

Nabendu Chaki
Soharab Hossain Shaikh
Khalid Saeed

Exploring Image Binarization Techniques

Studies in Computational Intelligence

Volume 560

Series editor

Janusz Kacprzyk, Polish Academy of Sciences Warsaw, Poland
email: kacprzyk@ibspan.waw.pl

For further volumes:
<http://www.springer.com/series/7092>

About the Series

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the worldwide distribution, which enable both wide and rapid dissemination of research output.

Nabendu Chaki · Soharab Hossain Shaikh
Khalid Saeed

Exploring Image Binarization Techniques



Springer

Nabendu Chaki
Computer Science and Engineering
University of Calcutta
Kolkata, West Bengal
India

Soharab Hossain Shaikh
A. K. Choudhury School of Information Technology
University of Calcutta
Kolkata, West Bengal
India

Khalid Saeed
Physics and Applied Computer Science
AGH University of Science
and Technology
Kraków
Poland

ISSN 1860-949X
ISBN 978-81-322-1906-4
DOI 10.1007/978-81-322-1907-1
Springer New Delhi Heidelberg New York Dordrecht London

ISSN 1860-9503 (electronic)

ISBN 978-81-322-1907-1 (eBook)

Library of Congress Control Number: 2014938212

© Springer India 2014

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law. The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*To my parents Md. Golam Hossain Shaikh
and Mrs. Krishna Shaikh for their warm
affection, loving indulgence, occasional
forgiveness, unconditional support and
invaluable contributions leading my life
towards a better tomorrow*

—Soharab Hossain Shaikh

*To my Grandson Gabriel Jan who had
opened a new chapter in my life*

—Khalid Saeed

*To my father Late Mugdhendu Sekhar Chaki
for always being there with all his support in
whatever I wanted to do in my life*

—Nabendu Chaki

Preface

It is a great pleasure to introduce this book on Image Binarization. The book is aimed to ease the job of future researchers who work in the field of image processing, especially one that requires segmentation of grayscale images. A grayscale image can be segmented into two groups as object and background by using a binarization technique. A threshold is calculated and all pixels with gray-level values above the threshold are set to build the background while pixels below the threshold are set to form the object. This bi-level segmentation is known as image binarization. It is an important step in the preprocessing stage performed in many image processing applications.

This book provides a comprehensive survey of existing binarization techniques for both document and graphic images. A number of evaluation techniques are presented for quantitative comparison of different binarization methods. It provides the results obtained comparing a number of standard and widely used binarization algorithms using standard evaluation metrics. The comparative results presented in tables and charts in this book facilitates to understand the process.

In addition to this, the book presents techniques for preparing a reference image, which is important for quantitative evaluation of the binarization techniques. The results are produced taking image samples from standard image databases.

It has been organized in the form of six chapters starting with an introduction and followed by a comprehensive review in the first two chapters. The most important contribution of the book is in [Chap. 3](#) where an iterative partitioning-based image binarization technique is introduced. In [Chap. 4](#), a method is proposed towards creation of reference image for degraded document images in the presence of various types of noises. We thank and appreciate Asis Kumar Maity and Ayan Dey for their contributions in implementing the proposed methodologies and experimental verification.

We express our sincere thanks to Aninda Bose, Publishing Editor from Springer India for his continual support and positive influence right from the point of offering us to work for a book on this topic.

Lastly, we thank all of our family members who spared us and sacrificed their valuable time to let us concentrate on the book. We will consider our effort to be successful if this book helps the budding scholars to explore the area of image processing and inspire them for greater contribution.

Kolkata, India, February 2014

Nabendu Chaki
Soharab Hossain Shaikh
Khalid Saeed

Contents

1	Introduction	1
1.1	Binarization and Image Segmentation	1
1.2	Binarization of an Image	2
1.3	Binarization of Graphic and Document Images	3
1.4	Calculating Threshold for Binarization	3
1.5	Applications of Binarization	4
References		4
2	A Comprehensive Survey on Image Binarization Techniques	5
2.1	Foundations of Image Binarization Techniques	5
2.2	Recent Works	11
References		14
3	A New Image Binarization Technique Using Iterative Partitioning	17
3.1	Image Binarization Using Iterative Partitioning	17
3.1.1	Motivation of the Work	17
3.1.2	Proposed Methodology: Binarization Using Iterative Partitioning	18
3.1.3	Evaluation Measures	25
3.1.4	Experimental Dataset	28
3.1.5	Experimental Verification	30
3.1.6	Conclusions	43
References		43
4	A Framework for Creating Reference Image for Degraded Document Images	45
4.1	Motivation of the Work	46
4.2	Proposed Methodology	47
4.3	Determining the Value of k	48
4.4	Benchmark Dataset	55

4.5	Experimental Verification	57
4.5.1	Majority Voting Method	57
4.5.2	Comparative Performance Analysis	61
4.6	Conclusions	62
	References	63
5	Applications of Binarization	65
5.1	Document Image Processing and OCR	65
5.2	Medical Image Processing	66
5.3	Video Processing	66
5.4	Face Detection	67
5.5	Hand Gesture Recognition	67
5.6	Fingerprint Recognition	68
5.7	Iris Recognition	68
5.8	Gait Recognition	69
	References	69
6	Conclusions	71
	Appendix A: Sample Test Images	75
	Index	81

About the Authors



Nabendu Chaki is a Senior Member of IEEE and an Associate Professor in the Department of Computer Science and Engineering, University of Calcutta, India. Besides editing several volumes in LNCS, Springer and other series, Nabendu has authored three textbooks with reputed publishers like Taylor and Francis (CRC Press), Pearson Education, etc. Dr. Chaki has published more than 100 refereed research papers in Journals and International conferences. His areas of research interests include image processing, distributed systems, and network security. Dr. Chaki has also served as a Research Assistant Professor in the Ph.D. program (Software Engineering) at the U.S. Naval Postgraduate School, Monterey, CA. He is a visiting faculty member for many universities including the University of Ca' Foscari, Venice, Italy. Dr. Chaki has contributed in SWEBOK v3 of the IEEE Computer Society as a Knowledge Area Editor for Mathematical Foundations. Besides, being an editorial board member of many international journals, he has also served in the committees of more than 50 international conferences. He is the founding Chapter Chair of ACM Professional Chapter in Kolkata, India, since January 2014.



Soharab Hossain Shaikh is a faculty member at the A. K. Choudhury School of Information Technology, University of Calcutta, India. After his B.Sc. Honors in Computer Science from University of Calcutta in 2001, he completed M.Sc. in Computer and Information Science in 2003 followed by M.Tech. in Computer Science and Engineering in 2005 from the Department of Computer Science and Engineering, University of Calcutta. He has received a fellowship from the Italian Ministry of Education for Universities and Research (MIUR) for pursuing research work at Ca' Foscari, University of Venice, Italy in 2006–2007. His research interests include image processing, computer vision, biometrics, and pattern recognition. He works in active collaboration with

AGH University of Science and Technology, Bialystok Technical University, Poland. Mr. Shaikh jointly holds a US-patent on Character Recognition. He has recently submitted his doctoral thesis in the domain of computer vision and image processing. He has served as the reviewer/committee member in many international conferences/symposiums and journals. He is a member of the IEEE Computer Society and ACM.



Khalid Saeed received the B.Sc degree in Electrical and Electronics Engineering in 1976 from Baghdad University, M.Sc and Ph.D. degrees from Wrocław University of Technology, in Poland in 1978 and 1981, respectively. He received his D.Sc degree (Habilitation) in Computer Science from the Polish Academy of Sciences in Warsaw in 2007. He is a Professor of Computer Science at AGH University of Science and Technology in Poland. He has authored more than 190 publications including 23 edited books, Journals and Conference Proceedings, 8 text and reference books. He supervised more than 110 M.Sc. and 12 Ph.D. theses. His areas of interest are Biometrics, Image Analysis and Processing, and Computer Information Systems. He gave 39 invited lectures and keynotes in different universities in Europe, China, India, South Korea, and Japan. The talks were on Biometric Image Processing and Analysis. He received about 16 academic awards. Khalid Saeed is a member the editorial boards of over 15 international journals and conferences. He is an IEEE Senior Member and has been selected as IEEE Distinguished Speaker for 2011–2013 and 2014–2016. Khalid Saeed is the Editor-in-Chief of International Journal of Biometrics with Inderscience Publishers.

Chapter 1

Introduction

Abstract Binarization is one of the most important preprocessing steps in most of the vision-based systems for object detection and classification. Application of binarization includes finding out the region of interest from a given image targeted for a particular application. This chapter presents introductory information to the main subject of the book—binarization.

Keywords Image segmentation • Binarization • Thresholding • Applications of binarization • Document image binarization • Threshold

Binarization is one of the most important preprocessing steps in most of the vision-based systems for object detection classification. Application of binarization includes finding out the region of interest from a given image targeted for a particular application.

1.1 Binarization and Image Segmentation

Binarization is one of the methods toward image segmentation. Image segmentation is the process of clustering the pixels depending on some property of the image, e.g., intensity gray levels, color, texture, depth, edge continuity. After segmentation, the whole image is partitioned into smaller regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects. Segmentation can be used for object recognition, estimation of occlusion boundary within motion or stereo systems, image compression, image editing, and/or image database look-up.

There are many approaches to segmentation, e.g., region segmentation (group of connected pixels with similar properties, e.g., color) and edge segmentation. In ideal image, regions are bounded by closed contours which may be obtained from edge detection and regions may be obtained by boundary filling. Segmenting

images is very important in scene interpretation and object recognition. Sometimes, only the image properties such as color and texture may not be sufficient for classifying an object. Therefore, the shape of the object may be an important cue in its identification. However, the shape can be detected only after the same scan be segmented and separated out from the background.

Binarization is one of such important image segmentation approaches for finding out the area of interest from an image by separating the pixels into two groups: one representing the object and the other background. A binary image $B(x, y)$ of a given grayscale image $I(x, y)$ is a representation of $I(x, y)$ with only two (bi) gray levels. In a binary image, there are only two gray levels; this is the reason why this is called a binary image. In a binary image, gray value 0 (black) generally represents object or foreground pixels and gray value 255 (white) represents the background pixels or vice versa. Representation can be done using gray value 0 for foreground and 1 for background. This makes every pixel to be stored with a single bit. A typical gray value known as threshold is to be selected to form a binary image from a grayscale image. After the threshold (T) is selected, all the pixels in the image having gray-level intensity value greater than or equal to T are set to 1 and the rest of the pixels are set to 0 to form the binary image.

For some threshold T (*gray level*)

$$\begin{aligned} B(x, y) &= 1, \quad \text{if } I(x, y) \geq T \\ &= 0, \quad \text{otherwise.} \end{aligned}$$

Binarization methods can be broadly categorized into two groups; global and local methods depending on how threshold value is calculated for the image to be segmented [1]. If a single threshold value is used for the entire image, the corresponding method is a global binarization method [2]. On the other hand, for a local method [3], a number of threshold values can be calculated for different regions of an image depending on some properties of the image.

1.2 Binarization of an Image

A simple way to binarize an image is through *thresholding* and separating the light and dark regions (background and foreground) according to the pixel intensities. In many image processing and pattern recognition applications, gray levels of pixels belonging to the object are substantially different from the gray levels of pixels belonging to the background. In such context, thresholding becomes a simple but effective tool for separating objects from the background. Thresholding creates binary images from a gray-level image by setting all the pixels below some threshold to zero and all pixels equal or above that threshold to one.

Binarization is used as a preprocessing step in several image processing applications. It is also used to find the region of interest (ROI) from an image. The use of binary images decreases computational load for the overall application. Figure 1.1



Fig. 1.1 Vision-based object recognition system

shows the block diagram of a typical vision-based object recognition system. It clearly shows that binarization is an important preprocessing step in such systems.

1.3 Binarization of Graphic and Document Images

Graphic images contain the picture of a person, scene, objects, etc. These images mostly contain different shades of gray-level intensities that depict a scene. On the other hand, a document image is a digitized version of a handwritten or printed text. Some document images contain graphical contents such as logos and symbols.

A document image analysis system includes several image processing tasks, beginning with digitization of the document and ending with character recognition and natural language processing. The thresholding step can affect quite critically the performance of successive steps such as classification of the document into text objects, and the correctness of the optical character recognition (OCR). Improper thresholding causes blotches, streaks, erasures on the document confounding segmentation, and recognition tasks. The merges, fractures, and other deformations in the character shapes as a consequence of incorrect thresholding are the main reasons of OCR performance deterioration [1].

1.4 Calculating Threshold for Binarization

A threshold value for binarization can be calculated by following a simple method [4] as follows:

1. Select an initial estimate for T . (A suggested initial estimate is the average of the minimum and maximum intensity values in the image).
2. Segment the image using T . This will produce two groups of pixels: G_1 , consisting of pixels with intensity values $\geq T$, and G_2 , consisting of pixels with values $< T$.
3. Compute the average intensity values μ_1 and μ_2 for the pixels in regions G_1 and G_2 .
4. Compute a new threshold value:

$$T = \frac{1}{2} * (\mu_1 + \mu_2)$$

5. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_0 .

1.5 Applications of Binarization

Some of the applications of binarization are listed below:

1. Document image analysis [5–7] (e.g., extraction of printed logos, graphical content from document images; finding lines, legends, and characters in map processing).
2. OCR and quality inspection of materials [1].
3. Foreground background classification and object recognition [8] and extraction of text embedded in images in hand-held devices [9]
4. Satellite image segmentation [10] and identification of objects of interest in medical imaging [11].
5. Video processing, moving object detection, scene matching, etc. [12, 13].
6. Finger-vein pattern extraction [14] and fingerprint preprocessing [15].

References

1. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging* **13**(1), 146–165 (2004)
2. Otsu, N.: A threshold selection method from gray-level histogram. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979)
3. Sauvola, J., Pietikainen, M.: Adaptive document image binarization. *Pattern Recogn.* **33**(2), 225–236 (2000)
4. Gonzalez, R., Woods, R.: *Digital Image Processing*. Pearson Prentice Hall, Upper Saddle River (2008). (International 3rd Revised Edition, Chapter: 10)
5. Datta, S., Singh, P.K., Sarkar, R., Nasipuri, M.: A new image binarization technique by classifying document images. In: 5th International Conference on Pattern Recognition and Machine Intelligence (PReMI), pp. 539–544 (2013)
6. Su, B., Lu, S., Tan, C.L.: Robust document image binarization technique for degraded document images. *IEEE Trans. Image Process.* **22**(4), 1408–1417 (2013)
7. Cheriet, M., Moghaddam, R.F., Hedjam, R.: A learning framework for the optimization and automation of document binarization methods. *Comput. Vis. Image Underst. (CVIU)* **117**(3), 269–280 (2013)
8. Shaikh, S.H., Maity, A.K., Chaki, N.: A new image binarization method using iterative partitioning. *Springer J. Mach. Vis. Appl. (MVAP)* **24**(2), 337–350 (2013)
9. Mollah, A.F., Basu, S., Nasipuri, M., Basu, D.K.: Handheld mobile device based text region extraction and binarization of image embedded text documents. *J. Intell. Syst.* **22**(1), 25–47 (2013)
10. Mitra, S., Kundu, P.P.: Satellite image segmentation with Shadowed C-Means. *J. Inform. Sci.* **181**, 3601–3613 (2011)
11. Rodriguez, R.: Binarization of medical images based on the recursive application of mean shift filtering: another algorithm. *J. Adv. Appl. Bioinf. Chem.* **1**, 1–12 (2008)
12. Rosin, P.L.: Thresholding for change detection. *Comput Vis Image Underst (CVIU)* **86**(2), 79–95 (2002)
13. Shaikh, S.H., Bhunia, S.K., Chaki, N.: On generation of silhouette of moving objects from video. In: Springer Proceedings of the 4th International Conference on Signal and Image Processing (ICSIP), vol. 1, pp. 213–223 (2012)
14. Walus, M., Kosmala, J., Saeed K.: Finger vein pattern extraction algorithm. In: 6th International Conference on Hybrid Artificial Intelligent Systems (HAIS), pp. 404–411 (2011)
15. Shaikh, S.H., Saeed, K., Chaki, N.: Performance benchmarking of different binarization techniques for fingerprint-based biometric authentication. In: Springer Proceedings of the 8th International Conference on Computer Recognition Systems (CORES), Part-II, pp. 237–246 (2013)

Chapter 2

A Comprehensive Survey on Image Binarization Techniques

Abstract A detailed survey about the principles of image binarization techniques is introduced in this chapter. A comprehensive review is given. A number of classical methodologies together with the recent works are considered for comparison and study of the concept of binarization for both document and graphic images.

Keywords Review of binarization methods • Global binarization • Image thresholding • Adaptive local binarization

2.1 Foundations of Image Binarization Techniques

A number of methodologies have been proposed by several researchers on image segmentation using binarization and its applications toward moving object detection and human gait recognition. This section presents a review of the classical methodologies found in the literature. Over past four decades, several researchers have proposed a variety of thresholding techniques for binarization of document images [1–7] as well as graphic images [8, 9]. Processing of documents that are of very poor quality due to seeping of ink from the other side of the page or general degradation of the paper and ink, background noise or variation in contrast and illumination are also found in the literature [10]. All the reported thresholding methods have been demonstrated to be effective in constrained processing environments with predictable images. However, pictures taken in real-life situations may contain different artifacts such as shadow, non-uniform illumination. Proper binarization of these images is very important for separating the foreground object from the background. A good binarization will result in better recognition accuracy for any pattern recognition application.

Binarization can become a challenging job [10] under varying illumination and noise. A number of factors contribute to complicate the thresholding scheme including ambient illumination, variance of gray levels within the object and the

background, inadequate contrast. A wrong selection of threshold value may misinterpret the background pixel and can classify it as object and vice versa, resulting in overall degradation of system performance. In document analysis, binarization is sensitive to noise, surrounding illumination, gray-level distribution, local shading effects, inadequate contrast, the presence of dense non-text components such as photographs. While at the same time, the merges, fractures, and other deformations in the character shapes affect the threshold value in OCR system.

The binarization methods can be categorized in different groups depending on which principal criteria they consider in calculating the threshold. The method proposed by Otsu [8] proposed a clustering analysis-based method based on image variance. Methods proposed by Johannsen et al. [11] and Kapur et al. [12] are entropy-based methods. Binarization methods based on image variance are proposed by Sauvola et al. [13] and Niblack [14]. Bernsen [9] proposed a thresholding approach based on image contrast. Kittler et al. [15] consider error measure in calculating the optimal threshold. Some of the methods are discussed below in brief.

Otsu's method [8] is the most successful global thresholding method. It automatically performs histogram shape-based image thresholding for the reduction of a gray-level image to a binary image. The algorithm assumes that the image for thresholding contains two classes of pixels (e.g., foreground and background) and then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal. It exhaustively searches for the threshold that minimizes the intra-class variance, defined as the weighted sum of variances of the two classes. The weighted within-class variance is $\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$ where the class probabilities of different gray-level pixels are estimated as:

$$q_1(t) = \sum_{i=0}^t P(i) \quad \text{and} \quad q_2(t) = \sum_{i=t+1}^{255} P(i)$$

And the class means are given by:

$$\mu_1(t) = \sum_{i=0}^t \frac{i * P(i)}{q_1(t)} \quad \text{and} \quad \mu_2(t) = \sum_{i=t+1}^{255} \frac{i * P(i)}{q_2(t)}$$

Total variance (σ^2) = Within-class variance ($\sigma_w^2(t)$) + Between-class Variance ($\sigma_b^2(t)$), where $\sigma_b^2(t) = q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2$

Since the total variance is constant and independent of t , the effect of changing the threshold is merely to move the contributions of the two terms back and forth. Between-class variance is $\sigma_b^2(t) = q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2$. Thus, minimizing the within-class variance is the same as maximizing the between-class variance. This method gives satisfactory results when the numbers of pixels in each class are close to each other.

