

9-2013

Document Binarization with Automatic Parameter Tuning

Nicholas Howe

Smith College, nhowe@smith.edu

Follow this and additional works at: https://scholarworks.smith.edu/csc_facpubs



Part of the [Computer Sciences Commons](#)

Recommended Citation

Howe, Nicholas, "Document Binarization with Automatic Parameter Tuning" (2013). Computer Science: Faculty Publications, Smith College, Northampton, MA.

https://scholarworks.smith.edu/csc_facpubs/128

This Article has been accepted for inclusion in Computer Science: Faculty Publications by an authorized administrator of Smith ScholarWorks. For more information, please contact scholarworks@smith.edu

Document Binarization with Automatic Parameter Tuning

Nicholas R. Howe

Received: date / Accepted: date

Abstract

Document analysis systems often begin with binarization as a first processing stage. Although numerous techniques for binarization have been proposed, the results produced can vary in quality and often prove sensitive to the settings of one or more control parameters. This paper examines a promising approach to binarization based upon simple principles, and shows that its success depends most significantly upon the values of two key parameters. It further describes an automatic technique for setting these parameters in a manner that tunes them to the individual image, yielding a final binarization algorithm that can cut total error by one-third with respect to the baseline version. The results of this method advance the state of the art on recent benchmarks.

1 Introduction

Physical documents in the real world exhibit a multitude of colors and shadings, and the appearance of even a single document may vary widely depending upon factors such as lighting, viewing angle, etc. Digital reproductions of physical documents usually make use of a 24-bit color representation, or perhaps 8-bit grayscale. Such formats omit some of the information present in the original, but retain more than enough for most applications. Certain document processing schemes go much further: they retain only a single bit per document pixel. Binary document representations are very useful despite their low fidelity because the information lost in binarization is more or less unrelated to the symbolic content of the document. Most documents are produced using

monochromatic ink on paper, and their meaning is embodied solely in the distribution of the ink, a pattern that the binary document explicitly represents.

Of course, inferring the correct binarization of a document from its color or grayscale representation can prove challenging. Physical degradation of the document, adverse lighting or imaging conditions, and limitations on resolution can all conspire to obscure the original pattern.¹ To meet this challenge, researchers have proposed many algorithms to binarize document images. Indeed, intense activity in this field has made the Document Image Binarization Contest (DIBCO) series one of the most popular competitions within the document analysis research community, attracting seventeen entries in its most recent iteration [19]. Nevertheless, the results from these contests reveal that there remains room for improvement in the quality of automatic binarization.

This paper makes two related contributions to the study of automatic document binarization. First, it describes a method that can achieve excellent results on a wide variety of challenging document images. Second, as a means to these results it investigates a method to automatically determine the best parameter values suited to each particular image. Automatic parameter estimation has received little research attention to date but offers two advantages when done well: it makes the method easier to employ since the

¹Some ambiguity remains over how to define the ground truth binarization. For instance, should an unintended ink spill made by a scribe be included? For the record, stipulate that the ground truth should identify the set of all pixels at least 50% covered with ink during the production of the document. In practice, the ground truth used on most benchmark datasets probably does not correspond perfectly with this ideal, but represents the imperfect judgment of a human expert [4].

user doesn't need to worry about finding the best parameter settings, and it also improves the average result because each image can use its own ideal value instead of a single compromise setting that works adequately but suboptimally for all images.

1.1 Prior Work

Before considering new approaches, it is worth surveying the history and development of binarization algorithms. Because most documents are printed in dark ink on light paper, many classic methods rely on thresholding the image intensity. Under ideal circumstances, as noted by Otsu [17] a simple global threshold suffices to distinguish markings from the background. In more difficult cases the overall intensity may vary over the document, so that a single global threshold may simultaneously prove too high in some areas and too low in others. Thus both Niblack [16] and Sauvola and Pietikainen [21] have proposed *locally adaptive* techniques that adjust the threshold used according to the mean and variance in some local region around each pixel. Although generally more accurate than Otsu for document images with varying intensities, locally adaptive thresholding can sometimes fail catastrophically in areas of low variance, motivating hierarchical approaches such as the *Decompose* algorithm [7]. Adaptive thresholding also presents the user with additional parameters to set, relating to the size of the local region and to the relative importance of the local mean vs. its deviation as the local threshold is determined.

The winners of recent DIBCO events go beyond locally adaptive thresholding. They achieve their results partially by using thresholds, but most also perform modeling of the ink and background classes to enable more accurate individual pixel classification. Lu et al. [14] model the document background via an iterative polynomial smoothing, and then choose local thresholds based on detected text stroke edges. This method won DIBCO 2009, and a revised version tied as a winner of H-DIBCO 2010 (handwritten documents only). The second winner in 2010 was a method described by Bar-Yosef et al. [3] that grows foreground regions iteratively based upon local modeling of the foreground and background within a 7×7

pixel window. Lelore and Bouchara [13] won DIBCO 2011 with a technique that first uses coarse thresholding to partition pixels into three groups: ink, background, and unknown. Models describe the ink and background clusters, and guide decisions on the unknown pixels. The method also employs a higher resolution grid generated from the original image via linear interpolation.

Several other ideas proposed in recent years provide motivation for the base algorithm employed in this paper. A number of projects have explored the use of Markov random fields (MRF) for binarization [12, 18, 15, 11]. Inspired by the human visual system, some work has employed center-surround filters to identify areas that are darker than their local environment, and thus likely to be ink [23]. These play a role similar to that of the Laplacian filter used herein. Finally, various approaches have used edge detection to improve binarization results [10, 20]. Although these ideas have been explored separately, the specific combination used in this work appears particularly effective.

Automatic estimation of appropriate parameter settings for document binarization has received limited attention to date. Gatos et. al. describe a parameter-free method that relies on detailed modeling of the document background [9]. Dawoud describes a method based upon cross section sequence that combines results at multiple threshold levels into a single binarization [8]. Badekas and Papamarkos [2] describe a method inspired by work on edge detection. Their technique first produces binarizations over a range of parameter settings, and estimates a ground truth via chi-square analysis on the cumulative votes of the multiple binarizations at each pixel. It then iteratively narrows the parameter range until unable to do so further, giving the final estimated best setting. Although the method described herein also begins by computing binarizations over a range of parameter values, it differs by not attempting to estimate a ground truth, and requires no further iteration to choose the ideal parameter value.

2 Approach

The base approach to binarization used in this paper was introduced recently [11] and rests on three mutually supporting strategies. First, it defines the target binarization as a labeling on pixels that minimizes a global energy function inspired by a Markov random field model. Second, in formulating the data-fidelity term of this energy it relies on the Laplacian of the image intensity to distinguish ink from background. This grants a crucial invariance to differences in contrast and overall intensity. Third, it incorporates edge discontinuities into the smoothness term of the global energy function, biasing ink boundaries to align with edges and allowing a stronger smoothness incentive over the rest of the image. The paragraphs below explain each of these points in greater detail.

