

8-2018

Information theoretic thresholding techniques based on particle swarm optimization.

Surina Surina
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>



Part of the [Other Applied Mathematics Commons](#)

Recommended Citation

Surina, Surina, "Information theoretic thresholding techniques based on particle swarm optimization." (2018). *Electronic Theses and Dissertations*. Paper 3040.
<https://doi.org/10.18297/etd/3040>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

INFORMATION THEORETIC THRESHOLDING TECHNIQUES BASED ON
PARTICLE SWARM OPTIMIZATION

By

Surina

B.S., Inner Mongolia University, 2007

M.S., Beihang University, 2012

M.A., University of Louisville, 2014

A Dissertation

Submitted to the Faculty of the
College of Arts and Sciences of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

in

Applied and Industrial Mathematics

Department of Mathematics

University of Louisville

Louisville, KY

August 2018

INFORMATION THEORETIC THRESHOLDING TECHNIQUES BASED ON
PARTICLE SWARM OPTIMIZATION

Submitted by

Surina

A Dissertation Approved on

July 10, 2018

by the following Dissertation Committee:

Dr. Yongzhi Xu, Dissertation Director
(Co-Advisor)

Dr. Prasanna Sahoo (Principal Advisor,
Deceased)

Dr. Thomas Riedel

Dr. Changbing Hu

Dr. El-Baz Ayman

DEDICATION

This dissertation is dedicated
to my beloved mother Hasigerile Bai, who has always loved me unconditionally
and taught me the purpose of life.
To my beloved father Zhurihe Borjigin, who has been my source of inspiration and
gave me strength in my life.
To my sister Arongna Borjigin and brother Suyila Borjigin, who always stand by
me and encourage me when things look bleak.
To my dearest husband Timothy Pervenecki, who constantly supports and
encourages me to achieve my goals and complete my dissertation successfully.

ACKNOWLEDGEMENTS

Foremost, I would like to express my deepest gratitude to my principal academic adviser, Prof. Prasanna Sahoo, for his valuable guidance and advice during the time of my Ph.D study and research, for his enthusiasm and immense knowledge to mathematics that had a great inspiratioin on both my doctoral research and life in general.

I would like to sincerely thank my thesis co-advisor, Dr. Yongzhi Xu for his guidance and support throughout my writing of this thesis.

I would also like to thank the rest of my thesis committee members: Dr. Thomas Riedel, Dr. Changbing Hu and Dr. El-Baz Ayman, for their generous help and insightful comments.

I am also grateful to all of my friends and colleagues who helped, supported and encouraged me during the time of graduate study.

I also thank my mother-in-law and father-in-law, Shelly Pervenecki and James Pervenecki, for their support and encouragement during my thesis defense.

Last but not least, I am grateful to my family: my parents, Hasigerile Bai and Zhurihe Borjigin, for giving birth to me at the first place and supporting me spiritually throughout my life; my sister and brother, Arongna Borjigin and Suyila Borjigin, for their continued love and support; my husband, Timothy Pervenecki, who experienced all of the ups and downs of my research. This dissertation would not have been possible without all of your warm love, continued patience and endless support.

All the images that are used in this thesis were downloaded from The Berke-

ley Segmentation Dataset and Benchmark.

ABSTRACT

INFORMATION THEORETIC THRESHOLDING TECHNIQUES BASED ON PARTICLE SWARM OPTIMIZATION

Surina

July 10, 2018

In this dissertation, we discuss multi-level image thresholding techniques based on information theoretic entropies.

In order to apply the correlation information of neighboring pixels of an image to obtain better segmentation results, we propose several multi-level thresholding models by using Gray-Level & Local-Average histogram (GLLA) and Gray-Level & Local-Variance histogram (GLLV). Firstly, a RGB color image thresholding model based on GLLA histogram and Tsallis-Havrda-Charvát entropy is discussed. We validate the multi-level thresholding criterion function by using mathematical induction. For each component image, we assign the mean value from each thresholded class to obtain three segmented component images independently. Then we obtain the segmented color image by combining the three segmented component images.

Secondly, we use the GLLV histogram to propose three novel entropic multi-level thresholding models based on Shannon entropy, Rényi entropy and Tsallis-Havrda-Charvát entropy respectively. Then we apply these models on the three components of a RGB color image to complete the RGB color image segmentation.

An entropic thresholding model is mostly about searching for the optimal threshold values by maximizing or minimizing a criterion function. We apply particle swarm optimization (PSO) algorithm to search the optimal threshold values

for all the models. We conduct the experiments extensively on The Berkeley Segmentation Dataset and Benchmark (BSDS300) and calculate the average four performance indices (Probability Rand Index, PRI , Global Consistency Error, GCE , Variation of Information, VOI and Boundary Displacement Error, BDE) to show the effectiveness and reasonability of the proposed models.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT	vi
LIST OF TABLES	xi
LIST OF FIGURES	xii
1. INTRODUCTION	1
1.1 Two-dimensional histogram	4
1.1.1 Gray-Level & Local Average histogram (GLLA)	5
1.1.2 Gray-Level & Local-Variance histogram (GLLV)	8
1.2 Image segmentation by thresholding	12
1.2.1 Bi-level image thresholding	12
1.2.2 Multi-level image thresholding	12
1.3 Shannon, Rényi and Tsallis-Havrda-Charvát entropies	14
1.3.1 Shannon entropy	14
1.3.2 Rényi entropy	14
1.3.3 Tsallis-Havrda-Charvát entropy	15
1.4 Particle swarm optimization algorithm	16
1.5 Thesis motivations, contributions and outline	19
2. ENTROPIC THRESHOLDING MODELS BASED ON ONE-DIMENSIONAL HISTOGRAM	22
2.1 Thresholding using shannon entropy	22
2.2 Thresholding using Rényi entropy	24

2.3	Thresholding using Tsallis-Havrda-Charvát entropy	26
2.4	Experiments	29
3.	ENTROPIC THRESHOLDING MODELS BASED ON TWO-DIMENSIONAL HISTOGRAMS	33
3.1	Entropic thresholding model based on Gray-Level & Local-Average histogram	33
3.1.1	Thresholding using Shannon entropy	35
3.1.2	Thresholding using Rényi entropy	37
3.1.3	Thresholding using Tsallis-Havrda-Charvát entropy	37
3.1.4	Experiments	38
3.2	Entropic thresholding model based on Gray-Level & Local-Variance histogram	41
3.2.1	Thresholding using Shannon entropy	44
3.2.2	Thresholding using Rényi entropy	45
3.2.3	Thresholding using Tsallis-Havrda-Charvát entropy	46
3.2.4	Experiments	47
4.	RGB COLOR IMAGE THRESHOLDING MODELS BASED ON ENTROPIES AND TWO-DIMENSIONAL HISTOGRAMS	50
4.1	The fusion of three thresholded component image	50
4.2	Experiments	51
5.	COMPARISON OF MODELS	58
5.1	Four performance indices	59
5.1.1	Probability rand index (PRI)	61
5.1.2	Global consistency error (GCE)	61
5.1.3	Variation of information (VOI)	63
5.1.4	Boundary displacement error (BDE)	64
5.2	Comparison of models	67

6. CONCLUSIONS AND FUTURE WORK	69
6.1 Conclusion	69
6.2 Future work	70
6.2.1 Symmetric padding method	70
6.2.2 Thresholding models based on other types of $2D$ his-	
tograms	71
REFERENCES	80
CURRICULUM VITAE	86

LIST OF TABLES

2.1	Mean values from each thresholded class.	31
2.2	Thresholds values for different entropic models.	32
3.1	Mean values from each thresholded class.	39
3.2	Thresholds values for different entropic models.	42
3.3	Mean values from each thresholded class.	48
3.4	Thresholds values for different entropic models.	48
4.1	Mean values and thresholds values from 1D Tsallis-Havrda-Charvát thresholding model.	52
4.2	21 colors from 1D Tsallis-Havrda-Charvát thresholding model.	53
4.3	Mean values and thresholds values from GLLA Tsallis-Havrda-Charvát thresholding model.	53
4.4	25 colors from GLLA Tsallis-Havrda-Charvát thresholding model.	54
4.5	Mean values and thresholds values from GLLV Tsallis-Havrda-Charvát thresholding model.	54
4.6	25 colors from GLLV Tsallis-Havrda-Charvát thresholding model.	54
5.1	Average performance indices for different algorithms.	68
6.1	Comparison of the average performance indices between the Tsallis-Havrda-Charvát based model and the Shannon based model.	70

LIST OF FIGURES

1.1	Gray-level image, its 1D histogram $h(t)$ and normalized 1D histogram $\hat{h}(t)$	4
1.2	Original RGB color image and its red, green and blue component images	5
1.3	Gray-level image, corresponding 1D histogram, GLLA histogram plane and 3D demonstration of GLLA.	7
1.4	Gray-level image, GLLV histogram plane, normalized GLLV histogram plane and 3D demonstration of normalized GLLV. . .	10
1.5	Gray-level and its corresponding g_{LA} image and g_{LV} image . .	11
1.6	Original image, bi-level image, three-level image, five-level image and their corresponding 1D histograms	13
1.7	Three segmented images	13
1.8	Illustration of PSO algorithm	18
2.1	Three-level segmentations with the corresponding 1D histograms	30
2.2	Ten-level segmentations with the corresponding 1D histograms	31
3.1	GLLA histogram planes with bi-level and three-level segmentation	36
3.2	Original image, the three-level thresholded images and their corresponding 1D and GLLA histograms	40
3.3	Original image, the ten-level thresholded images and their corresponding 1D and GLLA histograms	41
3.4	GLLV histogram planes with bi-level and 3-level segmentation	42

3.5	Original image, the ten-level thresholded images and their corresponding 1D and GLLV histograms	49
4.1	Red, Green and Blue component of 113016.jpg and their corresponding 1D histograms, GLLA histograms and GLLV histograms	55
4.2	Thresholded component images and the fusion RGB image from 1D Tsallis-Havrda-Charvát thresholding model	56
4.3	Three thresholded component images and the fusion RGB image from GLLA Tsallis-Havrda-Charvát model	56
4.4	Three thresholded component images and the fusion RGB image from GLLV Tsallis-Havrda-Charvát model	57
5.1	Benchmark images of 113016.jpg from BSDS300.	59
5.2	The original 113016.jpg, thresholded image and labeled image (thresholding levels are R: 3, G: 4, B: 3)	60
5.3	Label 2, 15 and 25 for the thresholded image	60
6.1	From left to right: first row: GLLE, GLSC and GLGM histogram planes; second row: local-entropy image, spatial-correlation image and gradient-magnitude image; third row: the corresponding 3D demonstrations for each histogram plane.	72

CHAPTER 1

INTRODUCTION

Image segmentation is a process that partitions an image into non-overlapping regions such that each region is homogeneous in terms of some features, such as color, texture and brightness. Image thresholding, because of its simplicity, is one of the most widely used segmentation techniques. Generally speaking, based on the gray level histogram of an image, thresholding models are grouped into two classes: bi-level thresholding and multi-level thresholding. Under the assumption that an image has only two homogeneous regions, bi-level thresholding models [1, 4, 9, 15, 28, 30, 31, 45, 49] classify the pixels of an image into two groups, called object and background, by using one threshold value. However, in many applications, one must deal with multi-modal images, such that multi-level thresholding models [7, 11, 18, 23, 34, 35, 37, 47, 48] are used to segment the pixels of an image into multiple classes by using more than two threshold values.

The notion of a one-dimensional ($1D$) histogram [9, 10, 15, 20, 28, 33, 37, 47] has been used in thresholding techniques for years. It is derived from the gray level information of an image, so it does not take into account the spatial correlation between a pixel and its neighbor pixels. In order to overcome this drawback, people presented several types of two-dimensional histograms ($2D$) in the past few years. The first thresholding model based on a $2D$ histogram and Shannon entropy was introduced in 1989 [1]. This $2D$ histogram was constructed by using the gray level of each pixel and the average gray level from the local neighborhood of the pixel and was named the Gray-Level & Local-Average histogram (GLLA). In 2017, a new

type of 2D histogram, named the Gray-Level & Local-Variance histogram (GLLV) was developed [49]. The GLLV histogram is constructed by applying the dispersion of gray level distribution of pixels in a neighborhood. In this dissertation, we apply GLLA and GLLV histograms to build the proposed models. In general, thresholding methods based on 2D histograms [1, 18, 19, 22, 30, 31, 34, 48] perform better than the 1D histogram methods [2, 9, 11, 28, 35, 37, 47].

Recently, entropic thresholding techniques have attracted more and more attention [2, 9, 11, 14, 18, 19, 29, 32, 42, 47, 48, 49], [28] - [37]. Entropy [8, 13, 16, 39] is originally from thermodynamics and is proposed as a measure of the information of a random signal. According to the information theory, an entropy of a random process is the amount of information in the process [13]. A measure of the information is defined as certain formulations of the probability distribution from the process [13, 36]. An entropic thresholding model is mostly about searching for optimal threshold values by maximizing or minimizing an entropic criterion function. In 2004 [9], a 1D bi-level thresholding model was presented based on the non-extensive property of the Tsallis entropy. In the same year, P. K. Sahoo and G. Arora [30] proposed a Rényi entropic bi-level thresholding method by using the GLLA histogram. In 2006 [31], they combined the GLLA histogram with the Tsallis-Havrda-Charvát entropy and proposed a Tsallis-Havrda-Charvát entropic bi-level thresholding model. As for the multi-level thresholding, normally formulating the multi-level criterion function based on the 1D Shannon entropy [11] is not a difficult task because of the extensive property of Shannon entropy. But formulating the multi-level criterion function based on the Tsallis-Havrda-Charvát entropy is not easy. A. C. Sparavigna [37], in 2015, formulated a multi-level thresholding model based on the 1D Tsallis entropy. Furthermore, in 2017, A. B. Ishak [18, 19] presented two multi-level thresholding models based on the GLLA histogram by using the Rényi entropy and the Tsallis entropy respectively. We point out that, in 2016, a

multi-level thresholding model based on the GLLA histogram and Kullback-Leibler divergence, instead of entropy, was developed in [48].

We know that most of the bi-level thresholding methods can be extended to multi-level with some appropriate modifications. But, the $2D$ multi-level extension gives rise to the exponential increase of computational time [30], since exhaustively searching for the optimal threshold values of a multi-level thresholding is an NP-hard combinatorial optimization problem [18]. In order to reduce the computational time, in recent years, people have been focusing on metaheuristic algorithms, such as, Artificial Bee Colony (ABC) approach [47], Differential Evolution (DE) approach [34], Quantum Genetic (QG) algorithm [18] and Particle Swarm Optimization (PSO) algorithm [11, 21, 35, 37, 48]. In this dissertation, we implement our experiments by applying PSO algorithm because of its simplicity in concept, time efficiency and highly convergent properties.

We test our method on The Berkeley Segmentation Dataset and Benchmark (BSDS300) extensively by computing the average four performance indices (Probability Rand Index, PRI , Global Consistency Error, GCE , Variation of Information, VOI and Boundary Displacement Error, BDE). We compare the average four performance indices of the models from our dissertation with the ones in [48] to illustrate the effectiveness and reasonability of our models.

Mathematically, a gray-level image is a function $f(x, y): \mathbb{Z}_M \times \mathbb{Z}_N \rightarrow G$, where $\mathbb{Z}_M = \{1, 2, \dots, M\}$ for $M \geq 2$, and $G = \{0, 1, \dots, 255\}$ is the gray levels of the image. For example, the image 113016.jpg in the Figure 1.1 (a) has 225 gray levels. The $1D$ histogram of a gray-level image $f(x, y)$ provides information about the gray level distribution of the image, so the $1D$ histogram is a map $h: G \rightarrow \mathbb{N}$ such that:

$$h(t) = \text{the number of pixels with gray level } t ,$$

where $t \in G = \{0, 1, \dots, 255\}$.

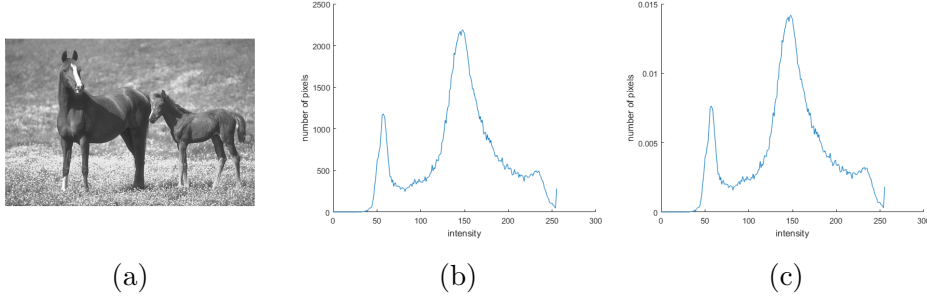


Figure 1.1 – Gray-level image, its 1D histogram $h(t)$ and normalized 1D histogram $\hat{h}(t)$

Then the normalized 1D histogram is:

$$\hat{h}(t) = \frac{h(t)}{\text{total \# of pixels : } M \times N} ,$$

where $t \in G = \{0, 1, \dots, 255\}$. What's more, we have $\sum_{t=0}^{255} \hat{h}(t) = 1$. Figure 1.1 (b) and (c) illustrate the 1D histogram $h(t)$ and normalized 1D histogram $\hat{h}(t)$ of the corresponding gray-level image respectively.

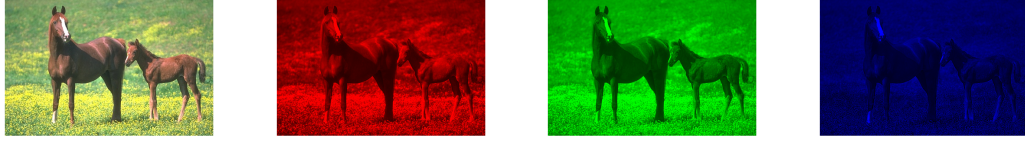
A RGB color image is a vector function $\vec{f}(x, y) : \mathbb{Z}_M \times \mathbb{Z}_N \rightarrow G \times G \times G$ such that:

$$\vec{f}(x, y) = [f_r(x, y), f_g(x, y), f_b(x, y)] ,$$

where $f_r(x, y)$, $f_g(x, y)$, $f_b(x, y)$ are red, green and blue components whose mixtures generate any color that can be displayed. We use $f_c(x, y)$ to represent an arbitrary (red, green or blue) component image. Thus a RGB color image is an $M \times N \times 3$ array of color pixels, where each color pixel is a triplet corresponding to the red, green and blue components at a specific spatial location. Figure 1.2 shows a RGB color image and its three component images.

1.1 Two-dimensional histogram

In this section, we introduce the Gray-Level & Local-Average histogram (GLLA) and the Gray-Level & Local-Variance histogram (GLLV) and use a simple



(a) RGB image (b) Red component (c) Green component (d) Blue component

Figure 1.2 – Original RGB color image and its red, green and blue component images

matrix f as an example to show how to construct each $2D$ histogram mathematically.

For the pixels that are from the top and bottom rows and right and left columns of the matrix f , since they do not have enough neighboring pixels, we pad zeros in their neighbors. For example, the entry $f(1,1) = 1$ only has three neighboring pixels $f(1,2) = 2$, $f(2,1) = 6$ and $f(2,2) = 7$, so we pad five zeros in its neighbor if we want to use a 3×3 neighborhood of the pixel. Thus we obtain a matrix $f_{zeropad}$, which is used in the $2D$ histogram construction. For example:

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{pmatrix} \Rightarrow f_{zeropad} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 0 \\ 0 & 6 & 7 & 8 & 9 & 10 & 0 \\ 0 & 11 & 12 & 13 & 14 & 15 & 0 \\ 0 & 16 & 17 & 18 & 19 & 20 & 0 \\ 0 & 21 & 22 & 23 & 24 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

1.1.1 Gray-Level & Local Average histogram (GLLA)

The Gray-Level & Local Average histogram (GLLA), known as the traditional two-dimensional histogram [1], is constructed from the gray level of each pixel and the average gray level value of its neighborhood.

Let $g(x, y)$ be the average gray level value of the 3×3 neighborhood of the

pixel that is located at the point (x, y) . Then it is calculated as:

$$g(x, y) = \left\lfloor \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j) \right\rfloor, \quad (1.1)$$

where $\lfloor r \rfloor$ denotes the floor function, or the integer part of the number r . Then the GLLA histogram is formulated as:

$$h(t, s) = \text{Prob}(f(x, y) = t \text{ and } g(x, y) = s),$$

where Prob refers to the number of pixels that satisfy $f(x, y) = t$ and $g(x, y) = s$ with $t, s \in G$.

For a given image, there are several methods to estimate this density function. One of the most frequently used methods is the *method of relative frequency* [30, 31]. The normalized histogram is given by the following formula:

$$\hat{h}(t, s) = \frac{h(t, s)}{\text{total \# of pixels of } f(x) (M \times N)}.$$

The joint probability mass function $p(t, s)$ is given by:

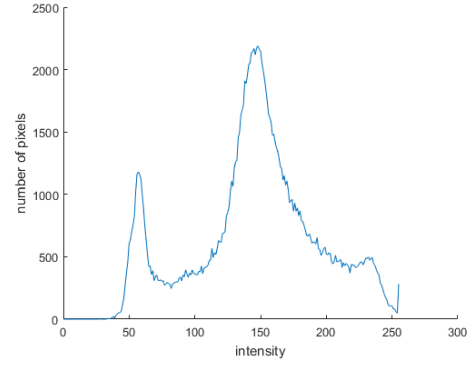
$$p(t, s) = \hat{h}(t, s),$$

where $t, s = 0, 1, \dots, 255$. Figure 1.3 illustrates the (a) gray-level image, (b) corresponding 1D histogram, (c) GLLA histogram plane and (d) 3D demonstration of the GLLA histogram. From Figure 1.3 (c), we notice the information we are interested in is mainly distributed along the main diagonal on the 2D histogram plane because the average gray level value from the 3×3 neighborhood of a pixel is very close to the original gray level. We consider this information as the background and object information.

