An Adaptively Iterative Method of Document Image Binarization

Ning Liu
Peking University
School of Mathematical Sciences
No.5 Yiheyuan Road, Beijing
China
liuning19880928@gmail.com

Guanxiang Wang
Peking University
School of Mathematical Sciences
No.5 Yiheyuan Road, Beijing
China
gxwang@math.pku.edu.cn

Caiping Lv
Peking University
School of Mathematical Sciences
No.5 Yiheyuan Road, Beijing
China
shualcp@126.com

Yu Wang
Peking University
School of Mathematical Sciences
No.5 Yiheyuan Road, Beijing
China
wangyu_amo@126.com

Abstract: Document image binarization is difficult when the image is affected by background noise or non-uniformly illuminated. In this paper, an adaptively method is proposed to address the above problems and thus get expected binarization results. This method begins by defining a filter window length with initialized stroke width, and then transforms the document image into a feature space with Gaussian kerneled Bhattacharyya distance and sets up a threshold with Ostu's method to temporarily binarize the image, and in the end, it uses the stroke width extracted from the newly obtained image to update the filter window length until the iteration convergent. The proposed method has been tested on the DIBCO(2009-2012) databases and get acceptable results on most images in these databases. In addition, expected results have been achieved when we tested this method on several other seriously noised and shadowed document images, even with various resolutions, which indicates our method is effective.

Key–Words: adaptive binarization, document image, Bhattacharyya distance, stroke width, run-length histogram

1 Introduction

Document image binarization is an important aspect of image processing and video processing, which is related to image segmentation and pattern recognition. Binarization technique has a wide range of applications such as optical character recognition (OCR) and content based image/video retrieval (CBIR/CBVR). Although there have already been some tools doing binarization, there are still some challenges when it comes to images acquired under non-uniform illumination conditions, images with additional noise such as blur and images of different resolutions resulted from various photographic equipments. As more and more document images produced on the Internet everyday, adaptive binarization methods are becoming increasingly important.

At present, there are mainly two kinds of binarization methods: global information based methods and local information based methods. Global methods find a single threshold value by a certain standard based on the gray levels of the image pixels. The classification may be based on histogram[1], clustering[2], entropy[3] and features[4]. Global thresholding methods are generally of high efficiency, and could achieve acceptable result when the gray levels of the text and background pixels are separable. But for document images suffering multiple degradations and non-uniform illuminations, global thresholding methods usually could not get expected results.

To overcome the drawback of global thresholding methods, various local thresholding methods have been proposed. In local thresholding methods, each pixel was binarized according to the information contained in the neighborhood of it. These methods are usually based on intensity information [5] or edge information[6]. And the intensity-based methods can be divided into two sub-groups: threshold-based methods[7, 8] and clustering methods[9]. The main drawback of local thresholding methods is high computational cost and high tendency to fail to extract the inner parts of thick characters.

In this paper, we introduce an adaptively iterative method to binarize document images. This method, based on stroke width information, transforms every pixel's local information into a feature space and uses Otsu's classification method on the feature space to distinguish the text pixels from background pixels. What's more important, is that this method could get stroke width information from the temparary binarizing result and could update the transformation and thus establish an iterative strategy. The feature space used in this paper is of local Bhattacharyya similarity with Gaussian kernel which improves the separability of the features greatly.

2 Proposed method

The binarization method discussed in this paper is a kind of local thresholding methods and more details will be introduced in this section.

For local thresholding methods, some kinds of filters are usually used to secure local features and thus the window size of the filter is of great importance. Actually, the edge of the binary result will be fuzzy when the window size of the filter is too big, while the inner part of the character will be hollowed out if the window size of the filter is too small. The reasonable definition of the window size of a filter, or the filter window length (FWL), should be closely related to stroke width (SW). We here take $FWL = 2*SW-1$ as in [10]. But unfortunately, we do not exactly know the stroke width before we make a proper binarization of the document image.

The most widely used method in extracting stroke width is Run-Length Histogram although there exist several other methods at present. For example, in paper[11], the authors use $Stroke\ Width\ Transform$ (SWT) as a local image operator which computes per pixel the width of the most likely stroke containing the pixel, where the Canny edge detector should be first used to collect the edge information of the images. Here we adopt Run-Length Histogram to extract stroke width.

The concept of run-length was proposed in the 1950s and has become the compression standard in fax transmissions and bitmap-file coding[12]. A run is defined as a string of consecutive pixels which have the same gray level of intensity along a specific linear orientation (typically in $0^o, 45^o, 90^o$, and 135^o).The length of the run is the number of repeating pixels in the run. For a given image, any element $p(i,j)$ of a run-length matrix is defined as the number of runs with pixels of gray level i and run length j. In a run-length matrix $(p_\theta(i,j))_{M\times N}$, M is the number of gray levels , N is the maximum run length and θ is the specific linear orientation. Then the run-length histogram(RLH) of an image is defined as a vector[13]:

$$H_\theta(j) = \sum_{i=1}^{M} p_\theta(i,j). \quad 1 < j < N \tag{1}$$

And the $stroke\ width$ (SW) of this image is defined as the run-length with the highest frequency in the run-length histogram excluding the unit run-length. That is,

$$SW = t,\ if\ H_{max}(t) = \max_{i\in I} H(i). \quad i \neq 1 \tag{2}$$

Although there are some methods aimed to classify pixels of document images directly according to their gray values, the results are usually not good enough due to background noise and non-uniform illumination. More promising methods are to extract certain features from the images and to classify pixels in this new feature space[14, 15, 16]. In paper [14], the author extracts the structural contrast and stroke width to form a 2D feature space, paper[15] extracts regions of interest (ROIs) and paper[16] proposes a stroke-model which depicts the local features of character objects as double-edges in a predefined size.