In locally adaptive thresholding algorithms, a threshold is calculated at each pixel, which depends on some local statistics such as range, variance, or

surface-fitting parameters of the pixel neighborhood. In what follows, the threshold $T(i, j)$ is indicated as a function of the coordinates (i, j) at each pixel, or if this is not possible, the object/background decisions are indicated by the logical variable $B(i, j)$. Niblack's method [14] calculates pixel-wise threshold by sliding a rectangular window over the gray-level image. This method adapts the threshold according to the local mean $m(i, j)$ and standard deviation $\sigma(i, j)$ and calculated a window size of $b \times b$. The threshold T is denoted as: $T(i, j) = m(i, j) + k \cdot \sigma(i, j)$.

Here, k is a constant, which determines how much of the total print object edge is retained, and has a value between 0 and 1. The value of k and the size of the sliding window define the quality of binarization. Binarization gives thick and unclear strokes with a small k value, and slim and broken strokes with a large k value. As for many applications, a 25×25 size for the sliding window and 0.6 as the value of k have been found to be heuristically optimal. The size of the neighborhood should be small enough to reflect the local illumination level and large enough to include both objects and the background.

The method proposed by Sauvola et al. [13] is local-variance-based method. It is an improvement on the method proposed by Niblack [14], especially when the background contains light texture, big variations, stained and badly and unevenly illuminated documents. It adapts the contribution of the standard deviation. For example, in the case of text on a dirty or stained paper, the threshold is lowered. The threshold is calculated as follows:

$$T(i, j) = m(i, j) * \left[1 + k \left(\frac{\sigma(i, j)}{R} - 1 \right) \right]$$

The typical values of $k = 0.5$ and $R = 128$ are suggested. Here, m and σ are again the mean and standard deviation of the whole window, and k is a fixed value. It was found that the value of R has a very small effect on the quality while the values of k and window size affect it significantly. The smaller the value of k , the thicker is the binarized stroke, and the more overlap exists between characters. A smaller window size will produce thinner strokes. An optimal combination of k and the sliding window will produce a good binary image.

Local adaptive method proposed by Bernsen [9] is based on contrast of an image. The threshold is set at the midrange value, which is the mean of the minimum $I_{\text{low}}(i, j)$ and maximum $I_{\text{high}}(i, j)$ gray values in a local window of suggested size $w = 31$. However, if the contrast $C(i, j) = I_{\text{high}}(i, j) - I_{\text{low}}(i, j)$ is below a certain contrast threshold k , the pixels within the window may be set to background or to foreground according to the class that most suitably describes the window. This algorithm is dependent on the value of k and also on the size of the window. $T(i, j) = 0.5 \{ \max_w[I(i + m, j + n)] + \min_w[I(i + m, j + n)] \}$, where $w = 31$, provided contrast $C(i, j) = I_{\text{high}}(i, j) - I_{\text{low}}(i, j) \geq 15$.

The method proposed by Kapur et al. [12] is an entropy-based method which considers the image foreground and background as two different signal sources, so that when the sum of the two class entropies reaches its maximum, the image is said to be under optimal thresholding.

In this method, two probability distributions (e.g., object distribution and background distribution) are derived from the original gray-level distribution of the image as follows:

$$\frac{p_0}{P_t}, \frac{p_1}{P_t}, \dots, \frac{p_t}{P_t} \quad \text{and} \quad \frac{p_{t+1}}{1-P_t}, \frac{p_{t+2}}{1-P_t}, \dots, \frac{p_l}{1-P_t}$$

where t is the value of threshold and $P_t = \sum_{i=0}^t p_i$

$$H_b(t) = - \sum_{i=0}^t \frac{p_i}{P_t} \log_e \left(\frac{p_i}{P_t} \right) \quad \text{and} \quad H_w(t) = - \sum_{i=t+1}^l \frac{p_i}{1-P_t} \log_e \left(\frac{p_i}{1-P_t} \right)$$

The optimal threshold t^* is defined as the gray level which maximizes $H_b(t) + H_w(t)$, i.e. $t^* = \arg \max \{H_b(t) + H_w(t)\}$ for all t belonging to the set of all gray values in the image.

Thresholding can be considered as a classification problem. If the gray-level distributions of the foreground object and background pixels are known or can be estimated, then the optimal, minimum error threshold can be obtained using statistical decision theory. This involves lots of computation. Therefore, it is realistic to assume that the respective populations are distributed normally with distinct means and standard deviations. Under this assumption, the parameters of the population can be inferred from the gray-level histogram by fitting. Afterward, the corresponding optimal threshold can be determined. A computationally efficient solution to the problem of minimum error thresholding has been derived by Kittler et al. [15] under the assumption of foreground object and background pixel gray-level values being normally distributed. The principal idea behind the method is to optimize the average pixel classification error rate directly, using either an exhaustive search or an iterative algorithm. The method is applicable in multi-threshold selection.

$$p(g) = \sum_{i=1}^2 P_i p(g|i), \quad \text{where, } p(g|i) = \frac{1}{\sqrt{2\Pi\sigma_i^2}} \exp \left(1 - \frac{(g - \mu_i)^2}{2\sigma_i^2} \right).$$

The threshold value can be selected by solving the quadratic equation

$$\frac{(g - \mu_1)^2}{\sigma_1^2} + \log_e \sigma_1^2 - 2 \log_e P_1 = \frac{(g - \mu_2)^2}{\sigma_2^2} + \log_e \sigma_2^2 - 2 \log_e P_2$$

However, the parameters μ_i, σ_i^2 and $P_i (i = 1, 2)$ of the mixture density $p(g)$ associated with an image for thresholding are not usually known. In order to overcome the difficulty of estimating these unknown parameters, Kittler et al. introduced a criterion function $J(t)$ given by

$$J(t) = 1 + 2 \{ P_1(t) \log_e \sigma_1(t) + P_2(t) \log_e \sigma_2(t) \} \\ - 2 \{ P_1(t) \log_e P_1(t) + P_2(t) \log_e P_2(t) \}$$

where

$$P_1(t) = \sum_{g=0}^t h(g), \quad P_2(t) = \sum_{g=t+1}^l h(g)$$

$$\mu_1(t) = \frac{\left\{ \sum_{g=0}^t h(g)g \right\}}{P_1(t)}, \quad \mu_2(t) = \frac{\left\{ \sum_{g=t+1}^l h(g)g \right\}}{P_2(t)},$$

$$\sigma_1^2(t) = \frac{\left\{ \sum_{g=0}^t (g - \mu_1(t))^2 h(g) \right\}}{P_1(t)}, \quad \sigma_2^2(t) = \frac{\left\{ \sum_{g=t+1}^l (g - \mu_2(t))^2 h(g) \right\}}{P_2(t)}.$$

The optimal threshold is obtained by minimizing $J(t)$, i.e., by finding $t^* = \arg \text{Min}\{J(t)\}$ for all gray levels t belonging to the image.

The method proposed by Johannsen et al. [11] uses the entropy of the gray-level histogram of the digital image as a measure of information. Essentially, it divides the set of gray levels into two parts so as to minimize the interdependence between them. This method chooses the threshold value t^* from the relation, $t^* = \arg \text{Min}\{S(t) + S'(t)\}$ for all possible gray levels t in the image. Here,

$$S(t) = \log_e \left(\sum_{i=0}^t p_i \right) - \frac{1}{\sum_{i=0}^t p_i} \left[p_t \log_e (p_t) + \left(\sum_{i=0}^{t-1} p_i \right) \log_e \left(\sum_{i=0}^{t-1} p_i \right) \right]$$

$$S'(t) = \log_e \left(\sum_{i=t}^{l-1} p_i \right) - \frac{1}{\sum_{i=t}^{l-1} p_i} \left[p_t \log_e (p_t) + \left(\sum_{i=t+1}^{l-1} p_i \right) \log_e \left(\sum_{i=t+1}^{l-1} p_i \right) \right]$$

A technique for determining a threshold for binarization of an image is presented in [16]. The method follows an iterative process and assumes that the image contains an object and background occupying different average gray levels. The iterative method provides a simple automatic selection of the optimum threshold. Assuming an object is located within a square region of the image; without any prior knowledge of the exact location of the objects, it is considered as a first approximation that the four corners of the scene contain only background pixels and the remainder contains the object. Thresholding is done to come up with a path image. This patch may then be used as a switching function $f(s)$ to route a digitized image into one of two integrators.

In [17] a multi-scale binarization framework is introduced, which can be used along with any adaptive threshold-based binarization method. This framework is able to improve the binarization results and to restore weak connections and strokes, especially in the case of degraded historical documents. The framework requires several binarization methods on different scales, which is addressed by introduction of fast grid-based models. This enables to explore high scales which are usually unreachable to the traditional approaches. In order to expand the set of adaptive methods, an adaptive modification of Otsu's method, called AdOtsu, is

introduced. In addition, in order to restore document images suffering from bleed-through degradation, the authors combine the framework with recursive adaptive methods. The framework shows promising performance in subjective and objective evaluations performed on available datasets.

An automatic histogram threshold approach based on a fuzziness measure is presented in [18]. Using the concepts of fuzzy logic, the problems involved in finding the minimum of a criterion function are avoided. Similarity between gray levels is the key to find an optimal threshold. Two initial regions of gray levels are defined at the boundaries of the histogram. After that using an index of fuzziness, a similarity process is started to find the threshold point. A significant contrast between objects and background is assumed. Histogram equalization is used in images having small contrast difference.

Paper [19] presents an adaptive algorithm for efficient document image binarization with low computational complexity and high performance. This is particularly suitable for use in portable devices such as PDA, mobile phones which are marked by their limited memory space and low computational capability. This method divides the document image into several blocks by integrating the concept of global and local methods. After that a threshold surface is constructed based on the diversity and the intensity of each region to derive the binary image. Experimental results show the effectiveness of the proposed method.

A binarization method is presented in [20] based on edge information for video text images. It attempts to handle images with complex background with low contrast. The contour of the text is detected, after that local thresholding method is used to look for the inner side of the contour; subsequently, the contours of the characters are filled up to form characters that are recognizable to OCR software.

A new document image binarization technique is presented in [21], as an improved version of the adaptive logical-level technique (ALLT). The original ALLT makes use of fixed windows for extracting essential features (e.g., the character stroke width). However, there are possibilities of characters with several different stroke widths within a region. This may lead to erroneous results. In [21], local adaptive binarization is used as a guide to adaptive stroke width detection. The skeleton and the contour points of the binarization output are combined to identify the stroke width locally. In addition, an adaptive local parameter is defined that enhances the characters and improves the overall performance achieving more accurate binarization results for both handwritten and printed documents with a particular focus on degraded historical documents.

In [22], the authors proposed a new technique for the validation of document binarization algorithms. Authors claim that the proposed method is simple in its implementation and can be performed on any binarization algorithm since it does not require anything more than the binarization stage. As a demonstration of the proposed technique, we use the case of degraded historical documents. The proposed technique is evaluated with 30 binarization algorithms for performance comparison.

Images with two dominant intensity levels are subjected to manual thresholding a ease. For automatic image thresholding, most of the effective techniques are either too complex or too eager of computer resources.

The balanced histogram thresholding method [23] is a very simple method used for automatic image thresholding. Like Otsu's method [8], this is a histogram-based thresholding method. Assuming that the image is divided into two main classes: the background and the foreground, this method tries to find the optimum threshold level that divides the histogram in two classes. This method weighs the histogram, checks which of the two sides is heavier, and removes weight from the heavier side until it becomes the lighter. It repeats the same operation until the edges of the weighing scale meet. This method may have problems when dealing with very noisy images, because the weighing scale may be misplaced. The problem can be minimized by ignoring the extremities of the histogram.

Evaluation of document image binarization techniques is a tedious task that is mainly performed by human experts or by involving an OCR engine. Paper [24] presents a methodology for objective evaluation of document image binarization algorithms. The methodology aims at reducing the human interference in the construction of the ground truth and testing. A skeletonized ground truth image is created by the user following a semiautomatic procedure. The estimated ground truth image can aid in evaluating the binarization result in terms of recall and precision as well as to further analyze the result by calculating broken and missing text, deformations, and false alarms.

Paper [25] presents a real-time adaptive using the integral image of the input. The technique proposed is robust to illumination changes in the image suitable for processing live video streams at a real-time frame-rate which makes it suitable for the interactive applications.

2.2 Recent Works

In Sect. 2.1, we have discussed the broad area of our research by citing some of the most significant works that have shaped the evolution in the relevant areas. In this section, the state-of-the-art for image binarization methods is discussed for all the areas considered in this work.

Binarization is an essential step for document image analysis. In general, different available binarization techniques are implemented for different types of binarization problems.

In [26], a learning framework for the optimization of the binarization methods is introduced, which is designed to determine the optimal parameter values for a document image. The framework works with any binarization method performs three main steps: extracts features, estimates optimal parameters, and learns the relationship between features and optimal parameters. An approach is proposed to generate numerical feature vectors from 2D data. The statistics of various maps are extracted and then combined into a final feature vector, in a nonlinear way. The optimal behavior is learned using support vector regression (SVR). The experiments are done using grid-based Sauvola's method and Lu's method on the DIBCO2009 and DIBCO2010 datasets.

A pixel-based binarization evaluation methodology for historical handwritten/machine-printed document images is presented in [3]. In the evaluation scheme in [3], the recall and precision evaluation measures are properly modified using a weighting scheme that diminishes any potential evaluation bias. Additional performance metrics of the proposed evaluation scheme consist of the percentage rates of broken and missed text, false alarms, background noise, character enlargement, and merging. The validity of the method is justified by several experiments conducted in comparison with other pixel-based evaluation measures.

An image binarization technique is proposed in [27] for degraded document images that takes into consideration the adaptive image contrast. The adaptive image contrast is a combination of the local image contrast and the local image gradient that is tolerant to text and background variation caused by different types of document degradations. An adaptive contrast map is first constructed for an input-degraded document image. The contrast map is then binarized and combined with Canny's edge map to identify the text stroke edge pixels. The document text is further segmented by a local threshold that is estimated based on the intensities of detected text stroke edge pixels within a local window. It has been tested on three public datasets achieving accuracies of around 90 %.

There are many challenges addressed in handwritten document image binarization, such as faint characters, bleed-through, and large background ink stains. Usually, binarization methods cannot deal with all the degradation types effectively. Motivated by the low detection rate of faint characters in binarization of handwritten document images, a combination of a global and a local adaptive binarization method at connected component level is proposed in [4] that aims in an improved overall performance. Initially, background estimation is applied along with image normalization based on background compensation. Afterward, global binarization is performed on the normalized image. In the binarized image, very small components are discarded and representative characteristics of a document image such as the stroke width and the contrast are computed. Furthermore, local adaptive binarization is performed on the normalized image taking into account the aforementioned characteristics. Finally, the two binarization outputs are combined at connected component level. Authors report good performance after extensive testing on the DIBCO series datasets which include a variety of degraded handwritten document images.

An adaptive binarization method inspired by Otsu's method is introduced in [1]. The method, called AdOtsu, uses the estimated background (EB) as a priori information to differentiate between text and non-text regions. The estimated background values are calculated in a boot-strap process implicitly incorporating the proposed binarization method. Also, a priori structural information, including the average stroke width and the average text height, is used to adapt the method on the input document image and to make it parameter-less. The method is generalized to a multi-scale binarization, which enables it to separate interfering patterns from the true text using higher scales. Postprocessing corrections, both topological and clustering, are considered to improve the final output.

Paper [5] proposes another algorithm for the binarization of degraded document images. The image is mapped into a 2D feature space in which the text and background pixels are separable, and then this feature space is partitioned into small regions. These regions are labeled as text or background using the result of a basic binarization algorithm applied on the original image. Finally, each pixel of the image is classified as either text or background based on the label of its corresponding region in the feature space.

An adaptive binarization method for historical manuscripts and degraded document images is reported in [6]. The method is based on maximum likelihood (ML) classification using a priori information and the spatial relationship on the image domain. The method performs a decision of thresholding based on a probabilistic model. It recovers the main text in the document image, including low intensity and weak strokes from an initialization map (under-binarization) containing only the darkest part of the text. Fast and robust local estimation of text and background features is obtained using grid-based modeling and in-painting techniques; afterward, the ML classification is performed to classify pixels into two classes (black and white). This method preserves weak connections and provides smooth and continuous strokes due to its correlation-based nature. Performance is evaluated both subjectively and objectively against standard databases. The method produces competitive results with state-of-the-art methods presented in the DIBCO2009 binarization contest.

The majority of binarization techniques are complex and are compounded from filters and existing operations. However, the few simple thresholding methods available cannot be applied to many binarization problems. In [7], a local binarization method is presented based on a simple, novel thresholding method with dynamic and flexible windows. The method is tested on selected samples of DIBCO 2009 benchmark dataset.

An adaptive water flow model for the binarization of degraded document images is presented in [28]. In this approach, the image surface is regarded as a three-dimensional terrain and water is poured on it. The water finds the valleys and fills them. The algorithm controls the rainfall process, pouring the water, in such a way that the water fills up to half of the valley depth. After stopping the rainfall, each wet region represents one character or a noisy component. To segment each character, the wet regions are labeled and regarded as blobs. Some of the blobs represent noisy components. A multilayer perceptron is trained to label each blob as either text or non-text. The algorithm is shown to preserve stroke connectivity. Experimental verification shows superior performance against six well-known algorithms on three sets of degraded document images with uneven illumination.

It is evident from the discussion in this chapter that there is a need of a binarization algorithm that would work well for both document and graphic images. In addition to this, we need a methodology for generating the reference image for quantitative evaluation among different image binarization methods. In [Chap. 3](#) of this text, we have documented works that addresses these issues.