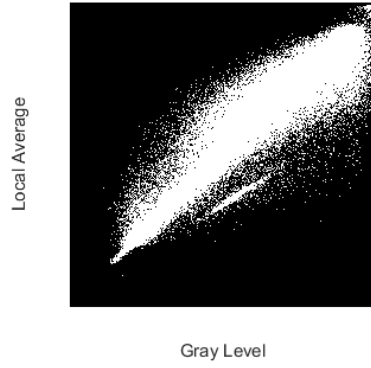
In the matrix $f_{zeropad}$, if we apply equation (1.1) to the entry $f(1, 1) = 1$, then the local average value $g(1, 1)$ from its 3×3 neighborhood is computed as



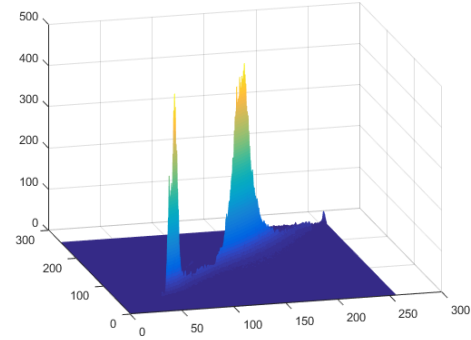
(a)



(b)



(c)



(d)

Figure 1.3–Gray-level image, corresponding 1D histogram, GLLA histogram plane and 3D demonstration of GLLA.

follows:

$$f_{zeropad} = \begin{pmatrix} \begin{matrix} 0 & 0 & 0 \\ 0 & \color{red}{1} & 2 \\ 0 & 6 & 7 \end{matrix} & \begin{matrix} 0 & 0 & 0 & 0 \\ 3 & 4 & 5 & 0 \\ 8 & 9 & 10 & 0 \end{matrix} \\ 0 & 11 & 12 & 13 & 14 & 15 & 0 \\ 0 & 16 & 17 & 18 & 19 & 20 & 0 \\ 0 & 21 & 22 & 23 & 24 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$g(1, 1) = \left\lfloor \frac{(1 + 2 + 6 + 7)}{9} \right\rfloor = \lfloor 1.7778 \rfloor = 2.$$

In the same manner, we can obtain the local average values for all the entries of f

such that the local average matrix g_{LA} for the original matrix f is as follows:

$$g_{LA} = \left[\begin{pmatrix} 1.7778 & 3 & 3.6667 & 4.3333 & 3.1111 \\ 4.3333 & 7 & 8 & 9 & 6.3333 \\ 7.6667 & 12 & 13 & 14 & 9.6667 \\ 11 & 17 & 18 & 19 & 13 \\ 8.4444 & 13 & 13.6667 & 14.3333 & 9.7778 \end{pmatrix} \right] = \begin{pmatrix} 2 & 3 & 4 & 4 & 3 \\ 4 & 7 & 8 & 9 & 6 \\ 8 & 12 & 13 & 14 & 10 \\ 11 & 17 & 18 & 19 & 13 \\ 8 & 13 & 14 & 14 & 10 \end{pmatrix}.$$

1.1.2 Gray-Level & Local-Variance histogram (GLLV)

The Gray-Level & Local-Variance histogram (GLLV) [49] is constructed from the gray level of each pixel and the local variance from its neighborhood. Local variance is a measure of the dispersion of the gray level distribution of a pixel in a neighborhood. If a pixel's gray level is close to the gray levels from its neighboring pixels, then its local variance has a small value, and vice versa. In general, the object and background of an image are homogeneous regions. A pixel from the object and background has a gray level which is close to its neighboring pixel's gray level, so that this pixel has small local variance value. At the same time, the pixel from the edges and noise regions of an image has a large local variance value. In this dissertaion, we use the local sample standard variance, instead of local sample variance that was used in [49], to construct GLLV histogram.

Let $g(x, y)$ be the local sample standard variance function from the 3×3 neighborhood of a pixel located at the point (x, y) . Then $g(x, y)$ is calculated as follows:

$$g(x, y) = \sqrt{\frac{1}{9-1} \sum_{i=-1}^1 \sum_{j=-1}^1 [f(x+i, y+j) - \bar{f}(x, y)]^2}, \quad (1.2)$$

where $\bar{f}(x, y)$ is the average gray level in the 3×3 neighborhood of this pixel, which

also is calculated by equation (1.1):

$$\bar{f}(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j) .$$

Let $g_{\min} = \min_{(x,y) \in \mathbb{Z}_M \times \mathbb{Z}_N} g(x, y)$ and $g_{\max} = \max_{(x,y) \in \mathbb{Z}_M \times \mathbb{Z}_N} g(x, y)$. Then the function $g(x, y)$ has boundaries as follows:

$$g_{\min} \leq g(x, y) \leq g_{\max} , \quad (1.3)$$

According to the paper [49], we normalize the local sample standard variance as:

$$g(x, y) = \left\lfloor \frac{g(x, y) - g_{\min}}{g_{\max} - g_{\min}} \times L' \right\rfloor . \quad (1.4)$$

where $L' = 64$ according to the experience. Thus $g(x, y)$ is bounded by 0 and 64 for all the experiments. We point out that if we use the original g_{\min} and g_{\max} as the boundaries of $g(x, y)$, then a different image has different boundaries of $g(x, y)$.

Thus the GLLV histogram is formulated as density function as follows:

$$h(t, s) = Prob(f(x, y) = t \text{ and } g(x, y) = s) ,$$

where $Prob$ refers to the number of pixels that satisfy $f(x, y) = t$ and $g(x, y) = s$ with $t = 0, 1, \dots, 255$, $s = 0, 1, \dots, 64$. So the normalized GLLV histogram is approximated by using the formula:

$$\hat{h}(t, s) = \frac{h(t, s)}{\text{total \# of pixels of } f(x) (M \times N)} .$$

The joint probability mass function $p(t, s)$ is given by:

$$p(t, s) = \hat{h}(t, s),$$

where $t = 0, 1, \dots, 255$, $s = 0, 1, \dots, 64$. Figure 1.4 illustrates the (a) gray-level image, (b) GLLV histogram plane, (c) normalized GLLV histogram plane and (d) 3D demonstration of normalized GLLV. We point out that the object and background information is mainly distributed at the lower part on the GLLV histogram

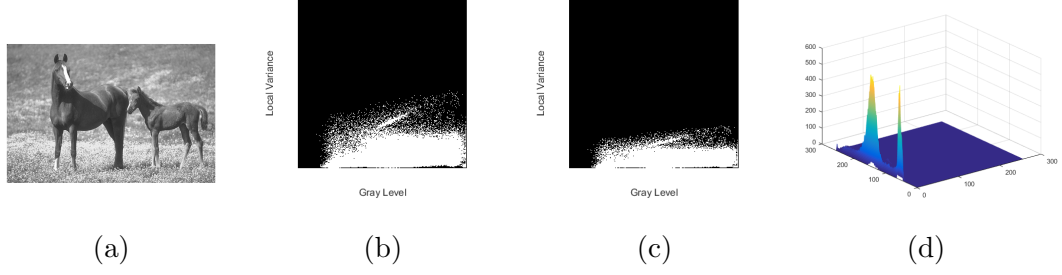


Figure 1.4 – Gray-level image, GLLV histogram plane, normalized GLLV histogram plane and 3D demonstration of normalized GLLV.

plane (Figure 1.4 (b)) with upper boundary of 117 and normalized GLLV histogram plane (Figure 1.4 (c)) with upper boundary of 64. Since GLLV and normalized GLLV don't have a big difference, we mainly use the normalized GLLV histogram to construct the thresholding models in this dissertation.

Since we already know $\bar{f}(1, 1) = 1.7778$ for the entry $f(1, 1) = 1$ in the matrix $f_{zeropad}$, we plug $\bar{f}(1, 1)$ into equation (1.2) to obtain the local sample standard variance $g(1, 1)$ from its 3×3 neighborhood as follows:

$$g(1, 1) = \sqrt{\frac{5 \cdot (-1.7778)^2 + (1 - 1.7778)^2 + (2 - 1.7778)^2 + (6 - 1.7778)^2 + (7 - 1.7778)^2}{8}}$$

$$= 2.7739 .$$

In the same manner, we can obtain the local sample standard variances for all the entries of f such that the local variance matrix g_{LV} for the original matrix f is as follows:

$$g_{LV} = \begin{pmatrix} \textcolor{red}{2.7739} & 3.2016 & 3.5707 & 3.9686 & 4.1062 \\ 4.8218 & 4.4159 & 4.4159 & 4.4159 & 5.9372 \\ 6.7639 & 4.4159 & 4.4159 & 4.4159 & 8.0777 \\ 8.9861 & 4.4159 & 4.4159 & 4.4159 & 10.3803 \\ 10.1749 & 10.0125 & 10.5000 & 10.9886 & \textcolor{blue}{11.7343} \end{pmatrix} ,$$

where $g_{\min} = 2.7739$ and $g_{\max} = 11.7343$. Then we apply the equation (1.4) to

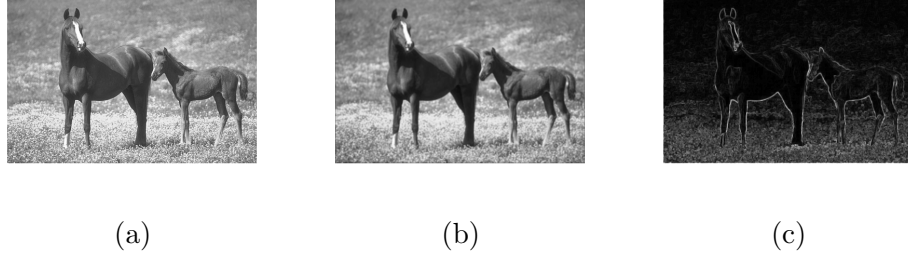


Figure 1.5– Gray-level and its corresponding g_{LA} image and g_{LV} image

normalize g_{LV} :

$$\begin{aligned}
 g_{LV} &= \left[\begin{pmatrix} 0 & 3.0547 & 5.6913 & 8.5334 & 9.5163 \\ 14.6274 & 11.7280 & 11.7280 & 11.7280 & 22.5938 \\ 28.4985 & 11.7280 & 11.7280 & 11.7280 & 37.8829 \\ 44.3708 & 11.7280 & 11.7280 & 11.7280 & 54.3286 \\ 52.8615 & 51.7018 & 55.1838 & 58.6739 & 64 \end{pmatrix} \right] \\
 &= \begin{pmatrix} 0 & 3 & 6 & 9 & 10 \\ 15 & 12 & 12 & 12 & 23 \\ 28 & 12 & 12 & 12 & 38 \\ 44 & 12 & 12 & 12 & 54 \\ 53 & 52 & 55 & 59 & 64 \end{pmatrix} .
 \end{aligned} \tag{1.5}$$

Figure 1.5 demonstrates the corresponding (b): g_{LA} image and (c): g_{LV} image for the gray-level image 1113016.jpg (a). Since the equation (1.1) has a smoothing effect on the original image (a), that means, the local average image (b) is a blurred version of (a). And the equation (1.2) measures the difference among gray levels, so the local variance image (c) shows the edges of the original image (a). We know in a gray-level image, 0 implies black color, and 255 implies white color. We can see from Figure 1.5 (c) that the edge pixels are illustrated in brighter colors because they have larger local variance values. What's more, the main body of the horses and the background pixels are in darker colors, because these pixels have smaller

local variance values.

1.2 Image segmentation by thresholding

1.2.1 Bi-level image thresholding

The image thresholding methods can be classified as bi-level and multi-level thresholding. It is well known that 1D bi-level thresholding methods [1, 9, 30, 31] require only one threshold value $\tau \in G$ to partition the gray level set G into two disjointed classes with levels $\{0, \dots, \tau\}$ and $\{\tau+1, \dots, 255\}$ respectively. For all the models in this dissertation, we assign the mean values from each class to the pixels from that class. Then the segmented image $Seg_2(x, y)$ has only two gray levels, m_1 and m_2 , which are the mean values from class one and class two respectively:

$$Seg_2(x, y) = \begin{cases} m_1, & \text{if } 0 \leq f(x, y) \leq \tau \\ m_2, & \text{if } (\tau + 1) \leq f(x, y) \leq 255 \end{cases} \quad (1.6)$$

1.2.2 Multi-level image thresholding

Generally speaking, n threshold values, $\{\tau_1, \dots, \tau_n\} \in G$, are required for the $(n + 1)$ -level thresholding methods [11, 35, 37, 47]. Then the segmented image $Seg_{n+1}(x, y)$ has $n + 1$ gray levels, $\{m_1, \dots, m_{n+1}\}$, which are the mean values from corresponding classes respectively:

$$Seg_{n+1}(x, y) = \begin{cases} m_1, & \text{if } 0 \leq f(x, y) \leq \tau_1 \\ m_2, & \text{if } (\tau_1 + 1) \leq f(x, y) \leq \tau_2 \\ \dots & \\ m_{n+1}, & \text{if } (\tau_n + 1) \leq f(x, y) \leq 255 \end{cases} \quad (1.7)$$

Figure 1.6 demonstrates (a) the original image, (b) the bi-level thresholded image S_2 with $\tau = 100$, (c) the three-level thresholded image S_3 with $\tau_1 = 100$, $\tau_2 =$

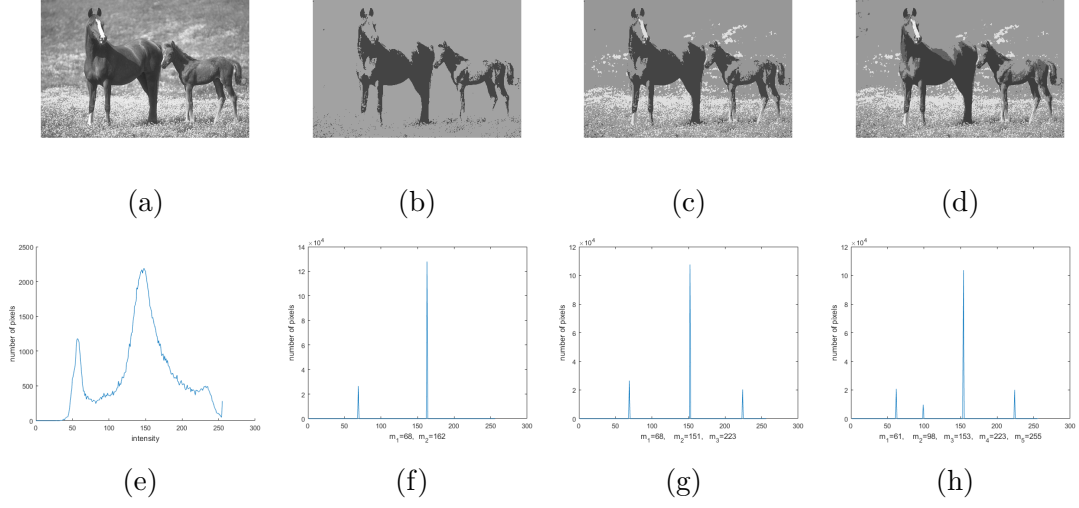


Figure 1.6–Original image, bi-level image, three-level image, five-level image and their corresponding 1D histograms

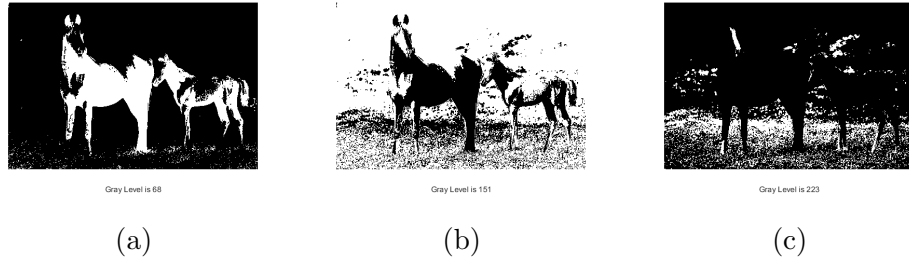


Figure 1.7–Three segmented images

200, (d) the five-level thresholded image S_5 with $\tau_1 = 83$, $\tau_2 = 110$, $\tau_3 = 200$, $\tau_4 = 254$, and their corresponding 1D histograms (e)-(h).

We know that Figure 1.6 (c) has only three gray levels: 68, 151 and 223. Figure 1.7 exhibits the three segmented binary images from Figure 1.6 (c). In Matlab, for a binary image, gray level 0 means “false” (black color), and 1 means “true” (white color). In Figure 1.7 (a), the main body parts of the two horses are shown in white. These pixels have gray level 68 in Figure 1.6 (c). In Figure 1.7 (b), most of the background pixels are shown in white. These pixels have gray level 151 in Figure 1.6 (c). In Figure 1.7 (c), most of the grasses are shown in white. These pixels have gray level 223 in Figure 1.6 (c). This is how we explain and apply the thresholded image Figure 1.6 (c) in a real application. We can use the same manner

to explain other thresholded images with different gray levels.

1.3 Shannon, Rényi and Tsallis-Havrda-Charvát entropies

Entropy [8, 13, 16, 39] is originally from thermodynamics, which is related to the internal energy of a system. According to information theory, entropy is defined as certain formulations of the probability distribution from a discrete system.

Assume that $P = \{p_i\}_{i=1}^{i=n}$ is the probability distribution of the discrete system A , where $0 \leq p_i \leq 1$, $i = 1, 2, \dots, n$, $n \geq 2$ and $\sum_{i=1}^n p_i = 1$.

1.3.1 Shannon entropy

In 1948, Shannon [36] introduced the seminal work of Shannon entropy, which is formulated as:

$$S(P) = - \sum_{i=1}^n p_i \ln p_i. \quad (1.8)$$

Shannon entropy has an extensive property, that is, its value depends on the amount of information that it presents. Let us consider a physical system that is decomposed into two statistical independent subsystems A and B . According to the extensive property, the combined system $A \cup B$ has a Shannon entropy $S(A + B) = S(A) + S(B)$.

1.3.2 Rényi entropy

In 1961, Rényi [27] extended the Shannon entropy to an entropy of order α , which is a real positive parameter and $\alpha \neq 1$:

$$R(P) = \frac{1}{1 - \alpha} \cdot \ln \sum_{i=1}^n p_i^\alpha. \quad (1.9)$$

Actually, the Rényi entropy becomes the Shannon entropy when α is close to 1. We prove it as follows:

$$\begin{aligned}
\lim_{\alpha \rightarrow 1} R(P) &:= \lim_{\alpha \rightarrow 1} \frac{1}{1 - \alpha} \cdot \ln \sum_{i=1}^n p_i^\alpha && (\text{L'Hôpital's rule}) \\
&= \lim_{\alpha \rightarrow 1} \frac{(\ln \sum_{i=1}^n p_i^\alpha)'}{(1 - \alpha)'} \\
&= \lim_{\alpha \rightarrow 1} \frac{\frac{1}{\sum_{i=1}^n p_i^\alpha} \cdot (\sum_{i=1}^n p_i^\alpha)'}{-1} \\
&= \lim_{\alpha \rightarrow 1} \frac{-\sum_{i=1}^n (p_i^\alpha)'}{\sum_{i=1}^n p_i^\alpha} && (1.10) \\
&= \lim_{\alpha \rightarrow 1} \frac{-\sum_{i=1}^n p_i^\alpha \ln p_i}{\sum_{i=1}^n p_i^\alpha} \\
&= \frac{-\sum_{i=1}^n p_i \ln p_i}{1} \\
&= -\sum_{i=1}^n p_i \ln p_i \\
&= S(p) .
\end{aligned}$$

The Rényi entropy also has an extensive property, that means the combined system $A \cup B$ has the Rényi entropy $R(A + B) = R(A) + R(B)$.

1.3.3 Tsallis-Havrda-Charvát entropy

In 1967, Havrda and Charvát [16] developed a structural α -entropy, where α is a real positive parameter and $\alpha \neq 1$:

$$H(P) = \frac{1}{1 - 2^{1-\alpha}} \left[1 - \sum_{i=1}^n p_i^\alpha \right] . \quad (1.11)$$

Independently, in 1988, Tsallis [39] proposed another α -entropy that has the same expression as equation (1.11) but a different coefficient:

$$H(P) = \frac{1}{\alpha - 1} \left[1 - \sum_{i=1}^n p_i^\alpha \right] . \quad (1.12)$$

In equations (1.9), (1.11) and (1.12), α is an arbitrary real positive parameter not equal to 1, and the limiting case for $\alpha \rightarrow 1$ is the Shannon entropy given by equation

(1.8).

P. K. Sahoo and G. Arora [31] concluded that the two entropies, equations (1.11) and (1.12), yield the same result and proposed the Tsallis-Havrda-Charvát entropy with degree α :

$$H(P) = N(\alpha) \left[1 - \sum_{i=1}^n p_i^\alpha \right], \quad (1.13)$$

where $N(\alpha)$ is a normalizing factor. Thus in equation (1.11), $N(\alpha) = \frac{1}{1-2^{1-\alpha}}$, and in equation (1.12), $N(\alpha) = \frac{1}{\alpha-1}$.

Actually, the Tsallis-Havrda-Charvát entropy becomes the Shannon entropy when α is close to 1. We prove it as follows:

$$\begin{aligned} \lim_{\alpha \rightarrow 1} T(P) &:= \lim_{\alpha \rightarrow 1} \frac{1}{\alpha - 1} \cdot \left[1 - \sum_{i=1}^n p_i^\alpha \right] && \text{(L'Hôpital's rule)} \\ &= \lim_{\alpha \rightarrow 1} \frac{(1 - \sum_{i=1}^n p_i^\alpha)'}{(\alpha - 1)'} \\ &= \lim_{\alpha \rightarrow 1} \frac{-\sum_{i=1}^n (p_i^\alpha)'}{1} \\ &= \lim_{\alpha \rightarrow 1} \frac{-\sum_{i=1}^n p_i^\alpha \ln p_i}{1} \\ &= -\sum_{i=1}^n p_i \ln p_i \\ &= S(p) . \end{aligned} \quad (1.14)$$

The Tsallis-Havrda-Charvát entropy has a non-extensive property. The entropy of the combined system $A \cup B$ follows the non-additivity rule [8, 39]:

$$H(A + B) = H(A) + H(B) + (1 - \alpha) \cdot H(A) \cdot H(B) .$$

1.4 Particle swarm optimization algorithm

The particle swarm optimization was discovered by J. Kennedy and R. Eberhart in 1995 [21] from the simulation of simplified swarming behaviors such as bird

flocking, fish schooling and bee swarming. Imagine a group of flying birds is searching for food in an area with only one piece of food available and each bird has a position and a velocity at any time. The bird changes its position by adjusting the velocity according to how far the food is from its position. The velocity changes based on its past experience and the feedbacks received from its neighbors.

In the application, each solution is considered as a bird, which is called a particle with a fitness value that can be calculated using an objective function. All the particles preserve their personal best positions, and there is a global best position for the entire group. They adjust their velocities by considering their personal best performances and the global best performance of the group and change their positions by adjusting the velocities. We point out that the entropic criterion function in our model is actually the objective function in the PSO algorithm.