The global energy function operates on binarizations B , which label each pixel indexed by (i, j) as either ink or background, $B_{ij} \in \{0, 1\}$. The energy takes on a typical additive form, with terms representing the fidelity of a particular labeling when compared to the intensity data, and other terms representing the smoothness or regularity of the solution. In particular, the energy includes a cost L_{ij}^0 or L_{ij}^1 capturing how well the label B_{ij} chosen for each pixel matches its appearance, and irregularity costs C_{ij}^h and C_{ij}^v for each pixel whose label differs respectively from those of its horizontal or vertical neighbor.

$$\begin{aligned} \mathcal{E}_I(B) = & \sum_{i=0}^m \sum_{j=0}^n [L_{ij}^0(1 - B_{ij}) + L_{ij}^1 B_{ij}] \\ & + \sum_{i=0}^{m-1} \sum_{j=0}^n C_{ij}^h (B_{ij} \neq B_{i+1,j}) \\ & + \sum_{i=0}^m \sum_{j=0}^{n-1} C_{ij}^v (B_{ij} \neq B_{i,j+1}) \end{aligned} \quad (1)$$

Assume that the Boolean expressions above evaluate to either 0 or 1 in the usual manner according to their truth value. With this energy the optimal binarization will tend to conform to the intensity contours while smoothing over small irregularities resulting from noise sources. The degree of smoothing

relative to data fidelity will depend on the relative magnitudes of L_{ij}^b to C_{ij}^h and C_{ij}^v .

The label costs L_{ij}^0 and L_{ij}^1 should be invariant to the local image illumination, and thus are taken from the Laplacian of the image intensity:

$$L_{ij}^0 = \nabla^2 I_{ij} \quad (2)$$

$$L_{ij}^1 = -\nabla^2 I_{ij} \quad (3)$$

Intuitively this will tend to separate ink from background because the Laplacian measures the divergence of the gradient. It will thus be positive at intensity valleys (ink) and negative at intensity peaks or plateaus (background). The data terms of the energy function become a summation of the label-signed Laplacian over all pixels in the image. For a particular component of ink or background, Green's theorem tells us that the summed Laplacian over all its pixels is mathematically equivalent to the gradient flux across its boundary. In other words, the energy contribution of each component is determined solely by what happens at its boundary. This makes intuitive sense but can cause trouble for components that intersect the image edges: such regions can occasionally be mislabeled because their entire natural boundary is not visible, and thus their true energy contribution can only be estimated. In practice, this causes trouble occasionally for noisy background areas that are largely isolated from the rest of the document background by ink markings. Several potential solutions exist. For example, one could simply fix L_{ij}^1 to a large negative value for all pixels (i, j) on the image border, under the assumption that all ink is framed by background regions. Rather than committing to such a strong assumption, this work adopts a more conservative strategy, looking for bright outlier pixels and applying a fixed constant L_{ij}^1 to them. This still ensures that large background regions will receive the proper label, while not preventing identification of ink pixels on the image boundary. To be precise, modify L_{ij}^1 for any pixels more than two standard deviations σ_{ij}^r brighter than the local mean μ_{ij}^r , as computed over nearby pixels weighted by a Gaussian of radius r . This may be viewed as an extremely conservative application of locally adaptive thresholding, where only pixels most certain to be background are

labeled as such. In the equation below, ϕ will take on a large negative value.

$$L_{ij}^1 = \begin{cases} -\nabla^2 I_{ij} & I_{ij} \leq \mu_{ij}^r + 2\sigma_{ij}^r \\ \phi & I_{ij} > \mu_{ij}^r + 2\sigma_{ij}^r \end{cases} \quad (4)$$

The neighbor mismatch penalties C_{ij}^h and C_{ij}^v offer the opportunity to employ the third strategy mentioned above, incorporating the Canny edge map. (Recall that Canny [6] first smooths the image with a Gaussian filter of small radius σ_E , then finds edges at local directed maxima of the image gradient, choosing to retain only those edges selected using a hysteresis procedure with two thresholds t_{hi} and t_{lo} .) The algorithm sets the mismatch penalties to a uniform value c everywhere except between pixels where a Canny edge has identified a likely discontinuity. To be more specific, Canny identifies pixels as edges, while Equation 1 requires locating discontinuities in the connections *between* pairs of pixels. To address this discrepancy, the formulation below zeros out the discontinuity penalty between Canny edge pixels and their brighter neighbors, effectively choosing to include the Canny edge pixels within the inked area. The opposite choice would also be self-consistent, but would inhibit detection of single pixel width strokes.

$$C_{ij}^h = \begin{cases} 0 & \text{if } E_{ij} \wedge (I_{ij} < I_{i+1,j}) \\ 0 & \text{if } E_{i+1,j} \wedge (I_{ij} \geq I_{i+1,j}) \\ c & \text{otherwise} \end{cases} \quad (5)$$

$$C_{ij}^v = \begin{cases} 0 & E_{ij} \wedge I_{ij} < I_{i,j+1} \\ 0 & E_{i,j+1} \wedge I_{ij} \geq I_{i,j+1} \\ c & \text{otherwise} \end{cases} \quad (6)$$

The choice of constant discontinuity penalties everywhere except at edges deserves a note. One might imagine using a penalty that varies continuously according to the similarity in intensity between the neighbors. Empirically this approach seems less successful, perhaps because it actually gives little guidance about the best precise location of the ink-background transition: the intensity differences tend to be fairly large everywhere within a few pixels of the actual boundary, and thus it becomes too easy to choose the wrong location.

2.1 Parameters

The algorithm described above includes six free parameters, of which only two strongly influence the binarization outcome and require varying settings for different images. The four less important parameters can be set to a constant with negligible consequences for all images tested, either because they do not strongly influence the result or because the best value appears not to fluctuate much for different images. Thus finding automatic values for the two important parameters suffices to create a method that can be applied as a “black box” with little or no parameter testing required.

Two of the less important parameters appear in Equation 4: r and ϕ . Of these, r must be large enough to encompass at least a few background pixels, and thus should be set to some value larger than the expected stroke width. ϕ can be any sufficiently negative value. No attempt is made to optimize these parameters, and the experiments all use $r = 20$ and $\phi = -500$, for images with grayscale intensity in the range from 0 to 255. The remaining less important parameters are two of the three that control the Canny edge detection algorithm, namely the lower of the two edge detection thresholds t_{lo} and the radius of smoothing applied prior to edge detection σ_E . Bad values for these parameters can certainly hurt binarization quality, but fortunately the experiments will show that there exists a uniform setting that works on all test images with near-optimal results.

The two important parameters that remain interact with each other to determine the final binarization result. Most crucial is c , whose magnitude determines the relative balance between data fidelity and regularization. The high Canny edge detection threshold t_{hi} matters most when c is large because the detection or nondetection of an edge can determine whether an entire ink component appears or disappears in the minimal energy solution. The high Canny threshold plays a particularly important role in images that exhibit ink bleeding through from the opposite side of the paper: since the bled ink tends to have weaker edges, fortuitous thresholding can eliminate most of the false ink components.

Early experiments with this binarization scheme,

and indeed for most approaches to binarization, have relied on choosing parameter values that work reasonably well across a wide range of images. This can be achieved by searching for the best values on a training set of images similar to the ones that will be used for testing, and yields good results [11]. However, even globally optimal parameter values embody some compromise on individual images, and tuning the settings to the image can offer significant gains. The next section explains how to do this.