References

1. Moghaddam, R.F., Cheriet, M.: AdOtsu: an adaptive and parameter less generalization of Otsu's method for document image binarization. *Pattern Recogn.* **45**(6), 2419–2431 (2012)
2. Gatos, B., Pratikakis, I., Perantonis, S.J.: Adaptive degraded document image binarization. *Pattern Recogn.* **39**(3), 317–327 (2006)
3. Ntirogiannis, K., Gatos, B., Pratikakis, I.: Performance evaluation methodology for historical document image binarization. *IEEE Trans. Image Process.* **22**(2), 595–609 (2013)
4. Ntirogiannis, K., Gatos, B., Pratikakis, I.: A combined approach for the binarization of handwritten document images. *Pattern Recogn. Lett.* **35**, 3–15 (2014). (ISSN 0167-8655, <http://dx.doi.org/10.1016/j.patrec.2012.09.026>)
5. Valizadeh, M., Kabir, E.: Binarization of degraded document image based on feature space partitioning and classification. *Int. J. Doc. Anal. Recogn. (IJDAR)* **15**(1), 57–69 (2012)
6. Hedjam, R., Moghaddam, R.F., Cheriet, M.: A spatially adaptive statistical method for the binarization of historical manuscripts and degraded document images. *Pattern Recogn.* **44**(9), 2184–2196 (2011)
7. Bataineh, B., Abdullah, S.N.H.S., Omar, K.: An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. *Pattern Recogn. Lett.* **32**(14), 1805–1813 (2011)
8. Otsu, N.: A threshold selection method from gray-level histogram. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979)
9. Bernsen, J.: Dynamic thresholding of gray level images. In: Proceedings of International Conference on Pattern Recognition (ICPR), pp. 1251–1255 (1986)
10. Gatos, B., Ntirogiannis, K., Perantonis S.J.: Improved document image binarization by using a combination of multiple binarization techniques and adapted edge information. In: Proceedings of 19th International Conference on Pattern Recognition (ICPR), pp. 1–4 (2008)
11. Johannsen, G., Bille, J.: A threshold selection method using information measures. In: 6th International Conference on Pattern Recognition, pp. 140–143 (1982)
12. Kapur, N.J., Sahoo, P.K., Wong, C.K.A.: A new method for gray-level picture thresholding using the entropy of the histogram. *J. Comput. Vis. Graph. Image Process.* **29**(3), 273–285 (1985)
13. Sauvola, J., Pietikainen, M.: Adaptive document image binarization. *Pattern Recogn.* **33**(2), 225–236 (2000)
14. Niblack, W.: An introduction to digital image processing, pp. 115–116. Prentice Hall, Eaglewood Cliffs (1986)
15. Kittler, J., Illingworth, J.: Minimum error thresholding. *Pattern Recogn.* **19**(1), 41–47 (1986)
16. Ridler, T., Calvard, S.: Picture thresholding using an iterative selection method. *IEEE Trans. Syst. Man Cyber.* **8**(8), 630–632 (1978)
17. Moghaddam, R.F., Cheriet, M.: A multi-scale framework for adaptive binarization of degraded document images. *Pattern Recogn.* **43**(6), 2186–2198 (2010)
18. Lopes, N.V., Mogadouro do Couto, P.A., Bustince, H., Melo-Pinto, P.: Automatic histogram threshold using fuzzy measures. *IEEE Trans. Image Process.* **19**(1), 199–204 (2010)
19. Pai, Y.T., Chang, Y.F., Ruan, S.J.: Adaptive thresholding algorithm: efficient computation technique based on intelligent block detection for degraded document images. *Pattern Recogn.* **43**(9), 3177–3187 (2010)
20. Zhou, Z., Li, L., Tan, C.L.: Edge based binarization for video text images. In: Proceedings of 20th International Conference on Pattern Recognition (ICPR), pp. 133–136 (2010)
21. Ntirogiannis, K., Gatos, B., Pratikakis, I.: A modified adaptive logical level binarization technique for historical document images. In: Proceedings of 10th International Conference on Document Analysis and Recognition, pp. 1171–1175 (2009)
22. Stathis, P., Kavallieratou, E., Papamarkos, N.: An evaluation technique for binarization algorithms. *J. Univers. Comput. Sci.* **14**(18), 3011–3030 (2008)
23. Anjos, A., Shahbazkia, H.: Bi-level image thresholding—a fast method. *Biosignals* **2**, 70–76 (2008)

24. Ntirogiannis, K., Gatos, B., Pratikakis, I.: An objective evaluation methodology for document image binarization techniques. In: 8th IAPR Workshop on Document Analysis Systems (2008)
25. Bradley, D., Roth, G.: Adaptive thresholding using the integral image. *J. Graph. Tools* **12**(2), 13–21 (2007)
26. Cheriet, M., Moghaddam, R.F., Hedjam, R.: A learning framework for the optimization and automation of document binarization methods. *Comput. Vis. Image Underst. (CVIU)* **117**(3), 269–280 (2013)
27. Su, B., Lu, S., Tan, C.L.: Robust document image binarization technique for degraded document images. *IEEE Trans. Image Process.* **22**(4), 1408–1417 (2013)
28. Morteza, V., Ehsanollah, K.: An adaptive water flow model for binarization of degraded document images. *Int. J. Doc. Analysis Recogn. (IJDAR)* **16**(2), 165–176 (2013)

Chapter 3

A New Image Binarization Technique Using Iterative Partitioning

Abstract In this chapter, a new technique for image binarization is presented and discussed intensively. The worked-out algorithm is mainly based on the idea of iterative partitioning. The authors show how this approach outperforms the existing and widely used binarization methods in terms of accuracy. The algorithm is introduced with its computer implementation to show how it performs in practice.

Keywords Iterative partitioning • Histogram analysis • Image partitioning • Noise reduction • Histogram peak • Multi-modal histogram • Reference image • Majority voting

3.1 Image Binarization Using Iterative Partitioning

In this section, a new method is presented for image binarization using iterative partitioning. The iterative partitioning method is tested for binarization of both document and graphic images. The quantitative comparisons with other standard methods reveal that the proposed method outperforms existing widely used binarization methods in terms of accuracy of binarization. The experimental results further establish the superiority of iterative partitioning method, especially for degraded document and graphic images. The new method also shows good noise immunity. The proposed algorithm is suitable for a multi-core processing environment as it can be split into multiple parallel units of executions after the initial partitioning.

3.1.1 Motivation of the Work

A number of image binarization algorithms have been proposed by researchers over past four decades as found in the literature [1, 2]. However, no single algorithm is found to be suitable for processing both document images (that contain mostly

printed and handwritten textual matters with other objects, e.g., logos and chart) and graphic images (a photograph, frames from videos and other images). The reason is that a binarization method that is suitable for processing document images has to deal with various artifacts such as ink seepage, non-uniform illumination, and noises incorporated by poor digitization. On the other hand, graphic images normally do not contain this kind of artifacts, rather different shades of gray levels.

In this work, a novel binarization method (binarization using iterative partitioning) is presented. The proposed method is suitable for processing both document and graphic images. The new method is compared with some standard binarization approaches on images taken from standard image database [3]. Multiple quantitative measures have been used toward relative performance evaluation. The results prove effectiveness of the new iterative partitioning method.

3.1.2 Proposed Methodology: Binarization Using Iterative Partitioning

The proposed method [4] is an improvisation on the Otsu's method [2] for image binarization. Otsu's method is a fast optimal thresholding that utilizes the concept of between-class variance in calculating the threshold from the whole image. In this process, this method makes use of one-dimensional histograms which lack spatial relationship between the pixels of an image. This results in very poor and incorrect segmentation of foreground and background when the image to be binarized contains noises.

The idea of the new method is to keep the spatial relationship between pixels by processing pixels in a local neighborhood by iteratively segmenting the whole image into logical partitions, depending on the intensity distributions of the pixels.

The histogram of the image to be binarized is first calculated. If a multi-modal histogram is generated, the image is partitioned into four logical sub-images and image histograms are calculated for each of the sub-images. If a multi-modal histogram is obtained for a partition, it is further divided into four rectangular partitions. This process of partitioning continues as long as either the size of each sub-image after partitioning is larger than or equal to some predefined size or for any partition, a bimodal histogram is achieved. A bimodal histogram of a sub-image implies that the pixel intensities tend to be distributed among two major groups (foreground object and background), so this is the ideal situation for computing a threshold. Therefore, optimal threshold is calculated for this sub-image. After partitioning is done, threshold is computed by maximizing the between-class variance of intensity levels of all the pixels in each partition.

The histogram of an image shows the distribution of gray-level intensities of different pixels. The range of pixel intensity level is between 0 and 255 for a grayscale image (8 bits/pixel). Here, 0 and 255 represent black and white pixels, respectively. All the gray levels in between 0 and 255 are shades of gray. When the histogram of an image has at most two sharp peaks (bimodal), the choice of a threshold to

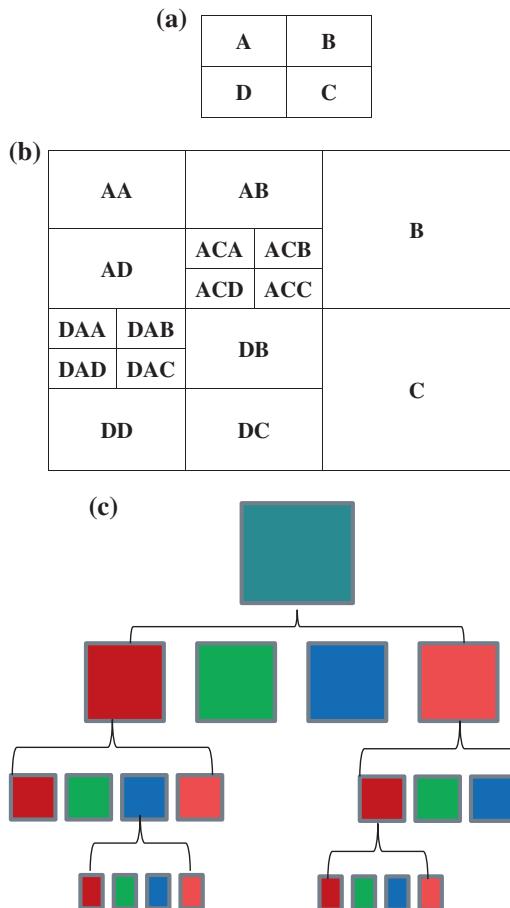


Fig. 3.1 The process of iterative partitioning **a** arrangement of partitions, **b** an arbitrary partitioning of an image, and **c** iterative partitioning in a treelike representation (Source [5])

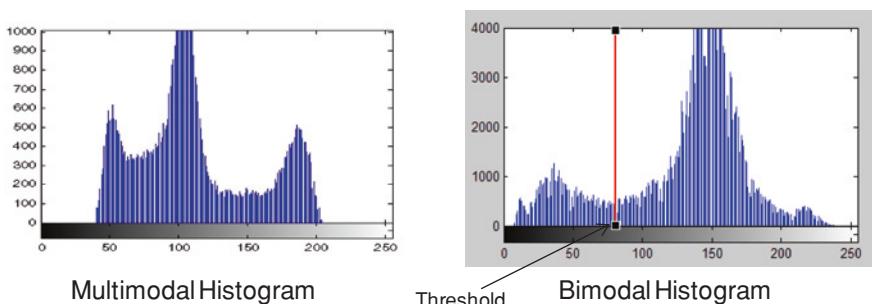


Fig. 3.2 Histogram analysis (Source [5])

binarize the image becomes trivial. This implies that the pixel intensities tend to be distributed among two major groups, one representing the object and the other representing the background. In such a situation, the midpoint between two peaks gives the threshold. This is an ideal case to calculate a threshold for binarization. However, when the number of peaks is more than two (multi-modal), then according to the proposed algorithm, the image is further divided into four rectangular partitions. The process continues when the sub-image size after partitioning is larger than or equal to some predefined size. All four logical partitions (sub-images) are treated separately in a similar fashion until a terminating condition is reached.

The concept of iterative partitioning has been depicted in Fig. 3.1. Figure 3.2 shows intensity distributions for images with multimodal and bimodal histograms. In Fig. 3.2 the value of an ideal threshold is also indicated by a vertical red line located at the valley of the bimodal histogram.

Otsu's method is known to produce very good results for the images with pixel values having a tendency of clustering toward two major gray levels. The histograms of such images yield only two sharp peaks. Thus, in a partition of the image, if the histogram has a maximum of two peaks, then this is logical to apply Otsu's method in that particular partition. On the other hand, if the number of peaks is more than two, then Otsu's method may not give satisfactory result for that partition. Thus, it may be partitioned again. Some parameters are to be defined before formally presenting the new algorithm.

Partition Ratio (PR): Let avg denote the average grayscale value of all pixels in any partition (sub-image). The *partition ratio* (PR) is defined as the ratio of the number of pixels having grayscale value greater than avg to the number of pixels having grayscale value less than avg present in any image partition.

Partition Parameter (PP): PP is used as terminating criteria of the iterative partitioning process. PP is obtained by multiplying *PR* by a constant value K . The value of constant depends on the type of image to be binarized (e.g., graphic or document image).

Algorithm for Iterative Partitioning

Begin

1. Input a gray scale image $I(x,y)$; x and y being spatial coordinates of the image.
2. /* The number of sharp peaks for the given image is computed using procedure ComputeSharpPeak() */
 $N = \text{ComputeSharpPeak}(\text{Image } I);$
3. /* The original image is directly binarized here using Otsu's method if the image does not have much variance in gray scale values for pixels. */
If $N \leq 2$

Find the threshold value $Th = \text{Compute_Threshold}(Image I)$;

Binarize the entire image and terminate the program;

Endif.

4. Partition the image into four rectangular partitions; A , B , C and D . (Fig. 4.1)

5. For each partition P (i.e., sub-images A , B C and D)

Repeat

5.1. /* Calculate sharp peaks of the current partition from histogram. */

$N = \text{ComputeSharpPeak}(\text{Partition } P)$;

5.2. If the number of sharp peaks $N > 2$ then

Calculate the Partition Ratio (PR) of the sub-image;

Calculate Partition Parameter, $PP = K * PR$;

If the number of rows and columns of the current partition > PP

Go to Step 4 to partition the current-partition P ;

Endif

Endif

5.3. /* The procedure reaches this point of execution when either the current partition has less than or equal 2 sharp peaks, or the size of the current partition becomes smaller than the predefined threshold of PP . */

Find the threshold for the partition $Th = \text{Compute_Threshold}(Image I)$;

Binarize the current partition P using Th .

End.

Procedure Integer ComputeSharpPeak (Image I)

/* Algorithm to calculate the number of sharp peaks from histogram of image. */

Begin

1. Calculate the Histogram of the image (I);
2. Compute the frequency of occurrences of distinct pixels $\text{freq}(k)$ for each gray-scale value k in the closed interval $[0..255]$;

3. /* The previous 2 and next 2 neighbor positions for each gray value k is taken in a circular manner in $p1$, $p2$, $n1$, and in $n2$ respectively. */

For $k = 0$ to 255,

If ($k=0$) then

$p1(k)=255$; $p2(k)=254$; $n1(k)=1$; $n2(k)=2$;

else if ($i=1$) then

$p1(k)=0$; $p2(k)=255$; $n1(k)=2$; $n2(k)=3$;

else if ($i=254$) then

$p1(k)=253$; $p2(k)=252$; $n1(k)=255$; $n2(k)=0$;

else if ($i=255$) then

$p1(k)=254$; $p2(k)=253$; $n1(k)=0$; $n2(k)=1$;

else

$p1(k)=(k-1)$; $p2(k)=(k-2)$; $n1(k)=(k+1)$; $n2(k)=(k+2)$;

Endif

Endfor

4. /* If the frequency of a gray value k is greater than that of its two adjacent left and two adjacent right gray value frequencies then k is identified as a **peak**. */
 $n = 0$; /* n is the number of peaks in the image, initialized to 0*/

For $k=0$ to 255

If $freq(k) > max(freq(p1(k)), freq(p2(k)), freq(n1(k)), freq(n2(k)))$ then

/* $max()$ returns the maximum value for the arguments */

$n = n + 1$; /* a peak is identified for gray value k */

$ph(n) = freq(k)$; /* Value of the i^{th} pick is recorded at $ph(i)$ */

Endif

Endfor

5. /* Average peak frequency is calculated from the peaks recorded in Step 4 */

$$\text{avg-peak} = \sum ph(i) / n;$$

6. A peak is defined as a **Sharp Peak**, where the frequency of occurrences of pixels for the corresponding gray-scale value is greater than the average peak frequency, $avg\text{-peak}$, computed above.

7. Return the number of sharp perks.

End.

Procedure Integer Compute_Threshold (Image I)*/*Algorithm to calculate the optimal threshold for image I. */***Begin**

1. /* Find the number of distinct gray levels k in image I ; also find the minimum (l_{min}) and maximum (l_{max}) intensity level in image I */

$$l_{min} = \min \text{Gray_Level}(I);$$

$$l_{max} = \max \text{Gray_Level}(I);$$

$$k = \text{Distinct_Gray_Level}(I);$$

2. /* Let $P(i)$ be the probability of each distinct gray level i present in image I . Let n_i is the number of pixels with gray level intensity value i and $N = mxn$, total number of pixels in the sub-image I of dimension mxn */

For $i = 1$ to k

$$P(i) = \frac{n_i}{N};$$

Endfor

3. /* Find the number of iterations */

$$no_itr = l_{max} - l_{min} + 1;$$

4. /* Iterate through all possible gray levels in search of the optimal threshold */

For $j = 0$ to no_itr

$$t = l_{min} + j;$$

- 4.1 /* Compute class probabilities of pixels $q_1(t)$ and $q_2(t)$ for threshold t */

$$q_1(t) = 0; q_2(t) = 0;$$

For $i = l_{min}$ to t

$$q_1(t) = q_1(t) + P(i);$$

Endfor

For $i = t+1$ to l_{max}

$$q_2(t) = q_2(t) + P(i);$$

Endfor

- 4.2 /* Calculate class mean */

For $i = l_{min}$ to t

$$\mu_1(t) = (i * P(i)) / q_1(t);$$

Endfor

For i = t+1 to l_{max}

$$\mu_2(t) = (i * P(i)) / q_2(t);$$

Endfor

4.3 / Find between-class variance for the selected threshold t */*

$$Variance_BC(j) = q_1(t)[1 - q_1(t)] [\mu_1(t) - \mu_2(t)]^2;$$

Endfor

5. / Find the gray level t that maximizes the between class variance */*

$$t = Max_Idx(Variance_BC);$$

Return (t);

End

Here, K is a constant in the calculation of the *partition parameter* (PP). The value of K has been selected as 20 for degraded document image binarization. This value has been chosen empirically. Other values (5, 10, 15, 20, 25, 30, etc.) have been tried with, and finally, 20 has been selected as it is giving the best result. The justification behind using PP is as follows: PP controls the size of the sub-image in the iterative partitioning process. If the size of the sub-image is too small, then there is no point in partitioning it again.

On the other hand, if the size of a partition is too large and there are significant variations in the gray levels in that partition, then directly applying thresholding in that partition may result in the significant loss of information. A window size of approximately 20×20 pixels is a good trade-off between the two extreme cases.

Several values of K (40, 60, 80) have been used for experimental purpose with graphic images. The value for $K = 60$ has been found to produce the best result among all. Here, K is the parameter that can be changed to fine-tune the proposed method. The detailed results are provided in the “Experimental Results” section (Figs. 3.16, 3.17).

Calculating Threshold for each Partition: Let there be L number of distinct gray levels in the input image. Let there be a total of n number of pixels in the image. Probability of a pixel with intensity value i is given by $p_i = n_i/n$ for $i = 0, 1, 2, \dots, L - 1$. The whole image can be divided into two groups ($C1$ and $C2$) based on a particular gray value $1 \leq k \leq L - 2$. Probability of set $C1$ is defined as follows:

$$P_1(k) = \sum_{i=0}^k p_i$$

Probability of set C_2 is similarly defined as follows:

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i$$

Mean intensities up to level k are defined as follows:

$$m(k) = \sum_{i=0}^k i \cdot p_i$$

Mean intensities of pixels in the whole image are defined as follows:

$$m_G(k) = \sum_{i=0}^{L-1} i \cdot p_i$$

The between-class variance is defined as follows:

$$\sigma_B^2(k) = P_1(k)[m_1(k) - m_G]^2 + P_2(k)[m_2(k) - m_G]^2$$

As we know, the total variance (σ^2) = within-class variance ($\sigma_W^2(t)$) + between-class variance ($\sigma_B^2(t)$) is constant for an image and independent of t . So, maximizing the between-class variance is the same as minimizing the within-class variance. The threshold is the value of k that maximizes the value of σ_B ; essentially, this is an iterative process which can be checked for each possible value of k from 0 to 255.

Analysis of Time Complexity: Let Otsu's method make I number of iterations for the whole image for computing the threshold. Let, after the partitioning is done by iterative partitioning method, us get a total of N partitions. Let $M = N \times I$.

After the partitioning is done, the calculation of threshold for each partition is an iterative process. However, the number of iterations required for computing threshold for each partition is generally much less than that of what is required by Otsu's method for the whole image. As shown in Fig. 3.3, with every logical partition, the number of distinct gray levels present in a sub-image gets reduced. This suggests that after the partitioning is done, the computation of threshold is much less than M .