Mathematically, PSO is a stochastic process of optimizing a continuous non-linear objective function by moving a number of particles in a n -dimensional searching space. Suppose that we want to maximize a real-valued multi-variable objective function $f(\vec{x}_i) : \mathbb{R}^n \rightarrow \mathbb{R}$ by using the PSO method. We need a swarm of m particles in the n -dimensional space and the i^{th} ($i = 1, \dots, m$) particle has:

- position vector $\vec{x}_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{in}(t))^T$
- velocity vector $\vec{v}_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{in}(t))^T$
- previously best visited position (personal best position)

$$\vec{p}_i(t) = (p_{i1}(t), p_{i2}(t), \dots, p_{in}(t))^T .$$

For the entire swarm, there is a global best position $\vec{g}(t) = (g_1(t), g_2(t), \dots, g_n(t))^T$. Then the velocity vector of the i^{th} particle is updated based on its personal best and global best position:

$$\vec{v}_i(t+1) = \omega \cdot \vec{v}_i(t) + c_1 \cdot r_1 \cdot [\vec{p}_i(t) - \vec{x}_i(t)] + c_2 \cdot r_2 \cdot [\vec{g}(t) - \vec{x}_i(t)] , \quad (1.15)$$

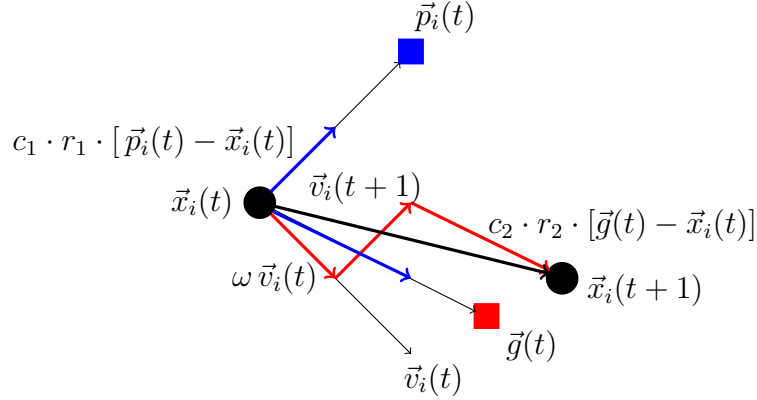


Figure 1.8 – Illustration of PSO algorithm

where ω is an inertia weight; c_1 is a cognitive parameter; c_2 is a social parameter; r_1, r_2 are random numbers in the interval $[0, 1]$, and t is the iteration number.

Then, the position vector \vec{x}_i of the i^{th} particle is updated according to the updated velocity vector $\vec{v}_i(t+1)$:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1). \quad (1.16)$$

Then, we obtain the fitness value of $\vec{x}_i(t+1)$ by plugging it into the objective function f . Figure 1.8 illustrates the dynamic movement of the particle \vec{x}_i . We can see that the particle \vec{x}_i moves along with the red line from moment t to moment $t+1$ according to equation (1.15). If the fitness value $f(\vec{x}_i(t+1))$ performs better than its previous best value, $f(\vec{p}_i(t))$, then the personal best position of the i^{th} particle is updated as $\vec{x}_i(t+1)$. Otherwise, we keep the current personal best position:

$$\vec{p}_i(t+1) = \begin{cases} \vec{x}_i(t+1) & \text{if } f(\vec{p}_i(t)) \leq f(\vec{x}_i(t+1)) \\ \vec{p}_i(t) & \text{if } f(\vec{x}_i(t+1)) < f(\vec{p}_i(t)) \end{cases} \quad (1.17)$$

The global best position $\vec{g}(t)$ is updated based on the best fitness value that is found among the entire swarm:

$$\vec{g}(t+1) = \arg \max_{\vec{p}_i} \{f(\vec{p}_1(t+1)), f(\vec{p}_2(t+1)), \dots, f(\vec{p}_m(t+1))\}. \quad (1.18)$$

The maximum of the function f is achieved when all candidate solutions \vec{p}_i yield the same fitness value within a tolerance.

1.5 Thesis motivations, contributions and outline

In this dissertation, we extend the multi-level thresholding technique to RGB color images by using the GLLA histogram firstly. What's more, we know that people haven't come up with any multi-level entropic thresholding models by using the GLLV histogram [49] yet. This gives us another motivation to explore and propose several novel multi-level thresholding models by using the GLLV histogram based on the Shannon entropy, the Rényi entropy and the Tsallis-Havrda-Charvát entropy respectively.

The main contributions of this dissertation are as follows:

1. We derive the generalized multi-level thresholding criterion functions based on the Tsallis-Havrda-Charvát entropy by using the GLLA and the GLLV histograms respectively and validate these formulations rigorously by applying the mathematical induction method.
2. We propose a multi-level thresholding scheme for RGB color images, which is the first attempt that has been done so far according to our research. We assign the mean values from each thresholded class to obtain three segmented component images independently. Then, we obtain a segmented RGB color image, which is very close to the original image and has fewer color levels than the original image.
3. We formulate three different multi-level thresholding models based on the Shannon entropy, the Rényi entropy and the Tsallis-Havrda-Charvát entropy respectively by using GLLV histogram.

4. We employ the PSO algorithm to seek the optimal threshold values in a very reasonable computational time.
5. We compare the labeled segmented image with the benchmark images (ground truth, human segmentation) from BSDS300 to evaluate the proposed model quantitatively and objectively. We calculate the average four performance indices (PRI , GCE , VOI and BDE) of the models from the dissertation and compare them with the results from paper [48]. We use Matlab R2016b on a computer with 6 GB memory and Intel Core i7, 2.00 GHz processor during the experiments.

The outline of this dissertation is as follows:

In chapter one, we discuss the background knowledge about the GLLA histogram and the GLLV histogram, the Shannon entropy, the Rényi entropy, the Tsallis-Havrda-Charvát entropy and the PSO algorithm respectively.

In chapter two, we discuss the multi-level entropic thresholding models by using the 1D histogram and give some experimental results.

In chapter three, firstly, we discuss the multi-level entropic thresholding models by using the GLLA histogram. Then, we discuss the multi-level entropic thresholding models by using the GLLV histogram. The Shannon entropy, the Rényi entropy and the Tsallis-Havrda-Charvát entropy are used in these models respectively. We formulate the multi-level thresholding criterion functions based on the Tsallis-Havrda-Charvát entropy and validate the formulation by using mathematical induction.

In chapter four, we discuss the thresholding scheme for RGB color images and show the experimental results on several images.

In chapter five, we discuss the average four performance indices (PRI , GCE , VOI and BDE) and report the effectiveness of our models by applying the PSO

algorithm on BSDS300. We calculate the average four performance indices for the 300 images from BSDS300 extensively to show to comparison results among different models.

In chapter six, we give a conclusion of this dissertation by discussing the advantages and drawbacks of each model. We also show some future works of this dissertation by illustrating several different types of $2D$ histograms and discuss the probability of using these $2D$ histograms to construct novel multi-level thresholding models.

CHAPTER 2

ENTROPIC THRESHOLDING MODELS BASED ON ONE-DIMENSIONAL HISTOGRAM

In this chapter, we examine entropic thresholding models based on one-dimensional ($1D$) histogram.

From the introduction, we know that an entropy of a discrete source is defined on probability distribution of that source. A two dimensional intensity image was viewed as an information source [13]. Then the entropy of a gray image is defined on the its $1D$ histogram. A thresholding model based on an entropy is mainly about searching for optimal threshold values by maximizing the entropy of the image. We call the entropy function in the thresholding model as the criterion function for that model.

2.1 Thresholding using shannon entropy

In this section, we discuss the bi-level thresholding technique based on the Shannon entropy first, then we move to the general multi-level situation.

If we consider the normalized $1D$ histogram of an image f as the probability distribution $P = \{p_i\}_{i=0}^{i=255}$, then for a bi-level thresholding, the threshold value τ groups the $1D$ histogram into two disjointed distribution class (1) and class (2):

$$\begin{aligned} (1) : & \left(\frac{p(0)}{P_1}, \frac{p(1)}{P_1}, \dots, \frac{p(\tau)}{P_1} \right), \\ (2) : & \left(\frac{p(\tau+1)}{P_2}, \frac{p(\tau+2)}{P_2}, \dots, \frac{p(255)}{P_2} \right). \end{aligned}$$

where P_1 and P_2 are two independent *posteriori* class probabilities:

$$P_1 = \sum_{i=0}^{\tau} p(i), \quad P_2 = \sum_{i=\tau+1}^{255} p(i). \quad (2.1)$$

We point out that the distributions in class (1) and class (2) are normalized probability distributions, and each class has a total probability that equals 1.

Then the Shannon entropy for distributions in class (1) and class (2) are as follows respectively:

$$S_1(\tau) = - \sum_{i=0}^{\tau} \left(\frac{p(i)}{P_1} \right) \ln \left(\frac{p(i)}{P_1} \right), \quad (2.2)$$

$$S_2(\tau) = - \sum_{i=\tau+1}^{255} \left(\frac{p(i)}{P_2} \right) \ln \left(\frac{p(i)}{P_2} \right). \quad (2.3)$$

According to the extensive property of Shannon entropy, for the entire image, we obtain the following bi-level Shannon entropy with threshold values τ . We call this formation as the criterion function Φ_2 , which will be used as the objective function in the PSO algorithm.

$$\Phi_2(\tau) := S_{1+2}(\tau) = S_1(\tau) + S_2(\tau). \quad (2.4)$$

Then we obtain the optimal threshold τ^* by maximizing Φ_2 :

$$\tau^* = \underset{\tau \in G}{\operatorname{argmin}} \Phi_2(\tau),$$

Then a segmented image Seg_2 with two gray levels, $\{m_1, m_2\}$, which are the means of the gray levels from class (1) and class (2) respectively, is obtained from equation (1.6) by using τ^* .

Now we consider the $(n+1)$ -level thresholding model. We need n threshold values, $\{\tau_1, \dots, \tau_n\}$, to classify the 1D histogram of the image into $n+1$ disjointed

distribution classes:

$$\begin{aligned}
(1) : & \left(\frac{p(0)}{P_1}, \frac{p(1)}{P_1}, \dots, \frac{p(\tau_1)}{P_1} \right), \\
(2) : & \left(\frac{p(\tau_1 + 1)}{P_2}, \frac{p(\tau_1 + 2)}{P_2}, \dots, \frac{p(\tau_2)}{P_2} \right), \\
& \dots \\
(n+1) : & \left(\frac{p(\tau_n + 1)}{P_{n+1}}, \frac{p(\tau_n + 2)}{P_{n+1}}, \dots, \frac{p(255)}{P_{n+1}} \right).
\end{aligned}$$

where P_1, P_2, \dots, P_{n+1} are $n+1$ independent *posteriori* class probabilities as follows:

$$P_1 = \sum_{i=0}^{\tau} p(i), \quad P_2 = \sum_{i=\tau_1+1}^{\tau_2} p(i), \quad \dots, \quad P_{n+1} = \sum_{i=\tau_n+1}^{255} p(i). \quad (2.5)$$

Once again, these distribution classes are normalized probability distributions, and the total probability of each class equals 1.

Then for the $(n+1)$ -level thresholding model, we have $n+1$ Shannon entropies corresponding to $n+1$ disjointed probability distribution classes respectively:

$$S_k(\tau_{k-1}, \tau_k) = - \sum_{i=\tau_{k-1}+1}^{\tau_k} \left(\frac{p(i)}{P_k} \right) \ln \left(\frac{p(i)}{P_k} \right), \quad (2.6)$$

where $k = 1, 2, \dots, n+1$ with the convention of $\tau_0 = -1, \tau_{n+1} = 255$.

By applying the extensive property of Shannon entropy again, the entire image has the criterion function Φ_{n+1} with threshold values $\tau = \{\tau_1, \dots, \tau_n\}$:

$$\Phi_{n+1}(\tau) := S_{1+2+\dots+(n+1)} = \sum_{k=1}^{n+1} S_k. \quad (2.7)$$

Thus by maximizing Φ_{n+1} , the optimal threshold τ^* is as follows:

$$\tau^* = \underset{\tau \in G}{\operatorname{argmin}} \Phi_{n+1}(\tau),$$

where $\tau^* = \{\tau_1^*, \dots, \tau_n^*\}$. Then we use equation (1.7) and τ^* to threshold the original image f to obtain the segmented image Seg_{n+1} .

2.2 Thresholding using Rényi entropy

In this section, we discuss the bi-level and the multi-level thresholding model based on the Rényi entropy.

For a bi-level thresholding, according to the extensive property of Rényi entropy, we have a criterion function Φ_2 as follows:

$$\Phi_2(\tau) := R_{1+2}(\tau) = R_1(\tau) + R_2(\tau) , \quad (2.8)$$

where $R_1(\tau)$ and $R_2(\tau)$ are Rényi entropies corresponding to distributions in class (1) and class (2) that are mentioned in section 2.1:

$$R_1(\tau) = \frac{1}{1-\alpha} \cdot \ln \sum_{i=0}^{\tau} \left(\frac{p(i)}{P_1} \right)^{\alpha} \quad \text{and} \quad R_2(\tau) = \frac{1}{1-\alpha} \cdot \ln \sum_{i=\tau+1}^{255} \left(\frac{p(i)}{P_2} \right)^{\alpha} , \quad (2.9)$$

with P_1 and P_2 which are two independent *posteriori* class probabilities from the equation (2.1).

The criterion function Φ_{n+1} for the $(n+1)$ -level thresholding model based on the Rényi entropy with threshold values $\tau = \{\tau_1, \dots, \tau_n\}$ is given by:

$$\Phi_{n+1}(\tau) := R_{1+2+\dots+(n+1)} = \sum_{k=1}^{n+1} R_k(\tau) , \quad (2.10)$$

where R_k is the Rényi entropy of the k^{th} disjointed probability distribution class:

$$R_k(\tau_{k-1}, \tau_k) = \frac{1}{1-\alpha} \cdot \ln \sum_{i=\tau_{k-1}+1}^{\tau_k} \left(\frac{p(i)}{P_k} \right)^{\alpha} , \quad (2.11)$$

for all $k = 1, 2, \dots, n+1$ with the convention of $\tau_0 = -1$, $\tau_{n+1} = 255$, and P_k is the k^{th} independent *posteriori* class probability, which is mentioned in the equation (2.5).

Thus the optimal threshold τ^* by maximizing Φ_{n+1} is given by:

$$\tau^* = \underset{\tau \in G}{\operatorname{argmin}} \Phi_{n+1}(\tau) ,$$

where $\tau^* = \{\tau_1^*, \dots, \tau_n^*\}$. Then we use equation (1.7) and τ^* to threshold the original image f and obtain the segmented image Seg_{n+1} .

2.3 Thresholding using Tsallis-Havrda-Charvát entropy

We point out that even though a criterion function for the multi-level thresholding by using the GLLA histogram and the Tsallis-Havrda-Charvát entropy was given in [18, 19], the detail on how to derive this function was not discussed. We believe that our work is the first attempt so far to illustrate the correctness of the criterion function.

This section is mainly about the bi-level and the multi-level thresholding models based on the Tsallis-Havrda-Charvát entropy. We use the non-additivity rule and mathematical induction to demonstrate how to derive the general criterion function Φ_{n+1} for a $(n+1)$ -level thresholding.

According to the non-extensive property of Tsallis-Havrda-Charvát entropy, the criterion function Φ_2 for the bi-level thresholding model with threshold values τ is given by:

$$\Phi_2(\tau) := T_{1+2} = T_1(\tau) + T_2(\tau) + (1 - \alpha) \cdot T_1(\tau) \cdot T_2(\tau) , \quad (2.12)$$

where T_1 and T_2 are Tsallis-Havrda-Charvát entropies corresponding to distributions in class (1) and class (2), which are mentioned in section 2.1:

$$T_1(\tau) = \frac{1}{\alpha - 1} \cdot \left[1 - \sum_{i=0}^{\tau} \left(\frac{p(i)}{P_1} \right)^{\alpha} \right] , \quad T_2(\tau) = \frac{1}{\alpha - 1} \cdot \left[1 - \sum_{i=\tau+1}^{255} \left(\frac{p(i)}{P_2} \right)^{\alpha} \right] ,$$

where P_1 and P_2 are defined in equation (2.1). Next we induce the criterion function for the three-level thresholding model with threshold values τ_1 and τ_2 :

$$\begin{aligned} \Phi_3 &:= T_{1+2+3} \\ &= T_{(1+2)+3} \\ &= T_{1+2} + T_3 + (1 - \alpha) \cdot T_{1+2} \cdot T_3 \\ &= T_1 + T_2 + (1 - \alpha) \cdot T_1 \cdot T_2 + T_3 + (1 - \alpha) \cdot [T_1 + T_2 + (1 - \alpha) \cdot T_1 \cdot T_2] \cdot T_3 \\ &= T_1 + T_2 + T_3 + (1 - \alpha) \cdot [T_1 T_2 + T_1 T_3 + T_2 T_3(t, s)] + (1 - \alpha)^2 \cdot T_1 \cdot T_2 \cdot T_3 , \end{aligned} \quad (2.13)$$

where T_1 , T_2 and T_3 are Tsallis-Havrda-Charvát entropies corresponding to three disjointed distribution classes respectively:

$$\begin{aligned} T_1(\tau_1) &= \frac{1}{\alpha - 1} \cdot \left[1 - \sum_{i=0}^{\tau_1} \left(\frac{p(i)}{P_1} \right)^\alpha \right], \\ T_2(\tau_1, \tau_2) &= \frac{1}{\alpha - 1} \cdot \left[1 - \sum_{i=\tau_1+1}^{\tau_2} \left(\frac{p(i)}{P_2} \right)^\alpha \right], \\ T_3(\tau_2) &= \frac{1}{\alpha - 1} \cdot \left[1 - \sum_{i=\tau_2+1}^{255} \left(\frac{p(i)}{P_3} \right)^\alpha \right], \end{aligned}$$

with

$$P_1(\tau_1) = \sum_{i=0}^{\tau_1} p(i), \quad P_2(\tau_1, \tau_2) = \sum_{i=\tau_1+1}^{\tau_2} p(i), \quad P_3(\tau_2) = \sum_{i=\tau_2+1}^{255} p(i).$$

In the same manner, for the $(n+1)$ -level thresholding model, we conclude that with threshold values $\tau = \{\tau_1, \tau_2, \dots, \tau_n\}$, the criterion function Φ_{n+1} for the $(n+1)$ -level thresholding method based on the Tsallis-Havrda-Charvát entropy is:

$$\begin{aligned} \Phi_{n+1} &:= T_{1+2+\dots+(n+1)} \\ &= \sum_{i=1}^{n+1} T_i + (1 - \alpha) \cdot \sum_{i=1}^n \sum_{j=i+1}^{n+1} T_i T_j \\ &\quad + (1 - \alpha)^2 \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=j+1}^{n+1} T_i T_j T_k \\ &\quad + (1 - \alpha)^3 \cdot \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n \sum_{l=k+1}^{n+1} T_i T_j T_k T_l \\ &\quad + \dots + (1 - \alpha)^n \cdot \prod_{i=1}^{n+1} T_i, \end{aligned} \tag{2.14}$$

where T_k is the Tsallis-Havrda-Charvát entropy corresponding to the k^{th} disjointed probability distribution class:

$$T_k(\tau) = \frac{1}{\alpha - 1} \cdot \left[1 - \sum_{i=\tau_{k-1}+1}^{\tau_k} \left(\frac{p(i)}{P_k} \right)^\alpha \right] \tag{2.15}$$

for all $k = 1, 2, \dots, n+1$ with the convention of $\tau_0 = -1$, $\tau_{n+1} = 255$ and P_k is the k^{th} independent *posteriori* class probability, which is mentioned in the equation

(2.5). Thus the optimal threshold τ^* by maximizing Φ_{n+1} is as follows:

$$\tau^* = \operatorname{argmin}_{\tau \in G} \Phi_{n+1}(\tau) ,$$

with $\tau^* = \{\tau_1^*, \dots, \tau_n^*\}$.

Now we prove the generalized formulation (2.14) by applying mathematical induction. Using the non-additivity rule, we have

$$\begin{aligned} \Phi_{n+1} &:= T_{1+2+\dots+n+(n+1)} \\ &= T_{(1+2+\dots+n)+(n+1)} \\ &= T_{1+2+\dots+n} + T_{n+1} + (1 - \alpha) \cdot T_{1+2+\dots+n} \cdot T_{n+1} . \end{aligned} \tag{2.16}$$

From the induction assumption, we have

$$\begin{aligned} \Phi_n &:= T_{1+2+\dots+n} \\ &= T_1 + T_2 + \dots + T_n \\ &+ (1 - \alpha) \cdot \sum_{i=1}^{n-1} \sum_{\substack{j>i \\ j=i+1}}^n T_i T_j \\ &+ (1 - \alpha)^2 \cdot \sum_{i=1}^{n-2} \sum_{\substack{j>i \\ j=i+1}}^{n-1} \sum_{\substack{k>j \\ k=j+1}}^n T_i T_j T_k \\ &+ \dots \\ &+ (1 - \alpha)^{n-1} \cdot T_1 \cdot T_2 \cdot \dots \cdot T_n \\ &= \sum_{s=1}^n (1 - \alpha)^{s-1} \cdot \sum_{i_1=1}^{n-s+1} \sum_{\substack{i_2>i_1 \\ i_2=i_1+1}}^{n-s+2} \dots \sum_{\substack{i_s>i_{s-1} \\ i_s=i_{s-1}+1}}^{n-s+s} T_{i_1} T_{i_2} \dots T_{i_s} . \end{aligned} \tag{2.17}$$

By plugging (2.17) in (2.16), we have

$$\begin{aligned}
\Phi_{n+1} &:= \left[\sum_{i=1}^n T_i + (1-\alpha) \cdot \sum_{i=1}^{n-1} \sum_{\substack{j>i \\ j=i+1}}^n T_i T_j + \cdots + (1-\alpha)^{n-1} \cdot \prod_{i=1}^n T_i \right] + T_{n+1} \\
&+ (1-\alpha) \cdot \left[\sum_{i=1}^n T_i + (1-\alpha) \cdot \sum_{i=1}^{n-1} \sum_{\substack{j>i \\ j=i+1}}^n T_i T_j + \cdots + (1-\alpha)^{n-1} \cdot \prod_{i=1}^n T_i \right] \cdot T_{n+1} \\
&= \sum_{i=1}^{n+1} T_i + (1-\alpha) \cdot \sum_{i=1}^n \sum_{\substack{j>i \\ j=i+1}}^{n+1} T_i T_j + \cdots + (1-\alpha)^n \cdot \prod_{i=1}^{n+1} T_i .
\end{aligned} \tag{2.18}$$

Thus we proved the general equation (2.14).

2.4 Experiments

In this section, we conduct our experiments on the image 113016.jpg mentioned in Fig. 1.1 (a).