To extract features from document images, in most cases, some kind of background is needed. Background estimation related to document images has been studied by many investigators. Gatos et al. proposed a method estimating the global background in his paper[18] based on the binary document image generated by Sauvola's thresholding method[8]. Moghaddam et al.[19] estimate the background through an adaptive and iterative image averaging procedure. In fact, what kind of background will be efficient is also dependent on the way of different measurement between the background and the foreground. In this paper, we choose Bhattacharyya distance[17], or Bhattacharyya similarity to measure the difference. So the background here is supposed to be flat.

Bhattacharyya distance[17] was proposed to measure the similarity of two discrete or continuous probability distributions, which is defined by the Bhattacharyya coefficient $BC(p,q)$.

$$BC(p,q) = \begin{cases} \sum_{i=1}^{n} \sqrt{p_i q_i}, & p=(p_1,\cdots,p_n) \text{ and } q=(q_1,\cdots,q_n) \text{ are two discrete probability distributions.} \\ \int \sqrt{p(x)q(x)}dx, & p=p(x) \text{ and } q=q(x) \text{ are two continuous probability distributions.} \end{cases} \tag{3}$$

And the Bhattacharyya similarity of p and q is defined as

$$SB(p,q) = \exp\{-\lambda * (1 - BC(p,q))\} \tag{4}$$

where $\lambda > 0$ is a fixed constant and thus $0 \le SB \le 1$. When applied to two images/matrixes $p(i,j)_{m\times n}$, $q(i,j)_{m\times n}$, the Bhattacharyya coefficient of discrete type is written as

$$BC(p,q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} \sqrt{p(i,j)q(i,j)}}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} p(i,j) \sum_{i=1}^{m}\sum_{j=1}^{n} q(i,j)}}. \tag{5}$$

In this paper we present a thresholding methods, and use Bhattacharyya distance formula to every local template of a document image and its related background. To take the positional information of the neighboring pixels of a template center into consideration, we add an Gaussian kernel to the above Bhattacharyya distance formula and thus the Bhattacharyya coefficient of a template from image $p(i,j)_{m\times n}$ and its corresponding background is modified as

$$BCG(x_0,y_0)=\frac{\sum\limits_{i=-W}^{W}\sum\limits_{j=-H}^{H}G(i,j)\sqrt{p(x_0+i,y_0+j)*L(i,j)}}{\sqrt{\sum\limits_{i=-W}^{W}\sum\limits_{j=-H}^{H}p(x_0+i,y_0+j)\sum\limits_{i=-W}^{W}\sum\limits_{j=-H}^{H}L(i,j)}} \quad (6)$$

where (x_0,y_0) is the center of the template $p(x_0+i,y_0+j)_{(2W+1)*(2H+1)}$ with width $2W+1$ and hight $2H+1$, $L(i,j)_{(2W+1)*(2H+1)}$ is the flat background template and the Gauss kernel $G(i,j)_{(2W+1)*(2H+1)}$ is taken as:

$$G(i,j)=\exp\{\frac{-(i^2+j^2)}{(2*\delta)^2}\}. \quad (7)$$

Then the local Bhattacharyya similarity with Gaussian kernel is expressed as:

$$SBG(x_0,y_0)=\exp\{-\lambda*(1-BCG(x_0,y_0))\} \quad (8)$$

in which,$W<x_0<n-W,\ H<y_0<m-H.$

And an Bhattacharyya similarity matrix $SBG(x_0,y_0)_{(m-2W)\times(n-2H)}$ is obtained as a features of the document image $p(i,j)_{m\times n}$.

After the access to local information or features, we next should classify the pixels of the document image, which means that we have to take a threshold to separate the pixels into two classes. In literatures, the choice of a threshold is made by several methods. Niblack's method sets threshold $T(x,y)$ for each pixel based on the pixel's local statistical information within a fixed local window.

$$T(x,y)=m(x,y)+k*s(x,y), \quad (9)$$

$$B(x,y)=\begin{cases}0. & \text{if } f(x,y)<T(x,y)\\ 1. & \text{else}\end{cases} \quad (10)$$

Where $f(x,y)$ is the gray value of the pixel $p(x,y)$, $m(x,y), s(x,y)$ represent the mean value and variance respectively of the gray values among the local window centered at $p(x,y)$, k is a fixed parameter, and $B(x,y)$ is the binary result. Sauvola improved Niblack's method and took the threshold formula as

$$T(x,y)=m(x,y)[1-k(s(x,y)/R)], \quad (11)$$

where $m(x,y)$ and $s(x,y)$ have the same meaning as the above statement, k is also a fixed parameter, and $R=128$ for images with gray value range (0,255).