Threshold selection takes linear time: $O(L)$, where L is the total number of gray levels present in the image. For P number of partitions and average G number of distinct gray levels per partition, a total of $O(PG)$ iterations are required. So, it is obvious that the time complexity of the proposed method is linear.

3.1.3 Evaluation Measures

Different evaluation techniques for document image binarization are found in the literature. Most of the authors employ recall- and precision-based metrics for comparison. In this work, nine evaluation measures, namely misclassification error (ME), relative foreground area error (RAE), recall, precision, F -measure, mean square error (MSE),



Fig. 3.3 Reduction in distinct gray levels in each sub-image due to partitioning (*Source [5]*)

pixel error rate (PERR), signal-to-noise ratio (SNR), and peak signal-to-noise ratio (PSNR), are used. All of these measures are defined in the following subsections.

3.1.3.1 Misclassification Error

Misclassification error [1] reflects the percentage of background pixels wrongly assigned to foreground and, conversely, foreground pixels wrongly assigned to background. ME can be simply expressed as following equation for the two-class segmentation problems:

$$\text{ME} = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O \cap F_O|}$$

where B_O and F_O denote the background and foreground of the original ground truth (reference) image, B_T and F_T denote the background and foreground area pixels in the test image, and $| \cdot |$ is the cardinality of the set. The ME varies from 0 for a perfectly classified image to 1 for a totally wrongly binarized image.

3.1.3.2 Relative Foreground Area Error

The next comparison is based on a measure for the area feature, the relative foreground area error which is stated below as defined in [1].

$$\begin{aligned} \text{RAE} &= \frac{A_0 - A_T}{A_0} \quad \text{if } A_T < A_0 \\ \dots &= \frac{A_T - A_0}{A_T} \quad \text{if } A_T \geq A_0 \end{aligned}$$

Here, A_0 is the area of reference image, and A_T is the area of thresholded image. Obviously, for a perfect match of the segmented regions, RAE is zero, while if there is zero overlap of the object areas, the penalty is the maximum one.

3.1.3.3 Recall, Precision, and F-Measure

In the context of binarization, the recall, precision, and *F*-measure can be defined as follows:

N_{Relevant} = Number of object pixels in the reference image.

$N_{\text{Retrieved}}$ = Number of object pixels in the binary image.

A = Number of common object pixels in the reference and the binary images.

B = Number of relevant object pixels of the reference image not retrieved
 $= N_{\text{Relevant}} - A$.

C = Number of irrelevant object pixels retrieved in the binary image
 $= N_{\text{Retrieved}} - A$.

Recall is the ratio of the number of relevant object pixels retrieved to the total number of relevant object pixels in the reference image. It is usually expressed as a percentage.

$$\text{Recall} = \frac{A}{A + B} \times 100$$

Precision is the ratio of the number of relevant object pixels retrieved to the total number of irrelevant and relevant object pixels retrieved. It is usually expressed as a percentage.

$$\text{Precision} = \frac{A}{A + C} \times 100$$

F-measure is defined as follows:

$$F\text{-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

A higher value of *F*-measure represents a better result.

3.1.3.4 MSE, PERR, SNR, and PSNR

This section defines the following measures: the mean-squared error (MSE), PERR, the SNR, and PSNR.

Let $x(i, j)$ represent the value of the i th row and j th column pixel in the reference image, and let $y(i, j)$ represent the value of the corresponding pixel in the output image y . Since it is all about black and white images, both values will be either 0 (black) or 255 (white). The local error is $e(i, j) = x(i, j) - y(i, j)$, and the total square error rate will be

$$\text{MSE} = \frac{\sum_i \sum_j e(i, j)^2}{M \times N}$$

Notice that if a pixel is correctly classified, the value of $e(i, j)^2$ will be 0, while if the pixel is wrongly classified, it will be 255^2 . Thus, taking into account the PERR definition, it will be

$$\text{PERR} = \frac{\text{MSE}}{255^2}$$

SNR is defined as the ratio of average signal power to average noise power. For an $M \times N$ image, it is defined as follows:

$$\begin{aligned}\text{SNR(DB)} &= 10 \log_{10} \frac{\sum_i \sum_j x(i, j)}{\sum_i \sum_j (x(i, j) - y(i, j))^2} \\ &= 10 \log_{10} \frac{\sum_i \sum_j x(i, j)}{\text{MSE}} = 10 \log_{10} \frac{\sum_i \sum_j x(i, j)}{\text{PERR} \cdot 255^2}\end{aligned}$$

The peak measure, PSNR is defined as the ratio of peak signal power to average noise power. For a grayscale image, it is defined as follows:

$$\begin{aligned}\text{PSNR(DB)} &= 10 \log_{10} \frac{255^2 \cdot MN}{\sum_i \sum_j (x(i, j) - y(i, j))^2} \\ &= 10 \log_{10} \frac{255^2 \cdot MN}{\text{MSE}} = 10 \log_{10} \frac{MN}{\text{PERR}}\end{aligned}$$

The above-mentioned measures (metrics) will be used for quantitative evaluations of binarization methods.

3.1.4 Experimental Dataset

USC-SIPI [3] graphic image dataset is provided by University of Southern California, Signal and Image Processing Institute. A set of images is taken from this dataset for experimentation purpose. The images are renamed for the sake of ease of reference. All the benchmark images contain different shades of gray-level intensity variations suitable for analyzing the performance of binarization algorithms. The set of test graphic images is shown in Fig. 3.4.

The test-degraded document images are shown in Fig. 3.5. These images are collected from different sources. Most of the images are used by researchers working on degraded document image processing. As shown in Fig. 3.5, the images are contaminated with different kinds of noise degradations such as non-uniform illumination, ink seepage, unwanted shades due to poor digitization of the document, and stains.

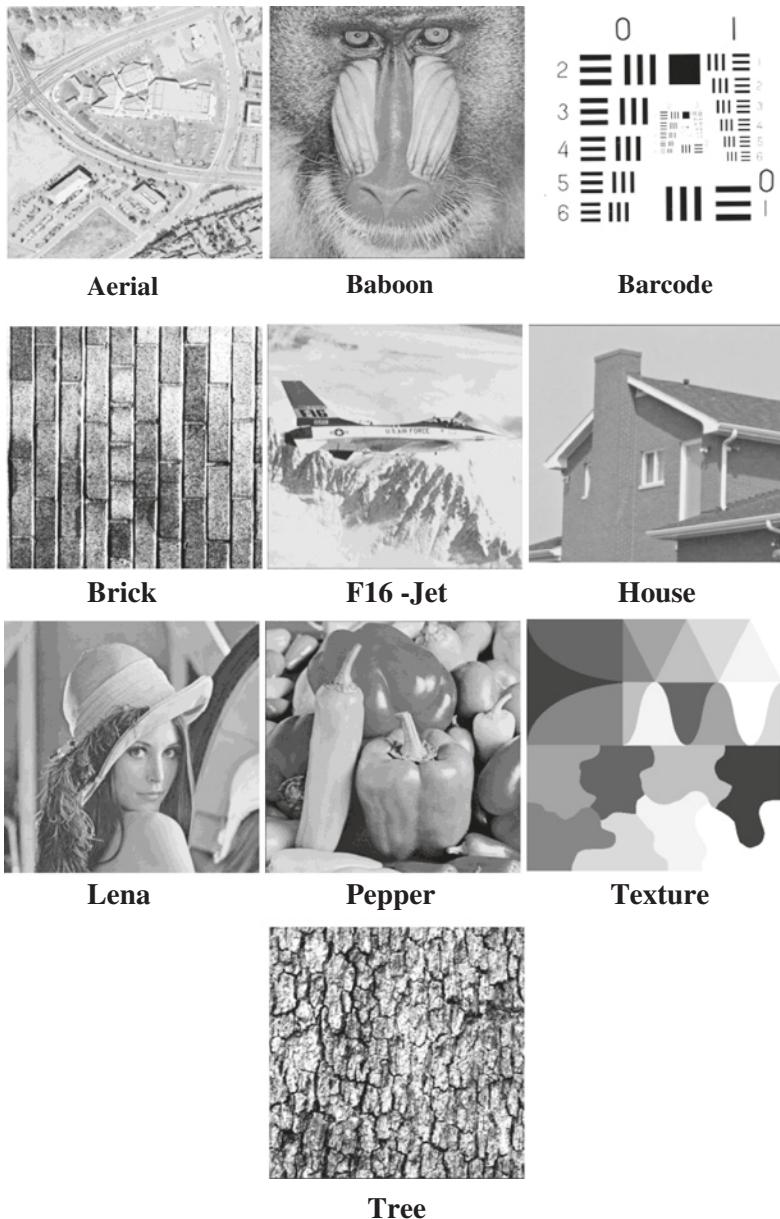


Fig. 3.4 Selected test images from USC-SIPI database (*Source [4]*)

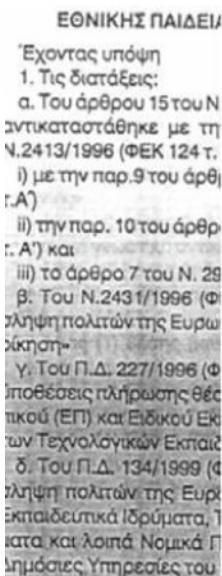


Image 1



Image 3

motivation for this co-segmentation and registration. By reading methods are appropriate of sources, occurring to the majority of data.

Image 4

για σωματινή, στα να πονεί μά τέτων της Μυσημένων. ειατικήν, σταν ακόση τὸν Φεῦγε τὴν ἀμέλειαν καὶ συνηρός εἰναι μάστερ χωράφιον μά τέτο λογεινή ή Σοφία, ὡνάχωσε, κα καλιέργα τὴν κλασίς ἐργασίας, διὰ νὰ κοπία. Εργάζεται για πάντα μὲν δείτην τόπον ὁ δαιμόνιος

Image 5

0	1820	1840	1900	193
0	2210	2240	2300	232
0	0200	0230	0300	032
1030	1100	1120	1140	
0.1510	1540	1600	162	
0.1900	1950	2020	204	

Image 7



Image 6

Fig. 3.5 Test document images (Source [4])

3.1.5 Experimental Verification

A number of degraded document images used for experimental verification purposes have been collected from different sources. Ten different graphic images have been taken from the USC-SIPI [3] database for the purpose of experimental verification. The images from this widely used database are taken as it is without any change in size or quality. The images have been renamed for better readability.

Table 3.1 Performance evaluation for document images based on ME

Image name	Otsu	Niblack	Bernsen	Sauvola	Iterative partitioning
Image1	23.75	8.32	7.42	11.43	0.73
Image2	17.89	16.74	8.22	14.18	4.66
Image3	3.02	7.83	24.38	2.36	1.61
Image4	33.15	7.88	17.44	8.47	3.5
Image5	2.55	5.37	2.68	6.62	0.97
Image6	3.48	15.08	4.38	11.47	2.97
Image7	17.88	8.3	10.47	12.09	0.98
Average	14.53	9.93	10.71	9.51	2.2

3.1.5.1 Creating Reference Image

A reference image is created to measure the performance of different binarization methods.

The proposed new method is compared with four popular and widely used image binarization techniques, namely Otsu [2], Niblack [6], Bernsen [7], and Sauvola [8]. In the present experiment, the reference images are created by following a majority voting scheme as follows: Five binarized images are created for each test image using the five binarization methods (Otsu, Niblack, Bernsen, Sauvola, and the proposed iterative partitioning method). In order to generate the reference image, all the binarized images are consulted pixel by pixel. Each pixel in the reference image is set to 1 if the majority of the five methods agree that the corresponding pixel is a 1 in their respective resultant binarized image. Otherwise, the pixel is set to 0 in the reference image.

A number of document and graphic images have been tested. The graphic images have been taken from the database provided by USC-SIPI [3]. Here, the results of a few selected images are produced.

3.1.5.2 Experimental Results

Results on Document Images

Table 3.1 shows the results of the ME measure on selected seven document images. Some of the images contain only text, and some of them contain both graphic objects and text. Most of the images are characterized by different shades of gray level, non-uniform illumination, shadows, and ink seepage. All the test document images are shown in Fig. 3.5.

Results show that for all the document images, the proposed iterative partitioning method performs better than the rest of the methods. The values of ME are in fact much better than the second closest method for many of the images (e.g., Image1, Image5, and Image7). The minimum value of the corresponding measure has been given in italic for each image. Table 3.1 shows that for all the test images, the proposed method outperforms other methods.

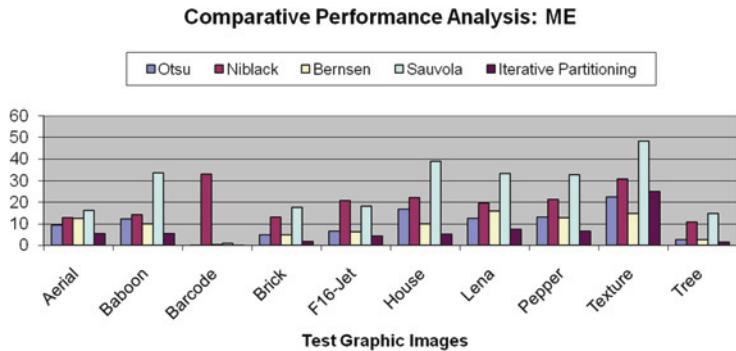


Fig. 3.6 Performance evaluation of binarization methods on ME

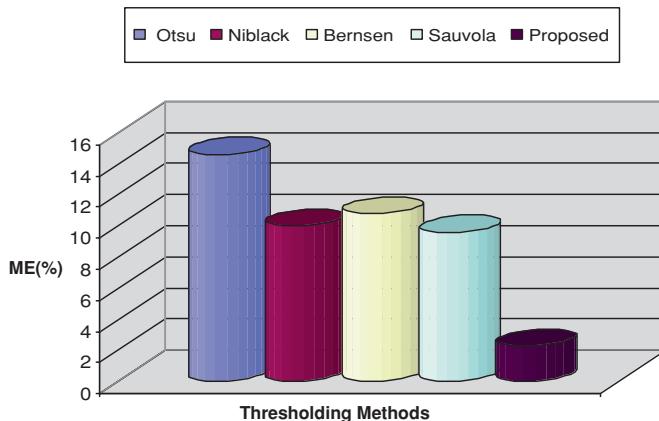


Fig. 3.7 Average performance evaluation of binarization methods on ME

Figure 3.6 presents a plot of how different methods perform for ME for different images on the basis of results shown in Table 3.1.

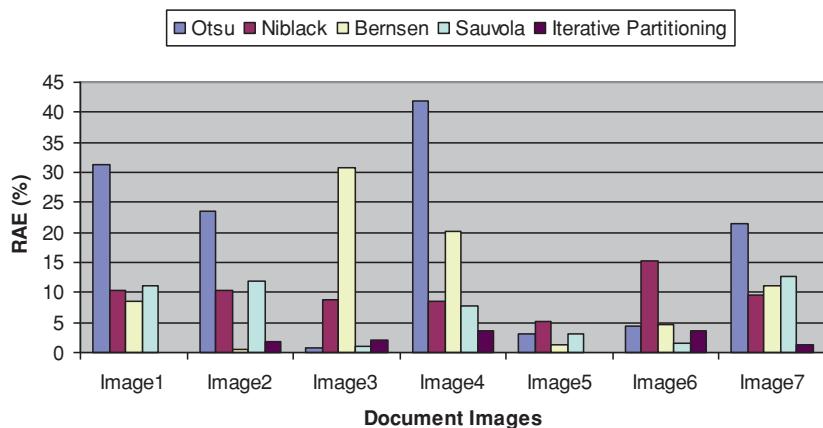
Figure 3.7 shows the average performance of all the binarization method for measure ME. Here, the performances for each method on all seven test images have been aggregated, and then, an average for all seven document images is computed on ME. The plot corresponds to the data in the last row of Table 3.1. According to this plot, the proposed method performs the best as average ME is minimum for it.

Table 3.2 collects the results of similar experiment for RAE measure. In Table 3.2, it is shown that for images 2, 3 and 6, methods proposed by Bernsen, Otsu and Sauvola yield the best performances respectively. However, the overall performance of the proposed method is much better than other four methods.

In Fig. 3.8, the performances of all the binarization methods are plotted with respect to measure RAE (corresponding to data provided in Table 3.2).

Table 3.2 Performance evaluation for document images based on RAE

Image name	Otsu	Niblack	Bernsen	Sauvola	Proposed
Image1	31.32	10.41	8.63	11	0.04
Image2	23.61	10.31	0.57	11.8	1.93
Image3	0.84	8.84	30.86	1.14	1.94
Image4	42.02	8.66	20.18	7.84	3.51
Image5	3.16	5.14	1.2	3.1	0.05
Image6	4.27	15.18	4.73	1.66	3.5
Image7	21.47	9.47	11.17	12.68	1.18
Average	18.09	9.72	11.05	7.03	1.74

**Fig. 3.8** Performance of binarization methods for different images on RAE**Table 3.3** Relative average performance on document images for nine metrics

Performance evaluation metrics	Binarization methods				
	Otsu	Niblack	Bernsen	Sauvola	Iterative partitioning
ME	14.53	9.93	10.71	9.51	2.20
RAE	18.09	9.72	11.05	7.03	1.74
Recall	82.96	91.48	92.57	80.23	99.44
Precision	97.87	97.33	93.03	98.49	95.88
F-measure	89.8	94.32	92.8	88.43	97.62
PERR	14.11	8.31	11.33	15.78	3.69
MSE	9.17	5.40	7.36	10.26	2.39
SNR	31.08	32.79	32.07	29.25	36.08
PSNR	57.13	58.34	57.39	55.43	62.13

Similar experiments are carried out for all the images for nine evaluation measures. The average performance of five binarization methods for all seven document images is summarized in Table 3.3 for nine evaluation measures.

The results in Table 3.3 show that the average performance of this method is better than other four binarization methods for all measures except precision. Iterative

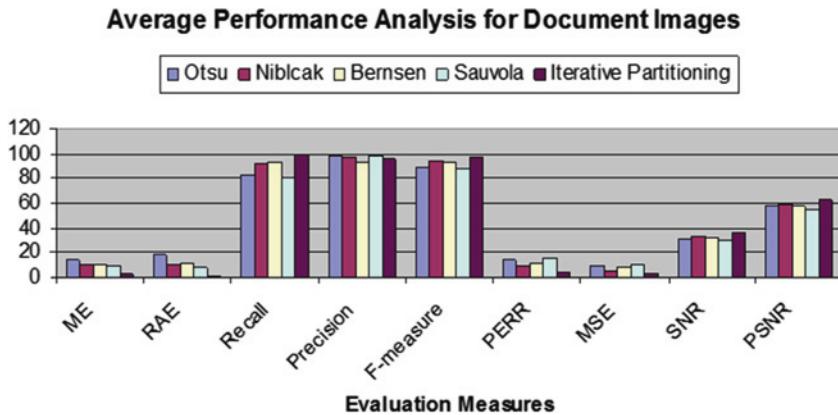


Fig. 3.9 Average performance evaluation for document images

partitioning is not giving the best score for precision; however, the score given by this method is very competitive to others. Besides, the proposed method produces the best scores for recall and the error metrics ME, RAE, PERR, and MSE. The proposed method is also found to yield the best average performance for SNR and PSNR for all the images in the test set. Figure 3.9 shows a visual representation of the average performance for all nine measures on the basis of results provided in Table 3.3.

Figure 3.10 shows comparative results of iterative partitioning method and the one proposed by Otsu [2] for all the degraded document images. As evidenced by the results, for all the test images, iterative partitioning outperforms Otsu's method.