For a $(n+1)$ -level thresholding, the position of the i^{th} particle in the PSO algorithm is a n -dimensional threshold vector of the original image $f(x, y)$:

$$\vec{x}_i(t) = (\tau_i) = (\tau_{i1}, \tau_{i2}, \cdots, \tau_{in}),$$

where $i = 1, \dots, m$, and m is the total number of particles in the n -dimensional searching space. The main parts of the PSO algorithm are given by the following steps:

Step 1: Initialize $\vec{x}_i(t)$ and use $\vec{x}_i(t)$ to initialize $\vec{p}_i(t)$

Step 2: Evaluate the criterion function Φ_{n+1} (according to which model we want to use) at $\vec{x}_i(t)$

Step 3: Update $\vec{g}(t)$ by equation (1.18)

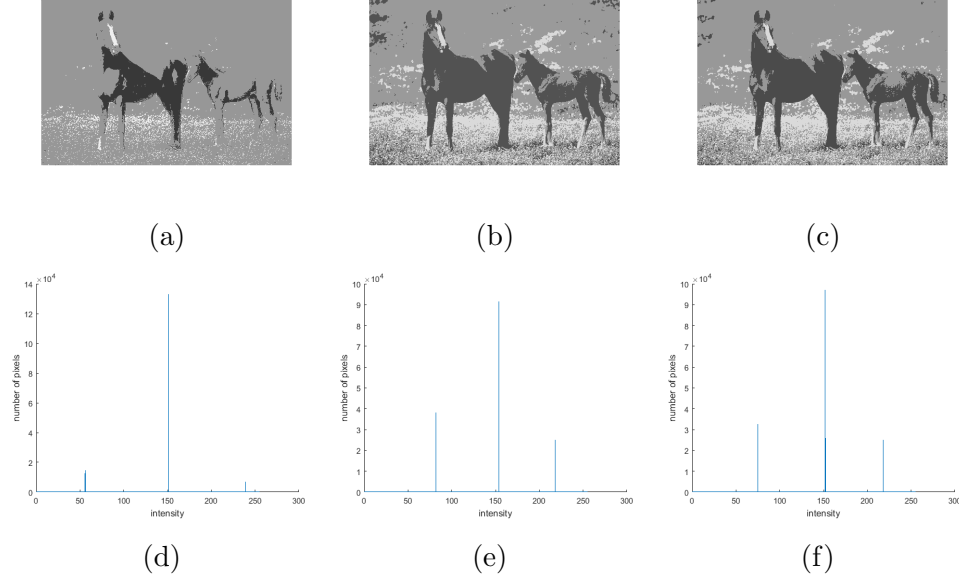


Figure 2.1 – Three-level segmentations with the corresponding 1D histograms

Step 4: Evaluate velocity $\vec{v}_i(t)$ by equation (1.15)

Step 5: Update $\vec{x}_i(t)$ by equation (1.16)

Step 6: Update $\vec{p}_i(t)$ and $\vec{g}(t)$ by (1.17) and (1.18) respectively

Step 7: Return to step 2 .

We assign $c_1 = 0.7, c_2 = 1.43$ and swarm size $m = 20$ for each experiment [11]. Experimental results in Figure 2.1 are from three-level thresholding models based on the Shannon entropy (a) (section 2.1), the Rényi entropy (b) (section 2.2) and the Tsallis-Havrda-Charvát entropy (c) (section 2.3) respectively. Their corresponding 1D histograms are shown in (d), (e) and (f).

Figure 2.2 demonstrates the segmentation results from ten-level thresholding models in which (a) is from the Shannon entropic model, (b) is from the Rényi entropic model and (c) is from the Tsallis-Havrda-Charvát entropic model. Their corresponding 1D histograms are shown in (d), (e) and (f).

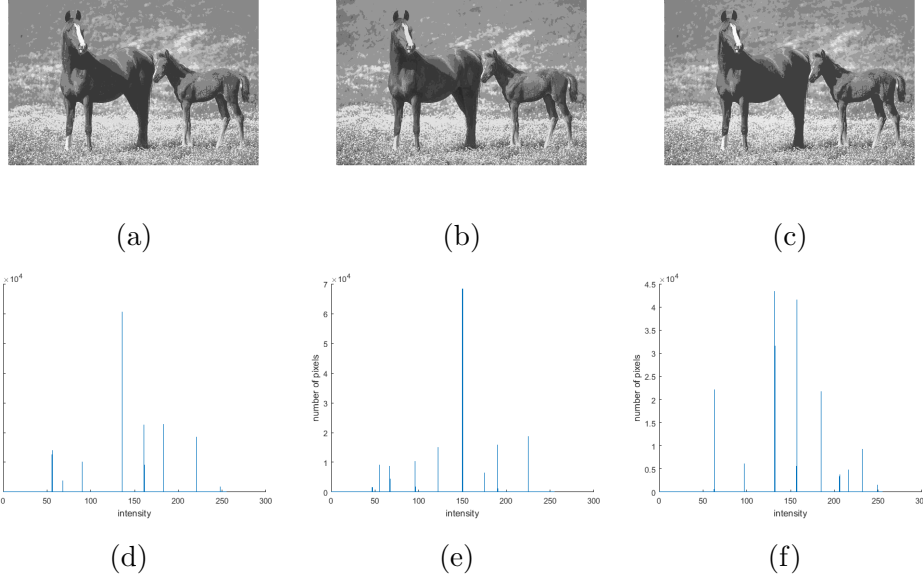


Figure 2.2 – Ten-level segmentations with the corresponding 1D histograms

Table 2.1 – Mean values from each thresholded class.

Models	Shannon	Rényi	Tsallis-Havrda-Charvát
3-level	56, 151, 239	82, 154, 218	75, 152, 218
10-level	28, 33, 56, 68, 90, 136, 161, 183, 221, 248	39, 47, 55, 67, 96, 122, 150, 175, 190, 225	28, 63, 97, 132, 157, 185, 206, 216, 232, 249

We notice that the Rényi based model and the Tsallis-Havrda-Charvát based model not only give similar results (see Figure 2.1 (b) and (c)) but also perform better than the Shannon based model (see Figure 2.1 (a)). This is because the Rényi and the Tsallis-Havrda-Charvát entropies become the Shannon entropy when α is close to 1 (see (1.10) and (1.14)). It implies that the Shannon entropy actually is a special case of Rényi and Tsallis-Havrda-Charvát entropies.

What's more, for an entropic thresholding model, the larger level we use to threshold an image, the more accurate result we obtain from the model (compare Figure 2.1 with Figure 2.2). Ten-level thresholding results are closer to the original image (see Figure 1.1 (a)) than three-level thresholding results.

Table 2.2 – Thresholds values for different entropic models.

Models	Shannon	Rényi	Tsallis-Havrda-Charvát
3-level	64, 230	123, 192	114, 192
10-level	29, 33, 63, 73, 104, 153, 169, 200, 242	41, 49, 59, 79, 109, 130, 171, 178, 203	31, 88, 105, 145, 171, 202, 210, 221, 243

Table 2.1 lists all the mean values that we used to exhibit images in Figure 2.1 and Figure 2.2. Table 2.2 lists all the threshold values that we obtained from three different entropic models. We point out that for 3-level thresholding models, we need two threshold values, and for 10-level thresholding models, we need nine threshold values.

CHAPTER 3

ENTROPIC THRESHOLDING MODELS BASED ON TWO-DIMENSIONAL HISTOGRAMS

From chapter one, we know that only the gray level information is used to build the one-dimensional ($1D$) histogram and the spatial correlation among pixels in a neighborhood is not considered. In this chapter, we are going to discuss the entropic image thresholding techniques by using two-dimensional ($2D$) histograms GLLA and GLLV, respectively, to illustrate that $2D$ histograms perform better than the $1D$ histogram.

People have been constructing several multi-level thresholding models by using the GLLA histogram [18, 19, 30, 31, 48]. However, very few multi-level thresholding models involve the GLLV histogram. So in the first section of this chapter, we discuss a variety of thresholding models by using the GLLA histogram, then in the next section, we develop three novel multi-level thresholding models by applying the GLLA. At the end of each section, we demonstrate the effectiveness of models by some experiments.

3.1 Entropic thresholding model based on Gray-Level & Local-Average histogram

Firstly, we illustrate thresholding models based on the Gray-Level & Local-Average histogram (GLLA). In the introduction chapter, we know that the GLLA is constructed from the gray level of each pixel and the average gray level value of its neighborhood.

Now we consider a GLLA histogram plane from Figure 1.3 (c) as the prob-

ability distribution plane in the following models. We claim that for a $(n + 1)$ -level thresholding, the threshold value is actually the threshold pair (τ, σ) , such that $\tau = (\tau_1, \tau_2, \dots, \tau_n)$ is the threshold from the original image $f(x, y)$ and $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$ is the threshold from the average image $g(x, y)$. We also point out that the optimal value τ^* that is obtained from the original image is the one we use to segment the image at the last step.

For the bi-level thresholding with $n = 1$, we assume that GLLA histogram surface $p(t, s)$ has two peaks and one valley (see Figure 1.3 (d)), then the threshold vector (τ, σ) divides the GLLA histogram plane into four quadrants (see Figure 3.1 (a)). In this case, the third and fourth quadrants contain information about edges and noise, so they are ignored in the calculation. The object and the background information are from the main diagonal quadrants or vice versa. We consider the first quadrant $([0, \tau] \times [0, \sigma])$ and the second quadrant $([\tau + 1, 255] \times [\sigma + 1, 255])$ as two different independent probability distributions, then we have two independent *posteriori* class probabilities P_1 and P_2 as follows:

$$P_1 = \sum_{i=0}^{\tau} \sum_{j=0}^{\sigma} p(i, j) , \quad P_2 = \sum_{i=\tau+1}^{255} \sum_{j=\sigma+1}^{255} p(i, j) .$$

What's more, a probability distribution must be normalized such that the total probability equals 1 in each quadrant. Thus two normalized probability distributions associated with first and second quadrants are:

$$\begin{aligned} (1) : & \left(\frac{p(0, 0)}{P_1}, \dots, \frac{p(0, \sigma)}{P_1}, \frac{p_c(1, 0)}{P_1}, \dots, \frac{p(\tau, \sigma)}{P_1} \right) , \\ (2) : & \left(\frac{p(\tau + 1, \sigma + 1)}{P_2}, \dots, \frac{p(\tau + 1, 255)}{P_2}, \frac{p(\tau + 2, \sigma + 1)}{P_2}, \dots, \frac{p(255, 255)}{P_2} \right) . \end{aligned}$$

Now we discuss the three-level thresholding with $n = 2$. Threshold value (τ, σ) with $\tau = (\tau_1, \tau_2)$ and $\sigma = (\sigma_1, \sigma_2)$ segments the 2D histogram plane into 9 divisions (see Figure 3.1 (b)). We use the object and the background information from the first, second and third quadrants to construct the model, meanwhile ignoring the rest of quadrants with edges and noise information. Three normalized

disjointed probability distributions associated with first, second and third quadrants are:

$$\begin{aligned}
(1) : & \left(\frac{p(0,0)}{P_1}, \dots, \frac{p(0,\sigma_1)}{P_1}, \frac{p(1,0)}{P_1}, \dots, \frac{p(\tau_1,\sigma_1)}{P_1} \right), \\
(2) : & \left(\frac{p(\tau_1+1,\sigma_1+1)}{P_2}, \dots, \frac{p(\tau_1+1,\sigma_2)}{P_2}, \frac{p(\tau_1+2,\sigma_1+1)}{P_2}, \dots, \frac{p(\tau_2,\sigma_2)}{P_2} \right), \\
(3) : & \left(\frac{p(\tau_2+1,\sigma_2+1)}{P_3}, \dots, \frac{p(\tau_2+1,255)}{P_3}, \frac{p(\tau_2+2,\sigma_2+1)}{P_3}, \dots, \frac{p(255,255)}{P_3} \right),
\end{aligned}$$

where P_1 , P_2 and P_3 are three independent *posteriori* class probabilities as follows:

$$P_1 = \sum_{i=0}^{\tau_1} \sum_{j=0}^{\sigma_1} p(i,j), \quad P_2 = \sum_{i=\tau_1+1}^{\tau_2} \sum_{j=\sigma_1+1}^{\sigma_2} p(i,j), \quad P_3 = \sum_{i=\tau_2+1}^{255} \sum_{j=\sigma_2+1}^{255} p(i,j).$$

In the same manner, for the $(n+1)$ -level thresholding model, we conclude that the generalized $(n+1)$ -level thresholding model has the threshold values $\tau = (\tau_1, \tau_2, \dots, \tau_n)$ and $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$. Therefore, $(n+1)$ normalized disjointed probability distributions associated with $n+1$ quadrants from the main diagonal are as follows:

$$\left(\frac{p(\tau_{k-1}+1, \sigma_{k-1}+1)}{P_k}, \dots, \frac{p(\tau_{k-1}+1, \sigma_k)}{P_k}, \frac{p(\tau_{k-1}+2, \sigma_{k-1}+1)}{P_k}, \dots, \frac{p(\tau_k, \sigma_k)}{P_k} \right),$$

where P_k is the k^{th} independent *posteriori* class probability:

$$P_k = \sum_{i=\tau_{k-1}+1}^{\tau_k} \sum_{j=\sigma_{k-1}+1}^{\sigma_k} p(i,j), \quad (3.1)$$

for all $k = 1, 2, \dots, n+1$ with the convention of $\tau_0 = -1$, $\sigma_0 = -1$, $\tau_{n+1} = 255$, and $\sigma_{n+1} = 255$.

3.1.1 Thresholding using Shannon entropy

From the discussion of Shannon entropy in section (2.1), we know that criterion function Φ_{n+1} for the $(n+1)$ -level thresholding model based on Shannon entropy is given by equation (2.7):

$$\Phi_{n+1} := S_{1+2+\dots+(n+1)} = \sum_{k=1}^{n+1} S_k,$$

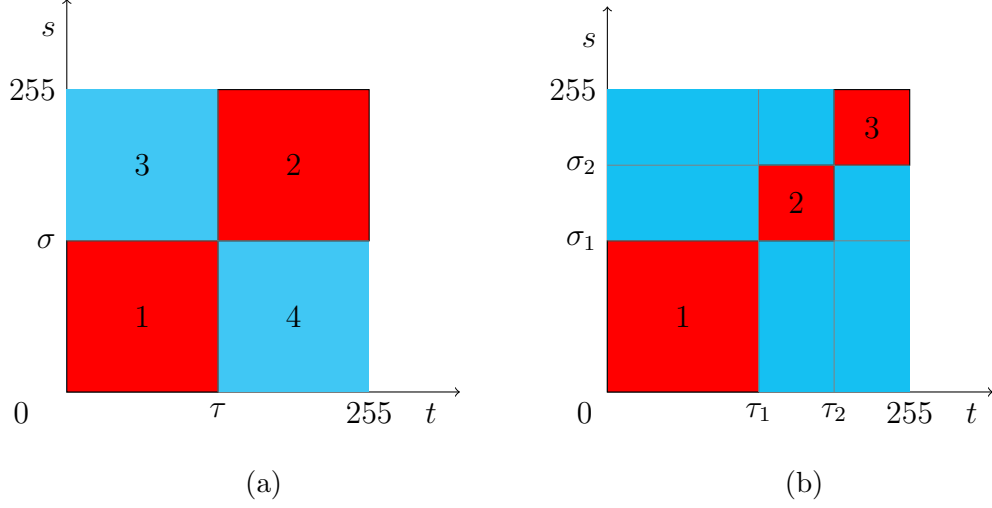


Figure 3.1 – GLLA histogram planes with bi-level and three-level segmentation

where S_k is the Shannon entropy corresponding to the k^{th} disjointed probability distribution class:

$$S_k = - \sum_{i=\tau_{k-1}+1}^{\tau_k} \sum_{j=\sigma_{k-1}+1}^{\sigma_k} \left(\frac{p(i,j)}{P_k} \right) \ln \left(\frac{p(i,j)}{P_k} \right)$$

and P_k , $k = 1, 2, \dots, n+1$ are given in the equation (3.1) with the convention of $\tau_0 = -1$, $\sigma_0 = -1$, $\tau_{n+1} = 255$, and $\sigma_{n+1} = 255$.

The optimal threshold pair (τ^*, σ^*) is obtained from:

$$(\tau^*, \sigma^*) = \underset{(\tau, \sigma) \in G^n \times G^n}{\operatorname{argmin}} \Phi_{n+1}(\tau, \sigma),$$

where $\tau^* = (\tau_1^*, \tau_2^*, \dots, \tau_n^*)$ and $\sigma = (\sigma_1^*, \sigma_2^*, \dots, \sigma_n^*)$. However, we only use $\tau^* = (\tau_1^*, \tau_2^*, \dots, \tau_n^*)$ to segment the image. Then the segmented image Seg_{n+1} is given by:

$$Seg_{n+1} = \begin{cases} m_1, & \text{if } 0 \leq f(x, y) \leq \tau_1^* \\ m_2, & \text{if } (\tau_1^* + 1) \leq f(x, y) \leq \tau_2^* \\ \dots & \\ m_{n+1}, & \text{if } (\tau_n^* + 1) \leq f(x, y) \leq 255, \end{cases} \quad (3.2)$$

where $\{m_1, \dots, m_{n+1}\}$ are the mean values from each corresponding class respectively (see section 1.2.2).

3.1.2 Thresholding using Rényi entropy

From the discussion of the Rényi entropy in section (2.2), we know that the criterion function Φ_{n+1} for the $(n+1)$ -level thresholding method based on the Rényi entropy is given by equation (2.10):

$$\Phi_{n+1} := R_{1+2+\dots+(n+1)} = \sum_{k=1}^{n+1} R_k ,$$

where R_k is the Rényi entropy corresponding to the k^{th} disjointed probability distribution class:

$$R_k = \frac{1}{1-\alpha} \left[\ln \sum_{i=\tau_{k-1}+1}^{\tau_k} \sum_{j=\sigma_{k-1}+1}^{\sigma_k} \left(\frac{p(i,j)}{P_k} \right)^\alpha \right]$$

and P_k , $k = 1, 2, \dots, n+1$ are given in the equation (3.1) with the convention of $\tau_0 = -1$, $\sigma_0 = -1$, $\tau_{n+1} = 255$, and $\sigma_{n+1} = 255$.

The optimal threshold pair (τ^*, σ^*) is obtained from:

$$(\tau^*, \sigma^*) = \underset{(\tau, \sigma) \in G^n \times G^n}{\operatorname{argmin}} \Phi_{n+1}(\tau, \sigma) ,$$

where $\tau^* = (\tau_1^*, \tau_2^*, \dots, \tau_n^*)$ and $\sigma^* = (\sigma_1^*, \sigma_2^*, \dots, \sigma_n^*)$. The segmented image is also obtained from equation (3.2).

3.1.3 Thresholding using Tsallis-Havrda-Charvát entropy

From the discussion of the Tsallis-Havrda-Charvát entropy in section (2.3), we know that the criterion function Φ_{n+1} for the $(n+1)$ -level thresholding method

based on the Tsallis-Havrda-Charvát entropy is given by equation (2.14):

$$\begin{aligned}
\Phi_{n+1} &:= T_{1+2\cdots+(n+1)} \\
&= \sum_{i=1}^{n+1} T_i + (1-\alpha) \cdot \sum_{i=1}^n \sum_{j=i+1}^{n+1} T_i T_j \\
&\quad + (1-\alpha)^2 \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=j+1}^{n+1} T_i T_j T_k \\
&\quad + (1-\alpha)^3 \cdot \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n \sum_{l=k+1}^{n+1} T_i T_j T_k T_l \\
&\quad + \cdots + (1-\alpha)^n \cdot \prod_{i=1}^{n+1} T_i,
\end{aligned}$$

where T_k is the Tsallis-Havrda-Charvát entropy corresponding to the k^{th} disjointed probability distribution class:

$$T_k = \frac{1}{\alpha - 1} \left[1 - \sum_{i=\tau_{k-1}+1}^{\tau_k} \sum_{j=\sigma_{k-1}+1}^{\sigma_k} \left(\frac{p(i,j)}{P_k} \right)^\alpha \right] \quad (3.3)$$

and P_k , $k = 1, 2, \dots, n+1$ are given in the equation (3.1) with the convention of $\tau_0 = -1$, $\sigma_0 = -1$, $\tau_{n+1} = 255$, and $\sigma_{n+1} = 255$.

The optimal threshold pair (τ^*, σ^*) is obtained from:

$$(\tau^*, \sigma^*) = \underset{(\tau, \sigma) \in G^n \times G^n}{\operatorname{argmin}} \Phi_{n+1}(\tau, \sigma),$$

where $\tau^* = (\tau_1^*, \tau_2^*, \dots, \tau_n^*)$ and $\sigma = (\sigma_1^*, \sigma_2^*, \dots, \sigma_n^*)$. The segmented image is also obtained from equation (3.2).

We claim that discussion on this model and its corresponding experiments is presented in our paper [3].

3.1.4 Experiments

The main part of the PSO algorithm is the same process as the one mentioned in section (2.4). However, for a $(n+1)$ -level thresholding based on the GLLA histogram, the i^{th} particle from the PSO algorithm is a $2n$ -dimensional vector instead

Table 3.1 – Mean values from each thresholded class.

Models	Shannon	Rényi	Tsallis-Havrda-Charvát
3-level	88, 158, 224	66, 147, 214	59, 134, 194
10-level	61, 97, 116, 150, 182, 200, 221, 234, 254	42, 56, 64, 87, 119, 147, 180, 214, 237, 253	35, 57, 94, 136, 160, 177, 186, 211, 232, 243

of n -dimensional vector:

$$\vec{x}_i(t) = (\tau_i, \sigma_i) = (\tau_{i1}, \tau_{i2}, \dots, \tau_{in}, \sigma_{i1}, \sigma_{i2}, \dots, \sigma_{in}),$$

where $i = 1, \dots, m$, and m is the total number of particles. The searching space is a $2n$ -dimensional space. We notice that $\tau_i = (\tau_{i1}, \tau_{i2}, \dots, \tau_{in})$ represents the threshold for the original image $f(x, y)$ and $\sigma_i = (\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{in})$ represents the threshold for the average image $g(x, y)$. According to section (1.4), the main parts of the PSO algorithm are as follows:

Step 1. Initialize $\vec{x}_i(t)$ and use $\vec{x}_i(t)$ to initialize $\vec{p}_i(t)$

Step 2. Evaluate the criterion function Φ_{n+1} (according to which model we want to use) at $\vec{x}_i(t)$

Step 3. Update $\vec{g}(t)$ by function (1.18)

Step 4. Evaluate velocity $\vec{v}_i(t)$ by function (1.15)

Step 5. Update $\vec{x}_i(t)$ by function (1.16)

Step 6. Update $\vec{p}_i(t)$ and $\vec{g}(t)$ by (1.17) and (1.18) respectively

Step 7. Return to step 2 .