2.2 Automatic Parameter Tuning

To understand how the crucial c parameter can be set automatically, it is worthwhile to examine the behavior of the base binarization algorithm under a range of different settings. Figure 1 shows the binarization results on a small image patch for a logarithmically increasing set of values. When c is very low, the binarization looks like a simple sign operator on the Laplacian, with many small noisy components. As c increases the noise components become more consolidated and many disappear. For a range of middle values of c , the binarization stays fairly stable, with only small changes as c increases. At the highest values, the result becomes unstable again as large ink components disappear, and occasionally join or have voids filled in. As c goes to infinity, the binarization will tend toward the majority label in each edge-isolated component.

The region of stable results at middle levels of c is significant, and turns out to show up consistently on ink-and-paper documents with reasons subject to explanation. To visualize the phenomenon, define the normalized *binarization instability* $\xi_\nu(c)$ in terms of $\Delta(B^c, B^{\nu c})$ the fraction of pixels that change labels between two values of c differing by a factor ν .

$$\Delta(B, B') = \sum_{i=0}^m \sum_{j=0}^n (B_{ij} \neq B'_{ij}) \quad (7)$$

$$\xi_\nu(c) = \frac{\Delta(B^c, B^{\nu c})}{mn(\ln(\nu c) - \ln(c))} \quad (8)$$

In the equation above, B_{ij}^c is the binarization label at pixel (i, j) using mismatch penalty c , and the Boolean

operator \neq converts to 0 or 1 in the customary fashion. As Figure 2 shows, the general shape of the instability curve is an intrinsic property of the image that does not depend on the exact value of ν chosen; thus in most cases ξ can drop its subscript. Histograms provide a useful analogy: although the details vary slightly, a histogram made using a sufficient number of samples from the same distribution always takes on the general appearance of that distribution for any reasonable choice of bin size. Indeed, the instability plots may be seen as histograms of pixel label transition events in respect to c , with logarithmically increasing bin sizes.

Figure 3 shows as solid lines the instability plots for several document images. Although the stable zone differs somewhat in appearance and location in each case, its nearly universal presence is striking. Yet in hindsight, the observation of such a stable zone at middling c values should not offer such a surprise. The twin peaks on either side arise from understandable mechanisms that apply for all document images, and thus the absence of a stable zone for any particular image would arise from exceptional circumstances.

The c parameter controls the incentive for neighboring patches to share the same label despite a data fidelity term indicating otherwise. The left peak in the instability curve appears at low levels of c , as it becomes large enough to overcome small noise fluctuations and label most large homogeneous foreground and background regions correctly. On the other hand, the right peak appears when c becomes large enough to force a uniform label even across regions of different underlying ground truth. Unless the noise amplitude exceeds the contrast between foreground and background, a zone of stability will normally separate the two. Furthermore, this stability zone coincides with binarizations that adhere to the underlying image structure while ignoring noise. Figure 3 shows as dotted lines the binarization error (based on the F-measure, as defined in Equation 14 below), which tends to reach a minimum at or near the lowest point in the instability curve. Computing the instability curve does not require access to the ground truth, yet observing the low point of the stable region reveals a value of c that typically achieves low ground-truth binarization error.

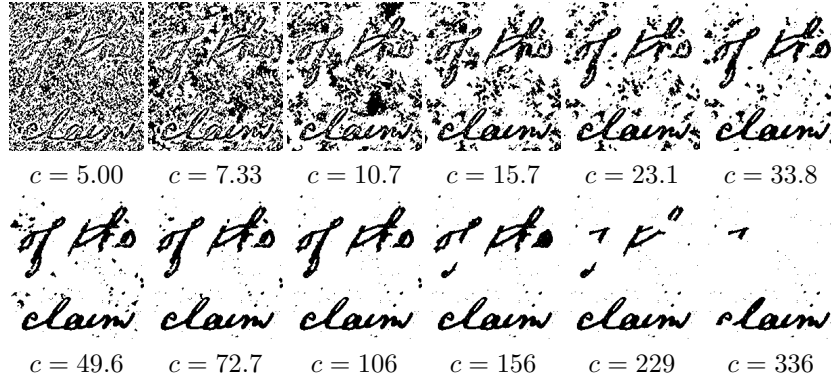


Figure 1: Binarization results for different values of c on one portion of a document image. A range of stability for intermediate c values corresponds to the most accurate result.

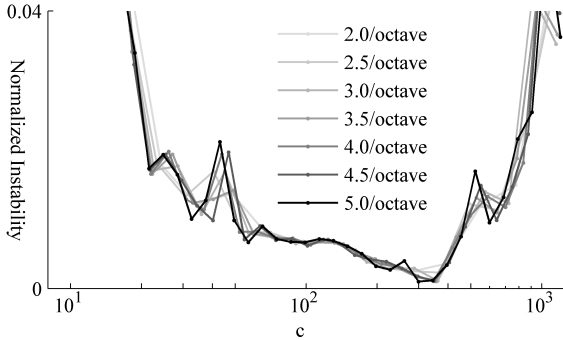


Figure 2: Binarization instability measured with varying step factor in c and normalized by the log of the step factor. The shape (and thus the location of the minimum) depends very little on the step factor chosen.

The link between the instability minimum and the binarization error minimum is at heart merely an empirical observation, but the philosophical considerations just given provide justification for the belief that it will apply to most document images. The algorithms developed below rest upon two related hypotheses, which the experiments in Section 3 largely bear out: First, the two instability peaks described above will reliably appear and can be consistently identified. Second, the point of maximum stability between the identified peaks will yield near-minimal binarization error. The truth of these hypotheses will depend upon the circumstances which produced a given document image. For example some documents include multiple types of markings at different contrast levels, such as those where ink bleeds through from the reverse side. These may muddy the situation by generating additional instability peaks, but in most cases the extra peaks don't alter the result greatly because they tend to be more diffuse and thus less prominent. Figure 5 below describes a few more difficult cases, including unusual images where the optimal c varies widely across the page. Nevertheless the experiments indicate that such problems remain mostly limited to unusual situations.

