-236.

[14] Jordan, J. (2018, May 30). *Evaluating image segmentation models*. (Proofpoint) Retrieved May 12, 2019, from https://www.jeremyjordan.me/evaluating-image-segmentation-models/

[15] Landini, G. (2017, April 29). *Auto Local Threshold*. (ImageJ) Retrieved March 10, 2019, from http://imagej.net/Auto_Threshold

[16] Landini, G. (2017, Apr 29). *Auto Threshold*. Retrieved 1 2019, Jan, from https://imagej.net/Auto_Threshold

[17] Liang, J. (n.d.). *CANNY EDGE DETECTION*. Retrieved 6 15, 2019, from http://justin-liang.com/tutorials/canny

[18] Mehmet Sezgin, B. S. (2004). Survey over Image Thresholding Techniques and Quantitiative Performance Evaluation. *Journal of Electronic Imaging, 13*(1), 146-165.

[19] Milan Sonka, V. H. (2015). Image Processing, Analysis, and Machine Vision. Stamford CT: Cengage Learning.

[20] N. Bonnet, J. C. (2002). A ‘no-threshold’ histogram-based image segmentation method. *Pattern Recognition, 35*, 2319 – 2322.

[21] Nagy, G. (2000). Twenty Years of Document Image Analysis in PAMI. *Pattern Analysis and Machine Intelligence, 22*(1), 38-62.

[22] Neerad Phansalkar, S. M. (2011). Adaptive local thresholding for detection of nuclei in diversity stained cytology images. *International Conference on Communications and Signal Processing.* Calicut, India.

[23] Nikil R Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition, 26*(9), 1277-1294.

[24] OpenCV. (2016). OpenCV. Retrieved from http://opencv.org

[25] Otsu, N. (1979). A Threshold Selection Method from Gray-level Histograms. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, SMC-9*(1), 62-66.

[26] PAL, N. R. (1996). ON MINIMUM CROSS-ENTROPY THRESHOLDING. *Pattern Recognition, 29*, 575-580.

[27] Paul L Rosin, E. I. (2003). Evaluation of global image thresholding for change detection. *Pattern Recognition Letters, 24*, 2345–2356.

[28] R. B. Fisher, K. K. (1998). *Interactive Textbooks; Embedding Image Processing Operator Demonstrations in Text.* Retrieved October 14, 2016, from Int. J. of Pattern Recognition and Artificial Intelligence, Vol 12, No 8, pp 1095-1123,: Images obtained from: http://homepages.inf.ed.ac.uk/rbf/HIPR2/

[29] Rashid Mehmood, J. C. (2005). *Parallel iterative solution method fo rlarge sparse linear equation systems.* Cambridge, UK.: University of Cambridge Computer Laboratory.

[30] S. D. Yanowitz, A. M. (1989). A New Method for Image Segmentation. *Computer Vision, Graphics, and Image Processing, 46*, 82-95.

[31] Salvatore Tabbone, L. W. (2001). Multi-scale binarization of images. *Pattern Recognition Letters, 24*, 403–411.

[32] Singh, T. R. (2012). A New Local Adaptive Thresholding Technique in Binarization. *International Journal of Computer Science IJSCI, 8*(6), 271.

[33] Tony F. Chan, L. A. (2001). Active Contours without Edges. *IEEE Transactions on Image Processing, 10*(2), 266-277.

[34] Wisconsin-Madison, U. o. (n.d.). Retrieved from https://homepages.cae.wisc.edu/~ece533/images