In 1979, in order to separate one group of numbers into two classes with a threshold, Otsu proposed a method [2] based on variance within classes and variance between classes. Denote the wanted threshold value k, that is to say, numbers bigger than or equal to k belong to class one and numbers less than k belong to class two. The mean values of the two classes are denoted by μ_0 and μ_1, the variances by σ_0 and σ_1. The mean value and variance of the whole group are μ_T and σ_T. Then the variance within classes is:

$$\sigma_W^2=\omega_0\sigma_0^2+\omega_1\sigma_1^2, \quad (12)$$

the variance between classes is:

$$\sigma_B^2=\omega_0(\mu_0-\mu_T)^2+\omega_1(\mu_1-\mu_T)^2=\omega_0\omega_1(\mu_1-\mu_0)^2, \quad (13)$$

and the total variance of the whole group is:

$$\sigma_T^2=\sum_{i=1}^{L}(i-\mu_T)^2p_i, \quad (14)$$

where ω_0,ω_1 represent the percentages of the two classes in the whole group and $\omega_0+\omega_1=1$. According to Otsu's theory, the optimal threshold k is determined by

$$k^*=\arg\max\sigma_B^2(k)/\sigma_T^2(k) \quad (15)$$

or

$$k^*=\arg\max\sigma_B^2(k)/\sigma_W^2(k). \quad (16)$$

In this paper, since the numbers to be separated are Bhattacharyya similarities, we use Otsu's formula (16) to determine the threshold.

Based on such gotten threshold, the document image is binarized preliminarily. But the result is usually not good enough since it may include points looking like pepper noise and some inner vacancy of characters. As a result, some post procession is needed.

Our strategy of post processing is based on local information as in [22].When there are many foreground pixels around a certain pixel, this pixel tends to be a foreground pixel; and in the other aspect, this pixel tends to be a background pixel. Suppose the binary image is $b(i,j)$ with the value 1 as foreground and 0 as background for each pixel, our post-processing method can be described as:

$$b(i,j)=\begin{cases}1, & \text{if } \frac{\sum_{i=-SW+1}^{SW-1}\sum_{j=-SW+1}^{SW-1}b(i,j)}{(2*SW-1)^2}>\text{maxThres}\\ 0, & \text{if } \frac{\sum_{i=-SW+1}^{SW-1}\sum_{j=-SW+1}^{SW-1}b(i,j)}{(2*SW-1)^2}<\text{minThres}\\ b(i,j), & \text{else}\end{cases} \quad (17)$$

where, $maxThres$ and $minThres$ are two self-defined parameters with the range of 0 to 1.

Affected by noises, the stroke width (SW) extracted by run-length histogram information is not always accurate, and the filter window length used to get the Bhattacharyya similarity matrix is dependent on the stroke width. So the binarized document image is basically rough. Actually, suppose that the stroke width extracted from one original document image by run-length histogram is SW_0, setting $2*SW_0-1$ as the filter window length and thresholding the obtained Bhattacharyya similarity matrix by Otsu's method, we may use run-length histogram again on the binarized document image and get a new stroke width SW_1. Fortunately, lots of computing experiments shows that, if SW_0 is too small, SW_1 is usually bigger than SW_0, and that SW_1 is usually less than SW_0 if SW_0 is too big. Such phenomenon might suggest that SW_1 would equal SW_0 if SW_0 is chosen properly. Inspired by such an observation, we propose the following iterative procedure to binary document images.

1. For a given document image, initialize stroke width using run-length histogram.
2. Calculate the local Bhattacharyya similarities of the image using a filter with Gaussian kernel, whose window size is based on the stroke width obtained above.
3. Classify the pixels by Otsu's thresholding method on the feature space, i.e., the Bhattacharyya similarities.
4. Post process the binarized image with (17).
5. Estimate the stroke width of the obtained binarization image by run-length histogram.
6. If the current stroke width is equal to that used in step 2, end the process; if not, update the stroke width and filter window size and go to step 2.

A demonstration of the iterative procedure is displayed in Fig.1. It must be pointed out that such an iterative process is not always convergent, so we end the iterating if it dose not convergent after a certain iterates.

3 Evaluation measures and Experimental results

There are many ways to measure the effect of an binarization method on document images. We here employ the seven most common measures as follows. Assume the size of the document image in processing is $m\times n$. Let us denote the reference document image, or the ground truth, by $GT(i,j)_{m\times n}$, the binarized final document image by $B(i,j)_{m\times n}$, the skeletonized ground truth image by $SG(i,j)_{m\times n}$. And TP,FP,TN, FN mean the number of pixels of true positives,false positives, true negatives and false negatives, respectively.