The parameter-setting technique for c summarized as Algorithm 1 computes curves like those in Figure 3 and chooses c at the minimum point between the two

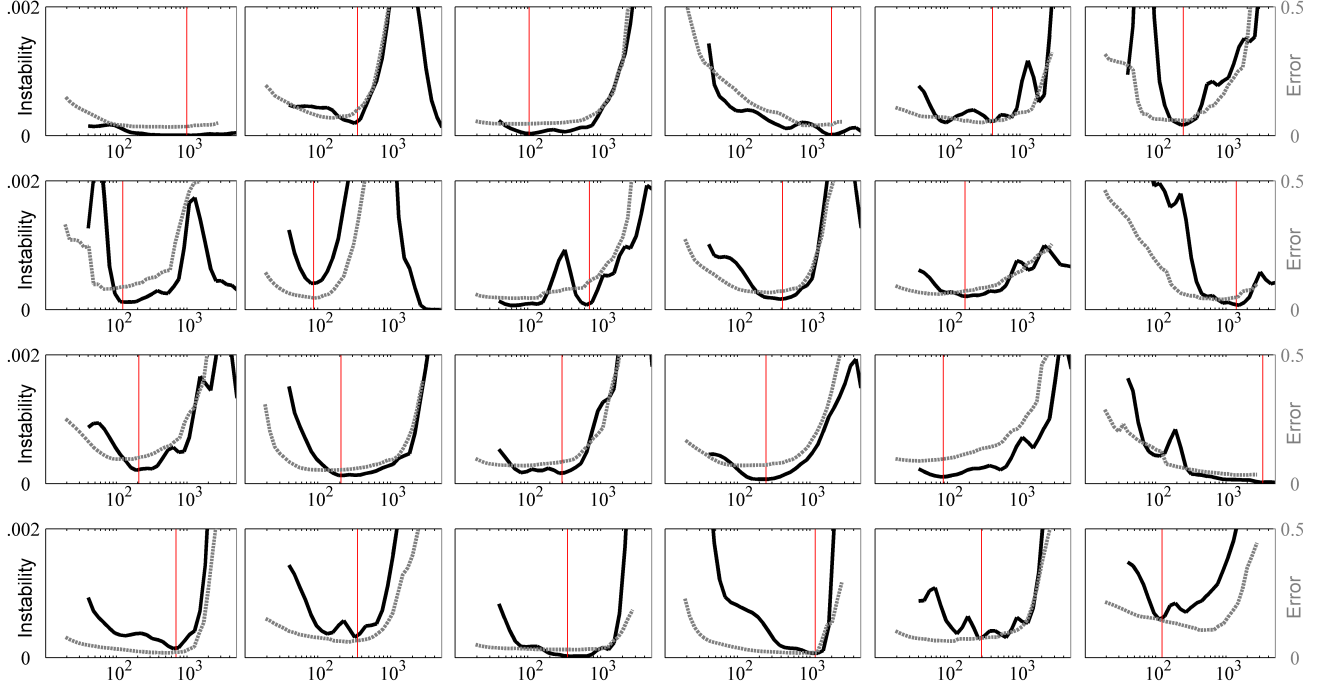


Figure 3: Binarization instability and error vs. c for the DIBCO 2009 and H-DIBCO 2010 images. The solid curve shows the fraction of the image area whose label changes between successive values of c . A vertical line identifies the value of c chosen by Algorithm 1 using this curve. The broken curve shows the error (difference from 1.0) of the the F-measure of the binarization. The horizontal axis shows c on a log scale. Images are given left to right in the order of Table 1.

peaks on either side. Minor parameter choices introduced here are the logarithmic separation between c values, the range of values to scan, and the amount of smoothing applied to the instability curve. Choosing the logarithmic separation is somewhat equivalent to choosing a histogram bin size; as Figure 2 demonstrates, the exact step factor chosen does not matter so long as it is not so large as to obscure the important minima and maxima, nor so small that there will often be no change observed from one value to the next. The experiments herein use 4 points per octave, or approximately $\nu = 1.19$. The other two choices may also be made in ways that affect the outcome minimally if at all. The range of c values scanned should be large enough to include any likely optimum, and the smoothing should even out small fluctuations in the curve without altering the overall shape significantly. The experiments in Section 3 use a unit Gaussian for smoothing, and values of c ranging from 20 to 5120.

Algorithm 1 Tune c to image I given t_{hi} , t_{lo} , and σ_E

```

for  $i = 0$  to 28 do
   $c_i \leftarrow 40 \cdot 2^{i/4}$ 
   $B_i \leftarrow \text{Binarize}(I, c_i, t_{hi}, t_{lo}, \sigma_E)$ 
end for
for  $i = 0$  to 27 do
   $D_i \leftarrow \Delta(B_i, B_{i+1})$ 
end for
 $G_0, G_1, \dots, G_5 \leftarrow 0.02, 0.13, 0.35, 0.35, 0.13, 0.02$ 
  {Gaussian of radius 1}
 $D_{-3}, D_{-2}, D_{-1} \leftarrow 0, 0, 0$ 
 $D_{28}, D_{29}, D_{30} \leftarrow 0, 0, 0$ 
for  $i = 0$  to 28 do
   $D'_i \leftarrow \sum_{j=0}^5 D_{i+j-3} G_j$ 
end for { $D'$  is now a smoothed version of  $D$ }
 $q, r, s \leftarrow \arg \min_{q,r,s} D'_q - 2D'_r + D'_s : q < r < s$ 
 $c \leftarrow c_r$ 

```

Picking c only partially solves the parameter-setting problem, since the high Canny threshold remains to be chosen. This upper threshold on edge detection rewards careful tuning, because properly chosen it can weed out the more diffuse edges of ink

bleeding through from the back side of the document, while retaining the sharper edges of ink shapes on the front side. Fortunately, the same stability criterion that guided the choice for c also serves to pick a good value within the likely range for t_{hi} . Algorithm 2 details the exact technique used in the experiments. Note that c and t_{hi} are set sequentially, rather than at the same time. The algorithm computes the best value of c for each possible t_{hi} , and then chooses the (c, t_{hi}) pair with greatest stability with respect to changes in t_{hi} . This simplifies each step in the process and makes it easier to find a near-optimum solution.

Algorithm 2 Tune t_{hi} to image I given t_{lo} , and σ_E

```

for  $i = 0$  to 10 do
   $\tau_i \leftarrow 0.15 + 0.05i$ 
   $c_i \leftarrow \text{TuneC}(I, \tau_i, t_{lo}, \sigma_E)$  {Algorithm 1}
   $B_i \leftarrow \text{Binarize}(I, c_i, t_{hi}, t_{lo}, \sigma_E)$ 
end for
for  $i = 0$  to 9 do
   $D_i \leftarrow \Delta(B_i, B_{i+1})$ 
end for
 $G_0, G_1, \dots, G_5 \leftarrow 0.02, 0.13, 0.35, 0.35, 0.13, 0.02$ 
  {Gaussian of radius 1}
 $D_{-3}, D_{-2}, D_{-1} \leftarrow D_2, D_1, D_0$ 
 $D_{10}, D_{11}, D_{12} \leftarrow D_9, D_8, D_7$ 
for  $i = 0$  to 10 do
   $D'_i \leftarrow \sum_{j=0}^5 D_{i+j-3} G_j$ 
end for { $D'$  is now a smoothed version of  $D$ }
 $k \leftarrow \arg \min_{j=0}^{10} D'_j$ 
 $t_{hi} \leftarrow \tau_k$ 
 $c \leftarrow c_k$ 

```

Arguments given previously in this section suggested reasons why a stability criterion might work well for choosing c . It is less clear why the same approach works for choosing an edge intensity threshold. Nevertheless a zone of stability with respect to both c and t_{hi} does seem to distinguish the best binarizations, as illustrated in Figure 4 for several representative images. Although the location of the best parameter setting varies widely between the images, each case displays a striking qualitative correlation between the binarization error and the two-way sta-