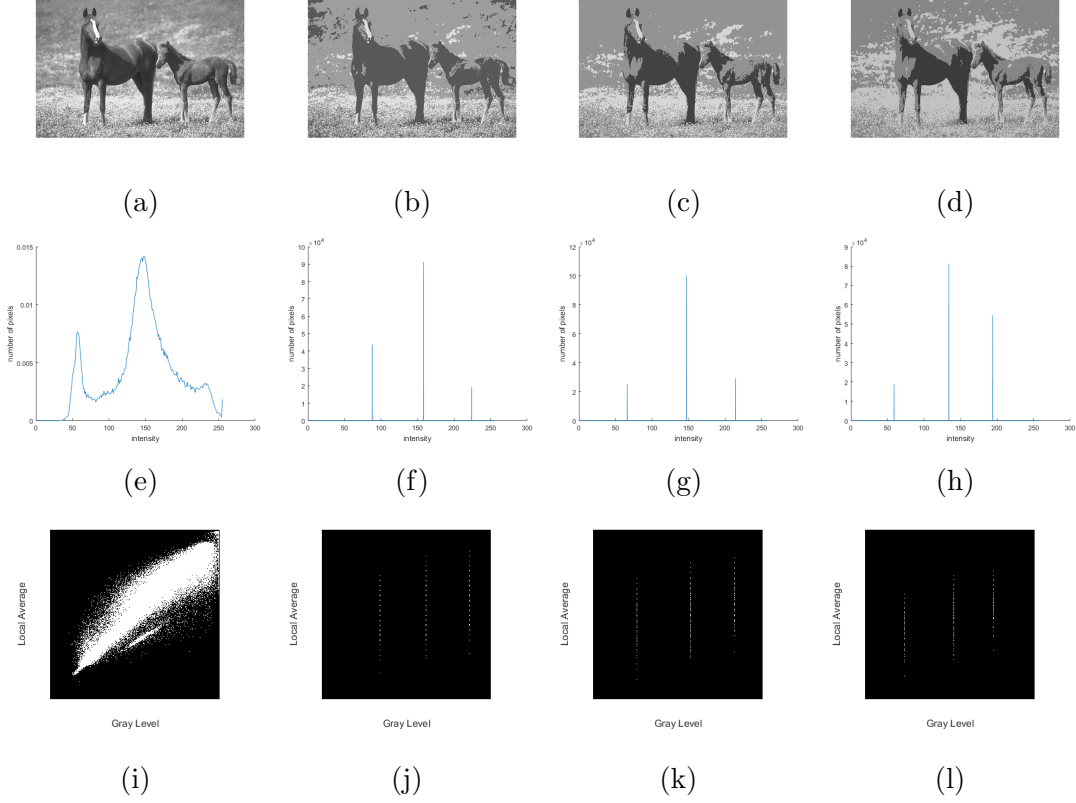


Figure 3.2 – Original image, the three-level thresholded images and their corresponding 1D and GLLA histograms

Figure 3.2 illustrates the experiment results from three-level models. (a) is the original image 113016.jpg, (b) is the thresholded image from the Shannon entropic model, (c) is the thresholded image from the Rényi entropic model and (d) is the thresholded image from the Tsallis-Havrda-Charvát entropic model. Images (e)-(h) in Figure 3.2 are the corresponding 1D histograms, and images (i)-(l) in Figure 3.2 are the corresponding GLLA histograms. Since all the thresholded images have only three gray levels, their GLLA histograms also have only three stripes.

Figure 3.3 illustrates the experiment results from ten-level models. (a) is the original image 113016.jpg, (b) is the thresholded image from the Shannon entropic model, (c) is the thresholded image from the Rényi entropic model and (d) is the thresholded image from the Tsallis-Havrda-Charvát entropic model. Images (e)-(h) in Figure 3.3 are the corresponding 1D histograms, and images (i)-(l) in Fig-

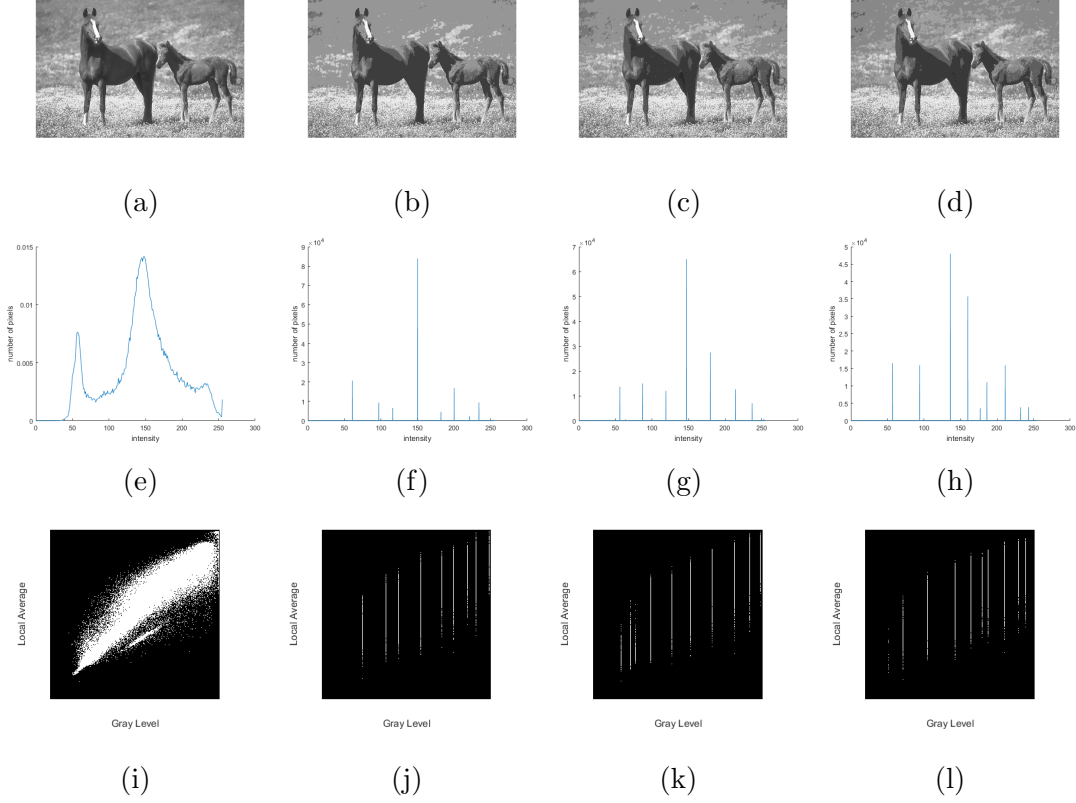


Figure 3.3–Original image, the ten-level thresholded images and their corresponding 1D and GLLA histograms

Figure 3.3 are the corresponding GLLA histograms. We can notice that their GLLA histograms have ten stripes because the level we used in the experiment is 10.

Table 3.1 lists all the mean values that we used to exhibit the images in Figure 3.2 and Figure 3.3. Table 3.2 lists all the threshold values that we obtained from three different entropic models. We point out that for 3-level thresholding models, we need two threshold values, and for 10-level thresholding models, we need nine threshold values.

3.2 Entropic thresholding model based on Gray-Level & Local-Variance histogram

In this section, we will discuss the thresholding method based on the Gray-Level & Local-Variance histogram (GLLV). It is a new type of two-dimensional ($2D$)

Table 3.2 – Thresholds values for different entropic models.

Models	Shannon	Rényi	Tsallis-Havrda-Charvát
3-level	129, 202	96, 185	76, 160
10-level	13, 83, 109, 121, 179,	45, 63, 64, 108, 127,	37, 69, 114, 149, 174,
	185, 217, 223, 251	165, 200, 228, 249	178, 194, 228, 236

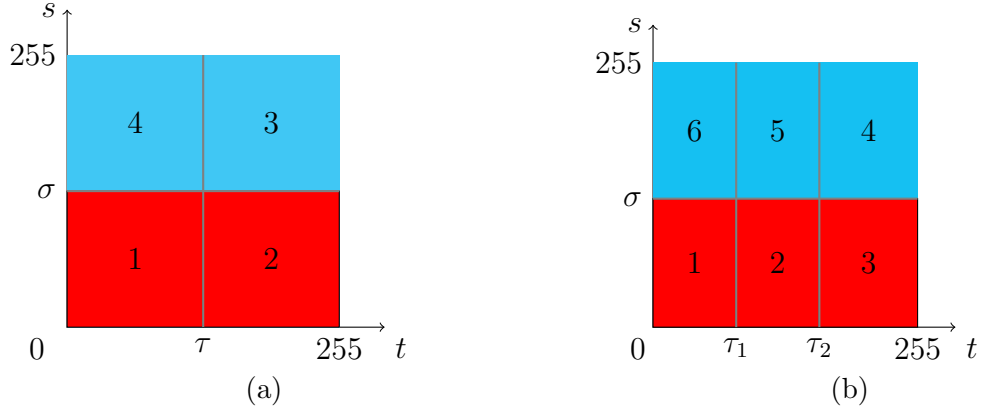


Figure 3.4 – GLLV histogram planes with bi-level and 3-level segmentation

histogram which was developed in paper [49]. The GLLV histogram is constructed by applying the dispersion of gray level distribution of pixels instead of the average gray level of a neighborhood which is used in the traditional 2D histogram, GLLA. In this section, we build three novel models under the framework of the GLLV histogram.

From the demonstration of a GLLV histogram (see Figure 1.4 (b) with upper boundary of $\sigma = 117$ and Figure 1.4 (c) with upper boundary of $\sigma = 64$), we notice that the object and the background information are mainly focused at the lower part on the GLLV histogram plane with small local variance values within $[0, \sigma]$ because object and background are homogenous regions with pixels whose gray levels are close to the average gray level from its 3×3 neighborhood. Thus we conclude that the object and the background pixels from the image have smaller

local variance values within $[0, \sigma]$ according to the definition (1.2) of local variance g .

For the bi-level thresholding with $n = 1$, we consider the threshold vector (τ, σ) . The value τ is from the original gray-level image $f(x, y)$, while the value σ is from the local variance image $g(x, y)$. The GLLV histogram plane, that is the joint probability mass function $p(t, s)$, gives us two peaks and one valley (see Figure 1.4 (d)). Then the GLLV histogram plane is divided into four quadrants (see Figure 3.4 (a)). Since the third and fourth quadrants contain information about edges and noise, they are ignored in the calculation. The object and the background information is from the first and second quadrants with small local variance values within $[0, \sigma]$. Then we have two independent probability distributions, denoted by first quadrant $[0, \tau] \times [0, \sigma]$ and second quadrant $[\tau + 1, 255] \times [0, \sigma]$. Also, they must be normalized such that the total probability in each the quadrant equals 1, thus two probability distributions associated with first and second quadrants are as follows:

$$\begin{aligned} (1) : & \left(\frac{p(0, 0)}{P_1}, \dots, \frac{p(0, \sigma)}{P_1}, \frac{p(1, 0)}{P_1}, \dots, \frac{p(\tau, \sigma)}{P_1} \right), \\ (2) : & \left(\frac{p(\tau + 1, 0)}{P_2}, \dots, \frac{p(\tau + 1, \sigma)}{P_2}, \frac{p(\tau + 2, 0)}{P_2}, \dots, \frac{p(255, \sigma)}{P_2} \right), \end{aligned}$$

where P_1 and P_2 are two independent *posteriori* class probabilities as follows:

$$P_1 = \sum_{i=0}^{\tau} \sum_{j=0}^{\sigma} p(i, j), \quad P_2 = \sum_{i=\tau+1}^{255} \sum_{j=0}^{\sigma} p(i, j).$$

For the three-level thresholding with $n = 2$, the GLLV histogram plane is subdivided by thresholds $\tau = (\tau_1, \tau_2)$ and σ into 6 divisions (see Figure 3.4 (b)). The first, second and third quadrants represent the object and the background information of the original image $f(x, y)$ because pixels from these three quadrants have smaller local variance values within $[0, \sigma]$.

Then three normalized disjointed probability distributions associated with

first, second and third quadrants are:

$$\begin{aligned}
(1) &: \left(\frac{p(0,0)}{P_1}, \dots, \frac{p(0,\sigma)}{P_1}, \frac{p(1,0)}{P_1}, \dots, \frac{p(\tau_1,\sigma)}{P_1} \right), \\
(2) &: \left(\frac{p(\tau_1+1,0)}{P_2}, \dots, \frac{p(\tau_1+1,\sigma)}{P_2}, \frac{p(\tau_1+2,0)}{P_2}, \dots, \frac{p(\tau_2,\sigma)}{P_2} \right), \\
(3) &: \left(\frac{p(\tau_2+1,0)}{P_3}, \dots, \frac{p(\tau_2+1,\sigma)}{P_3}, \frac{p(\tau_2+2,0)}{P_3}, \dots, \frac{p(255,\sigma)}{P_3} \right),
\end{aligned}$$

where P_1 , P_2 and P_3 are three independent *posteriori* class probabilities as follows:

$$P_1 = \sum_{i=0}^{\tau_1} \sum_{j=0}^{\sigma} p(i,j), \quad P_2 = \sum_{i=\tau_1+1}^{\tau_2} \sum_{j=0}^{\sigma} p(i,j), \quad P_3 = \sum_{i=\tau_2+1}^{255} \sum_{j=0}^{\sigma} p(i,j).$$

Then we conclude that the generalized $(n+1)$ -level thresholding model has threshold values $\tau = (\tau_1, \tau_2, \dots, \tau_n)$ and σ . Therefore, $(n+1)$ normalized disjointed probability distributions associated with $(n+1)$ quadrants from the bottom of the histogram plane are as follows:

$$\left(\frac{p(\tau_{k-1}+1,0)}{P_k}, \dots, \frac{p(\tau_{k-1}+1,\sigma)}{P_k}, \frac{p(\tau_{k-1}+2,0)}{P_k}, \dots, \frac{p(\tau_k,\sigma)}{P_k} \right),$$

where P_k is the k^{th} independent *posteriori* class probability:

$$P_k = \sum_{i=\tau_{k-1}+1}^{\tau_k} \sum_{j=0}^{\sigma} p(i,j), \quad (3.4)$$

with the convention of $\tau_0 = -1$, $\tau_{n+1} = 255$.

We point out that the local variance g always only requires a single value σ according to the definition of the GLLV histogram.

3.2.1 Thresholding using Shannon entropy

From the discussion of the Shannon entropy in section (2.1) as well as section (3.1.1), we know that the criterion function for the multi-level thresholding model based on the Shannon entropy is given by equation (2.7):

$$\Phi_{n+1} := S_{1+2+\dots+(n+1)} = \sum_{k=1}^{n+1} S_k,$$

where S_k is the Shannon entropy corresponding to the k^{th} disjointed probability distribution class:

$$S_k = - \sum_{i=\tau_{k-1}+1}^{\tau_k} \sum_{j=0}^{\sigma} \left(\frac{p(i,j)}{P_k} \right) \ln \left(\frac{p(i,j)}{P_k} \right)$$

and $P_k : k = 1, 2, \dots, n+1$ are given in the equation (3.4) with the convention of $\tau_0 = -1, \tau_{n+1} = 255$.

Therefore the optimal threshold pair (τ^*, σ^*) is obtained from:

$$(\tau^*, \sigma^*) = \underset{(\tau, \sigma) \in G^n \times G}{\operatorname{argmin}} \Phi_{n+1}(\tau, \sigma),$$

where $\tau^* = (\tau_1^*, \tau_2^*, \dots, \tau_n^*)$. For two-dimensional thresholding models, we only use $\tau^* = (\tau_1^*, \tau_2^*, \dots, \tau_n^*)$ to segment the image. The segmented image is also obtained from equation (3.2).

3.2.2 Thresholding using Rényi entropy

From the discussion of the Rényi entropy in section (2.2), as well as section (3.1.2), we know that the criterion function for the multi-level thresholding method based on the Rényi entropy is given by equation (2.10):

$$\Phi_{n+1} := R_{1+2+\dots+(n+1)} = \sum_{k=1}^{n+1} R_k,$$

where R_k is the Rényi entropy corresponding to the k^{th} disjointed probability distribution class:

$$R_k = \frac{1}{1-\alpha} \left[\ln \sum_{i=\tau_{k-1}+1}^{\tau_k} \sum_{j=0}^{\sigma} \left(\frac{p(i,j)}{P_k} \right)^\alpha \right]$$

and $P_k : k = 1, 2, \dots, n+1$ are given in the equation (3.4) with the convention of $\tau_0 = -1, \tau_{n+1} = 255$.

Therefore the optimal threshold pair (τ^*, σ^*) is obtained from:

$$(\tau^*, \sigma^*) = \underset{(\tau, \sigma) \in G^n \times G}{\operatorname{argmin}} \Phi_{n+1}(\tau, \sigma),$$

where $\tau^* = (\tau_1^*, \tau_2^*, \dots, \tau_n^*)$. The segmented image is also obtained from equation (3.2).

3.2.3 Thresholding using Tsallis-Havrda-Charvát entropy

From the discussion of the Tsallis-Havrda-Charvát entropy in section (2.3), as well as section (3.1.3), we know that the criterion function for the multi-level thresholding method based on the Tsallis-Havrda-Charvát entropy is given by equation (2.14):

$$\begin{aligned}\Phi_{n+1} &:= T_{1+2+\dots+(n+1)} \\ &= \sum_{i=1}^{n+1} T_i + (1-\alpha) \cdot \sum_{i=1}^n \sum_{j=i+1}^{n+1} T_i T_j \\ &\quad + (1-\alpha)^2 \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=j+1}^{n+1} T_i T_j T_k \\ &\quad + (1-\alpha)^3 \cdot \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n \sum_{l=k+1}^{n+1} T_i T_j T_k T_l \\ &\quad + \dots + (1-\alpha)^n \cdot \prod_{i=1}^{n+1} T_i ,\end{aligned}$$

where T_k is the Tsallis-Havrda-Charvát entropy corresponding to the k^{th} disjointed probability distribution class:

$$T_k = \frac{1}{\alpha - 1} \left[1 - \sum_{i=\tau_{k-1}+1}^{\tau_k} \sum_{j=0}^{\sigma} \left(\frac{p(i, j)}{P_k} \right)^\alpha \right]$$

and $P_k : k = 1, 2, \dots, n+1$ are given in the equation (3.4) with the convention of $\tau_0 = -1, \tau_{n+1} = 255$.

The optimal threshold pair (τ^*, σ^*) is obtained from:

$$(\tau^*, \sigma^*) = \underset{(\tau, \sigma) \in G^n \times G}{\operatorname{argmin}} \Phi_{n+1}(\tau, \sigma) ,$$

where $\tau^* = (\tau_1^*, \tau_2^*, \dots, \tau_n^*)$. The segmented image is also obtained from equation (3.2).

3.2.4 Experiments

The main part of the PSO algorithm is the same process as the one mentioned in section (2.4). However, for a $(n + 1)$ -level thresholding based on the GLLV histogram, the i^{th} particle from the PSO algorithm is a $(n+1)$ -dimensional threshold vector:

$$\vec{x}_i(t) = (\tau_i, \sigma_i) = (\tau_{i1}, \tau_{i2}, \dots, \tau_{in}, \sigma_i),$$

where $i = 1, \dots, m$, and m is the total number of particles. The searching space is a $(n + 1)$ -dimensional space. We notice that $\tau_i = (\tau_{i1}, \tau_{i2}, \dots, \tau_{in})$ represents the threshold for the original image $f(x, y)$ and σ_i represents the threshold for the local variance image $g(x, y)$. According to section (1.4), the main parts of the PSO algorithm are as follows:

Step 1. Initialize $\vec{x}_i(t)$ and use $\vec{x}_i(t)$ to initialize $\vec{p}_i(t)$

Step 2. Evaluate the criterion function Φ_{n+1} (according to which model we want to use) at $\vec{x}_i(t)$

Step 3. Update $\vec{g}(t)$ by function (1.18)

Step 4. Evaluate velocity $\vec{v}_i(t)$ by function (1.15)

Step 5. Update $\vec{x}_i(t)$ by function (1.16)

Step 6. Update $\vec{p}_i(t)$ and $\vec{g}(t)$ by (1.17) and (1.18) respectively

Step 7. Return to step 2 .

We continue to use the image 113016.jpg to illustrate the experiment results from GLLV based models. We demonstrate ten-level thresholded results in Figure 3.5. (b) is the result from the Shannon based model, (c) is the result from the Rényi

Table 3.3 – Mean values from each thresholded class.

Models	Shannon	Rényi	Tsallis-Havrda-Charvát
10-level	28, 55, 67, 96, 119, 137,	50, 61, 78, 94, 115, 148,	57, 81, 103, 142, 164
	148, 161, 192, 228	182, 198, 204, 226	169, 187, 215, 230, 242

Table 3.4 – Thresholds values for different entropic models.

Models	Shannon	Rényi	Tsallis-Havrda-Charvát
10-level	31, 61 , 73, 114, 122,	54, 73, 83, 104, 123,	69, 92, 112, 161, 167,
	146, 148, 178, 208	171, 193, 202, 205	171, 206, 224, 234

based model and (d) is the result from the Tsallis-Havrda-Charvát based model. Images (e)-(h) in Figure 3.5 are their 1D histograms and images (i)-(l) in Figure 3.5 are their corresponding GLLV histograms. We notice that since (a), (b) and (c) are ten-level results, their GLLV histogram have only 10 stripes and each stripe has a height of at most 64 (see section 1.1.2).

Table 3.3 lists all the mean values that we used to exhibit the images in Figure 3.5. Table 3.4 lists all the threshold values that we obtained from the three different entropic models. We point out that for 10-level thresholding models, we need nine threshold values.