The first three measures are Recall, Precision and F-measure[23]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{18}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{19}$$

$$\text{F-measure} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall}+\text{Precision}}. \tag{20}$$

The following two are p-Recall and p-FMeasure[24]:

$$\text{p-Recall} = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n} SG(x,y)\cdot B(x,y)}{\sum_{x=1}^{m}\sum_{y=1}^{n} SG(x,y)}\cdot 100\%, \tag{21}$$

$$\text{p-FMeasure} = \frac{2\cdot\text{p-Recall}\cdot\text{Precision}}{\text{p-Recall}+\text{Precision}}. \tag{22}$$

The sixth is PSNR[25]:

$$\text{PSNR} = 10*\log(\frac{C^2}{\text{MSE}}), \tag{23}$$

$$\text{MSE} = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n}(B(x,y)-GT(x,y))^2}{mn}, \tag{24}$$

where the parameter C is set to 1 for the gray values of the document images are between 0 and 1.

The last measure used here is the Distance Reciprocal Distortion Metric (DRD)[26]:

$$DRD = \frac{\sum_{k=1}^{S} DRD_k}{NUBN}, \tag{25}$$

where DRD_k is the distortion of the k-th flipped pixel and it is calculated using a 5x5 normalized weight matrix W_{Nm} as defined in [26]. DRD_k equals to the weighted sum of the pixels in the 5x5 block of GT that differ from the centered k^{th} flipped pixel at (x,y) in the binarization result image,

$$DRD_k = \sum_{i=-2}^{2}\sum_{j=-2}^{2}|GT_k(i,j)-B_k(x,y)|*W_{Nm}(i,j). \tag{26}$$

And NUBN is the number of the non-uniform (not all black or white pixels) blocks in GT.

Our algorithm was tested in the DIBCO (2009-2012) which includes different types of representative document degradations. The seven depicted measures of the results shown in Tab.1. The DIBCO competition summary of each year is presented in [27, 28, 29, 30].

Remark 1 *Though factors in Tab.1 are not all better than algorithms submitted in the competitions DIBCO 2009-2012, our algorithm can obtain acceptable results on most of the images of the four databases and does not need to adjust any parameters. Besides, the images in these four databases haven't shown many about non-uniformly illumination and multiresolution problems, which are the main focuses of our paper.*

Remark 2 *Our algorithm is applicable to document images with any resolution. Document images taken by different camera equipments have vast differences in resolution. The stroke width may have only two or three pixels in low resolution images, while it can be over ten pixels in high resolution ones. By adaptive iteration, our method can approximate the true stroke width and thus the result is almost independent of the resolution of the images. An example of document images with same content and different resolutions are shown in Fig.2.*

Remark 3 *Our method is feasible when dealing with document images with non-uniform illumination. Unless taken in lab, document images taken under natural light are usually not uniformly illuminated. By local thresholding and local filter with Gaussian cornel, our algorithm overcomes the difficulty due to non-uniform illumination. Fig.3 shows some examples of our algorithm on seriously non-uniformly illuminated pictures which were taken by a mobile phone.*

Remark 4 *Our method performs well on noisy document images. Two badly noised images are bianrized with our algorithm and the results are shown in Fig.4.*

Remark 5 *About the choice of the parameters. Generally, in Otsu's method of selecting threshold, three variables, σ_B^2, σ_W^2 and σ_B^2/σ_W^2 are chosen to be optimized. Theoretically speaking, for the best choice of threshold k^*, σ_W^2 gets the minimum value, and σ_B^2 gets the maximum value, thus σ_B^2/σ_W^2 also gets the maximum value. However, lots of experiments on DIBCO 2009-2012 database show that maximizing σ_B^2/σ_W^2 gives the best results for both F-measure and p-Fmeasure. The details are displayed in Fig.5 and Fig.6.*

4 Conclusion

In this paper, we proposed a new and efficient algorithm for the binarization of document images, especially for the ones with non-uniform illumination or/and noisy background. This method utilizes an adaptive iterating procedure and makes binarization based on a feature space gotten by Bhattacharyya distance/similarity. To better the results, an Gaussian kernel is added to the Bhattacharyya distance formula. The main characteristic of this method is that a more accurate stroke width of the document image can be obtained by such an iterating procedure. Moreover, it works independently on the resolutions of the images and so we don't need to set up any parameter in advance to a certain document image. Experiments show that this method is effective.

References:

[1] Weszka, J.S., Rosenfield, A.: Histogram modification for threshold selection. IEEE Trans. Syst. Man Cybernet. 9, 38C52 (1979)

[2] Otsu, N.: A threshold selection method from grey level histogram.IEEE Trans. Syst. Man Cybernet. 9, 62C66 (1979)

[3] Kapur, J.N., Sahoo, P.K., Wong, A.K.C.: A new method for gray level picture thresholding using the entropy of the histogram. Comput. Vis. Graph. Image Process. 29, 273C285 (1985)

[4] Dawoud, A., Kamel, M.S.: Iterative multimodel subimage binarization for handwritten character segmentation. IEEE Trans. Image Process. 13, 1223C1230 (2004)

[5] Farrahi Moghaddam, R., Cheriet,M.: A multi-scale framework for adaptive binarization of degraded document images. Pattern Recognit. 43, 2186C2198 (2010). doi:10.1016/$j.patcog$.2009.12.024

[6] Tan, C.L., Cao, R., Shen, P., Wang, Q., Chee, J., Chang, J.:Removal of interfering strokes in double-sided document images. In: Fifth IEEE Workshop on Applications of Computer Vision, 2000, pp. 16C21. Palm Springs, CA (2000). doi:10.1109/$WACV$.2000.895397