Results on Graphic Images

The selected ten graphic images (names of the files have been changed for the sake of improved readability) from USC-SIPI database [3] are shown in Fig. 3.4. The following tables present the results obtained for the graphic images.

Table 3.4 shows the results of applying the proposed algorithm and quantitative values obtained from other methods for graphic images. Except for the image Texture, for which Bernsen's method gives the best result, the proposed method exhibits better performance (for each entry, the best score is given in italic).

Similar to what is done for document images, the relative average performance for the five binarization methods is computed for graphic images on ME as well. The performances for each method on different graphic images have been aggregated, and then, an average for all ten images is computed. This corresponds to the data in the last row of Table 3.4. It is evident from the result that the proposed method performs the best for graphic images as well, for the ME measure.

The bar chart presented in Fig. 3.11 shows the performance of the five methods for all the selected graphic images for ME measure. It also shows the superiority of the proposed iterative partitioning method to other four methods.



Fig. 3.10 Comparison with Otsu's method

Table 3.4 Performance evaluation for graphic images based on ME

Image name	Otsu	Niblack	Bernsen	Sauvola	Iterative partitioning
Aerial	9.37	12.69	12.58	16.24	5.34
Baboon	12.17	14.19	9.89	33.69	5.51
Barcode	0.03	33	0.27	0.98	0.02
Brick	4.9	13.11	4.9	17.57	1.8
F-Jet-16	6.66	20.88	6.27	18.13	4.32
House	16.91	22.24	9.84	39	5.08
Lena	12.54	19.62	15.79	33.28	7.42
Pepper	13.11	21.43	12.85	32.58	6.43
Texture	22.5	30.65	14.67	48.39	25.01
Tree	2.57	10.87	2.57	14.76	1.51
Average	10.08	19.87	8.96	25.46	6.24

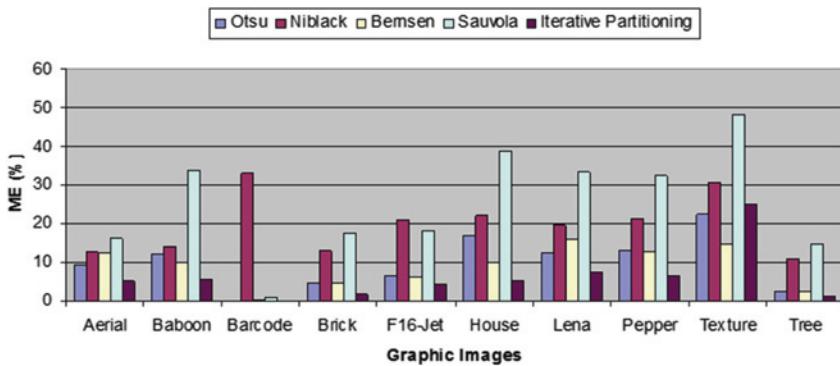
**Fig. 3.11** Performance of binarization methods for different images on ME

Table 3.5 shows the results of applying the proposed algorithm and quantitative values obtained from other methods using the RAE measure. The average performance of the proposed method is better than that of others.

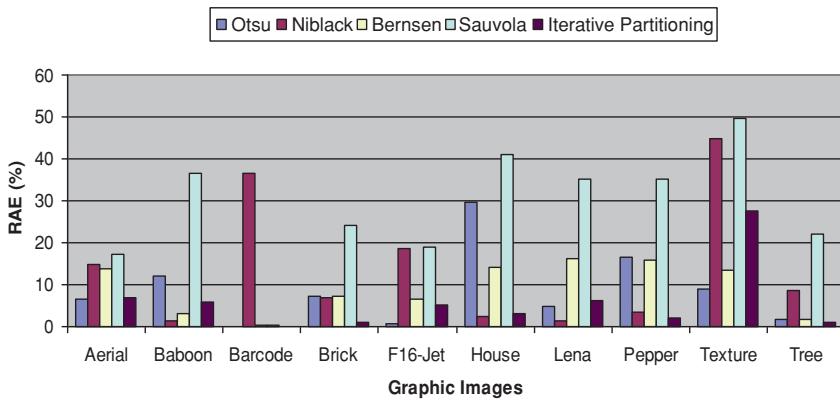
In Fig. 3.12, a comparative plot is presented corresponding to the data provided in Table 3.5. This plot clearly shows how the proposed method beats most of other four binarization methods for different images. Similar to what is done earlier for ME, the relative average performance for the five binarization methods is summarized in the last row of Table 3.5 on RAE as well for the ten graphic images. The performances for each method on the images have been aggregated, and then, an average for all the images is computed. As evident by the results, the proposed method performs the best on RAE for graphic images as well.

The results of binarization on graphic images have been summarized in Table 3.6 for the nine metrics. As before, the detailed results for these nine metrics are not presented to keep the manuscript brief. Table 3.6 records the average performances of all the binarization methods for the given test set of ten benchmark images.

In Tables 3.4 and 3.5, for ME and RAE, respectively, it is found that some of the existing techniques have occasionally produced better scores than the proposed

Table 3.5 Performance evaluation for graphic images based on RAE

Image name	Otsu	Niblack	Bernsen	Sauvola	Iterative partitioning
Aerial	6.42	14.97	13.96	17.31	6.88
Baboon	11.98	1.21	3.20	36.42	5.89
Barcode	0.00	36.47	0.31	0.35	0.01
Brick	7.17	7.03	7.17	24.25	0.99
F-Jet-16	0.77	18.57	6.63	19.03	5.16
House	29.67	2.57	13.98	40.87	3.24
Lena	4.69	1.37	16.15	35.24	6.25
Pepper	16.46	3.41	15.82	35.01	1.91
Texture	9.04	44.99	13.28	49.76	27.72
Tree	1.74	8.74	1.74	22.16	1.15
Average	8.79	13.93	9.22	28.04	5.92

**Fig. 3.12** Performance evaluation of the binarization methods on RAE**Table 3.6** Relative average performance on graphic images for seven metrics

Performance evaluation metrics	Binarization methods				
	Otsu	Niblack	Bernsen	Sauvola	Iterative partitioning
ME	10.08	19.87	8.96	25.46	6.24
RAE	8.79	13.93	9.22	28.04	5.92
Recall	91.20	79.68	93.76	66.70	95.93
Precision	91.2	79.68	93.76	66.7	95.93
<i>F</i> -measure	91.2	79.68	93.76	66.7	95.93
PERR	7.75	20.70	8.72	24.83	7.34
MSE	5.041	13.46	5.66	16.14	4.76
SNR	40.53	33.43	39.13	31.59	41.4
PSNR	67.02	60.07	65.34	59.36	67.89

Average Performance Analysis for Graphic Images

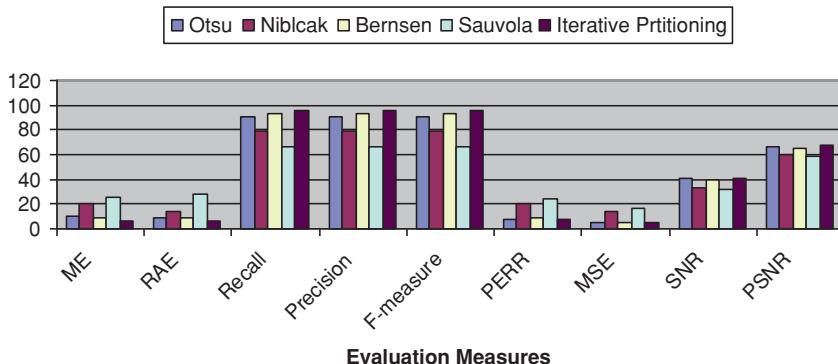


Fig. 3.13 Average performance evaluation for graphic images

iterative method for some graphic images. A similar phenomenon is witnessed for all these nine metrics. This shows that any single binarization method does not stand superior for all the test images for all evaluation metrics. However, it is evident from the results summarized in Table 3.6 that the proposed method produces best average performance for each of the nine metrics considered for evaluation performance of all the binarization methods for graphic images.

Figure 3.13 shows a visualization of the average performances based on the results presented in Table 3.6.

Figure 3.14 shows the binarized images produced by all five methods compared in this report for the image *F-Jet-16*. It is evident that Niblack's method fails to make proper foreground–background segmentation. It wrongly detects background pixels as foreground objects. Sauvola's method also misses a significant portion of the foreground object. However, Bernsen's method provides a good segmentation. Results of Otsu's method and that of iterative partitioning method are visibly not distinguishable.

Figure 3.15 shows the results of all the five methods for the test image *Lena*. Here again, it is seen that the performances of Niblack misclassify a significant portion of background as foreground and Sauvola's method misses a significant portion of the foreground. Bernsen's method is performing better than the methods by Niblack and Sauvola. Performance of Otsu's method is good. Iterative partitioning also segments more of the foreground pixels properly. However, some unwanted blocklike effects are there in the results. These artifacts appear due to the local processing of the different portions of the original image.

3.1.5.3 Selecting Parameter for the Algorithm

Choosing the value of K : The parameter K is very important for computing the *partition parameter* (PP). The results presented in this section justify why the

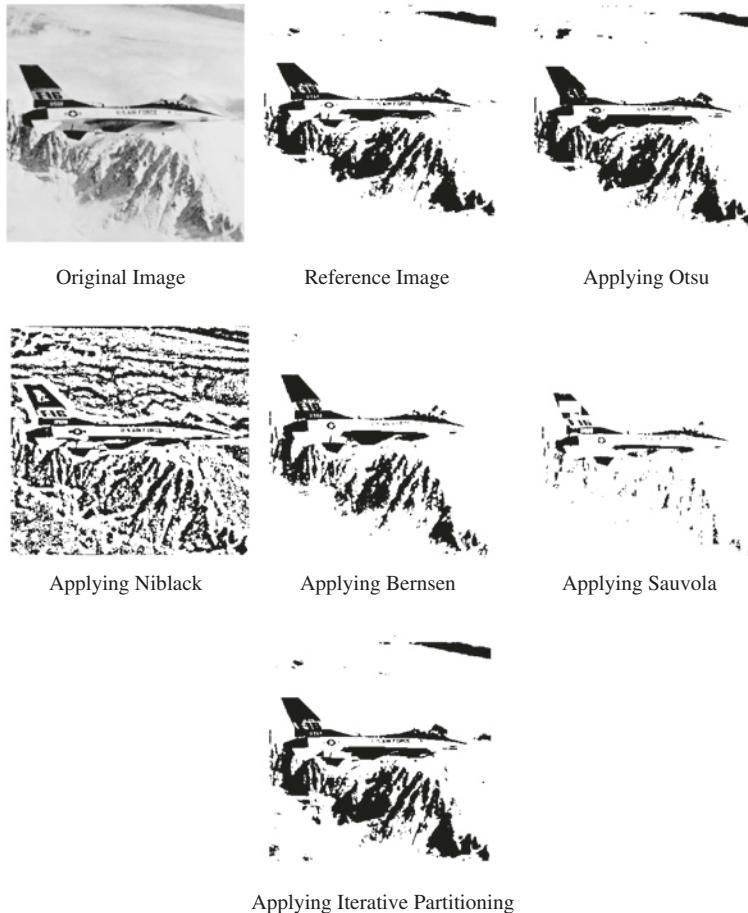


Fig. 3.14 Results of the binarization methods for F-Jet-16 (Source [4])

value of K is chosen to be 20 for binarizing document images using iterative partitioning.

3.1.5.4 Performance for Document Images

Table 3.7 shows how the result of the proposed method varies for document images with different values of K with respect to ME and RAE measures. The optimal result (given in italic) is obtained for $K = 20$ for document images. So, this value is used for experimenting with the document images.

Figure 3.16 shows this variation for ME and RAE corresponding to data presented in Table 3.7.



Fig. 3.15 Results of the binarization methods for Lena (Source [4])

Table 3.7 Performance evaluation on document images against K

Value of K	ME	RAE
$K = 5$	5.57	4.13
$K = 10$	2.41	1.51
$K = 15$	1.89	1.53
$K = 20$	1.63	1.41
$K = 25$	1.76	1.69
$K = 30$	1.88	1.82

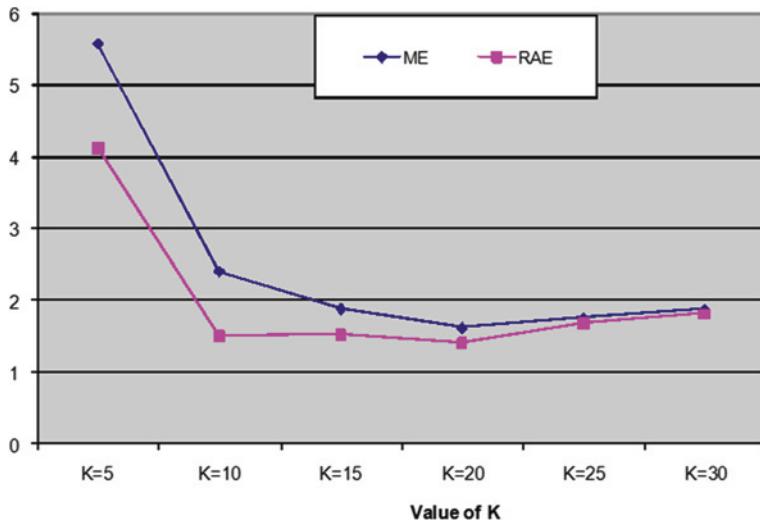


Fig. 3.16 Performance evaluation for document images against K (Table 4.7)

Table 3.8 Performance evaluation for graphic images against K

Value of K	ME	RAE
$K = 40$	6.69	5.88
$K = 60$	5.8	5.67
$K = 80$	6.24	5.92

Performance for Graphic Images

Table 3.8 shows how the result of the proposed method varies for graphic images with different values of K with respect to both ME and RAE measures. This is graphically presented in Fig. 3.17. The optimal result (given in italic) is obtained for $K = 60$ for graphic images. So this value is chosen for experimenting with the graphic images.

3.1.5.5 Analysis of Noise Immunity of the Proposed Method

The noise immunity of the proposed method has also been tested for both document and graphic images for salt-and-pepper and Gaussian noises. The results are presented in this section. Tables 3.9 and 3.10 present the average performance of the proposed method along with other four methods on document images with salt-and-pepper and Gaussian noises, respectively.

Results show that the average performance of the proposed method outperforms other four binarization methods in the presence of salt-and-pepper noises in terms of both ME and RAE measures. A similar inference may be obtained from the results for and Gaussian noises as recorded in Table 3.10.

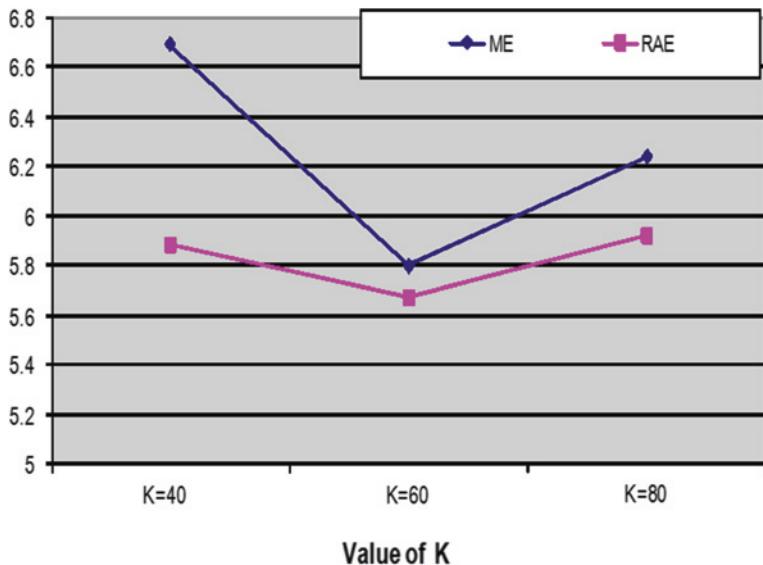


Fig. 3.17 Performance evaluation for graphic images against K (Table 4.8)

Table 3.9 Performance evaluation on document images for salt-and-pepper noise

Noise level	Otsu		Niblack		Bernsen		Sauvola		Iterative partitioning	
	ME	RAE	ME	RAE	ME	RAE	ME	RAE	ME	RAE
0.02	14.63	16.78	5.38	4.64	10.47	10.82	8.64	6.10	1.24	1.00
0.03	14.06	16.40	5.05	3.88	10.01	10.54	8.23	5.76	1.22	0.96
0.05	12.50	14.87	4.66	3.21	9.18	9.80	7.27	5.11	1.15	0.94

Table 3.10 Performance evaluation on document images for Gaussian noise

Noise level	Otsu		Niblack		Bernsen		Sauvola		Iterative partitioning	
	ME	RAE	ME	RAE	ME	RAE	ME	RAE	ME	RAE
0.02	12.84	10.91	9.80	10.40	11.92	11.46	15.10	13.91	2.34	1.94
0.03	12.67	10.79	9.82	10.66	11.77	11.31	14.83	13.51	2.26	1.84
0.05	12.35	10.41	9.78	10.76	11.53	10.63	14.81	13.39	2.13	1.51

Tables 3.11 and 3.12 accumulate the results of average performance of the proposed iterative partitioning method along with other four binarization methods for graphic images with salt-and-pepper and Gaussian noises, respectively.

Results show that the average performance of the proposed method outperforms other four binarization methods in the presence salt-and-pepper and Gaussian noises in terms of both ME and RAE measures. This also demonstrates

Table 3.11 Performance evaluation on graphic images for salt-and-pepper noise

Noise level	Otsu		Niblack		Bernsen		Sauvola		Iterative partitioning	
	ME	RAE	ME	RAE	ME	RAE	ME	RAE	ME	RAE
0.02	4.84	7.80	19.81	16.35	2.66	4.71	23.58	26.39	0.88	1.06
0.03	4.58	7.47	18.91	16.35	2.57	4.57	23.17	26.13	0.87	1.15
0.05	3.89	6.56	18.09	16.88	2.23	3.99	22.49	25.73	0.88	1.21

Table 3.12 Performance evaluation on graphic images for Gaussian noise

Noise level	Otsu		Niblack		Bernsen		Sauvola		Iterative partitioning	
	ME	RAE	ME	RAE	ME	RAE	ME	RAE	ME	RAE
0.02	9.77	14.01	20.65	12.34	4.71	7.40	17.45	21.77	2.13	1.24
0.03	4.58	7.47	18.91	16.35	2.57	4.57	23.17	26.13	0.87	1.15
0.05	3.89	6.56	18.09	16.88	2.23	3.99	22.49	25.73	0.88	1.21

that the proposed method shows more robustness in the presence of noise compared to other four binarization methods.

3.1.6 Conclusions

This section presents a new method for image binarization based on iterative partitioning of an image into several logical partitions. The proposed method is particularly significant for its non-overlapping threshold computation on the different partitions. This makes it a good choice for implementation in a multi-core computing system. This also opens up the possibility of extending the work for a large variety of applications including city surveillance and medical image analysis where binarization is very important as a preprocessing step for subsequent identification of object recognition. The proposed method produces better average results for nine different performance evaluation metrics as compared to other widely used binarization methods. The proposed method shows good noise immunity in the presence of salt-and-pepper and Gaussian noises. Moreover, the method is suitable for binarizing graphic and document images.