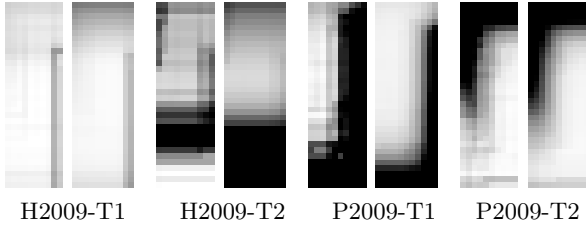


Figure 4: Binarization instability (left in pair) and error (right in pair) visualized with respect to varying values of c (vertical axis; logarithmic range from 20 to 5120) and t_{hi} (horizontal axis; linear range from 0.15 to 0.65) for four representative images. Despite the very different patterns on display, the parameter neighborhoods with lower error coincide in each instance with higher stability (both indicated by lighter shades). The converse does not always hold: the second image shows two areas of stability, with only the more central one corresponding to low error. The second, more peripheral stability zone is rejected by Algorithm 1.

bility, measured using the simple unnormalized formula below.

$$\begin{aligned} \zeta_{\nu,\tau}(c, t_{hi}) = & \frac{1}{4}[\Delta(B^{c,t_{hi}}, B^{\nu c, t_{hi}}) \\ & + \Delta(B^{c,t_{hi}}, B^{c/\nu, t_{hi}}) \\ & + \Delta(B^{c,t_{hi}}, B^{c, t_{hi}-\tau}) \\ & + \Delta(B^{c,t_{hi}}, B^{c, t_{hi}+\tau})] \end{aligned} \quad (9)$$

Other DIBCO images not shown behave similarly. Full explanation of this phenomenon lies beyond the scope of this paper, which merely observes and seeks to exploit the pattern.

2.3 Computational Efficiency

Completing Algorithm 2 requires 363 trial binarization computations (33 candidates for c times 11 candidates for t_{hi}) and thus runs much more slowly than a single binarization with static parameter settings. However, two modifications to the algorithm can substantially mitigate the speed difference.

The first takes advantage of the fact that the binarization result does not usually change substantially between successive values of c . Data structures built to minimize Equation 1 for one value of c can be modified and reused with a similar c value, achieving noticeable economies. This strategy, introduced by Boykov and Kolmogorov [5], minimizes Equation 1 by finding the minimum cut on a graph derived from the image. The method grows search trees from both the source and the sink of the image graph to help find augmenting paths, and it turns out that the trees grown for one value of c can be largely reused at nearby values to realize significant time savings. Empirical tests of this implementation show that it can speed computation by a factor of fifteen or more. In other words, it computes all 33 trial binarizations required for one iteration of Algorithm 1 in little more than the time that it would normally take to compute just two or three.

The second modification relies on a heuristic trick, and stems from the observation that the presence or absence of ink bleed-through significantly influences the optimal value of t_{hi} . Images showing bleed-through typically require a larger t_{hi} to avoid identifying the spurious bled ink components as real. By contrast, in images without bleed-through the algorithm achieves its best results with a lower t_{hi} , because this allows it to detect fainter edges. Thus it appears that an explicitly bimodal algorithm allowing only two possible values $\{\tau_1, \tau_2\}$ for t_{hi} might still perform well as compared with Algorithm 2, which tests eleven. The experimental results below confirm this hypothesis.

Detecting the presence or absence of bleed-through in a document is a nontrivial task in the general case, so it might seem impossible to determine which of even two candidate t_{hi} values to use. Fortunately, an extension of the stability criterion provides a simple heuristic that does fairly well empirically. In addition to the two candidate values, Algorithm 3 computes the binarization using their mean value $\tau_0 = (\tau_1 + \tau_2)/2$. This midpoint binarization can be compared to the other two, and the closer is judged the more stable and thus selected for the final result.

Algorithm 3 computes results for only three t_{hi} values and thus runs more than three times as fast as

Algorithm 2. The metaparameters τ_1 and τ_2 are chosen for best performance on a training set; analysis of the DIBCO 2009 and H-DIBCO 2010 image sets indicates that values of 0.25 and 0.50 respectively achieve the best results across the full set. (Using only 23 of the 24 images in cross-fold training occasionally yields other values.) Employing a training set to choose τ_1 and τ_2 diminishes the automatic nature of the full algorithm, but represents a tradeoff made in this particular algorithm for the sake of computation speed.

Algorithm 3 Pick t_{hi} from $\{\tau_1, \tau_2\}$ for image I given $\tau_1, \tau_2, t_{lo},$ and σ_E

```

 $\tau_0 \leftarrow (\tau_1 + \tau_2)/2$ 
 $c_0 \leftarrow \text{TuneC}(I, \tau_0, t_{lo}, \sigma_E)$ 
 $c_1 \leftarrow \text{TuneC}(I, \tau_1, t_{lo}, \sigma_E)$ 
 $c_2 \leftarrow \text{TuneC}(I, \tau_2, t_{lo}, \sigma_E)$ 
 $B_0 \leftarrow \text{Binarize}(I, c_0, \tau_0, t_{lo}, \sigma_E)$ 
 $B_1 \leftarrow \text{Binarize}(I, c_1, \tau_1, t_{lo}, \sigma_E)$ 
 $B_2 \leftarrow \text{Binarize}(I, c_2, \tau_2, t_{lo}, \sigma_E)$ 
 $D_1 \leftarrow \Delta(B_0, B_1)$ 
 $D_2 \leftarrow \Delta(B_0, B_2)$ 
if  $D_1 < D_2$  then
   $t_{hi} \leftarrow \tau_1$ 
else
   $t_{hi} \leftarrow \tau_2$ 
end if

```

These two innovations in combination mean that a result originally requiring 363 trial binarizations to compute may be reached in the time required for just eight or nine. This is still slower than using static parameter values, but investing the extra time may be worthwhile for more accurate results. Alternately, in situations where a number of similar documents must be binarized, the parameter tuning may be run for a few trial pages to find appropriate values, which are then set statically for the remainder of the set.

All the algorithms display more or less linear dependence of computation time on the number of pixels in the image. Executing Algorithm 2 takes an average of 892 seconds per megapixel on a 2.4 GHz Xeon processor running as a single thread (without parallelism). By contrast, Algorithm 3 runs in 18.1 sec-

onds per megapixel under the same conditions, while a single execution of the base algorithm under static parameters takes 2.12 seconds per megapixel.

2.4 Further Algorithmic Variants

For best results, many binarization methods compute an initial labeling using some base technique and then apply one or more postprocessing algorithms to improve it. For example, Su et al [22] remove components of three pixels or less. More complicated modeling and classification algorithms can identify and remove noise components while retaining real ink [1]. The unknown pixel classification of Lelore and Bouchara [13] may also be viewed as post-processing of a sort.

Similar techniques can be applied to the results from the method described herein. Indeed, both the three-pixel component filter and a more complex classification-based approach reminiscent of Agrawal & Doermann have been tested informally and found to reduce binarization error for the DIBCO test images, relative to ground truth. Unfortunately gains in training can come at the potential cost of a loss of generality, in practice. This suspicion is borne out by the DIBCO 2011 contest results. The two algorithms that ranked in first and second place according to the contest scoring methodology perform very well on most of the test images, but fail severely on one or two examples (PR6 and PR7). One might view this as a form of overfitting the problem: these algorithms are specialized to do very well on images that match the expectations of the designers, but cannot handle the full range of images that might be encountered “in the wild”. This paper aims to develop binarization techniques that work automatically and reliably on as wide a range of images as possible. Thus the experiments eschew post-processing heuristics of the type described above because they might decrease error on a given test set at the cost of generality. Such tricks are nevertheless worth mentioning because they may prove useful in whenever the images to be binarized are amenable.