[7] Trier, O., Jain, A.: Goal-directed evaluation of binarization methods. IEEE Trans. Pattern Anal. Mach. Intell. 17(12), 1191C1201. doi:10.1109/34.476511 (1995)

[8] Sauvola, J., Pietikainen, M.: Adaptive document image binarization. Pattern Recognit. 33(2), 225C236 (2000)

[9] Drira, F., Le Bourgeois, F., Emptoz, H.: Restoring ink bleedthrough degraded document images using a recursive unsupervised classification technique. Doc. Anal. Syst. VII, 38C49. doi:10.1007/11669487_4(2006)

[10] Yibing Yang*, Hong Yan.: An adaptive logical method for binarization of degraded document images. Pattern Recognition 33 787-807 (2000)

[11] Epshtein, B. ; Ofek, E. ; Wexler, Y. : Detecting text in natural scenes with stroke width transform. Computer Vision and Pattern Recognition (CVPR), IEEE Conference. (2010)

[12] M. M. Galloway, Texture analysis using gray level run lengths, in Cornput. Graph. Image Proc., vol. 4, pp. 171C179.(1975)

[13] Y. Liu, S.N. Srihari, Document image binarization based on texture features, IEEE Trans. Pattern Anal. Mach. Intell. 19 (5) 540-544. (1997)

[14] Morteza Valizadeh, Ehsanollah Kabir.: Binarization of degraded document image based on feature space partitioning and classification. International Journal on Document Analysis and Recognition (IJDAR), Volume 15, Issue 1, pp 57-69 (2012)

[15] HH Oh, KT Lim, SI Chien : An improved binarization algorithm based on a water flow model for document image with inhomogeneous backgrounds. Pattern Recognit. 38, 2612-2625 (2005)

[16] Ye, X., Cheriet, M., Suen, C.Y.: Stroke-model-based character extraction from gray-level document images. IEEE Trans. Image Process. 10, 1152-1161 (2001)

[17] Bhattacharyya, A. "On a measure of divergence between two statistical populations defined by their probability distributions". Bulletin of the Calcutta Mathematical Society 35: 99C109. MR 0010358 (1943)

[18] B. Gatos, I. Pratikakis, S.J. Perantonis. : Adaptive degraded document image binarization. Pattern Recognition 39, 317C327 (2006)

[19] Moghaddam, R.F., Cheriet, M.: Rsldi: restoration of singlesided low-quality document images. Pattern Recogn. 42, 3355C3364 (2009)

[20] Van Rijsbergen, C. J. Information Retrieval (2nd ed.). Butterworth.(1979)

[21] Jong-Sen Lee. : Digital Image Enhancement and Noise Filtering by Use of Local Statistics. Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:PAMI-2, Issue: 2) p165-168 (1980)

[22] Basilios Gatos, Ioannis Pratikakis, and Stavros J. Perantonis. "An Adaptive Binarization Technique for Low Quality Historical Documents". Document Analysis Systems VI Lecture Notes in Computer Science Volume 3163, pp 102-113. (2004)

[23] Van Rijsbergen, C. J. Information Retrieval (2nd ed.). Butterworth. (1979)

[24] K. Ntirogiannis. B. Gatos and I. Pratikakis. An Objective Evaluation Methodology for Document Image Binarization Techniques. The 8th IAPR International Workshop on Document Analysis Systems (DAS 2008). Nara Prefectural New Public Hall. Nara. Japan. September 17-19. pp.217-224. (2008)

[25] Huynh-Thu, Q.; Ghanbari, M. "Scope of validity of PSNR in image/video quality assessment". Electronics Letters 44 (13): 800. doi:10.1049/el:20080522 (2008)

[26] H. Lu, A. C. Kot and Y.Q. Shi, Distance-Reciprocal Distortion Measure for Binary Document Images, IEEE Signal Processing Letters, vol. 11, No. 2, pp. 228-231, (2004)

[27] B. Gatos, K. Ntirogiannis, I. Pratikakis. : DIBCO 2009: document image binarization contest. International Journal on Document Analysis and Recognition (IJDAR) Volume 14, Issue 1 , pp 35-44 (2009)

[28] Ioannis Pratikakis, Basilis Gatos and Konstantinos Ntirogiannis. H-DIBCO 2010CHandwritten Document Image Binarization Competition. 2010 12th International Conference on Frontiers in Handwriting Recognition. (2010)

[29] Ioannis Pratikakis, Basilis Gatos, Konstantinos Ntirogiannis. ICDAR 2011 Document Image Binarization Contest (DIBCO 2011). International Conference on Document Analysis and Recognition. (2011)

[30] Ioannis Pratikakis, Basilis Gatos, Konstantinos Ntirogiannis. ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012). International Conference on Frontiers in Handwriting Recognition. (2012)