References

1. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging* **13**(1), 146–165 (2004)
2. Otsu, N.: A threshold selection method from gray-level histogram. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979)

3. USC-SIPI Image Database. University of Southern California, Signal and Image Processing Institute. <http://sipi.usc.edu/database/>
4. Shaikh, S.H., Maity, A.K., Chaki, N.: A new image binarization method using iterative partitioning. Springer J. Mach. Vis. Appl. (MVAP) **24**(2), 337–350 (2013)
5. Shaikh, S.H.: Image binarization and analysis towards moving object detection and human gait recognition. Ph.D. thesis, Department of Computer Science and Engineering, University of Calcutta (2014)
6. Niblack, W.: An Introduction to Digital Image Processing, pp. 115–116. Prentice Hall, Eaglewood Cliffs (1986)
7. Bernsen, J.: Dynamic thresholding of gray level images. In: Proceedings of International Conference Pattern Recognition (ICPR), pp. 1251–1255 (1986)
8. Sauvola, J., Pietikainen, M.: Adaptive document image binarization. Pattern Recogn. **33**(2), 225–236 (2000)

Chapter 4

A Framework for Creating Reference Image for Degraded Document Images

Abstract In this chapter, a method for creating reference image for degraded documents binarization is suggested. A reference image is required for quantitative performance evaluation of binarization methods. The proposed method computes a threshold for the reference image. In this method, seven binarization algorithms are considered in an initial pool. The final binarized reference image is created by selecting only five out of these seven algorithms. The selection is based on the proximity of the thresholds produced by these algorithms from the median of all the thresholds. The threshold value is calculated taking the absolute difference between the standard deviation and median from these selected thresholds and multiplying with a tuning parameter. The value of the tuning parameter depends on the type of degradation present in the image to be binarized. Examples are given to show how the new method and its implemented algorithm deal with images taken from a benchmark database of degraded document images.

Keywords Document image binarization • Reference image • Degradation of images • Non-uniform illumination • Ink seepage • Noisy images • Majority voting

A binarization algorithm for degraded document processing requires a reference image for quantitative evaluation. Most of the papers found in the literature rely on creating reference images based on manual processing.

These are often subjective and prone to human error. This chapter presents a method for creating reference image for degraded documents binarization. The reference image produced by the majority voting scheme [1] does not discriminate the wrong estimation from the correct one. There is a possibility of strong bias due to poor estimation of threshold by some methods for a particular image. This leads to a bad estimation for the threshold of the reference image. The method proposed in this chapter overcomes the biasness of majority voting to a great extent by discarding some of the binarization algorithms participating in the creation of the final reference image. This method gives a semiautomated approach in

selecting a threshold value for creating a reference image for degraded document images. The value of the threshold depends on the type of degradation present in the input image for which a reference is to be created. Experimental verification using images from a standard database illustrates the effectiveness of the proposed method.

4.1 Motivation of the Work

Document image binarization is a challenging task due to the presence of different kinds of degradation in the digitized images of the documents. Degradation can be observed in different forms, e.g., bad-quality paper and illegible text on it, ink seepage from backside of the page, the presence of stains and noises imposed by bad and non-uniform illumination due to poor digitization of the documents, wear, and tear of the page due to aging (yellowish paper). This makes the task of binarization challenging as it is very difficult to select a threshold that differentiates between the foreground object and the background pixels toward proper image segmentation. There exist a number of other factors that complicate the matter (e.g., use of ink of different colors).

The quantitative evaluation of different binarization techniques to measure their comparative performance is indeed an important aspect toward avoiding subjective evaluation. However, in majority of the methods found in the literature, creating a reference image is based on manual processing [2]. This requires human experts for creating the reference images. Thus, creating a reference image indeed remains an unsolved problem.

A quantitative evaluation is preferred over a qualitative one as the later measure is subjective. This calls for the requirement of preparing a reference image with which different binarization methods should be compared to produce quantitative performance evaluation. Some techniques are available for preparing reference image for document images as in [3]. However, this approach also requires human intervention in the process of preparing the reference image. In most of the approaches, found in the literature, a reference image has been formed manually which involves explicit human interference. This approach is highly subjective, so the quality of the reference image varies with a large extent from person to person.

Therefore, it seems that there is a gap in the literature toward creating reference image for degraded document images. An alternative technique is required for automatic creation of a reference image to be used for objective performance analysis.

Keeping this in view, this section of the dissertation introduces a novel methodology for creating a reference image for degraded documents binarization. The proposed method is a semiautomatic method that does not require involved human interference in the process of creating the reference unlike what is required for some of the methods found in the literature [2, 3]. A number of binarization methods are used for computing a threshold. The methods to be used for computing

the threshold for the reference image are selected automatically from the pool of a number of seven binarization methods. The threshold is computed making use of a tuning parameter. This parameter depends on the type of degradation present in the input image. Proper degradation categorization has been made for the test images toward finding the value of a tuning parameter, which is to be supplied by the user. The degradation categorization of the test images highlights the different possible values the tuning parameter can have depending on the presence of various types of degradations in the input image.

4.2 Proposed Methodology

In recent past, attempts have been made by the authors for creating reference images using majority voting [1]. In majority voting method, all the participating binarization algorithms have been taken into consideration when creating a reference image. There is an inherent biasness of this method. It is possible that a pixel may be wrongly chosen as object in the reference image if most of the algorithms take a wrong decision.

The proposed method considers seven binarization methods in the pool of initial methods. The final binarized reference image is created by selecting only five out of these seven methods. The selection is based on the proximity of the thresholds produced by these methods from the median of all the thresholds. A new threshold value is calculated taking the absolute difference between the standard deviation and median from these selected thresholds and multiplying with a tuning parameter. The value of the tuning parameter depends on the type of degradation present in the image to be binarized.

Seven binarization algorithms are initially considered to take part. The seven methods considered are the following: Otsu [4], iterative thresholding [5], balanced histogram [6], Kapur [7], Kittler [8], Johannsen [9], and Ridler [10].

The objective of the proposed methodology is to find a threshold that is suggested by the majority of the methods considered initially. Two among the seven participating binarization methods are discarded from being considered in the computation of the final threshold for generating the reference image. This method is presented in [11]. The steps of the algorithm are stated as follows:

Algorithm:

- Step 1: A grayscale image $I(x, y)$ is taken as input. (x, y) is the spatial coordinates of the image in a 2D plane. If the input is a color image, it needs to be converted to gray scale for further processing.
- Step 2: Seven binarization algorithms namely, Otsu, iterative thresholding, balanced histogram, Kapur, Johannsen, Ridler, and Kittler are used to calculate seven threshold values for image I , stored in the threshold vector t_i for $1 \leq i \leq 7$.

- Step 3: The median of t_i is calculated as $\text{med}_1 = \text{median}(t_1, t_2, \dots, t_7)$.
- Step 4: The difference (D_i) of each threshold t_i from med_1 is calculated as $D_i = |t_i - \text{med}_1|$ for $1 \leq i \leq 7$.
- Step 5: D_i is sorted in descending order. The first two elements of D_i represent those methods whose thresholds deviate most from med_1 among all seven binarization methods. These two methods and their corresponding thresholds are discarded from t_i . The adjusted range of i is $1 \leq i \leq 5$.
- Step 6: The median of t_i for $1 \leq i \leq 5$ is calculated as $\text{med}_2 = \text{median}(t_1, t_2, \dots, t_5)$. Let the standard deviation of all t_i for $1 \leq i \leq 5$ be σ .
- Step 7: The final threshold for the reference image is computed as follows:

$$t_{\text{ref}} = |\text{med}_2 - \sigma| \times k$$

- Step 8: The binary reference image $B(x, y)$ is created by setting all the pixels of $I(x, y)$ having gray-level intensity $\geq t_{\text{ref}}$ to one and other pixels to zero. Here, zero and one represent object and background pixels, respectively.

$$\begin{aligned} B(x, y) &= 1, & \text{if } I(x, y) \geq t_{\text{ref}} \\ &= 0, & \text{if } I(x, y) < t_{\text{ref}} \end{aligned}$$

The presented methodology [11] is a semiautomatic process in which the binarization algorithms that will finally be used for calculating the threshold for the reference image are selected automatically. However, the value of the parameter k is selected depending upon the type of degradation present in the concerned image. Therefore, a degradation categorization has been made for all the test images depending upon the prominent type of degradation present in the concerned image. The justification of choosing the values of k in step 7 of the proposed method is presented in Sect. 4.3.

4.3 Determining the Value of k

Selection of a value for parameter k is important for calculating the final threshold using the proposed algorithm. The value of k depends on the type of degradation present in the image to be binarized. A proper degradation categorization is required to get the possible values of this tuning parameter k . This section reports the findings of the authors on deciding the value of k on the basis of test images from DIBCO [12] benchmark database.

The objective of choosing the value for parameter k is quite simple. Let us consider image 3 that contains text in dark color with some text that appears from the backside due to poor-quality paper. The suitable value for this image is $k = 0.64$. The idea is to keep the final value of the computed threshold as low as possible so that pixels with lighter shades of gray which mostly represent the degradations (text from backside) do not qualify as foreground in the binarized image.

Table 4.1 DIBCO images and types of degradations

Degradation category type	Image no.	Description of degradations	Value of k
1	15, 16, 27	Shadow/illumination problem, stains due to ink/water seepage on the page	0.51
2	3, 11, 26	Text appears from the backside due to poor-quality paper, text written on a dark-color page, shadow covering a major proportion of the whole image	0.64
3	7, 8, 12, 25	Matter appears from the backside of the page	0.7
4	9, 10, 13, 24, 28, 29, 40	Relatively less amount of ink seepage and shadow change, noise due to folding of page	0.79
5	14, 19, 39	Ink spreads over the text and patches on the page	0.88
6	2, 37	Handwritten text on a light background paper.	0.91
7	6, 17, 18, 23, 35, 36	Text written on a relatively darker background, varying hand pressure during writing, red-color graphic symbol/logo, etc.	0.97
8	1, 30, 38	Low contrast difference between background and foreground	1.01
9	4, 21, 22, 31, 32, 33	Poor-quality paper with light texture	1.07
10	5, 20, 34	Varying hand pressures, thick and thin letters	1.16

A completely opposite scenario is observed for image 5. This image contains text written with varying hand pressures with thick and thin letters. The objective is to keep the threshold as high as possible so that more and more pixels (text having dark as well as lighter shades of gray) qualify as foreground. $k = 1.16$ is therefore found to be suitable for image 5.

The value of k is set experimentally. A 50 % swing on both sides of actual threshold (computed at step 7 of the algorithm presented in Sect. 4.2. before multiplying with k). Therefore, the value of the parameter k will vary from 0.5 to 1.5 during experiment. The experimentation is done with a step size of 0.01 to ascertain that the selected values of the parameter k for a typical type of degradation (as mentioned in Table 4.1) are actually the best choice to be used for binarization.

Table 4.1 presents the different types of degradations present in the test images collected from DIBCO [12] database for experimental verification. The fourth column in the table presents the value of the parameter k found suitable for that particular category of image degradation.

Table 4.2 Performance evaluation for image 3

Measures	Values of parameter k							
	0.51	0.60	0.64	0.70	0.79	0.88	0.97	1.16
ME	0.68	0.64	0.52	0.75	0.60	0.94	1.71	6.89
F -measure	97.76	98.72	99.74	98.77	97.69	97.52	96.12	95.35
PSNR	80.51	82.45	83.96	83.26	82.03	82.00	81.38	80.11

An image may not be contaminated with all types of degradations stated in column three of Table 4.1. More than one type of degradations can be present in an image. The goal of binarization in this situation would be to reduce the effect of one of degradations (which affects the image most) in the final output image. Different types of degradations sometimes may be conflicting. Reducing one type of degradation may cause the other one to increase. Sometimes, it is also found that a careful selection of threshold may significantly reduce noise/degradation in the final output with a certain loss of the actual document content (e.g., handwritten text).

Table 4.2 reports the results obtained for image 3 of test images from DIBCO database (as presented in Fig. 4.1) on three metrics (ME, F -measure, and PSNR as defined in Sect. 4.1) for different values of the parameter k . The actual experiment ranges from 0.5 to 1.5 with a step value of 0.01. In the present context, for the sake of brevity, the partial results are presented in Table 4.2. The values of k for other image degradations are summarized in Table 4.1.

As shown by the results presented in Table 4.2, for image 3, $k = 0.64$ is the most suitable value. As the values of k increase from 0.64, the scores of the three metrics gradually degrade representing a bad choice for k .

The similar experiments are performed for all 40 test images for all seven metrics (ME, RAE, Recall, Precision, F -measure, SNR, and PSNR). Results for all degradation category types are summarized in Table 4.3 through Table 4.12 for three measures (ME, F -measure, and PSNR).

Category-1 degradations contain illumination, shadow, and ink seepage problems as shown in images 15, 16, and 27. Results for these images are reported in Table 4.3, the suitable value of $k = 0.51$ is giving the optimal performance.

For images falling in category-2 types of degradation, $k = 0.64$ is found to be the most suitable value for the parameter. The corresponding results, for images 3, 11, and 26, are reported in Table 4.4.

Table 4.5 summarizes the results obtained for images of degradation category-3 (images 7, 8, 12, and 25). This category of images mostly contains documents where some text appears from the backside of the page. A higher value of the threshold is desired to suppress the kind of degradation present. The value of the parameter $k = 0.7$ is found to be suitable for this purpose.

As shown by the results for images 9, 10, 13, 24, 28, 29, and 40, $k = 0.79$ gives the best results for all three metrics as reported in Table 4.6.

**Fig. 4.1** Test images from DIBCO

Table 4.3 Performance evaluation for category-1

Image no.	Metrics	Parameter k				
		0.5	0.51	0.53	0.55	0.5
15	ME	5.43	5.21	5.69	6.00	5.43
	F -measure	97.03	97.20	96.84	96.69	97.03
	PSNR	67.13	67.30	66.28	66.18	66.98
16	ME	5.65	5.23	5.85	6.09	5.65
	F -measure	96.76	97.03	96.63	96.48	96.76
	PSNR	68.29	69.62	68.60	68.41	68.29
27	ME	4.09	3.98	4.07	4.13	4.09
	F -measure	96.35	97.90	96.36	96.32	96.35
	PSNR	71.11	72.02	71.29	71.18	71.11

Table 4.4 Performance evaluation for category-2

Image no.	Metrics	Parameter k				
		0.6	0.62	0.64	0.66	0.68
3	ME	0.64	0.63	0.52	0.73	0.74
	F -measure	98.72	98.78	99.74	98.77	98.76
	PSNR	82.45	82.46	83.96	82.44	82.37
11	ME	1.89	1.83	1.43	1.92	2.27
	F -measure	97.54	97.57	98.23	97.52	97.03
	PSNR	69.45	69.74	70.77	69.28	69.84
26	ME	2.84	2.79	1.98	2.80	2.86
	F -measure	97.05	97.07	98.98	97.07	97.04
	PSNR	75.33	75.43	76.84	75.39	75.25

Table 4.5 Performance evaluation for category-3

Image no.	Metrics	Parameter k				
		0.66	0.68	0.7	0.72	0.74
7	ME	6.11	6.03	5.77	6.11	6.28
	F -measure	95.19	95.23	96.19	95.19	95.07
	PSNR	65.32	65.41	66.68	65.32	65.96
8	ME	4.66	4.65	4.37	4.66	4.72
	F -measure	97.05	97.21	97.60	97.05	97.41
	PSNR	70.76	70.76	71.80	70.72	70.63
12	ME	1.98	1.89	1.73	1.79	1.94
	F -measure	98.34	98.35	99.10	98.34	98.32
	PSNR	74.87	74.90	75.71	74.84	74.69
25	ME	1.89	1.87	1.40	1.88	1.92
	F -measure	98.53	98.54	99.26	98.53	98.51
	PSNR	75.94	75.03	76.83	75.97	75.77

In case of images falling in category-5 types of degradation, 0.88 is found to be the most suitable value for k . The corresponding results, for images 14, 19, and 39, are reported in Table 4.7.

Images 2 and 37 fall in category-6. For these images, $k = 0.91$ is the suitable value of the parameter as shown in Table 4.8.

Table 4.6 Performance evaluation for category-4

Image no.	Metrics	Parameter k				
		0.75	0.77	0.79	0.81	0.83
9	ME	5.57	5.45	5.18	5.41	5.50
	F -measure	96.02	96.09	96.83	96.11	96.05
	PSNR	67.76	67.08	68.64	67.16	67.88
10	ME	3.09	2.98	2.84	3.05	3.23
	F -measure	96.82	96.88	97.32	96.83	95.73
	PSNR	67.57	67.77	68.38	67.57	67.09
13	ME	6.69	6.68	6.35	6.76	6.92
	F -measure	95.93	95.93	96.64	95.87	95.77
	PSNR	68.41	68.40	69.30	68.29	68.14
24	ME	4.65	4.53	3.25	4.49	4.56
	F -measure	96.54	96.61	97.70	96.62	96.57
	PSNR	67.61	67.79	68.28	67.81	67.66
28	ME	1.98	1.88	1.72	1.81	1.82
	F -measure	98.05	98.07	99.08	98.01	98.5
	PSNR	78.02	78.02	79.12	78.07	78.12
29	ME	2.01	2.02	1.98	2.10	2.01
	F -measure	96.06	97.01	98.96	97.02	96.98
	PSNR	80.02	81.12	81.73	81.01	79.01
40	ME	1.88	1.78	1.11	1.67	1.44
	F -measure	98.51	98.51	99.41	98.51	98.51
	PSNR	77.51	77.55	78.29	77.56	77.57

Table 4.7 Performance evaluation for category-5

Image no.	Metrics	Parameter k				
		0.84	0.86	0.88	0.9	0.92
14	ME	2.69	2.66	2.62	2.72	2.73
	F -measure	97.81	97.87	98.54	97.80	97.74
	PSNR	71.64	71.65	72.08	71.56	71.36
19	ME	2.17	1.97	1.87	1.98	2.66
	F -measure	96.84	96.94	97.95	96.93	96.55
	PSNR	72.91	72.32	73.92	72.24	72.92
39	ME	1.86	1.88	1.59	1.78	1.86
	F -measure	98.07	98.0	99.15	98.02	98.33
	PSNR	77.51	77.51	78.19	77.51	77.13

Table 4.8 Performance evaluation for category-6

Image no.	Metrics	Parameter k				
		0.87	0.89	0.91	0.93	0.95
2	ME	2.19	2.08	1.90	2.14	2.61
	F -measure	98.38	98.44	99.01	98.40	98.15
	PSNR	76.26	76.68	78.41	76.40	76.87
37	ME	1.92	1.94	1.57	1.86	1.77
	F -measure	96.04	97.03	99.18	98.76	98.45
	PSNR	80.34	81.03	81.36	81.02	81.03

Table 4.9 Performance evaluation for category-7

Image no.	Metrics	Parameter k				
		0.93	0.95	0.97	0.99	1.01
6	ME	2.07	2.03	1.99	2.04	2.08
	F -measure	98.36	98.38	98.81	98.37	98.35
	PSNR	74.10	74.24	74.55	74.19	74.03
17	ME	2.14	2.18	2.13	2.17	2.18
	F -measure	97.87	97.90	98.36	97.88	97.84
	PSNR	74.40	74.51	75.46	74.43	74.24
18	ME	0.76	0.88	0.73	0.89	0.94
	F -measure	98.62	98.64	99.10	98.74	98.65
	PSNR	76.84	76.13	77.41	76.33	76.19
23	ME	1.34	1.30	1.06	1.28	1.30
	F -measure	98.28	97.30	98.90	97.31	97.21
	PSNR	74.95	74.08	75.97	74.12	74.04
35	ME	2.48	2.45	2.16	2.57	2.58
	F -measure	97.56	97.66	98.86	97.56	97.43
	PSNR	75.12	75.11	75.66	75.02	75.04
36	ME	3.57	3.59	3.50	3.65	3.59
	F -measure	97.51	97.51	98.19	97.88	97.53
	PSNR	72.33	74.39	76.14	73.34	74.87

Table 4.10 Performance evaluation for category-8

Image no.	Metrics	Parameter k				
		0.97	0.99	1.01	1.03	1.05
1	ME	2.59	2.02	1.72	1.92	2.88
	F -measure	98.57	98.88	99.04	98.92	98.37
	PSNR	74.78	74.80	75.17	74.93	74.11
30	ME	3.36	3.34	3.12	3.65	3.69
	F -measure	97.65	97.23	98.35	97.25	97.66
	PSNR	69.24	70.18	70.65	68.34	68.22
38	ME	1.34	1.34	1.30	1.32	1.32
	F -measure	98.03	98.01	99.31	97.62	97.88
	PSNR	82.18	82.01	82.55	82.52	82.51

Images 6, 17, 18, 23, 35, and 36 fall in category-7 degradations. $k = 0.97$ is chosen as the value of the parameter as reported in Table 4.9.