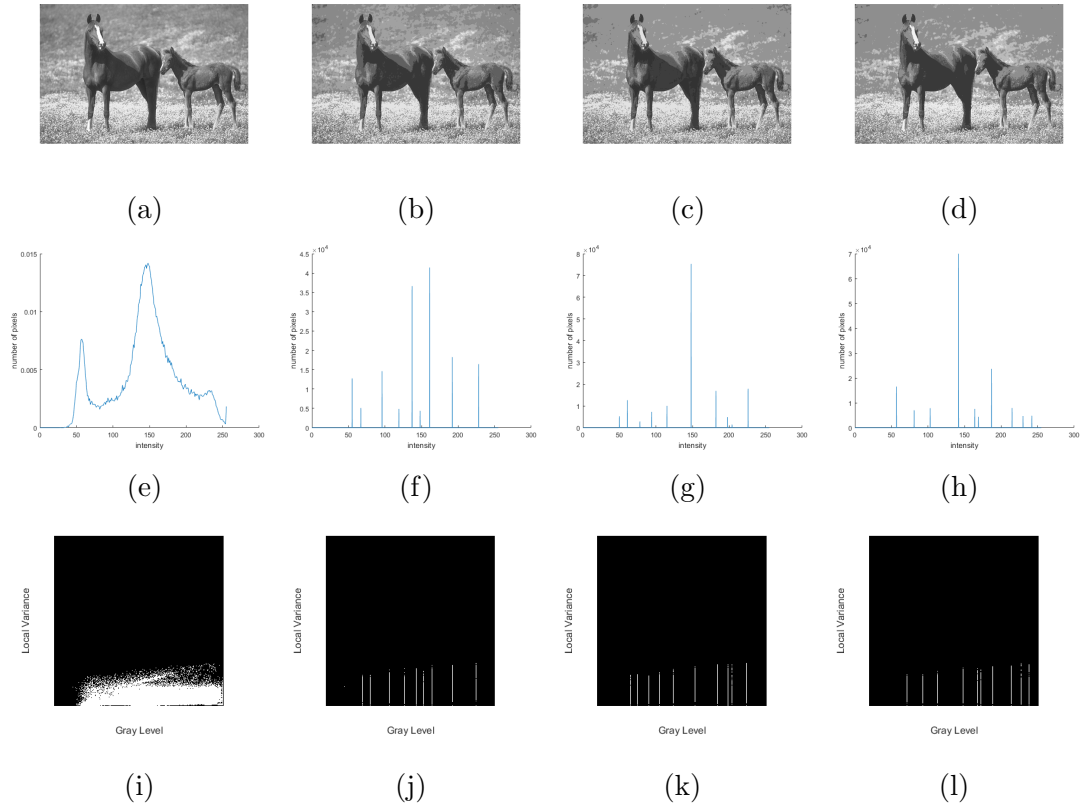


Figure 3.5 – Original image, the ten-level thresholded images and their corresponding 1D and GLLV histograms

CHAPTER 4

RGB COLOR IMAGE THRESHOLDING MODELS BASED ON ENTROPIES AND TWO-DIMENSIONAL HISTOGRAMS

Generally speaking, a color image provides a better description of a scene than the gray-level image [4, 6, 22, 23, 28, 38, 45]. It is known that most of the segmentation methods for a gray-level image can be directly applied to each component of a RGB color image [6]. However, so far only a limited amount of studies [23, 22] mentioned how to employ multi-level thresholding techniques on a color image. We point out that all the thresholding models mentioned in the introduction section didn't deal with a color image segmentation. Under this motivation, we come up with a RGB color image segmentation model under the framework of a multi-level thresholding technique by using the Shannon, the Rényi and the Tsallis-Havrda-Charvát Entropies as well as GLLA and GLLV histograms.

By observing each $1D$ histogram, we select a proper threshold level for each component (R, G and B) image. Then, we obtain the thresholded RGB image which has less colors than the original RGB image. Thus our work implies a new scheme for the RGB color image segmentation.

4.1 The fusion of three thresholded component image

In this part, we discuss how to threshold a RGB color image by using the multi-level thresholding technique based on entropies and $2D$ histograms.

Firstly, by observing the $1D$ histograms of the three components, we choose $(n_1 + 1)$, $(n_2 + 1)$ and $(n_3 + 1)$ as thresholding levels for the red, green and blue

components respectively and apply the PSO algorithm to search for the optimal threshold values for each component.

Then the thresholded red component T_r has $(n_1 + 1)$ gray levels, the green component T_g has $(n_2 + 1)$ gray levels, and the blue component T_b has $(n_3 + 1)$ gray levels. Then, by combining three thresholded components, we obtain the thresholded RGB image $\vec{T}(x, y)$:

$$[\vec{T}(x, y)] = [T_r(x, y), T_g(x, y), T_b(x, y)] .$$

Thus the segmented RGB image has at most $(n_1 + 1) \times (n_2 + 1) \times (n_3 + 1)$ color levels, which is a much smaller number than the number of colors from the original RGB image.

In general, the original RGB image has about 10^7 color levels. However color levels for the segmented color image are between 8 and 1000 in our model if the thresholding level n satisfies the condition $2 \leq n \leq 10$ for each component. We comment that the proposed scheme reduces the number of distinct colors to such a large amount that our models are reasonable methods for the RGB color image segmentation.

4.2 Experiments

We claim that all the 1D thresholding models mentioned in chapter 2 as well as 2D thresholding models mentioned in chapter 3 can be directly applied on each component of a RGB image. In this chapter, we continue to use the image 113016.jpg (see Figure 1.2 (a)) to test all the models.

In Figure 4.1, we list the red, green and blue components in (a), (b) and (c) respectively. Images (d)-(f) are their corresponding 1D histograms. Images (g)-(i) are their corresponding GLLA histograms. Images (j)-(l) are their corresponding GLLV histograms. According to the 1D histogram of the red component (see Figure

Table 4.1 – Mean values and thresholds values from 1D Tsallis-Havrda-Charvát thresholding model.

	Red	Green	Blue
Mean values	68, 134, 208	68, 117, 170, 237	68, 110, 230
Threshold values	100, 168	96, 136, 211	93, 182

4.1 (d)), we use 3 as the thresholding level. In the same manner, we choose 4 as the thresholding level for the green component and 3 as the thresholding level for the blue component. Then the segmented RGB image has at most $3 \times 4 \times 3 = 36$ color levels.

We use Tsallis-Havrda-Charvát based models from section (2.3), section (3.1.3) and section (3.2.3) to show experiment results for the RGB image of 113016.jpg.

Experiment results in Figure 4.2 are obtained by using the model in section (2.3): 1D Tsallis-Havrda-Charvát thresholding model. Images (a)-(c) in Figure 4.2 are the red, green and blue thresholded images respectively. Figure 4.2 (d) is a fusion of the three components. Their 1D histograms are listed in (e)-(h) respectively. Table 4.1 lists all the mean values and threshold values for each component image from Figure 4.2. Table 4.2 lists 21 different color levels from the fusion of three component images. We read the data from Table 4.2 vertically in groups of three. For example, the first color level is a triplet (68, 68, 68), and the last color level is a triplet (208, 237, 230).

Experiment results in Figure 4.3 are obtained by using the model in section (3.1.3): GLLA Tsallis-Havrda-Charvát thresholding model. Images (a)-(c) in Figure 4.3 are the red, green and blue thresholded images respectively. Figure 4.3 (d) is a fusion of the three components. Their 1D histograms are listed in (e)-(h) respectively. Table 4.3 lists all the mean values and threshold values for each component image from Figure 4.3. Table 4.4 lists 25 different color levels from the fusion of

Table 4.2–21 colors from 1D Tsallis-Havrda-Charvát thresholding model.

Red	68, 68, 68, 68, 68, 134, 134, 134, 134, 134, 134, 134, 208, 208, 208, 208, 208, 208, 208, 208, 208
Green	68, 117, 117, 170, 170, 68, 68, 117, 117, 170, 170, 237, 68, 117, 117, 170, 170, 170, 237, 237, 237
Blue	68, 68, 110, 68, 110, 68, 110, 68, 110, 68, 110, 110, 68, 68, 110, 68, 110, 230, 68, 110, 230

Table 4.3–Mean values and thresholds values from GLLA Tsallis-Havrda-Charvát thresholding model.

	Red	Green	Blue
Mean values	60, 137, 220	65, 121, 171, 234	22, 59, 105
Threshold values	85, 184	87, 146, 206	32, 80

three component images. We read the data from Table 4.4 vertically in groups of three. For example, the first color level is a triplet (60,65,22), and the last color level is a triplet (220,234,105).

Experiment results in Figure 4.4 are obtained by using the model in section (3.2.3): GLLV Tsallis-Havrda-Charvát thresholding model. Images (a)-(c) in Figure 4.4 are the red, green and blue thresholded images respectively. Figure 4.4 (d) is the fusion of the three components. Their 1D histograms are listed in (e)-(h) respectively. Table 4.5 lists all the mean values and threshold values for each component image from Figure 4.3. Table 4.6 lists 25 different color levels from the fusion of three component images. We read the data from Table 4.6 vertically in groups of three. For example, the first color level is a triplet (70,62,51), and the last color level is a triplet (225,203,133).

Table 4.4 – 25 colors from GLLA Tsallis-Havrda-Charvát thresholding model.

Red	60, 60, 60, 60, 60, 60, 137, 137, 137, 137, 137, 137, 137, 137, 137, 137, 137, 220 , 220, 220, 220, 220,220, 220, 220
Green	65, 65, 65, 121, 121, 121, 65, 65, 65, 121, 121, 121, 171, 171, 171, 234, 234, 121, 121, 171, 171, 171, 234, 234, 234
Blue	22, 59, 105, 22, 59, 105, 22, 59, 105, 22, 59, 105, 22, 59, 105, 59, 105, 59, 105, 22, 59, 105, 22, 59, 105

Table 4.5 – Mean values and thresholds values from GLLV Tsallis-Havrda-Charvát thresholding model.

	Red	Green	Blue
Mean values	70, 142, 225	62, 105, 154, 203	51, 96, 133
Threshold values	103, 191	79, 132, 167	69, 116

Table 4.6 – 25 colors from GLLV Tsallis-Havrda-Charvát thresholding model.

Red	70, 70, 70, 70, 70, 70, 70, 142, 142, 142, 142, 142, 142, 142, 142, 142, 142, 142, 225 , 225, 225, 225, 225, 225, 225
Green	62, 62, 105, 105, 154, 154, 203, 62, 62, 105, 105, 105, 154, 154, 154, 203, 203, 203, 105, 105, 154, 154, 203, 203, 203
Blue	51, 96, 51, 96, 51, 96, 96, 51, 96, 51, 96, 133, 51, 96, 133, 51, 96, 133, 96, 133, 96, 133, 51, 96, 133

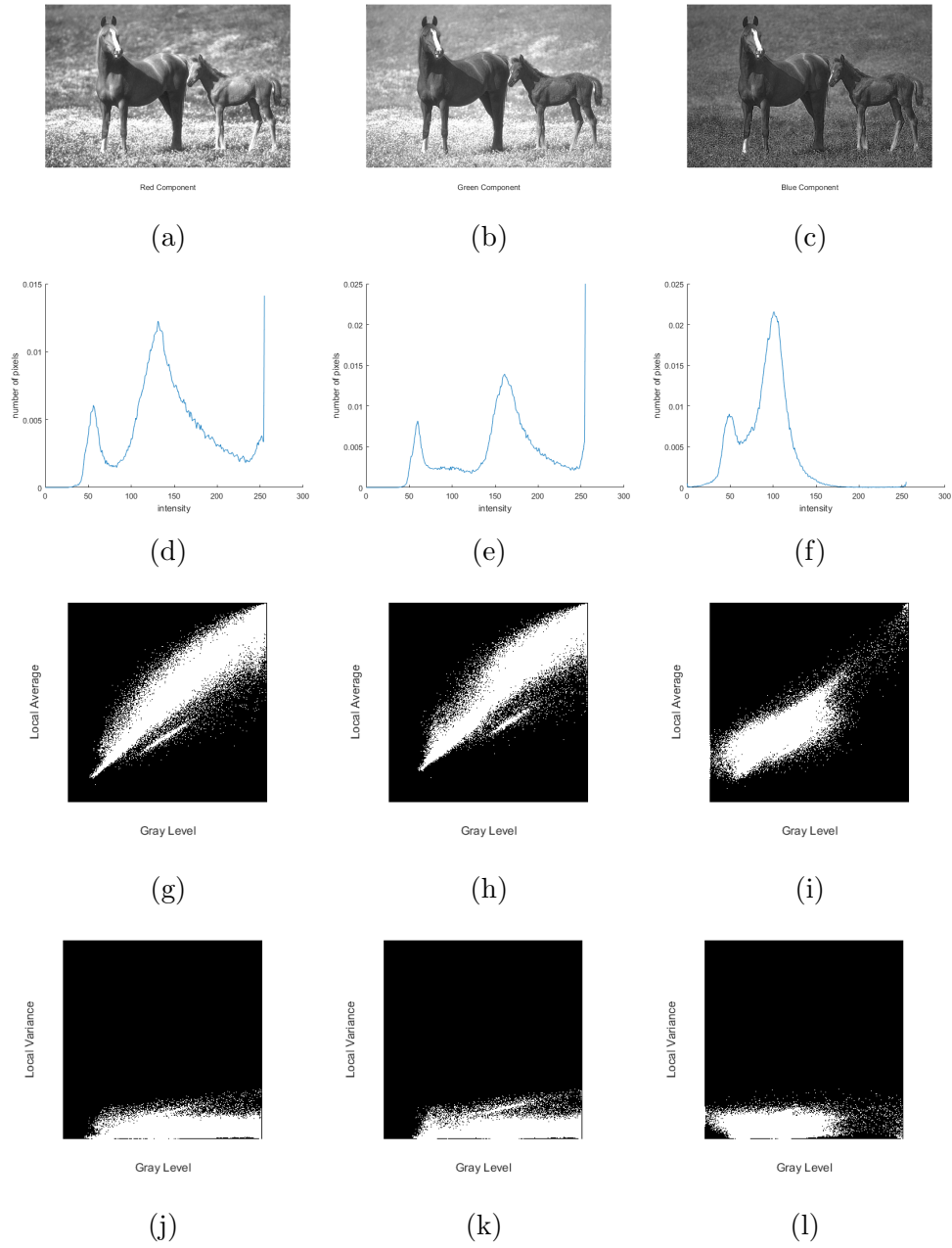


Figure 4.1 – Red, Green and Blue component of 113016.jpg and their corresponding 1D histograms, GLLA histograms and GLLV histograms

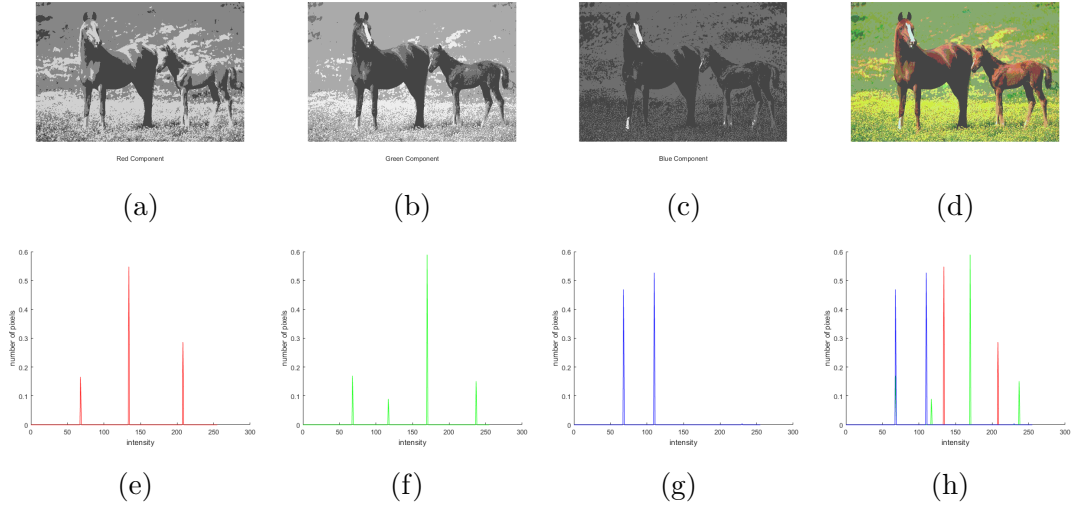


Figure 4.2—Thresholded component images and the fusion RGB image from 1D Tsallis-Havrda-Charvát thresholding model

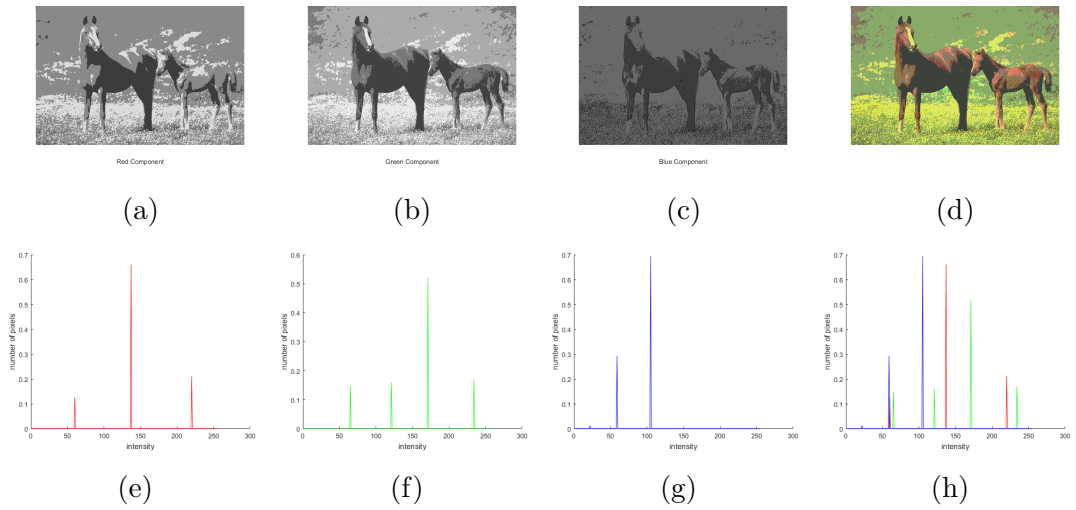


Figure 4.3—Three thresholded component images and the fusion RGB image from GLLA Tsallis-Havrda-Charvát model

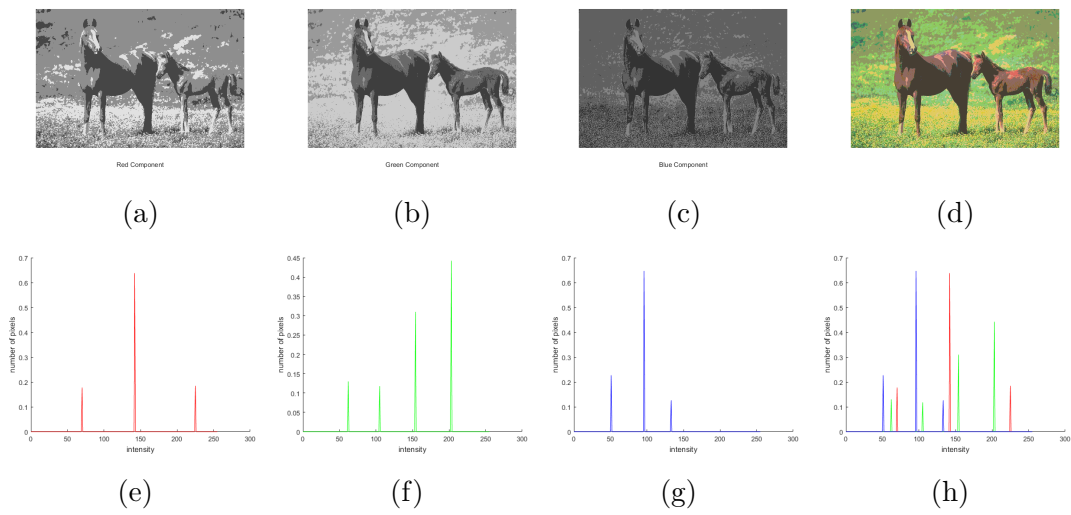


Figure 4.4 – Three thresholded component images and the fusion RGB image from GLLV Tsallis-Havrda-Charvát model

CHAPTER 5

COMPARISON OF MODELS

In this chapter, we evaluate the effectiveness of each model objectively by comparing the averages of the four performance indices (PRI , GCE , VOI , and BDE) [44, 48] of the 300 images from The Berkeley Segmentation Dataset and Benchmark (BSDS300). We claim that there are 300 RGB images in the BSDS300. For each image from the BSDS300, we apply the proposed model to obtain the corresponding thresholded image. Then we convert the thresholded image into the corresponding labeled image, whose number of gray levels is the same as the thresholded image's gray level. Next we compare each labeled image with the corresponding benchmark images from BSDS300 to calculate the four performance indices.

We point out that the PSO algorithm does a random search for the optimal thresholding values, and, as a result, the four performance indices change values after each running of the program. This implies that comparing the four performance indices of a single image among different models does not give any practical guidance. Therefore, we repeat this process for all 300 images from BSDS300 to calculate the average four performance indices (PRI , GCE , VOI , and BDE) which are used in comparison among different models.

All the images have sizes 481×321 or 321×481 and are normalized to have the sizes 320×214 or 214×320 in the calculation. For each image, a set of benchmark images (ground truth, human segmentation), which are compiled by different human observers, are provided. Figure 5.1 illustrates six benchmark images for the image



Figure 5.1 – Benchmark images of 113016.jpg from BSDS300.

113016.jpg.

We start this chapter by introducing the four performance indices firstly. Then we compute the averages of the four performance indices of 300 images for different models from this dissertation (see chapter 2 and chapter 3). Then, we give a conclusion based on the quantitative analysis.

We continue to use image 113016.jpg as an example. Figure 5.2 demonstrates the original image (a), the thresholded image (b) and the labeled image (c) with their corresponding 1D histograms. We claim that Figure 5.2 (b) is the same image as Figure 4.3 (d). From Table 4.4, we know that Figure 5.2 (b) has 25 color levels; that means the labeled image Figure 5.2 (c) also has 25 different labelings. We choose three labels (Figure 5.3 (a) with label 2, (b) with label 15 and (c) with label 25) to illustrate how we apply the labeled image in future applications.

5.1 Four performance indices

Consider N pixels $\{x_m : m = 1, \dots, N\}$ and two clusterings of these pixels: $C = \{C_1, \dots, C_{K_1}\}$ (benchmark image) and $C' = \{C'_1, \dots, C'_{K_2}\}$ (segmented image). Thus each segment ($C_i, i = 1, \dots, K_1$ or $C'_j, j = 1, \dots, K_2$) is a set of some pixels.