Table 1: Experimental results in DIBCO 2009-2012 database

Year	F-measure(%)	p-FMeasure(%)	PSNR	DRD	Recall	Precision	p-Recall
DIBCO 2009 database	83.1223	84.2805	15.8418	7.2607	84.8125	84.6958	86.9865
DIBCO 2010 database	81.1573	88.5472	16.5405	5.2690	75.1116	90.5817	87.6775
DIBCO 2011 database	78.0188	80.2528	14.8248	38.4523	82.1330	78.8107	86.7235
DIBCO 2012 database	73.0121	75.8937	15.7558	10.2791	71.8111	79.5967	77.3296

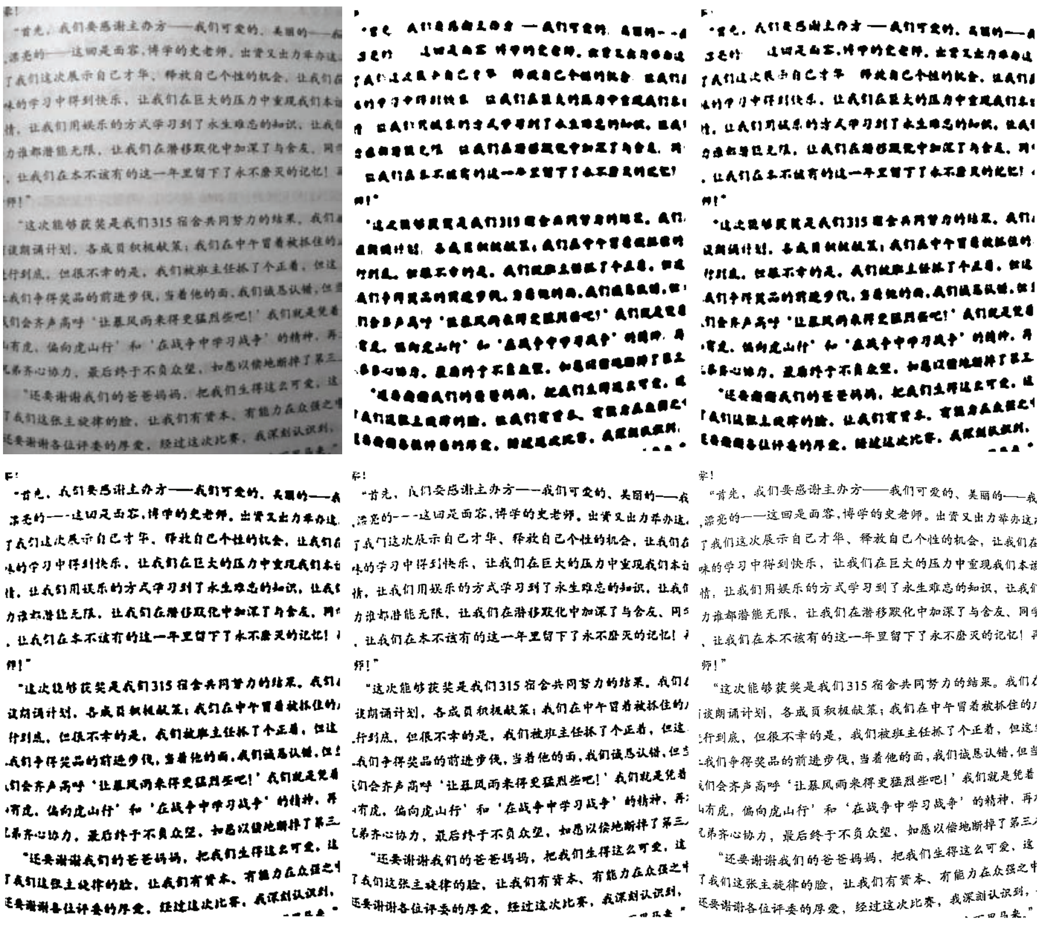

Figure 1: An example of the iteration procedure: the first one in the top row is the original image, from left to right by row are binary results during iteration, the last one is the final result.