In category-8, images 1, 30, and 38 are found. $k = 1.01$ is found to be suitable as shown in Table 4.10.

$k = 1.07$ is the parameter value found suitable for category-9 images (images 4, 21, 22, 31, 32, and 33). This is summarized in Table 4.11.

Table 4.11 Performance evaluation for category-9

Image no.	Metrics	Parameter k				
		1.03	1.05	1.07	1.09	1.11
4	ME	2.90	2.79	2.22	2.73	2.85
	F -measure	98.09	98.14	98.82	98.07	98.01
	PSNR	70.15	70.39	71.75	70.51	70.20
21	ME	2.89	2.88	2.76	2.79	3.06
	F -measure	98.36	98.37	98.50	98.31	97.83
	PSNR	73.07	73.41	74.31	73.20	73.05
22	ME	2.89	2.55	2.30	2.42	2.58
	F -measure	97.88	97.91	98.78	97.87	97.78
	PSNR	70.68	70.78	71.77	70.60	70.26
31	ME	2.98	2.97	2.78	2.89	2.87
	F -measure	97.66	97.41	98.45	97.97	97.98
	PSNR	73.04	73.34	74.2	73.79	73.86
32	ME	2.55	2.58	2.42	2.49	2.46
	F -measure	97.36	97.89	98.69	97.45	97.06
	PSNR	75.58	75.78	76.52	76.11	76.29
33	ME	2.49	2.48	2.45	2.65	2.67
	F -measure	97.98	98.02	98.64	98.03	98.07
	PSNR	77.02	77.14	78.59	77.26	77.35

Table 4.12 Performance evaluation for category-10

Image no.	Metrics	Parameter k				
		1.12	1.14	1.16	1.18	1.2
5	ME	2.48	2.36	2.17	2.17	2.21
	F -measure	98.66	98.72	98.82	98.82	98.02
	PSNR	72.49	72.08	73.64	73.03	73.01
20	ME	1.29	1.34	1.28	1.34	1.66
	F -measure	98.37	99.04	99.33	98.40	98.40
	PSNR	76.31	76.53	77.62	76.54	76.54
34	ME	1.06	1.08	0.99	1.05	1.07
	F -measure	98.77	98.87	99.48	98.82	98.23
	PSNR	78.12	78.32	79.20	79.11	79.04

Finally, for category-10 (images 5, 20, and 34), $k = 1.16$ is found to be suitable as reported in Table 4.12.

4.4 Benchmark Dataset

The experimental verification is performed by taking 40 images from DIBCO [12] database. Figure 4.1 shows the images used for testing purpose. DIBCO database contains benchmark-degraded document images. These images suffer from



Fig. 4.2 Comparative results

different kinds of degradations such as ink seepage, broken and missing text, non-uniform illumination, shadows due to poor digitization of the documents, etc. The dataset includes both grayscale and color images.

Figure 4.2 presents the results of a few selected images from the test samples. First column in Fig. 4.2 shows the reference images provided by DIBCO, the images in the second column are the references produced by the majority voting approach, and the third column contains the results of the proposed method.

This is obvious from the output that for some test images, the proposed method is a clear winner over the majority voting approach. This is evidenced by images shown in 4, 5, and 6th rows of Fig. 4.2. The dark stains are not present in the output of the proposed method for the 4th image. The noise due to non-uniform illumination is almost removed for the 5th image. For the 6th image, the dark patch due to ink stains is very prominent in case of majority voting. However, the proposed method does not produce any patch of course at the cost of loss of some foreground text.

In case of the last image (7th) in the sequence shown in Fig. 4.2, the degradation is the text that appears from the backside of the page. This degradation is almost removed in the output of the proposed method. Whereas in case of majority voting, some patches of text from the background still exist in the final result.

4.5 Experimental Verification

A reference image is created to measure the performance of different binarization methods. Methods for creating reference images are found in the literature for document image analysis. However, creating a proper reference is a challenging task indeed. In most of the cases, reference images have been generated by human expert. This method of creating a reference is highly subjective and prone to error and human bias. The following subsection presents an automated method of creating reference images.

4.5.1 Majority Voting Method

The reference image using majority voting [1] is created as follows: Seven binarized images are created for each test image using the seven binarization methods (Otsu, iterative thresholding, balanced histogram, Kapur, Johannsen, Ridler, and Kittler). In order to generate the reference image, all the binarized images are consulted pixel by pixel. Each pixel in the reference image is set to 1 if the majority of the seven methods agree that the corresponding pixel is a 1 in their respective resultant binarized image. Otherwise, the pixel is set to 0 in the reference image.

Table 4.13 Performance evaluation on ME and RAE

Image no.	ME		RAE	
	Majority voting	Proposed	Majority voting	Proposed
1	1.92	1.72	0.78	0.67
2	1.09	1.90	0.23	1.78
3	5.99	0.52	2.75	0.46
4	1.90	2.22	1.64	2.15
5	2.22	2.17	1.29	1.11
6	1.11	1.99	0.79	2.19
7	3.00	5.77	0.35	7.31
8	4.22	4.37	3.71	2.93
9	1.40	5.18	0.37	6.12
10	2.31	2.84	1.40	5.16
11	0.97	1.43	0.81	3.47
12	12.01	1.73	13.01	1.75
13	5.99	6.35	2.75	3.56
14	2.26	2.62	0.48	2.52
15	16.85	5.21	16.1	4.79
16	12.01	5.23	13.01	1.30
17	2.20	2.13	1.58	3.13
18	0.92	0.73	0.83	1.77
19	3.68	1.87	3.42	4.01
20	1.50	1.28	1.11	0.66
21	2.27	2.76	0.06	1.14
22	2.13	2.30	1.36	1.83
23	1.33	1.06	0.45	1.96
24	3.68	3.25	3.42	4.19
25	2.12	1.40	2.02	1.35
26	18.74	1.98	19.14	1.66
27	21.23	3.98	22.70	3.98
28	6.33	1.72	6.43	1.14
29	2.24	1.98	0.39	1.99
30	2.08	3.12	1.32	3.05
31	1.86	2.78	1.86	2.98
32	2.09	2.42	0.70	1.89
33	5.12	2.45	3.54	0.84
34	0.88	0.99	0.56	0.86
35	6.33	2.16	6.43	2.11
36	1.52	3.50	1.19	3.51
37	6.33	1.57	6.43	1.61
38	5.77	1.30	7.31	0.58
39	9.08	1.59	9.31	0.28
40	2.57	1.11	2.45	0.43
Average	4.68	2.52	4.09	2.36

Table 4.14 Performance evaluation on Recall, Precision, and *F*-measure

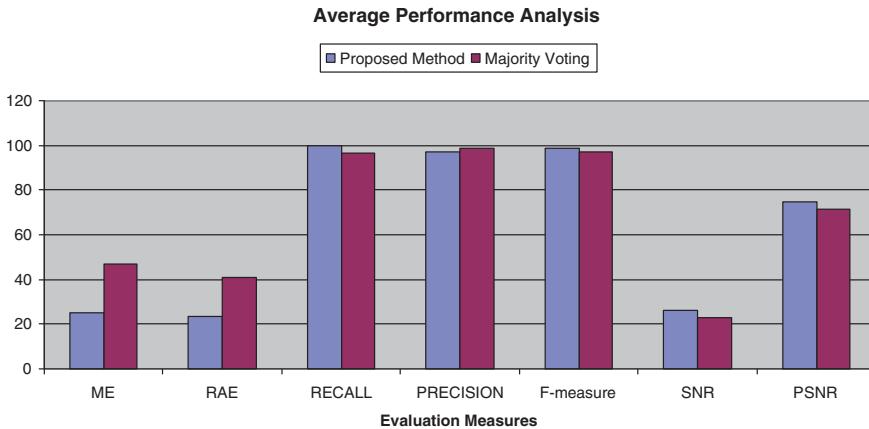
Image no.	Recall		Precision		<i>F</i> -measure	
	Majority voting	Proposed	Majority voting	Proposed	Majority voting	Proposed
1	98.53	99.38	99.31	98.71	98.92	99.04
2	99.32	99.91	99.54	98.13	99.43	99.01
3	99.14	99.96	99.89	99.51	99.52	99.74
4	99.81	99.90	98.18	97.76	98.99	98.82
5	99.44	99.38	98.16	98.27	98.80	98.82
6	99.73	99.92	98.95	97.73	99.34	98.81
7	98.41	99.98	98.07	92.67	98.24	96.19
8	95.79	99.07	99.48	96.17	97.60	97.60
9	99.30	99.99	98.93	93.87	99.12	96.83
10	97.98	99.97	99.38	94.81	98.68	97.32
11	99.90	100.00	99.09	96.53	99.50	98.23
12	99.26	99.99	99.19	98.24	99.23	99.10
13	95.39	98.42	98.09	94.92	96.72	96.64
14	98.49	99.81	98.96	97.30	98.72	98.54
15	82.66	99.64	98.52	94.87	89.89	97.20
16	86.62	97.67	99.58	96.40	92.65	97.03
17	99.63	99.95	98.06	96.82	98.84	98.36
18	96.93	100.00	94.10	98.22	95.49	99.10
19	97.48	100.00	95.51	95.98	96.48	97.95
20	95.77	99.66	94.67	99.01	95.21	99.33
21	98.74	97.93	98.79	99.06	98.77	98.50
22	99.55	99.71	98.20	97.88	98.87	98.78
23	99.51	99.89	99.06	97.94	99.29	98.90
24	96.26	99.83	99.66	95.65	97.93	97.70
25	97.86	99.94	99.88	98.59	98.86	99.26
26	80.69	99.81	99.79	98.16	89.23	98.98
27	77.20	99.93	99.87	95.95	87.08	97.90
28	93.40	99.65	99.81	98.52	96.50	99.08
29	98.62	99.96	99.00	97.98	98.81	98.96
30	99.56	99.90	98.24	96.85	98.89	98.35
31	99.89	99.96	98.04	96.99	98.96	98.45
32	99.21	99.64	98.52	97.76	98.86	98.69
33	95.39	99.06	98.89	98.23	97.11	98.64
34	99.82	99.91	99.26	99.06	99.54	99.48
35	99.55	99.93	99.11	97.82	99.33	98.86
36	95.80	99.98	94.62	96.47	95.20	98.19
37	95.87	100.00	95.27	98.38	95.56	99.18
38	98.87	99.60	99.58	99.02	99.22	99.31
39	90.52	99.29	99.81	99.02	94.94	99.15
40	97.39	99.62	99.84	99.19	98.60	99.41
Average	96.33	99.65	98.52	97.36	97.32	98.48

Table 4.15 Performance evaluation on SNR and PSNR

Image no.	SNR		PSNR	
	Majority voting	Proposed	Majority voting	Proposed
1	26.04	26.58	74.69	75.17
2	32.49	30.16	80.83	78.41
3	33.11	35.75	81.37	83.96
4	24.05	23.39	72.43	71.75
5	25.09	25.18	73.54	73.64
6	28.20	25.70	77.11	74.55
7	21.41	18.20	70.21	66.68
8	23.17	23.31	71.94	71.80
9	25.20	19.78	74.32	68.64
10	22.84	19.92	71.59	68.38
11	27.97	22.58	76.27	70.77
12	18.20	27.47	66.68	75.71
13	20.97	21.00	69.56	69.30
14	15.17	23.54	64.75	72.08
15	12.89	18.96	62.20	67.30
16	16.69	20.96	66.01	69.62
17	28.67	27.20	77.00	75.46
18	27.97	29.15	76.27	77.41
19	22.84	25.63	71.59	73.92
20	24.06	29.31	72.73	77.62
21	12.89	25.77	62.20	74.31
22	23.74	23.44	72.10	71.77
23	29.45	27.60	77.88	75.97
24	20.19	19.89	68.91	68.28
25	26.55	28.50	75.03	76.83
26	17.86	28.61	67.08	76.84
27	15.17	23.74	64.75	72.02
28	24.75	30.74	73.47	79.12
29	32.80	33.43	81.20	81.73
30	24.04	22.36	72.41	70.65
31	27.36	25.67	75.94	74.20
32	12.22	28.09	62.00	76.52
33	26.65	30.04	75.39	78.59
34	19.89	30.89	68.28	79.20
35	19.89	27.37	68.28	75.66
36	24.04	27.95	72.41	76.14
37	12.89	33.10	62.20	81.36
38	18.20	34.16	66.68	82.55
39	15.17	29.81	64.75	78.19
40	26.06	29.86	74.62	78.29
Average	22.67	26.37	71.42	74.76

Table 4.16 Average performance evaluation

	ME	RAE	Recall	Precision	F-measure	SNR	PSNR
Proposed method	2.52	2.36	99.65	97.36	98.48	26.37	74.76
Majority voting	4.68	4.09	96.33	98.52	97.32	22.67	71.42

**Fig. 4.3** Average performance analysis

4.5.2 Comparative Performance Analysis

The following section presents the results obtained when the proposed method is compared with majority voting. The comparisons are performed on the basis of seven evaluation measures defined in Sect. 3.1.3.

Table 4.13 summarizes the measures of two metrics namely misclassification error (ME) and relative foreground area error (RAE) comparing majority voting and the proposed method. In terms of both ME and RAE, the proposed method outperforms majority voting approach. As shown by the results in Table 4.13 that there is no significant difference in the average scores of ME and RAE obtained by the proposed method compared to that of by majority voting. However, for some of the images (e.g., 15, 26, 27, etc.), the performance of the proposed method is far better than majority voting method on the basis of ME.

Table 4.14 summarizes the values obtained for Recall, Precision, and *F*-measure. The results show that average scores for Recall and *F*-measure are better for the proposed method than majority voting. However, the average Precision score of the proposed method is slightly lower than what is reported by the majority voting approach.

Table 4.15 collects the results for two metrics namely signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR). As shown by the results, the average

performance of the proposed method is far better than the majority voting for both the measures SNR and PSNR.

The average scores for the measures are marked in bold face in the last row of Tables 4.14 and 4.15.

Table 4.16 summarizes the average performances of the majority voting approach and the proposed method for all seven evaluation measures. The average values of the scores are marked in bold face.

As shown in Table 4.16, as far as the average performance is concerned, the proposed method outperforms majority voting for all the measures except Precision. In Table 4.16, for each of the measures, the best scores are marked in bold face.

Figure 4.3 shows a plot of the comparative average performances of majority voting and the proposed method on the basis of data presented in Table 4.16. The values of measures ME and RAE are scaled up in the plot for the sake of better visualization.

4.6 Conclusions

Evaluation of binarization algorithms for degraded document images is a challenging task due to the presence of various kinds of noises such as non-uniform illumination, ink seepage, broken text, bad-quality papers, etc. A reference image is to be created for quantitative evaluation of binarization algorithms. In this section, a method is presented for creating reference image for quantitative evaluation of binarization algorithms for degraded document images. The proposed method is a semiautomatic method that selects the binarization methods that will take part in creating the final reference image based on the deviation from the median threshold of seven different binarization methods. The final threshold for the reference image is calculated by taking into account a tuning parameter. The value of this parameter depends on the type of degradation present in the image for which the reference is to be created. The proposed method is compared with the majority voting method for creating reference images. Manually generated reference images provided by DIBCO are considered as the references for the sake of comparative performance analysis. A number of evaluation measures such as misclassification error, relative foreground area error, Recall, Precision, *F*-measure, SNR, and PSNR are considered for evaluation purpose. Results show the effectiveness of the proposed methodology.

References

1. Shaikh, S.H., Maity, A.K., Chaki, N.: A new image binarization method using iterative partitioning. Springer J. Mach. Vis. Appl. (MVAP) **24**(2), 337–350 (2013)
2. Rodriguez, R.: Binarization of medical images based on the recursive application of mean shift filtering: another algorithm. J. Adv. Appl. Bioinform. Chem. **1**, 1–12 (2008)
3. Ntirogiannis, K., Gatos, B., Pratikakis, I.: An objective evaluation methodology for document image binarization techniques. In: 8th IAPR Workshop on Document Analysis Systems (2008)
4. Otsu, N.: A threshold selection method from gray-level histogram. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
5. Gonzalez, R., Woods, R.: Book: Digital Image Processing, International 3rd revised edn. Pearson Prentice Hall, Upper Saddle River (2008) (Chapter 10)
6. Anjos, A., Shahbazkia, H.: Bi-level image thresholding—a fast method. Biosignals **2**, 70–76 (2008)
7. Kapur, N.J., Sahoo, P.K., Wong, C.K.A.: A new method for gray-level picture thresholding using the entropy of the histogram. J. Comput. Vis. Graph. Image Process. **29**(3), 273–285 (1985)
8. Kittler, J., Illingworth, J.: Minimum error thresholding. Pattern Recogn. **19**(1), 41–47 (1986)
9. Johannsen, G., Bille, J.: A threshold selection method using information measures. In: 6th International Conference on Pattern Recognition, pp. 140–143 (1982)
10. Ridler, T., Calvard, S.: Picture thresholding using an iterative selection method. IEEE Trans. Syst. Man Cybern. **8**(8), 630–632 (1978)
11. Shaikh, S.H., Dey, A., Saeed, K., Chaki, N.: On creation of reference image for degraded documents binarization. In: Submitted to the International Journal of Computational Vision and Robotics (IJCVR) (2014)
12. Document Image Binarization Contest (DIBCO): <http://utopia.duth.gr/~ipratika/DIBCO2013/>. Accessed Dec 2013

Chapter 5

Applications of Binarization

Abstract This chapter shows the applications of binarization in image processing and analysis. A particular consideration is given toward some selected biometric applications. Use of binarization in medical image processing is also stated.

Keywords Applications of binarization • Image thresholding • OCR • Medical image processing • Magnetic resonance imaging (MRI) • Face detection • Video processing • Moving object detection • Biometric fingerprint analysis • Gait analysis • Iris recognition

Binarization is one of the most important steps in almost all image processing systems. In most of the vision-based systems, application of binarization includes finding out the region of interest from a given image targeted for a particular application. Some of the widely used applications of binarization are summarized in this chapter.

5.1 Document Image Processing and OCR

Binarization is widely used in document image analysis [1] and in optical character recognition (OCR) systems. In document images, adaptive binarization techniques are generally used to get rid of different kinds of degradations present in the image [2]. In printed and handwritten text recognition, binarization is used to identify the different characters. It is also used in automated recognition for number plates of vehicles [3]. Figure 5.1 shows two sample images (document and number plate) along with the binarized counterparts.