Assume that n_{ij} counts the number of pixels simultaneously in the i^{th} cluster

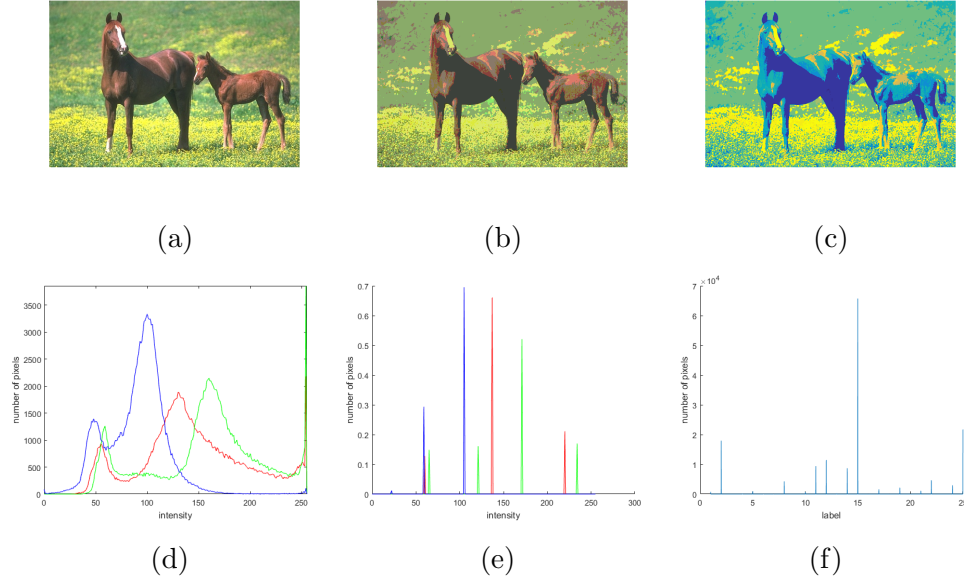


Figure 5.2–The original 113016.jpg, thresholded image and labeled image (thresholding levels are R: 3, G: 4, B: 3)

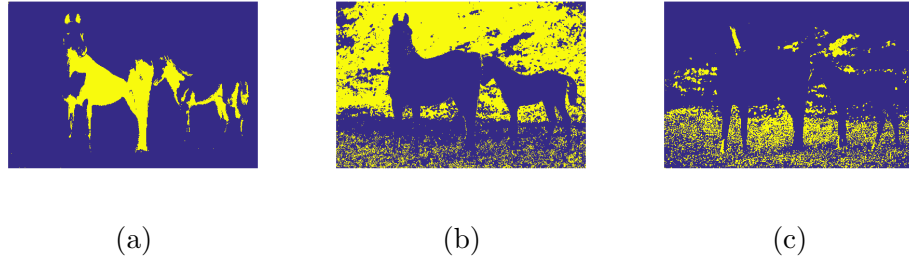


Figure 5.3–Label 2, 15 and 25 for the thresholded image

of C and the j^{th} cluster of C' ($i = 1, \dots, K_1$ and $j = 1, \dots, K_2$). We use a one-dimensional clustering example below to show the idea behind each performance index. Let $C = \{(a, b, c), (d, e, f)\}$ and $C' = \{(a, b), (c, d, e), (f)\}$ with $N = 6$; then $n_{11} = 2$ because pixels a and b are in the 1st cluster of C and the 1st cluster of C' simultaneously. In the same way, we calculate the values for all the n_{ij} , $\{i = 1, 2; j = 1, 2, 3\}$. Then it gives us a matrix:

$$[n_{ij}] = \begin{pmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}.$$

5.1.1 Probability rand index (PRI)

PRI [26, 40] counts the fraction of pairs of pixels whose labelings are consistent between C' and C . William Rand [26] proposed a similarity function that converted the problem of comparing two partitions with possibly different numbers of classes into a problem of computing pairwise label relationships. In one word, *PRI* counts the probability of pairs of pixels whose labelings are consistent between C' and C . *PRI* has values that are between 0 (when C and C' are totally different) and 1 (when C and C' are identical). We conclude that the higher the value of *PRI* is, the better the segmentation result is.

When C consists of a single cluster $\{(a, b, c, d, e, f)\}$ and C' consists of clusters containing only single points $\{(a), (b), (c), (d), (e), (f)\}$, or vice versa, we think of C and C' as two totally different clusterings. William Rand defined *PRI* between C and C' as:

$$PRI(C, C') := \frac{1}{\binom{n}{2}} \left[\binom{N}{2} - \frac{1}{2} \sum_i \left(\sum_j n_{ij} \right)^2 - \frac{1}{2} \sum_j \left(\sum_i n_{ij} \right)^2 + \sum_i \sum_j n_{ij}^2 \right]$$

$$= \frac{15 - 9 - 7 + 10}{15} = 0.6,$$

where $i = 1, 2$ and $j = 1, 2, 3$.

5.1.2 Global consistency error (GCE)

GCE measures the extent to which one segmentation can be viewed as a refinement of the other [24, 40]. Segmentations which are related in this manner are considered to be consistent since they could represent the same natural image segmentation at different scales.

For a given pixel x_m , we consider the segments in C and C' that contain that pixel. If one segment is a proper subset of the other, then the pixel lies in an area of refinement, and *GCE* is zero. If there is no subset relationship, then the two

regions overlap in an inconsistent manner. In this case, GCE is non-zero. When C and C' don't have any overlapped region, GCE is 1. Thus, the lower the value of GCE is, the better the segmentation result is. GCE between C and C' is defined as:

$$GCE(C, C') := \frac{1}{N} \min \left\{ \sum_{m=1}^N LRE(C, C', x_m), \sum_{m=1}^N LRE(C', C, x_m) \right\}, \quad (5.1)$$

where LRE is the local refinement error:

$$LRE(C, C', x_m) := \frac{|R(C, x_m) \setminus R(C', x_m)|}{|R(C, x_m)|},$$

with $R(C, x_m)$ as the set of pixels corresponding to the segment in C that contains pixel x_m . \setminus is set difference, and $| \cdot |$ is the cardinality of a set. For the 1D clustering example with $C = \{(a, b, c), (d, e, f)\}$ and $C' = \{(a, b), (c, d, e), (f)\}$, we have:

$$R(C, a) = R(C, b) = R(C, c) = (a, b, c),$$

$$R(C, d) = R(C, e) = R(C, f) = (d, e, f),$$

$$R(C', a) = R(C', b) = (a, b),$$

$$R(C', c) = R(C', d) = R(C', e) = (c, d, e),$$

$$R(C', f) = (f).$$

Then

$$LRE(C, C', a) := \frac{|R(C, a) \setminus R(C', a)|}{|R(C, a)|} = \frac{|(a, b, c) \setminus (a, b)|}{|(a, b, c)|} = \frac{|(c)|}{|(a, b, c)|} = \frac{1}{3}.$$

In the same manner, we calculate the local refinement error, LRE , for all the pixels:

$$LRE(C, C', b) = \frac{1}{3}, \quad LRE(C, C', c) = \frac{2}{3},$$

$$LRE(C, C', d) = \frac{1}{3}, \quad LRE(C, C', e) = \frac{1}{3}, \quad LRE(C, C', f) = \frac{2}{3}.$$

Since LRE is not symmetric in terms of C and C' , then we have another six different local refinement errors:

$$\begin{aligned} LRE(C', C, a) &= 0, \quad LRE(C', C, b) = 0, \quad LRE(C', C, c) = \frac{2}{3}, \\ LRE(C', C, d) &= \frac{1}{3}, \quad LRE(C', C, e) = \frac{1}{3}, \quad LRE(C', C, f) = 0. \end{aligned}$$

Then finally, according to definition (5.1), we have:

$$GCE(C, C') := \frac{1}{6} \min \left\{ \frac{8}{3}, \frac{4}{3} \right\} = \frac{1}{6} \cdot \frac{4}{3} = \frac{2}{9} = 0.222 \, .$$

5.1.3 Variation of information (VOI)

The VOI , introduced by Meila in 2003 [25], measures the distance between two clustering C and C' in terms of the information difference between them. VOI has values in $[0, \infty]$. VOI is 0 when C and C' are identical segmentations. So, the lower the value of VOI is, the better the segmentation result is. In [25], VOI is defined as:

$$VOI = H(C) + H(C') - 2 \cdot I(C, C') \, , \quad (5.2)$$

where H and I represent respectively the Shannon entropies and the mutual information between the two clusterings.

Assume that

$$a_i = \sum_j n_{ij}, \quad b_j = \sum_i n_{ij} \, ,$$

for the 1D clustering example with $C = \{(a, b, c), (d, e, f)\}$ and $C' = \{(a, b), (c, d, e), (f)\}$, we have: $a_1 = 3, a_2 = 3$ and $b_1 = 2, b_2 = 3, b_3 = 1$. Then $H(C)$, $H(C')$ and $I(C, C')$

are given by:

$$\begin{aligned} H(C) &= - \sum_i \frac{a_i}{N} \cdot \log_2 \frac{a_i}{N} \\ &= - \left(\frac{3}{6} \cdot \log_2 \frac{3}{6} + \frac{3}{6} \cdot \log_2 \frac{3}{6} \right) = \log_2 2 = 1, \end{aligned}$$

$$\begin{aligned} H(C') &= - \sum_j \frac{b_j}{N} \cdot \log_2 \frac{b_j}{N} \\ &= - \left(\frac{2}{6} \cdot \log_2 \frac{2}{6} + \frac{3}{6} \cdot \log_2 \frac{3}{6} + \frac{1}{6} \cdot \log_2 \frac{1}{6} \right) = 1.4591, \end{aligned}$$

$$\begin{aligned} I(C, C') &= \sum_i \sum_j \frac{n_{ij}}{6} \cdot \log_2 \left(\frac{\frac{n_{ij}}{6}}{\frac{a_i}{6} \cdot \frac{b_j}{6}} \right) \\ &= \left(\frac{1}{3} \log_2 2 + \frac{1}{6} \log_2 \frac{2}{3} \right) + \left(\frac{1}{3} \log_2 \frac{4}{3} + \frac{1}{6} \log_2 2 \right) \\ &= 0.5409. \end{aligned}$$

Finally, according to definition (5.2), we have :

$$VOI = H(C) + H(C') - 2 \cdot I(C, C') = 1 + 1.4591 - 2 * 0.5409 = 1.3773,$$

where we assume that $0 * \log_2 0 = 0$.

5.1.4 Boundary displacement error (BDE)

BDE [12, 17] measures the average displacement error of boundary pixels between two segmented images by defining the error of a boundary pixel as the distance between it and the closest pixel in the other boundary image. In this paper, we use the weighted boundary segmentation error rates [17]. When C and

C' are totally different, then BDE is 1. When C and C' are identical, then BDE is 0. We conclude that the lower the value of BDE is, the better the segmentation result is.

Assume that G^B is the boundary for the benchmark image C , and B is the boundary for the segmented image C' . Then for the 1D clustering example with $C = \{(a, b, c), (d, e, f)\}$ and $C' = \{(a, b), (c, d, e), (f)\}$, we can label them as $C_L = (1, 1, 1, 2, 2, 2)$ and $C'_L = (1, 1, 2, 2, 2, 3)$. This is because, in the clustering C , pixels a, b and c are classified into the first cluster (a, b, c) . We label these three pixels as 1 in C_L , and at the same time, pixels d, e and f are in the second cluster (d, e, f) . We label them as 2 in C_L . For the clustering C' , we use the same method to get C'_L .

Next we apply the difference method to approximate the boundaries G^B and B . For pixel a , we apply backward difference on the first element 1 from C_L , then we have $1 - 1 = 0$. For pixel f , we apply forward difference on the last element 2 from C_L , then we have $2 - 2 = 0$. For pixels b, c, d and e , we apply central difference on the non-boundary elements from C_L . Then for pixel b , we have $\frac{1 - 1}{2} = 0$. For pixels c and d , we have $\frac{2 - 1}{2} = 0.5$. For pixels e , we have $\frac{2 - 2}{2} = 0$. Thus we obtain the boundary $G^B = (0, 0, 0.5, 0.5, 0, 0)$. In the same manner, we can derive $B = (0, 0.5, 0.5, 0, 0.5, 1)$.

We point out that the positions of zero elements in G^B and B represent the corresponding non-boundary pixels in C and C' respectively, while the positions of non-zero elements in G^B and B represent the corresponding boundary pixels in C and C' respectively. If we assign '1' to the non-zero elements in G^B and B , then we have boundaries as follows: $G^B = (0, 0, 1, 1, 0, 0)$ and $B = (0, 1, 1, 0, 1, 1)$. Then $|G^B| = 2$ and $|B| = 4$.

In [17], BDE is defined as:

$$BDE = \frac{e_B^m + e_B^f}{2},$$

where $e_B^m = \frac{|T_1|}{|G^B|}$ is the missing boundary rate, and $T_1 = \{x|(x \in G^B) \wedge (x \notin B)\}$ is a set of pixels where the boundary pixels in G^B (or C) are mistakenly classified as non-boundary pixels in B (or C'). $e_B^f = \frac{|T_2|}{|B|}$ is the false boundary rate, and $T_2 = \{x|(x \in B) \wedge (x \notin G^B)\}$ is a set of pixels where the non-boundary pixels in G^B (or C) are mistakenly classified as boundary pixels in B (or C').

We claim that \wedge means ‘and’ in T_1 and T_2 . Then we understand T_1 as a set of pixels that change from 1 in G^B to 0 in B and T_2 as a set of pixels that change from 0 in G^B to 1 in B . Accordingly, in the 1D clustering example, we calculate T_1 and T_2 as follows:

$$T_1 = (G^B). * (\sim B) = (0, 0, 1, 1, 0, 0). * (1, 0, 0, 1, 0, 0) = (0, 0, 0, 1, 0, 0),$$

$$T_2 = (B). * (\sim G^B) = (0, 1, 1, 0, 1, 1). * (1, 1, 0, 0, 1, 1) = (0, 1, 0, 0, 1, 1),$$

where “ \sim ” is a complement operation. Then $|T_1| = 1$ and $|T_2| = 3$. In T_1 , pixel d is a boundary pixel in the benchmark C , but it is mistakenly classified as a non-boundary pixel in the segmented C' . So we think of pixel d as the missing boundary pixel. In T_2 , pixels b , e and f are non-boundary pixels in the benchmark C , but it is mistakenly classified as boundary pixels in the segmented C' . So we think of b , e and f as the false boundary pixels.

We also notice that pixel a is classified as a non-boundary pixel in both the benchmark C and the segmented C' , and pixel c is classified as a boundary pixel in both C and C' . So pixels a and c are the only two correctly classified pixels in the example.

Next, we calculate the missing boundary rate e_B^m and false boundary rate e_B^f

as follows:

$$e_B^m = \frac{|T_1|}{|G^B|} = \frac{1}{2}; \quad e_B^f = \frac{|T_2|}{|B|} = \frac{3}{4}$$

Finally we have:

$$BDE = \frac{e_B^m + e_B^f}{2} = \frac{\frac{1}{2} + \frac{3}{4}}{2} = 0.625.$$

5.2 Comparison of models

In order to compare the proposed method with [48], we convert the BSDS300 RGB images to their corresponding gray-level images firstly, then apply the 3-level segmentation on the gray-level images.

Table 5.1 lists the average four performance indices of the 1D Tsallis-Havrda-Charvát based model (section 2.3), GLLA Rényi based model (section 3.1.2), GLLA Tsallis-Havrda-Charvát based model (section 3.1.3), GLLV Rényi based model (section 3.2.2), GLLV Tsallis-Havrda-Charvát based model (section 3.2.3) and the paper [48]. The bold values from the table demonstrate the best results of the comparison among four models. In “No free lunch theorems for optimization” [41], one mentioned that an optimization algorithm cannot perform its best in every aspect for a problem. We notice from the Table 5.1 that basically each model performs better than other models in terms of one performance index. Thus, we conclude that all the models mentioned in this dissertation are reliable and effective image segmentation methods.

Table 5.1 – Average performance indices for different algorithms.

Model	PRI	GCE	VOI	BDE
1D Tsallis-Havrda-Charvát based model	0.5849	0.3122	2.7840	10.0331
GLLA Rényi based model	0.5517	0.2885	2.7738	10.6270
GLLA Tsallis-Havrda-Charvát based model	0.5574	0.2762	2.7270	10.3590
GLLV Rényi based model	0.5574	0.2873	2.7728	11.4973
GLLV Tsallis-Havrda-Charvát based model	0.5585	0.2934	2.7908	11.1799
GLLA K-L divergence model [48]	0.5975	0.4012	4.2763	11.5203

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this chapter, we give a conclusion on the entire dissertation firstly, then we give some idea for our future work.

6.1 Conclusion

In this dissertation, we mention nine different types of multi-level thresholding models based on the Shannon entropy, the Rényi entropy and the Tsallis-Havrda-Charvát entropy. We use the 1D histogram, the GLLA histogram and the GLLV histogram to construct each model. We apply the PSO algorithm to obtain the thresholded images in a reasonable time period.

Because the Rényi and the Tsallis-Havrda-Charvát entropy become the Shannon entropies when degree α is close to 1 [18] (see equation (1.10) and equation (1.14)), Rényi and Tsallis-Havrda-Charvát entropic models have better performances than the Shannon entropic model. We claim that we use $\alpha = 0.1$ for all the Rényi and Tsallis-Havrda-Charvát entropic models. We compare average four performance indices from the GLLA Tsallis-Havrda-Charvát based model (section 3.1.3) with the indices from the Shannon based model (section 3.1.1) in Table 6.1 to demonstrate that Tsallis-Havrda-Charvát entropic model is more useful than the Shannon based model. Values in bold imply better results. We choose 100 as the iteration number for all the experiments after testing different values. In the future, we will do more research on how to choose the most proper values for α and how

Table 6.1 – Comparison of the average performance indices between the Tsallis-Havrda-Charvát based model and the Shannon based model.

Model	PRI	GCE	VOI	BDE
GLLA Tsallis-Havrda-Charvát based model	0.5574	0.2762	2.7270	10.3590
GLLA Shannon based model	0.5432	0.2685	2.7357	10.6473

to decide the iteration value rigorously.

6.2 Future work

In this section, we discuss the symmetric padding method for the image boundary pixels and several different types 2D histograms.

6.2.1 Symmetric padding method

In section 1.1, we mentioned the zero padding method for the boundary pixels of a matrix f ; and we use the matrix $f_{zeropad}$ to build the GLLA and GLLV histograms. Now we consider using the symmetric padding method on the image boundary pixels to build each thresholding model. Symmetric padding is mainly about padding the matrix boundaries with the mirror reflections of itself. We use the matrix f from section 1.1 as an example to demonstrate this idea. For example:

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{pmatrix} \Rightarrow f_{sympad} = \begin{pmatrix} 1 & 1 & 2 & 3 & 4 & 5 & 5 \\ 1 & 1 & 2 & 3 & 4 & 5 & 5 \\ 6 & 6 & 7 & 8 & 9 & 10 & 10 \\ 11 & 11 & 12 & 13 & 14 & 15 & 15 \\ 16 & 16 & 17 & 18 & 19 & 20 & 20 \\ 21 & 21 & 22 & 23 & 24 & 25 & 25 \\ 21 & 21 & 22 & 23 & 24 & 25 & 25 \end{pmatrix}.$$

6.2.2 Thresholding models based on other types of 2D histograms

In recent years, several thresholding models by applying other types of 2D histograms, such as Gray-Level & Local-Variance histogram (GLLV) [49], Gray-Level & Local-Entropy histogram (GLLE) [5], Gray-Level & Spatial-Correlation histogram (GLSC) [14, 42], Gray-Level & Gradient-Magnitude histogram (GLGM) [43] and 2D Direction histogram (2DD) [46] and so on are developed.

We already applied the GLLV histogram in section 3.2 to construct three thresholding models based on Shannon, Rényi and Tsallis-Havrda-Charvát entropies. Our future work will focus on applying these 2D histograms (GLLE, GLSC and GLGM) to construct novel thresholding models. This can be done by moving all the thresholding techniques based on the GLLA histogram to the ones based on these 2D histogram in parallel. We take the image 113016.jpg for example again. Figure 6.1 illustrates histograms' planes of GLLE, GLSC and GLGM in (a)-(c); the local-entropy image, the spatial-correlation image and the gradient-magnitude image in (d)-(f); and their corresponding 3D demonstrations of the histogram planes (g)-(i).

For the GLSC histogram, the object and background information are mostly concentrated at the lower part on the histogram plane with only a few values, which are $\{1, \dots, 9\}$. We can give the same discussion to the GLLE and the GLGM histograms. Thus, one of the benefits of applying these 2D histograms in the model is that the algorithm will become more time-efficient because we will obtain more simplified versions of the equation (3.3) as well as the equation (2.14).

Next, we give a brief introduction on GLLE, GLSC and GLGM histograms and discuss how to construct them individually.

1. The Gray-Level & Local-Entropy histogram (GLLE)

We know that the entropy is a measure of histogram uniformity. The Gray-

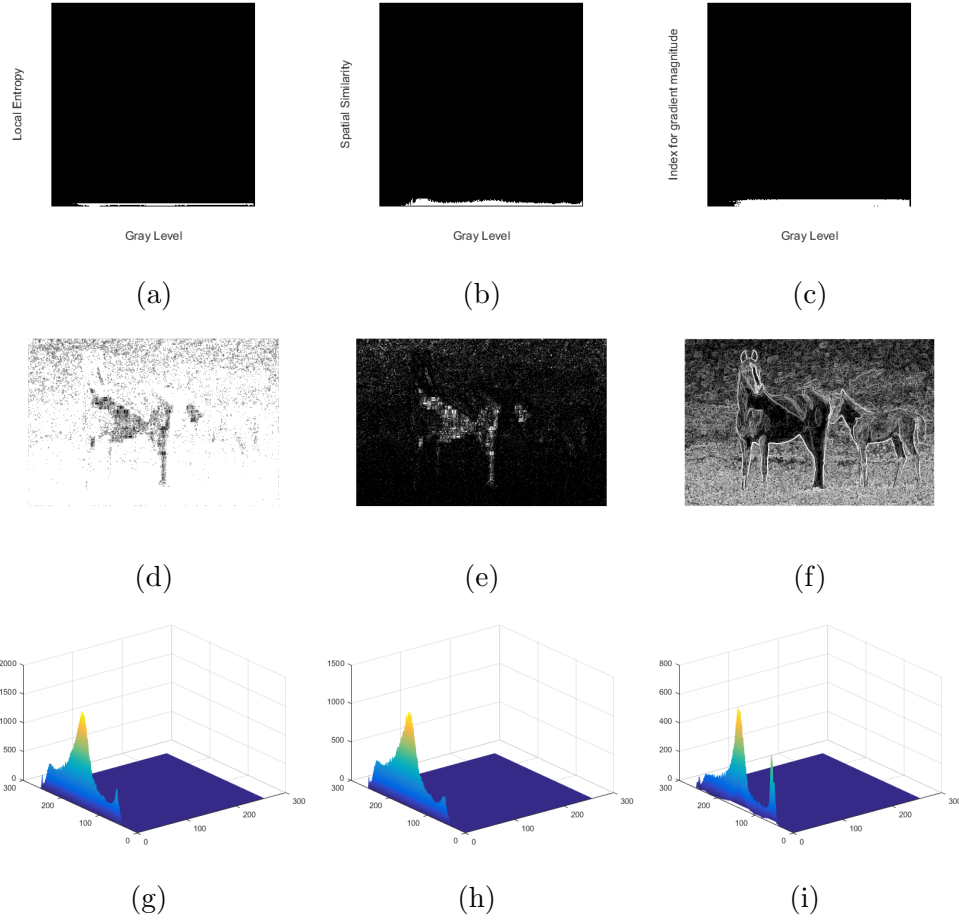


Figure 6.1—From left to right: first row: GLLE, GLSC and GLGM histogram planes; second row: local-entropy image, spatial-correlation image and gradient-magnitude image; third row: the corresponding 3D demonstrations for each histogram plane.