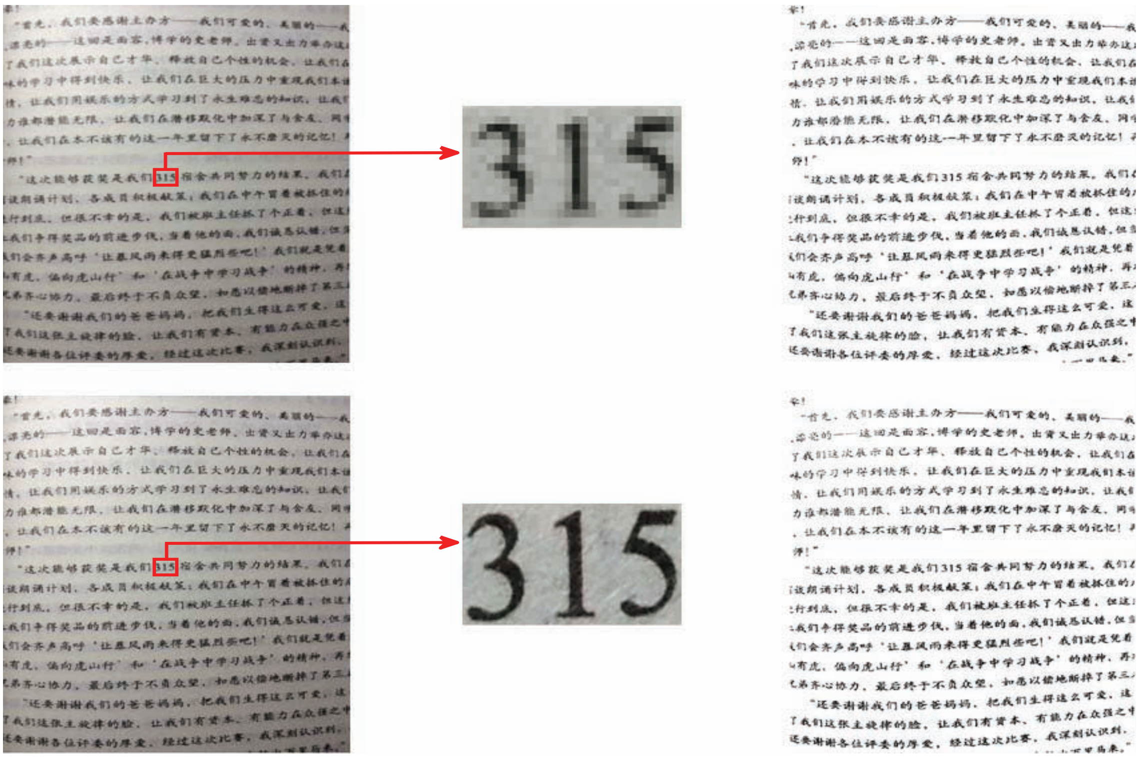

Figure 2: Binary results of two images with the same content but in different resolutions. In the upper row from left to right: the original image with resolution 456x604, the detail of the image shows the stroke width is about only three pixels, the binary result image. In the lower row from left to right: the original image with resolution 2448x3264, the detail of the image shows the stroke width is about ten pixels, the binary result image. The results are indenpent of the resolution of the images.

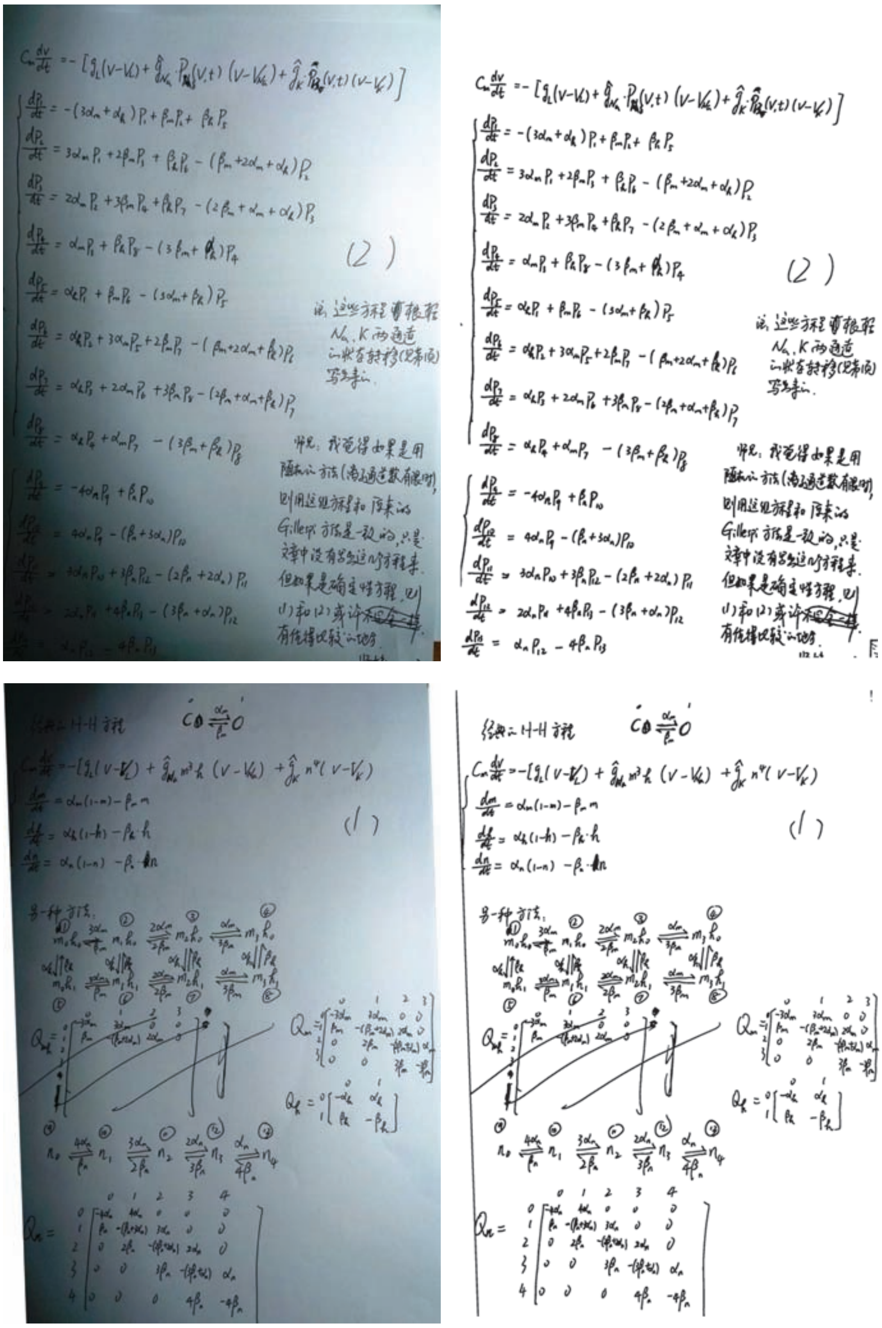

Figure 3: Two images in non-uniform illumination and their binary results by our algorithm.