Fig. 5.1 Document image analysis and OCR systems

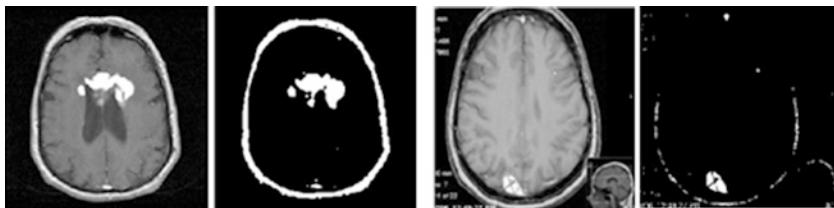


Fig. 5.2 Brain images (MRI). *Source* [4]

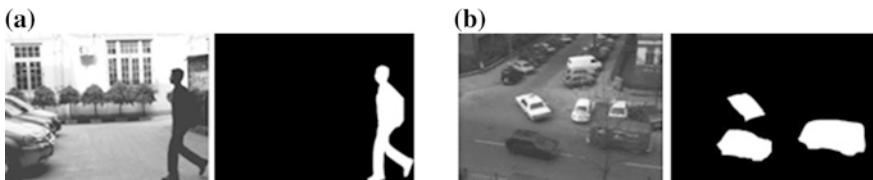


Fig. 5.3 **a.** Detection of human being. **b.** Detection of vehicles in a traffic video. Moving object detection from video. *Source b.* [15]

5.2 Medical Image Processing

In medical image processing, binarization is used to find the region of interest (ROI) from the given image. Figure 5.2 shows two brain images (MRI) [4, 5]. They are interesting because of the existence of tumors. Binarization along with other filter techniques for image de-noising can be used for identifying the diseased portions of the scanned image.

5.3 Video Processing

In video processing, one of the applications is the detection of moving objects from a sequence of video frames. Background removal techniques are well known in this context. Most of the background removal techniques generate a binary mask identifying the portion of the moving objects from a sequence of video frames. Figure 5.3 shows two such frames from two different video sequences.



Fig. 5.4 Face Detection. *Source* California Institute of Technology (CIT) Face Database

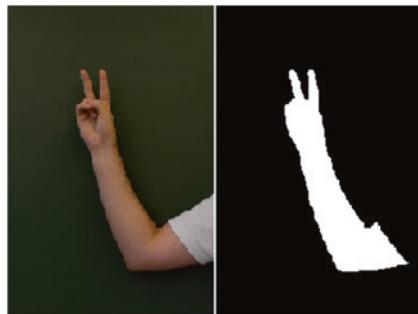


Fig. 5.5 Hand Gesture Recognition. *Source* Hand Gesture Recognition (HGR) Database, Silesian University of Technology, Institute of Informatics, Poland. <http://sun.aei.polsl.pl/~mkawulok/gestures/>. Last Accessed: May 2014

5.4 Face Detection

There are various techniques for the detection of face. Some of the techniques require identifying the locations of eyes, lips, etc., from a given facial image [6, 7]. Binarization can be used as an important tool in identifying the location of eyes and lips. Skin color-based segmentation techniques are also very common in face detection. However, almost every technique makes use of binarization as one of the processing steps for generating a binary mask identifying the region of face in an image (Fig. 5.4).

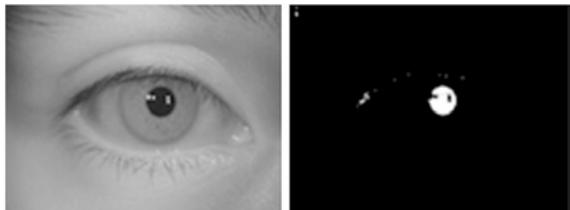
5.5 Hand Gesture Recognition

Gesture recognition is an important area of research towards building better HCI systems [8, 9]. Binarization is also used in segmenting the region of hands for hand gesture recognition. Color-based image segmentation is performed to identify the skin regions; afterwards, a binary hand image is generated (Fig. 5.5), locations of the finger tips are found out and tracked in subsequent image frames to identify one hand gesture.



Fig. 5.6 Fingerprint images

Fig. 5.7 Pupil localization for iris recognition. Original image source: CASIA [16]



5.6 Fingerprint Recognition

In the preprocessing steps, image binarization is used for finding the locations of ridges and valleys from a finger impression. Binarization helps in identifying the location of ridge bifurcation, presence of arch, and loops in the fingerprint images. Two such samples are shown in Fig. 5.6. In minutia-based fingerprint recognition systems [10, 11], binarization along with thinning is used as preprocessing steps before detecting start or end points of ridges and bifurcations.

5.7 Iris Recognition

Iris recognition for human identification is a well-known biometric approach [12, 13]. Localizing the iris is an important task. Scanned image can be binarized with a suitable threshold to identify the location of the pupil (Fig. 5.7). The locations of pupil boundary and the eye lids give helps in localizing the iris. During generation of iris code, the binarization of iris pattern is essential from grayscale images because comparing Boolean vector is less complex and needs less computational power than with number vectors.



Fig. 5.8 Gait cycle detection in silhouette-based gait analysis. *Source [1]*

5.8 Gait Recognition

In silhouette-based gait analysis [14], the binary silhouettes are generated from the sequence of video frames using background subtracting and binarization [1]. Silhouettes of human beings are analyzed for cycle detection and shape feature extraction for human recognition. A sequence of binary silhouettes is shown in Fig. 5.8. The temporal sequence of human silhouettes present in between two consecutive heel strikes by the same foot during walking, form a gait cycle [14]. Human silhouettes are needed for silhouette based gait recognition. Therefore, extraction of silhouettes of moving objects from video is an important area of research as found in the literature [17].

References

1. Shaikh, S. H.: Image binarization and analysis towards moving object detection and human gait recognition. Ph.D. Thesis, Department of Computer Science and Engineering, University of Calcutta (2014)
2. Document Image Binarization Contest (DIBCO), <http://utopia.duth.gr/~ipratika/DIBCO2013/>. Accessed in Dec 2013
3. Markus, F., Prokop, J., Johannes, S.: Automatic number plate recognition for the observance of travel behavior. In: 8th international conference on survey methods in transport: harmonisation and data comparability (2008)
4. Brain MRI Images: <http://overcode.yak.net/15>; Accessed in Jan 2014
5. Boberek, M., Saeed, K.: Segmentation of MRI brain images for automatic detection and precise localization of tumor. In: Choras, R.S. (ed.) Image Processing and Communications Challenges 3, AISC 102. Advances in soft computing series, pp. 333–341. Springer, Berlin (2011). (Kacprzyk J.—EiC)
6. Lin, C., Fan, K.-C.: Triangle-based approach to the detection of human face. Pattern Recogn. **34**, 1271–1284 (2001)
7. Kocjan, P., Saeed, K.: Face recognition in unconstrained environment. In: Biometrics and Kansei Engineering. Springer Science and Business Media, New York (2012)
8. Chen, F.-S., Fu, C.-M., Huang, C.-L.: Hand gesture recognition using a real-time tracking method and hidden Markov models. Image Vis. Comput. **21**, 745–758 (2003)
9. Nalepa, J., Grzejszczak, T., Kawulok, M.: Wrist localization in color images for hand gesture recognition, Proceedings of the International Conference on Man-Machine Interaction (ICMMI 2013), vol. **242**, pp. 79–86. Advances in Intelligent and Soft Computing. Springer (2013)
10. Wieclaw, L.: A minutiae-based matching algorithms in fingerprint recognition systems. J. Med. Inf. Tech. **13**, 65–71 (2009)

11. Surmacz, K., Saeed, K., Raptis, Piotr: An improved algorithm for feature extraction from a fingerprint fuzzy image. *Optica Applicata* **43**(3), 515–527 (2013)
12. Daugman, J.: How iris recognition works. *IEEE Trans. Circuits Syst. Video Tech.* **14**(1), (2004)
13. Misztal, K., Saeed, E., Tabor, J., Saeed, K.: Iris pattern recognition with a new mathematical model to its rotation detection. In: *Biometrics and Kansei Engineering*. Springer Science and Business Media, New York (2012)
14. Shaikh, .S. H., Saeed, K., Chaki, N.: Gait recognition using partial silhouette-based approach. In: *IEEE proceedings of the international conference on signal processing and integrated networks (SPIN)* (2014)
15. Image Sequence Server, Universitat Karlsruhe (TH), Institut Algorithmen und Kognitive Systeme (Group Prof. Dr. H.-H. Nagel); http://i21www.ira.uka.de/image_sequences/ Accessed in Jan 2014
16. Chinese Academy of Sciences, Institute of Automation (CASIA) IRIS Database. <http://biometrics.idealtest.org/dbDetailForUser.do?id=4>. Accessed in Jan 2014
17. Shaikh, S. H., Bhunia, S. K., Chaki, N.: On generation of silhouette of moving objects from video In: *Springer proceedings of the 4th international conference on signal and image processing (ICSIP)*, vol. 1, pp. 213–223. (2012)

Chapter 6

Conclusions

Binarization, as an early stage of image processing, is quite crucial in terms of reducing redundant information from the image. A grayscale image can be segmented in two groups as object and background by using a binarization technique. In other words, binarization reduces the space of pixel intensities to {0, 1} from a broader range of values [0, 255] in case of 8-bit grayscale images.

In small spatial regions of images, there may exist large variations of pixel intensities. The sharper the edges in an image, the higher is the local contrast, suggesting better result in identifying the foreground from the background. This forms the fundamental idea behind most of the binarization algorithms. A threshold value is calculated, and all the pixels having a gray level above the threshold is set to white (background), and all the pixels below the threshold is set to black (object) or vice versa. Binarization is nothing but this bi-level segmentation of images. This kind of segmentation helps in finding the region of interest from a given image. After the region of interest has been identified, the image may be processed locally taking the original grayscale or the color image. Image binarization is also very important as it reduces the size of an image for further processing.

The new binarization method in [Chap. 3](#) in its current state could be used for binarization with good performance as compared to many other common methods, especially for degraded text images. The iterative partitioning method performs better than some of the widely used binarization methods such as Bernsen, Niblack and Sauvola. Bernsen's method is based on image contrast. This method depends on the window size and the contrast threshold parameter k . In image regions with sharp contrast difference, this method proves to be suitable. However, for a low-contrast image, selecting a proper value for the parameter k is crucial. Niblack's dynamic thresholding method calculates a separate threshold for each pixel by shifting a window across the image. The window size and another constant k are important parameters for this method. k is set to -0.2 . This method can separate foreground object from background in the area around the object. However, in case where there is no object inside the local window, some parts of the background are regarded as foreground and the background noise is magnified. This artifact can be observed in Figs. [3.14](#) and [3.15](#) in Chap. 3.

Sauvola solved the problem of Niblack's method by assuming that the object and background pixels have gray values close to 0 and 255, respectively. In this method, the equation for threshold calculation is modified from Niblack's method by introducing another parameter R , a constant set to 128 for an image with 256 gray levels. The other parameter k is set to 0.5. This procedure gives satisfactory results in case of high contrast between the foreground and background pixels. However, the optimal values for R and k are proportional to the contrast of the object with respect to the background. In case of poor contrast images, Sauvola's method fails to detect foreground if the parameters are not set properly. This is evident from the results of Figs. 3.14 and 3.15 in Chap. 3.

The iterative partitioning method is also tested on degraded document images. The results are remarkably good when compared with Otsu's method for images containing noise/degradation in the form of non-uniform illumination, shadow, and ink-stain as observed in Fig. 3.10 in Chap. 3.

Future Works

The new methodology for image binarization that has been described in this text holds good potential to be used for various applications to offer reasonably good performance with low storage and time complexity.

First of all, iterative partition can be thought of as a framework for binarization. The concept of partitioning an image is a simple but effective methodology for retaining the relations among the pixels in a spatial neighborhood based on their intensity distribution. This concept can also be adopted for analyzing the performance of other global binarization algorithms.

Secondly, there is one problem observed in the iterative partitioning method due to local partitioning of the images. Sometimes, unwanted artifacts appear in the binarized image in form of “blocks” (Fig. 3.15 in Chap. 3). This problem can be solved by adjusting the size of the partitions and applying a block level smoothing in places that contain the block edges.

The concept of rectangular partition is quite common in image segmentation due to low computational cost in the process. However, analyzing the effect of partitions of different shapes can be another possible extension to the present work. The nature of partition actually depends on the intensity distribution of the pixels in the images. Analyzing the shapes of partition is itself another segmentation problem. Determination of non-uniform partitions can be computationally intensive. However, this is logical to consider this task targeting a particular application where the knowledge of different types of possible noises and shapes of the objects of interest is available a priori.

Finally, in iterative partitioning approach, Otsu's method has been used for calculating threshold for each partition. Otsu's method performs clustering based on image variance and is proved to be optimal where intensity distribution of the

pixels is bimodal. However, in the presence of noises, this method has failed badly. Therefore, it would be logical to calculate threshold based on a thresholding methodology that mostly suits the nature of noises present in a partition.

Appendix A

Sample Test Images

See Figs. [A.1](#), [A.2](#), [A.3](#), [A.4](#).

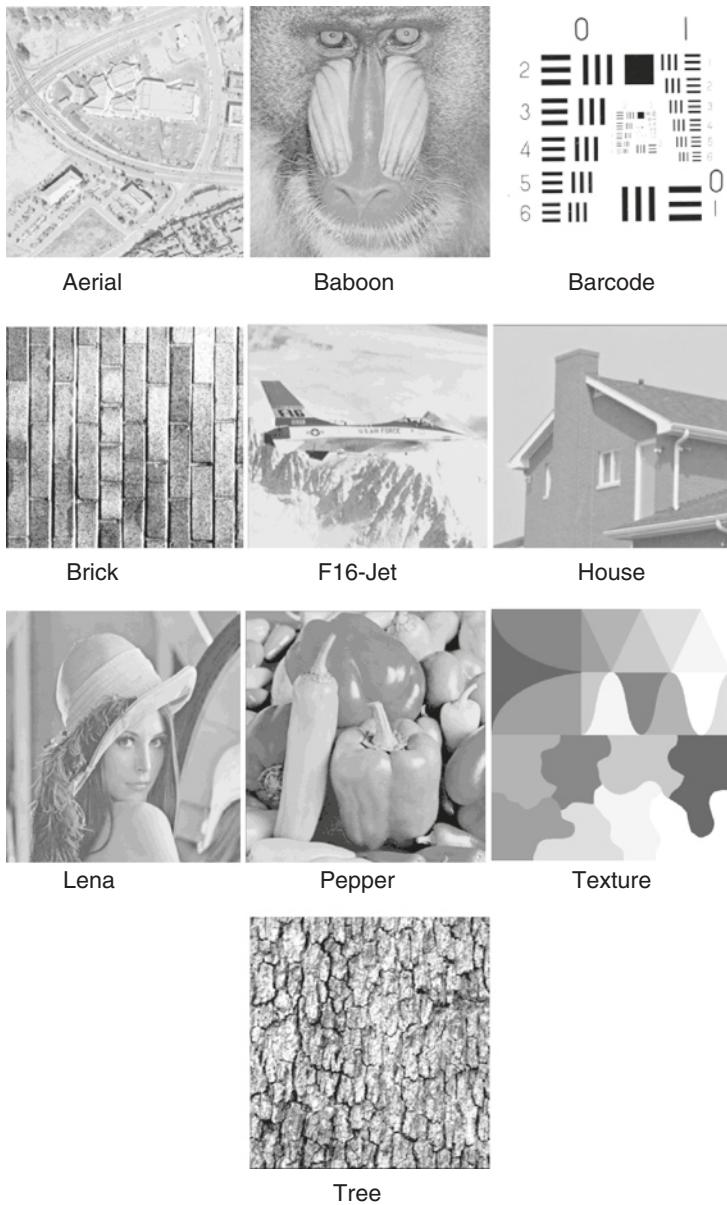


Fig. A.1 Selected Test Images from USC-SIPI Database *Source* USC-SIPI Image Database, <http://sipi.usc.edu/database/>, University of Southern California, Signal and Image Processing Institute

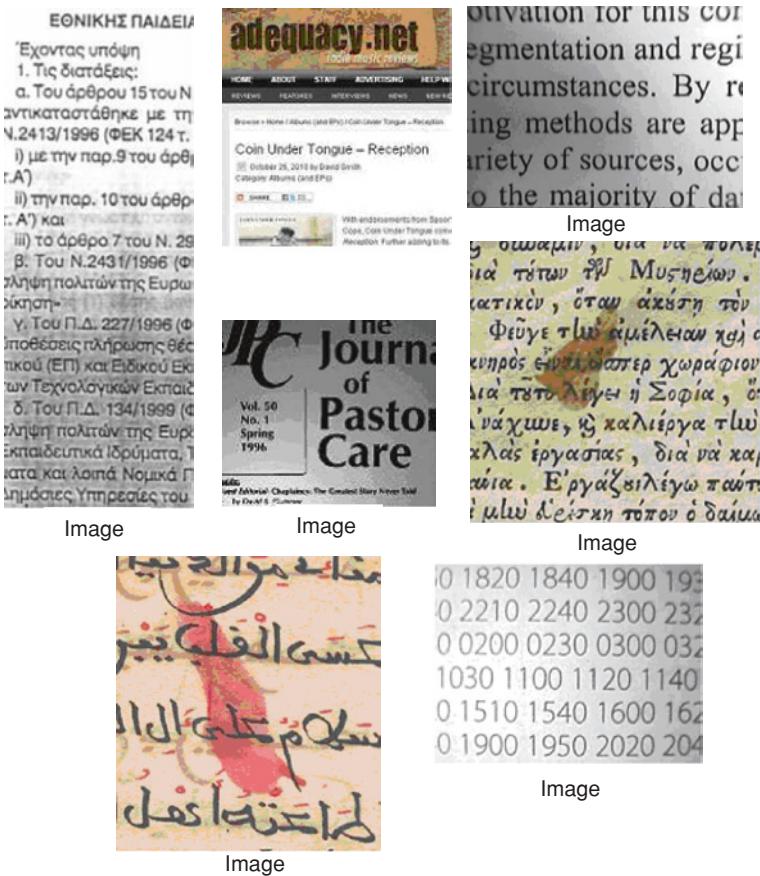


Fig. A.2 Test Document Images *Source* S. H. Shaikh, A. K. Maity, N. Chaki; “A New Image Binarization Method using Iterative Partitioning”; Springer Journal on Machine Vision and Applications (MVAP)



Fig. A.3 Test images from DIBCO Source Document Image Binarization Contest (DIBCO), <http://utopia.duth.gr/~ipratika/DIBCO2013/>: Accessed in December 2013



Fig. A.4 Reference images from DIBCO database Source Document Image Binarization Contest (DIBCO), <http://utopia.duth.gr/~ipratika/DIBCO2013/>: Accessed in December 2013

Index

A

AdOtsu, 9
ALLT, 10
Applications, 4, 65–68

B

Balanced histogram thresholding, 11
Bernsen, 6, 7
Binarization, 1, 2

D

Document image processing, 65
DIBCO, H-DIBCO, 11, 13

E

Evaluation Measures, 25–28

F

Face Detection, 68
Fingerprint recognition, 68
F-measure, 27

G

Gait recognition, 69

H

Hand gesture recognition, 67

I

Iris recognition, 68
Iterative Partitioning, 17–23

J

Johannsen, 6, 9

K

Kapur, 6, 7
Kittler, 6, 8

M

Medical image processing, 66
Misclassification Error (ME), 25
MSE, 27

N

Niblack, 6, 7
Noise immunity, 41–43

O

OCR, 65
Otsu, 6

P

Precision, 27
PERR, 27
PSNR, 27

R

Recall, [27](#)
Reference image, [31](#)
Relative foreground Area Error (RAE), [25](#)

S

Sauvola, [6, 7](#)
SNR, [27](#)
SVR, [11](#)

T

Threshold, [3](#)

U

USC-SIPI, [28](#)

V

Video processing, [66](#)