3 Experimental Results

The DIBCO events have provided a platform for side-by-side comparison of different binarization results to a ground truth created by a team of human referees. Three contests have been held to date: two included both handwritten and printed document images, and one consisted of just handwritten documents. Images from the first two contests were used to develop the algorithms described in this paper, with the third set held out for *post-hoc* testing.

A comprehensive survey of the parameter space has been prepared for this paper, providing the context necessary to evaluate the effectiveness of the automatic parameter tuning algorithms. It samples the 4-dimensional parameter space on a regular grid, surrounding and including the region found to produce the best binarizations in prior work [11]. The specific values tested for each parameter are:

$$t_{hi} \in T_{hi} \equiv \{0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65\} \quad (10)$$

$$t_{lo} \in T_{lo} \equiv \{0, 0.05, 0.10, 0.15, 0.20\} \quad (11)$$

$$\sigma_E \in S_E \equiv \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\} \quad (12)$$

$$c \in C \equiv \{40 \cdot 2^{i/4} \parallel i = 0..28\} \quad (13)$$

Results from the parameter survey can answer several interesting questions. They reveal both the best possible performance with a fixed set of parameters over all images (*static* parameters), and the best possible performance in hindsight using the optimal parameter settings for each individual image (*ideal* parameters). The difference between these two values is the space where parameter-tuning algorithms operate: they cannot aspire to do better than the ideal, although they may certainly do worse than the best static setting. If the difference between the static and ideal numbers is small, then parameter tuning may not be worthwhile because any possible gains are limited.

A number of different statistics can measure and compare the effectiveness of binarization algorithms. The DIBCO events compute some half-dozen numbers, with four used to determine the winners. The initial experiments herein focus on the F-measure,

defined in Equation 14, because its interpretation is simple and it serves as a proxy for the other measures of binarization quality. An F-measure of 100% represents perfection. In this paper the term *error* refers to the difference between the observed F-measure and a perfect 100% score.

$$F = \frac{2 \cdot R \cdot P}{R + P} \quad (14)$$

where the recall R and precision P are defined in terms of the number of true positive pixels N_{TP} , the number of false positive pixels N_{FP} , and the number of false negative pixels N_{FN} in the binarization B as compared to ground truth G .

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad (15)$$

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (16)$$

Table 1 summarizes binarization results on the DIBCO 2009 and H-DIBCO 2010 images for various parameter-tuning algorithms. The static parameters for the experiments represented in this table are determined in leave-one-out style. In other words, for the i th image, the computation uses whatever static parameter setting would have given the best mean result on the other 23 images. Table 1 shows the range of parameter values chosen via this technique for the various algorithms. (Note that even algorithms with static parameter settings may show a range in Table 1 if the static values computed differ for various leave-one-out training folds.)

The second column of the table gives the F-measure when all parameters are static; this is the baseline algorithm [11]. The third column gives the result achieved with an oracle that knows the ideal values of c and t_{hi} for each image; this represents the best possible outcome achievable by tuning the two parameters, but is not realizable in practice without prior access to ground truth. The ideal parameter settings reduce the overall error by more than one-third compared to static parameters, showing that tuning methods have significant potential if they can come close to this ideal. The table does not show results from an oracle that optimizes all four parameters. If available, such an oracle could achieve a

mean F-measure of 95.3% across all 24 images, hardly more than the two-parameter oracle. This confirms the conjecture that tuning all four parameters offers little reward in exchange for the risk.

The remaining columns show results for the various parameter-tuning strategies described in Section 2. Column four tunes c using Algorithm 1 and employs static settings for t_{hi} , t_{lo} , and σ_E . It does somewhat better than the all-static settings, but still shows room for improvement. Column five tunes both c and t_{hi} using Algorithm 2 and employs static settings for t_{lo} , and σ_E . This method achieves 83% of the level of improvement made by the oracle. The final column show the results of the two-state tuning for t_{hi} described in Algorithm 3. This method does nearly as well as the fully tuned t_{hi} , despite its much faster computation time.

Table 3 shows the results for the DIBCO 2011 images, used as a hands-off test set. Unlike the previous table, these are not trained leave-one-out style. To ensure an unbiased test, the algorithms were fixed before any results were examined, and static parameter values are set using the first two sets of contest images. The numbers here look more diverse because some of the new images are more extreme and therefore more challenging than the training images. However, the general picture remains the same: tuning c gives better results than static parameter settings, and tuning both c and t_{hi} improves the binarization quality still further for most images. A few exceptions lower the overall average, particularly H2011-6. Close inspection of this case reveals that the stability curve for t_{hi} on this image contains two basins of stability, and the minimization criterion chooses the wrong one by a slight margin. The two-state tuning does not suffer from this problem, because it selects t_{hi} from a more constrained set of choices. In hindsight this may be an unforeseen practical advantage of Algorithm 3.

Despite the success of the tuning methods described, the best F-measure achieved by tuning still lies farther from the oracular best for the DIBCO 2011 images than in the experiments with earlier images. This reflects the diversity of the test set. Some of the images represent adverse examples for the stability criterion advocated in this paper; the two worst

Table 1: F-Measure for DIBCO 2009 and H-DIBCO 2010 Images

Image	Static	Oracle	Alg. 1	Alg. 2	Alg. 3
H2009-T1	96.4	96.6	96.6	96.1	96.6
H2009-T2	93.3	93.4	92.5	93.1	93.1
H2009-1	92.7	95.9	95.4	95.8	95.8
H2009-2	90.1	96.4	88.8	96.2	95.7
H2009-3	94.7	95.6	92.0	94.8	95.1
H2009-4	92.1	94.7	93.9	94.6	94.5
H2009-5	86.5	92.7	91.9	92.3	92.6
H2010-1	93.7	96.2	95.0	95.4	96.0
H2010-2	89.0	96.1	95.4	95.7	95.4
H2010-3	94.4	94.7	93.3	94.7	94.7
H2010-4	92.9	94.4	93.5	93.8	93.9
H2010-5	92.9	96.5	96.1	96.5	96.4
H2010-6	90.9	91.2	90.4	90.9	90.9
H2010-7	95.0	95.2	94.8	95.0	95.1
H2010-8	92.6	93.7	92.2	93.5	93.4
H2010-9	92.6	93.8	92.9	92.1	93.5
H2010-10	88.8	92.7	90.8	92.5	87.2
P2009-T1	88.5	97.4	96.7	97.3	97.2
P2009-T2	98.0	98.5	98.1	98.4	98.5
P2009-1	93.9	94.3	90.4	94.0	94.2
P2009-2	96.8	96.9	96.8	96.8	96.9
P2009-3	97.6	98.4	98.2	98.2	98.3
P2009-4	93.5	93.8	92.8	92.9	92.8
P2009-5	88.6	91.5	84.7	91.2	84.3
Hand	92.3	94.7	93.3	94.3	94.1
Print	93.9	95.8	93.9	95.6	94.6
All	92.7	95.0	93.5	94.7	94.3