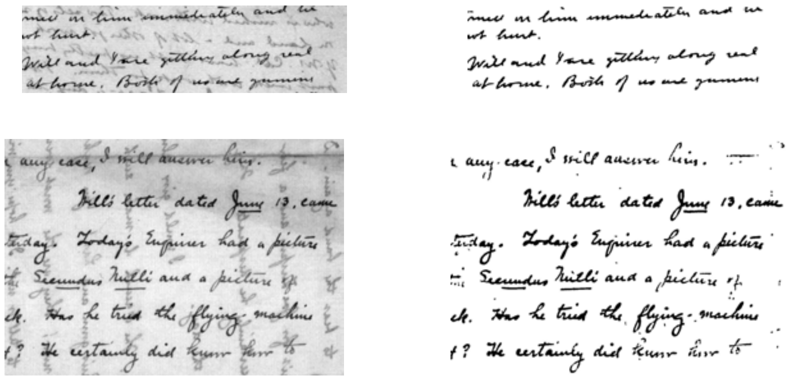

Figure 4: Two seriously noised images and their binary results by our algorithm.

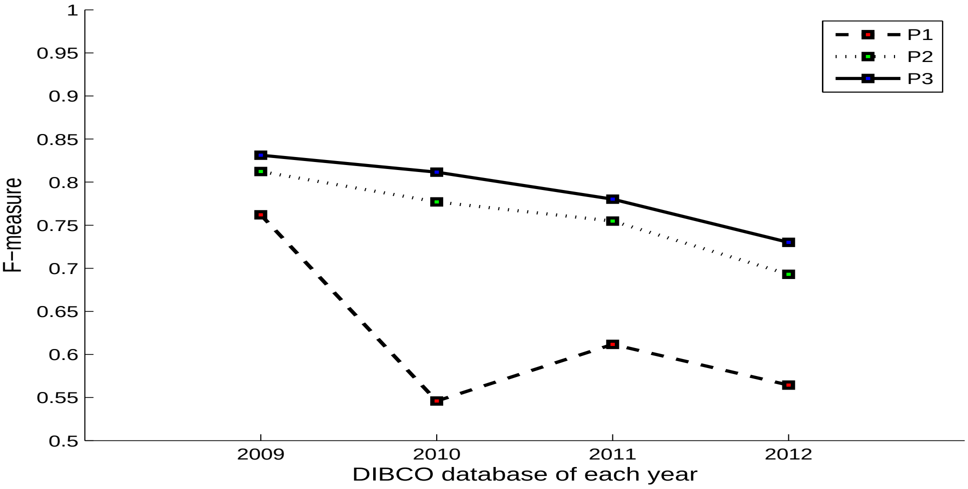

Figure 5: The comparation of F-Measures obtained by different optimizations. P1: average results by optimizing δ_B^2 P2: average results by optimizing δ_W^2 P3: average results by optimizing δ_B^2/δ_W^2

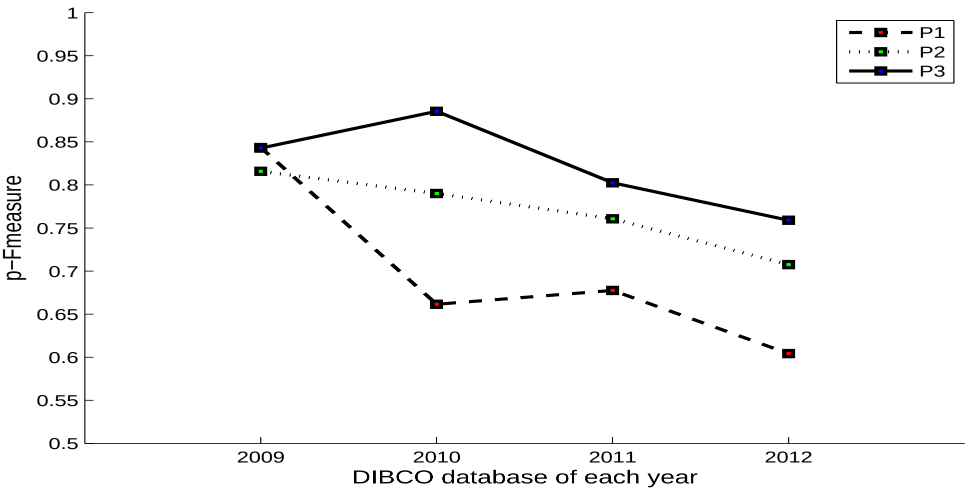

Figure 6: The comparation of p-FMeasures obtained by different optimizations. P1,P2 and P3 are same to those in Fig.5