Table 2: Parameter Ranges for Table 1

	Static	Oracle	Tune c	Tune t_{hi}	Dual
c	160	47.6–1280	56.6–3044	80–1076	67.3–3044
t_{hi}	0.4	0.15–0.65	0.2–0.3	0.15–0.6	0.25 or 0.5
t_{lo}	0.1	0–0.2	0.1–0.15	0.1	0.1–0.15
σ_E	0.6	0.3–0.8	0.5–0.6	0.6	0.5–0.6

Table 3: F-Measure for DIBCO 2011 Images

Image	Static	Oracle	Tune c	Tune t_{hi}	Dual
H2011-1	77.3	93.9	88.2	90.2	90.0
H2011-2	97.4	97.5	96.6	97.3	97.4
H2011-3	93.2	95.2	94.9	94.5	93.5
H2011-4	92.5	92.7	92.5	92.0	92.0
H2011-5	92.4	97.1	96.2	97.1	96.2
H2011-6	88.2	94.4	91.1	60.6	92.1
H2011-7	87.6	92.5	92.3	89.6	89.8
H2011-8	95.3	95.7	94.9	95.3	95.4
P2011-1	93.0	95.3	94.3	95.1	94.7
P2011-2	77.6	90.8	73.5	71.5	72.3
P2011-3	95.6	96.6	96.3	96.4	96.4
P2011-4	95.0	95.2	94.5	95.2	95.1
P2011-5	94.8	94.9	93.7	94.2	94.6
P2011-6	66.7	93.1	89.9	86.0	86.3
P2011-7	90.2	93.3	89.7	93.1	93.1
P2011-8	90.3	92.2	88.5	91.3	91.5
Hand	90.5	94.9	93.3	89.6	93.3
Print	87.9	93.9	90.0	90.3	90.5
All	89.2	94.4	91.7	90.0	91.9

appear in Figure 5.

3.1 Comparison with Other Algorithms

Results available for the DIBCO 2011 image set make possible a quantitative comparison between the the algorithms developed herein and other recent work. The DIBCO 2011 contest used three additional primary criteria in addition to the F-measure. Of these, the peak signal-to-noise ratio (PSNR) correlates fairly strongly with the F-measure. If B is the binarization and G the ground truth binary image,

$$PSNR = -10 \log(\Delta(B, G)) \quad (17)$$

Misclassification penalty metric (MPM) penalizes errors according to their distance from the ink boundary. Define D_{ij} as the distance of pixel (i, j) from this boundary in the ground truth.

$$MPM = \frac{\sum_{i=0}^m \sum_{j=0}^n D_{ij}(B_{ij} \neq G_{ij})}{2 \sum_{i=0}^m \sum_{j=0}^n D_{ij}(1 - G_{ij})} \quad (18)$$

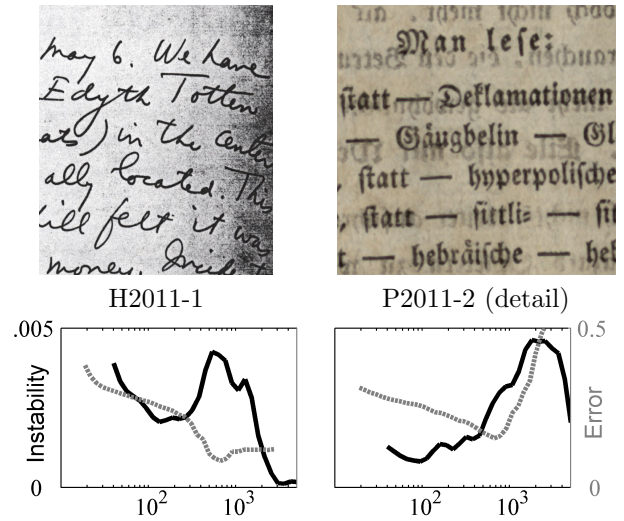


Figure 5: The DIBCO 2011 test images contain several challenging examples. Both these images have unusual stability curves. At left, the high-amplitude noise on the right side generates instability at values of c that are optimal for the rest of the page. At right, the stability criterion chooses a c value that preserves the bleed-through ink.

Distance reciprocal distortion metric (DRD) attempts to account for human perceptual salience by weighting errors according to pixel values in a local 5×5 neighborhood, using weight matrix W computed as the normalized inverse distance.

$$DRD = \frac{1}{N_8} \sum_{i=0}^m \sum_{j=0}^n \Gamma_{ij}(B_{ij} \neq G_{ij}) \quad (19)$$

$$\Gamma_{ij} = \sum_{h=-2}^2 \sum_{k=-2}^2 W_{hk}(B_{ij} \neq G_{i+h,j+k}) \quad (20)$$

N_8 divides G into 8×8 blocks and counts the number that are non-uniform. Note that better binarizations will have higher F-measure and PSNR, but lower MPM and DRD. For further details on these metrics, please read the DIBCO 2011 contest report [19].

Algorithm 3 is similar to entry #11 that was actually submitted to the DIBCO 2011 contest. That entry was based on a less comprehensive survey of the parameter space and thus differs in the setting of the static parameter values and in the range of c values tested. Entry #11 nevertheless had the highest mean F-measure across all the images, the lowest mean peak signal-to-noise ratio, the best mean misclassification penalty metric, and the fifth-best distance reciprocal distortion metric. Strangely, despite these successes, entry #11 ranked only third in the official DIBCO 2011 contest results, and the two other entries that placed first and second had mean scores over the full image set that were uniformly worse on all four measures. The scoring methodology of the contest chose the winner based upon summed ranks over all images on various quality measures, rather than directly on the mean metric scores. The two higher-rated methods ranked well on most images, and while they conversely showed extremely poor performance on a few of the printed documents this ultimately had limited effect on the final score [19]. With rank-based scoring the comparison of one method relative to another is affected by the performance (and presence or absence) of all the other methods in the contest.

Table 4 compares the results of the algorithms developed in this paper with other recent methods on

the DIBCO 2011 image set. The first section shows selected entries from the contest (those that beat Entry #11 on at least one measure), the second section shows other recent algorithms², and the third section shows the algorithms described in this paper, trained using the DIBCO 2009 and H-DIBCO 2010 images for a fair comparison. The table gives the mean result for each of the four metrics averaged across all the test images, and also the rank score total used to determine the DIBCO 2011 winner. For algorithms not entered in the contest, the rank score is simulated by treating the algorithm in question as an additional entrant, indicated with an asterisk in the table. Of course, adding a contestant changes the rank scores of the actual entrants, making them worse whenever the new entry does better on an image. The modified scores of existing entries are not shown in the table, but the information is accounted for in the parenthetical overall rank shown in the rightmost column. As the table reveals, either of Algorithms 2 or 3 would have won the contest on rank scoring, and also easily place first in mean score on each of the four quality measures as compared with all previous algorithms.

4 Conclusion

Tuning parameters offers both risks and potential rewards. Choosing the parameter values that perform best for a given image can lower binarization error by substantial amounts as compared with a static setting suitable for generic images. On the other hand, poorly chosen parameter values can sabotage the result: the potential losses in binarization quality generally dwarf the potential gains. Thus one must be careful to tune only where there is reasonable confidence of success.

This paper has introduced a stability heuristic criterion that helps to choose suitable parameter values for individual images. The approach hypothesizes that good parameter values are marked by low variability in the binarization solution with respect to changes in the parameter values. This criterion leads to an algorithm that successfully picks good c

²The numbers for Lelore & Bouchara exclude image PR6 because its results were not available.

Table 4: Algorithms compared by scores on the DIBCO 2011 image set.

Method	F	PSNR	MPM $\times 10^{-2}$	DRD $\times 10^{-3}$	Ranks
Entry 11	88.7	17.8	5.36	8.67	429 (3)
Entry 10	80.9	16.1	104.48	64.42	309 (1)
Entry 8	85.2	17.2	15.66	9.07	346 (2)
Entry 3	85.1	16.4	5.88	8.09	649 (12)
Entry 4	85.2	16.6	6.28	4.43	489 (5)
Entry 6	83.6	16.7	8.08	4.57	470 (4)
Entry 14	78.0	14.9	7.62	7.35	835 (17)
Gatos [9]	84.3	16.3	6.39	6.08	663 (11)*
Lelore [12]	56.2	12.5	2.97	13.30	811 (17)*
Lu [14]	79.7	15.5	39.67	21.47	579 (8)*
Su [22]	87.8	17.7	5.38	4.65	435 (3)*
Static	89.2	18.2	9.05	5.76	438 (3)*
Alg. 1	91.7	19.2	4.43	3.40	322 (2)*
Alg. 2	90.0	19.1	3.89	3.78	323 (1)*
Alg. 3	91.7	19.3	3.87	3.48	317 (1)*

values on all the images tested. A similar heuristic applied to the choice of t_{hi} also chooses good parameter values in most cases, although serious failures were observed with this technique for a handful of cases. The most successful method tested uses the stability criterion to choose c , and selects t_{hi} from a constrained set of two possible values. This approach delivers a substantial fraction of the maximum gain possible from tuning both parameters, and can be computed at about one eighth the speed of a simple binarization with static parameters. Reference code for the technique will be available on the author’s web site.

The parameter survey results make it clear that for many images the tuning algorithms given herein come close to maximizing the potential of the base binarization algorithm. Further improvements in result quality will likely have to come through development of new base algorithms. Some of these new algorithms may be application-specific, whereas this work has striven for broad applicability. Whether the stability criterion that has proven so useful here

will apply to other approaches remains to be seen, and is a topic for future work. In any case, tuning parameters with the algorithms described herein substantially advances the current state of the art in document binarization, as evidenced by comparative results on the DIBCO 2011 test images.

Acknowledgment

The author thanks those who kindly shared results or implementation details of their algorithms for comparison purposes, including Basilis Gatos, Su Bolan, and Frédéric Bouchara.

References

- [1] Agrawal, M., Doermann, D.: Stroke-like pattern noise removal in binary document images. In: International Conference on Document Analysis and Recognition, pp. 17–21 (2011)
- [2] Badekas, E., Papamarkos, N.: Estimation of proper parameter values for document binarization. International Journal of Robotics and Automation, Vol. 24, No. 1, 2009 **24**(1), 66–78 (2009)
- [3] Bar-Yosef, I., Beckman, I., Kedem, K., Dinstein, I.: Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents. Int. J. Doc. Anal. Recognit. **9**(2), 89–99 (2007)
- [4] Barney-Smith, E.: An analysis of binarization ground truthing. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pp. 27–33. Boston (2010)
- [5] Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. on Pattern Analysis and Machine Intelligence **26**(9), 1124–1137 (2004)
- [6] Canny, J.: A computational approach to edge detection. IEEE Trans. on Pattern Analysis and Machine Intelligence **8**(6), 679–714 (1986)

- [7] Chen, Y., Leedham, D.: Decompose algorithm for thresholding degraded historical document images. *IEEE Proceedings on Vision, Image and Signal Processing* **152**(6), 702–714 (2005)
- [8] Dawoud, A.: Iterative cross section sequence graph for handwritten character segmentation. *IEEE Transactions on Image Processing* **16**(8), 2150–2154 (2007)
- [9] Gatos, B., Pratikakis, I., Perantonis, S.J.: Adaptive degraded document image binarization. *Pattern Recognition* **39**(3), 317–327 (2006)
- [10] Gatos, B., Pratikakis, I., Perantonis, S.J.: Improved document image binarization by using a combination of multiple binarization techniques and adapted edge information. In: *International Conference on Pattern Recognition*, pp. 1–4 (2008)
- [11] Howe, N.: A Laplacian energy for document binarization. In: *International Conference on Document Analysis and Recognition*, pp. 6–10 (2011)
- [12] Lelore, T., Bouchara, F.: Document image binarization using Markov field model. In: *International Conference on Document Analysis and Recognition*, pp. 551–555 (2009)
- [13] Lelore, T., Bouchara, F.: Super-resolved binarization of text based on FAIR algorithm. In: *International Conference on Document Analysis and Recognition*, pp. 839–843 (2011)
- [14] Lu, S., Su, B., Tan, C.L.: Document image binarization using background estimation and stroke edges. *International Journal on Document Analysis and Recognition* **13**(4), 303–314 (2010)
- [15] Mishra, A., Alahari, K., Jawahar, C.V.: An MRF model for binarization of natural scene text. In: *International Conference on Document Analysis and Recognition* (2011)
- [16] Niblack, W.: *An Introduction to Digital Image Processing*. Prentice-Hall, Englewood Cliffs, New Jersey (1986)
- [17] Otsu, N.: A threshold selection method from graylevel histogram. *IEEE Trans. on System, Man, Cybernetics* **19**(1), 62–66 (1978)
- [18] Peng, X., Setlur, S., Govindaraju, V., Sitaram, R.: Markov random field based binarization for hand-held devices captured document images. In: *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 71–76 (2010)
- [19] Pratikakis, I., Gatos, B., Ntirogiannis, K.: IC-DAR 2011 document image binarization contest (DIBCO 2011). In: *International Conference on Document Analysis and Recognition*, pp. 1506–1510 (2011)
- [20] Ramírez-Ortegón, M.A., Tapia, E., Ramírez-Ramírez, L.L., Rojas, R., Cuevas, E.: Transition pixel: A concept for binarization based on edge detection and gray-intensity histograms. *Pattern Recognition* pp. 1233–1243 (2010)
- [21] Sauvola, N., Pietikainen, M.: Adaptive document image binarization. *Pattern Recognition* **33**(2), 225–236 (2000)
- [22] Su, B., Lu, S., Tan, C.L.: Binarization of historical document images using the local maximum and minimum. In: *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 159–166 (2010)
- [23] Vonikakis, V., Andreadis, I., Papamarkos, N., Gasteratos, A.: Adaptive document binarization: A human vision approach. In: *2nd International Conference on Computer Vision Theory and Applications*, pp. 104–110. Barcelona (2007)