Level & Local-Entropy histogram (GLLE) is constructed from the gray level of each pixel and the local entropy in its neighborhood [5]. Let $g(x, y)$ be the local entropy function from the 3×3 neighborhood of a pixel located at the point (x, y) . Assume that $\{q_1, \dots, q_n\}$ is the normalized probability for the gray levels from the 3×3 neighborhood of pixel (x, y) . For a gray-level image with $n = 256$, the local entropy function $g(x, y)$ is calculated as

$$g(x, y) = \left[- \sum_{i=1}^n q_i \cdot \log_2 q_i \right] , \quad (6.1)$$

then the GLLE histogram is formulated as density function:

$$h(t, s) = Prob(f(x, y) = t \text{ and } g(x, y) = s) ,$$

where $Prob$ refers to the number of pixels that satisfy $f(x, y) = t$ and $g(x, y) = s$ with $t, s \in G$. Then the normalized GLLE histogram is approximated by using the formula as follows:

$$\hat{h}(t, s) = \frac{h(t, s)}{\text{total \# of pixels of } f(x) (M \times N)} .$$

The joint probability mass function $p(t, s)$ is given by:

$$p(t, s) = \hat{h}(t, s),$$

where $t, s = 0, 1, \dots, 255$.

Now we consider the 3×3 neighborhood of the entry $f(1, 1) = 1$ with zero padding:

$$f_{zeropad} = \begin{pmatrix} \begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \textcolor{red}{1} & 2 & 3 & 4 & 5 & 0 \\ 0 & 6 & 7 & 8 & 9 & 10 & 0 \\ \hline 0 & 11 & 12 & 13 & 14 & 15 & 0 \\ 0 & 16 & 17 & 18 & 19 & 20 & 0 \\ 0 & 21 & 22 & 23 & 24 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \end{pmatrix} .$$

Because in the 3×3 neighborhood for entry 1, there are five entries with gray level of 0, one entry with gray level of 1, one entry with gray level of 2, one entry with gray level of 6 and one entry with gray level of 7, the normalized probability from this 3×3 neighborhood is given by $q_1 = \frac{5}{9}, q_2 = \frac{1}{9}, q_3 = \frac{1}{9}, q_7 = \frac{1}{9}, q_8 = \frac{1}{9}$, and the rest of the $q_i = 0, i \in \{1, \dots, 26\}$. Then the local entropy for entry 1 is

$$g(1, 1) = \left\lceil - \sum_{i=1}^{26} q_i \cdot \log_2 q_i \right\rceil = \left\lceil - \left(\frac{5}{9} \right) \cdot \log_2 \left(\frac{5}{9} \right) - 4 \cdot \left(\frac{1}{9} \right) \cdot \log_2 \left(\frac{1}{9} \right) \right\rceil = \lceil 1.88 \rceil = 2 .$$

In the same manner, we can obtain the local entropies for all the entries of f , so the local entropy matrix g_{LE} for the original matrix f is as follows:

$$g_{LE} = \left[\begin{pmatrix} 1.88 & 2.6416 & 2.6416 & 2.6416 & 1.88 \\ 2.6416 & 3.1699 & 3.1699 & 3.1699 & 2.6416 \\ 2.6416 & 3.1699 & 3.1699 & 3.1699 & 2.6416 \\ 2.6416 & 3.1699 & 3.1699 & 3.1699 & 2.6416 \\ 1.88 & 2.6416 & 2.6416 & 2.6416 & 1.88 \end{pmatrix} \right] = \begin{pmatrix} 2 & 3 & 3 & 3 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 \\ 2 & 3 & 3 & 3 & 2 \end{pmatrix} .$$

2. The Gray-Level & Spatial-Correlation histogram (GLSC)

The Gray-Level & Spatial-Correlation histogram (GLSC) is constructed from the gray level of each pixel and its correlation with other pixels from its $N \times N$ neighborhood, where N is a positive odd integer [14, 42]. Normally, this correlation is named as a local similarity function.

Let $g(x, y)$ be the local similarity function from the 3×3 neighborhood of a pixel located at the point (x, y) . Then $g(x, y)$ is calculated as the number of pixels of which the gray levels are close to $f(x, y)$, which is the gray level of the pixel located at the point (x, y) :

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 \#(|f(x+i, y+j) - f(x, y)| \leq \zeta), \quad (6.2)$$

where ζ is the level of the similarity in the neighborhood and defined as:

$$\#(|f(x+i, y+j) - f(x, y)| \leq \zeta) = \begin{cases} 1, & \text{if } |f(x+i, y+j) - f(x, y)| \leq \zeta \\ 0, & \text{if } |f(x+i, y+j) - f(x, y)| > \zeta \end{cases} \quad (6.3)$$

with $i, j \in \{-1, 0, 1\}$. For the 3×3 neighborhood, the maximum value of $g(x, y)$ is 9 when $\zeta \geq \max |f(x+i, y+j) - f(x, y)|$ and the minimum value of $g(x, y)$ is 0 when $\zeta < \min |f(x+i, y+j) - f(x, y)|$.

Then the GLSC histogram is formulated as density function:

$$h(t, s) = Prob(f(x, y) = t \text{ and } g(x, y) = s),$$

where $Prob$ refers to the number of pixels that satisfy $f(x, y) = t$ and $g(x, y) = s$ with $t \in G$, $s \in \{0, \dots, 9\}$. Then the normalized GLSC histogram is approximated by using the formula:

$$\hat{h}(t, s) = \frac{h(t, s)}{\text{total \# of pixels of } f(x) (M \times N)}.$$

The joint probability mass function $p(t, s)$ is given by:

$$p(t, s) = \hat{h}(t, s),$$

where $t = 0, 1, \dots, 255$, $s = 0, \dots, 9$.

We use $f_{zeropad}$ mentioned before as an example. If we choose $\zeta = 2$, then the local similarity value for entry $f(1, 1) = 1$ is 7. Because in its 3×3 neighborhood, except for entries 6 and 7, there are 7 entries $(0, 0, 0, 0, 0, 1, 2)$ with distances from 1 that are less than 2.

$$DistanceMatrix = \left| \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 6 & 7 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \right| = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 5 & 6 \end{pmatrix}.$$

In the same manner, we can obtain the local entropies for all the entries of f , so the spatial correlation matrix g_{SC} for the original matrix f is as follows:

$$g_{SC} = \begin{pmatrix} 7 & 6 & 3 & 3 & 2 \\ 2 & 3 & 3 & 3 & 2 \\ 2 & 3 & 3 & 3 & 2 \\ 2 & 3 & 3 & 3 & 2 \\ 2 & 3 & 3 & 3 & 2 \end{pmatrix}.$$

3. The Gray-Level & Gradient-Magnitude histogram (GLGM)

The Gray-Level & Gradient-Magnitude histogram (GLGM) is constructed from the gray level and labeled gradient magnitude of each pixel. We use the

convolution of the original image with two 3×3 kernels to approximate the derivative of an image. Assume that we have two 3×3 kernels which are sobel operators: one for horizontal changes, and one for vertical:

$$h_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad h_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

If we define f as the source image, then we use G_x to present the horizontal derivative approximation of f and G_y to present the vertical derivative approximation of f respectively. G_x and G_y are computed as follows:

$$G_x = h_x * f, \quad G_y = h_y * f,$$

where $*$ denotes the 2-dimensional signal processing convolution operation. Then the gradient magnitude matrix is given by $GM = \sqrt{G_x^2 + G_y^2}$.

We apply *Fibonacci* numbers to label the gradient magnitude matrix GM into a matrix whose entries have only 9 values that are assigned from $\{1, 2, \dots, 9\}$. We name this labeled matrix as GML . Thus a sequence of 9 *Fibonacci* numbers is used to set up the quantization bins: $\vec{Fibo} = \{1, 1, 2, 3, 5, 8, 13, 21, 34\}$.

Suppose that GM_{min} is the minimum value of GM and GM_{max} is the maximum value of GM , then the entire gradient magnitude entries from GM are scaled into $1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34 = 88$ segments uniformly without overlap as follows:

$$GMS = \frac{GM - GM_{min}}{step},$$

where $step = \frac{GM_{max} - GM_{min}}{88}$, and we name this matrix as GMS .

Consequently, the gradient magnitudes can be quantized as the index numbers of the bins where they are located. So the threshold values for partitioning the gradient magnitude are $\{1, 2, 4, 7, 12, 20, 33, 54, 88\}$. For example, if the gradient magnitude of a pixel is GM_{min} , then we have $\frac{GM_{min} - GM_{min}}{step} = 0$. So we

label 1 at this pixel. If the gradient magnitude of a pixel is GM_{max} , then we have $\frac{GM_{max}-GM_{min}}{step} = 88$. So we label 9 at this pixel. In this way, we obtain the GML .

Then, the segments are merged from low value to high value areas respectively to form the quantization bins. Then the GLGM histogram is formulated as density function:

$$h(t, s) = Prob(f(x, y) = t \text{ and } g(x, y) = s),$$

where $Prob$ refers to the number of pixels that satisfy $f(x, y) = t$ and $g(x, y) = s$ with $t \in G$, $s \in \{1, \dots, 9\}$. Then the normalized histogram is approximated by using the formula:

$$\hat{h}(t, s) = \frac{h(t, s)}{\text{total \# of pixels of } f(x) (M \times N)}.$$

The joint probability mass function $p(t, s)$ is given by:

$$p(t, s) = \hat{h}(t, s),$$

where $t = 0, 1, \dots, 255$, $s = 1, \dots, 9$.

We continue to use matrices f and $f_{zeropad}$ below as examples to illustrate how to construct the GLGM histogram:

$$f = \begin{pmatrix} \textcolor{red}{1} & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{pmatrix}, \quad f_{zeropad} = \begin{pmatrix} \begin{array}{|ccc|} \hline 0 & 0 & 0 \\ 0 & \textcolor{red}{1} & 2 \\ 0 & 6 & 7 \\ \hline \end{array} & 0 & 0 & 0 & 0 \\ 0 & 11 & 12 & 13 & 14 & 15 & 0 \\ 0 & 16 & 17 & 18 & 19 & 20 & 0 \\ 0 & 21 & 22 & 23 & 24 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

For entry 1 of matrix f , its horizontal derivative approximation is calculated as

follows:

$$h_x * \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 6 & 7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 6 & 7 \end{pmatrix} = 0 + (-2)*6 + (-1)*7 = -19.$$

For entry 1 of matrix f , its vertical derivative approximation is calculated as follows:

$$h_y * \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 6 & 7 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 6 & 7 \end{pmatrix} = 0 + 0 + (-2)*2 + (-1)*7 = -11.$$

Then $GM(1, 1) = \sqrt{(-19)^2 + (-11)^2} \approx 21.9545$. In the same manner, we obtain the horizontal and vertical derivative approximations of f as well as the gradient magnitude matrix GM as follows:

$$G_x = h_x * f = \begin{pmatrix} -19 & -28 & -32 & -36 & -29 \\ -30 & -40 & -40 & -40 & -30 \\ -30 & -40 & -40 & -40 & -30 \\ -30 & -40 & -40 & -40 & -30 \\ 49 & 68 & 72 & 76 & 59 \end{pmatrix},$$

$$G_y = h_y * f = \begin{pmatrix} -11 & -6 & -6 & -6 & 17 \\ -28 & -8 & -8 & -8 & 36 \\ -48 & -8 & -8 & -8 & 56 \\ -68 & -8 & -8 & -8 & 76 \\ -61 & -6 & -6 & -6 & 67 \end{pmatrix},$$

$$GM = \sqrt{G_x^2 + G_y^2} = \begin{pmatrix} 21.9545 & 28.6356 & 32.5576 & 36.4966 & 33.6155 \\ 41.0366 & 40.7922 & 40.7922 & 40.7922 & 46.8615 \\ 56.6039 & 40.7922 & 40.7922 & 40.7922 & 63.5295 \\ 74.3236 & 40.7922 & 40.7922 & 40.7922 & 81.7068 \\ 78.2432 & 68.2642 & 72.2496 & 76.2365 & 89.2749 \end{pmatrix}.$$

Since $GM_{min} = 21.9545$ and $GM_{max} = 89.2749$, we use a small step, $\frac{89.2749-21.9545}{88} = 0.7650$, to scale the entries of GM within $[0, 88]$. Thus we obtain the GMS matrix as follows:

$$GMS = \frac{GM - 21.9545}{0.7650} = \begin{pmatrix} 0 & 8.7335 & 13.8602 & 19.0091 & 15.2430 \\ 24.9438 & 24.6243 & 24.6243 & 24.6243 & 32.5580 \\ 45.2931 & 24.6243 & 24.6243 & 24.6243 & 54.3461 \\ 68.4560 & 24.6243 & 24.6243 & 24.6243 & 78.1072 \\ 73.5796 & 60.5352 & 65.7448 & 70.9565 & 88.0000 \end{pmatrix}.$$

Then according to the $\vec{Fibo} = \{1, 1, 2, 3, 5, 8, 13, 21, 34\}$, we partition the gradient magnitude matrix into $1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34 = 88$ segments in total by using the threshold values $\{1, 2, 4, 7, 12, 20, 33, 54, 88\}$. So we assign label 0 to the entries from GMS with values in $[0, 1]$. We assign label 1 to the entries from the GMS with values in $[1, 2]$. We assign label 2 to the entries from the GMS with values in $[2, 4]$, and so on, until we assign label 9 to the entries from GMS with values in $[54, 88]$.

Finally the gradient magnitude matrix is as follows:

$$g_{GM} = \begin{pmatrix} 1 & 5 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 \\ 8 & 7 & 7 & 7 & 9 \\ 9 & 7 & 7 & 7 & 9 \\ 9 & 9 & 9 & 9 & 9 \end{pmatrix}.$$

We notice that for the original matrix f , the pixels have values from 1 to 25. But, the corresponding local entropy values are either 2 or 3 (see local entropy matrix g_{LE}). For the spatial correlation matrix g_{SC} , pixels have values from $\{2, \dots, 7\}$, and for the gradient magnitude matrix g_{GM} , pixels have values from $\{1, \dots, 9\}$. In conclusion, GLLE, GLSC and GLGM histograms based models will simplify the entire image thresholding algorithms.

REFERENCES

- [1] Ahmed S. Abutaleb, *Automatic thresholding of gray-level pictures using two-dimensional entropy*, Computer Vision, Graphics, and Image Processing **47** (1989), 22–32.
- [2] Gamil Abdel Azim and Z. A. Abo-Eleneen, *A novel algorithm for image thresholding using non parametric fisher information*, 1st International electronic conference on entropy and its application, November 2014, pp. 1–15.
- [3] Surina Borjigin and Prasanna K. Sahoo, *Color image segmentation based on multi-level tsallis-havrda-charvát entropy and 2d histogram using pso algorithms, revised and resubmitted*, Pattern Recognition (2017).
- [4] M. Celenk and M. U. Haag, *Optimal thresholding for color images*, vol. 3304, SPIE, Nonlinear Image Processing IX, April 1998, pp. 250–255.
- [5] Jiaquan Chen, Binglei Guan, Hailun Wang, Xuguang Zhang, Yinggan Tang, and Wenzhao Hu, *Image thresholding segmentation based on two dimensional histogram using gray level and local entropy information*, **PP** (2017), no. 99, 1–7.
- [6] H. D. Cheng, X. H. Jiang, Y. Sun, and Jing Li Wang, *Color image segmentation: Advances and prospects*, Pattern Recognition **34** (2001), 2259–2281.
- [7] H. D. Cheng, X. H. Jiang, and J. Wang, *Color image segmentation based on*

- homogram thresholding and region merging*, Pattern Recognition **35** (2002), 373–393.
- [8] Z. Daróczy, *Generalized information functions*, Information and Control **16** (1970), 36–51.
- [9] M. Portes de Albuquerque, I.A. Esquef, and A.R. Gesualdi Mello, *Image thresholding using tsallis entropy*, Pattern Recognition Letters **25** (2004), 1059–1065.
- [10] Omer Demirkaya, Musa H. Asyali, and Prasanna K. Sahoo, *Image processing with matlab: Applications in medicine and biology*, 1 ed., CRC Press, 12 2008.
- [11] Sathya P. Duraisamy and Ramanujam Kayalvizhi, *A new multilevel thresholding method using swarm intelligence algorithm for image segmentation*, Journal of Intelligent Learning Systems and Applications **2** (2010), 126–138.
- [12] J. Freixenet, X. Munoz, D. Raba, and X. Cufi J. Marti, *Yet another survey on image segmentation: Region and boundary information integration*, Proceedings of European Conference on Computer Vision (Bonn, Germany), 2002, pp. 408–422.
- [13] Robert M. Gray, *Entropy and information theory*, 2 ed., Springer, 2 2011.
- [14] Weiya Guo, Xiaofei Wang, and Tianxu Zhang, *Entropic thresholding based on gray-level spatial correlation histogram*, International Conference on Pattern Recognition, Dec 2008, pp. 1–4.
- [15] R. Harrabi and E. Ben Braie, *Color image segmentation using automatic thresholding techniques*, 8th International Multi-Conference on Systems, Signals and Devices, March 2011.
- [16] J. Havrda and F. Charvat, *Quantification method of classification processes: Concept of structural α -entropy*, Kybernetika **3** (1967), no. 1, 30–35.

- [17] Qian Huang and B. Dom, *Quantitative methods of evaluating image segmentation*, Proceedings of International Conference on Image Processing (Washington, DC, USA), 1995, pp. 23–26.
- [18] Anis Ben Ishak, *Choosing parameters for rényi and tsallis entropies within a two-dimensional multilevel image segmentation framework*, Physica A: Statistical Mechanics and its Applications **466** (2017), 521–536.
- [19] ———, *A two-dimensional multilevel thresholding method for image segmentation*, Applied Soft Computing **52** (2017), 306–322.
- [20] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, *A new method for gray-level picture thresholding using the entropy of the histogram*, Computer Vision, Graphics, and Image Processing **29** (1985), 273–285.
- [21] James Kennedy and Russell C. Eberhart, *Particle swarm optimization*, Proceedings of the 1995 IEEE International Conference on Neural Networks (Perth, Australia, IEEE Service Center, Piscataway, NJ), vol. 4, 1995, pp. 1942–1948.
- [22] R. Sukesh Kumar, Abhisek Verma, and Jasprit Singh, *Color image segmentation and multi-level thresholding by maximization of conditional entropy*, International Journal of Computer, Electrical, Automation, Control and Information Engineering **1** (2007), 1633–1641.
- [23] E. Kurugollu, B. Sankur, and A. E. Harmanci, *Color image segmentation using histogram multithresholding and fusion*, Image & Vision Computing **19** (2001), 915–928.
- [24] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik, *A database of human segmented natural images and its applications to evaluating segmentation algorithms and measuring ecological statistic*, International Conference on Computer Vision, July 2001, pp. 1–8.

- [25] Marina Meila, *Comparing clusterings-an axiomatic view*, Proceedings of 22nd International Conference on Machine Learning (Bonn, Germany), 2005, pp. 577–584.
- [26] William M. Rand, *Objective criteria for the evaluation of clustering methods*, Journal of the American Statistical Association **66** (1967), no. 336, 846–850.
- [27] Alfréd Rényi, *On measures of entropy and information*, Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability (University of California Press, Berkeley), vol. 1, 1961, pp. 547–561.
- [28] Samy Sadek and Ayoub Al-Hamadi, *Entropic image segmentation: A fuzzy approach based on tsallis entropy*, International Journal of Computer Vision and Signal Processing **5** (2015), no. 1, 1–7.
- [29] Prasanna Sahoo, Carrye Wilkins, and Jerry Yeager, *Threshold selection using renyi's entropy*, Pattern Recognition **30** (1997), no. 1, 71–84.
- [30] Prasanna K. Sahoo and Gurdial Arora, *A thresholding method based on two dimensional rényi's entropy*, Pattern Recognition **37** (2004), 1149–1161.
- [31] ———, *Image thresholding using two dimensional tsallis-havrda-charvát entropy*, Pattern Recognition Letters **27** (2006), 520–528.
- [32] Prasanna K. Sahoo, Dick W. Slaaf, and Thomas A. Albert, *Threshold selection using a minimal histogram entropy difference*, Society of Photo-Optical Instrumentation Engineers **36** (1997), no. 7, 1976–1981.
- [33] Prasanna K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen, *A survey of the thresholding techniques*, Computer Vision, Graphics, and Image Processing **41** (1988), 233–260.

- [34] Soham Sarkar and Swagatam Das, *Multilevel image thresholding based on 2d histogram and maximum tsalli entropy - a differential evolution approach*, Transactions on Image Processing **22** (2013), no. 12, 4788–4797.
- [35] P. D. Sathya and R. Kayalvizh, *Pso-based tsallis thresholding selection procedure for image segmentation*, International Journal of Computer Applications **5** (2010), no. 4, 0975–8887.
- [36] C. E. Shannon, *A mathematical theory of communication*, The Bell System Technical Journal **27** (1948), 379–423, 623–656.
- [37] A. C. Sparavigna, *Tsallis entropy in bi-level and multi-level image thresholding*, International Journal of Sciences **4** (2015), no. 1, 40–49.
- [38] Khang Siang Tan and Nor Ashidi Mat Isa, *Color image segmentation using histogram thresholding - fuzzy c-means hybrid approach*, Pattern Recognition **44** (2011), 1–15.
- [39] C. Tsallis, *Possible generalization of boltzmann-gibbs statistics*, Journal of Statistical Physics **52** (1988), no. 1/2, 479–487.
- [40] Ranjith Unnikrishnan and Martial Hebert, *Measures of similarity*, 7th IEE workshop on Applications of Computer Vision, 2005, pp. 394–400.
- [41] David H. Wolpert and William G. Macready, *No free lunch theorems for optimization*, IEEE Transactions on evolutionary computation **1** (1997), no. 1, 67–82.
- [42] Yang Xiao, Zhigou Cao, and Sheng Zhong, *New entropic thresholding approach using gray-level spatial correlation histogram*, Optical Engineering **49** (2010), no. 12, 1–13.

- [43] Yang Xiao, Zhiguo Cao, and Junsong Yuan, *Entropic image thresholding based on glgm histogram*, Pattern Recognition Letters **40** (2014), 47–55.
- [44] Allen Y. Yang, John Wright, Yi Ma, and S. Shankar Sastry, *Unsupervised segmentation of natural images via lossy data compression*, Computer Vision and Image Understanding **110** (2008), 212–225.
- [45] Chen-Kuei Yang and Wen-Hsiang Tsai, *Reduction of color space dimensionality by moment - preserving thresholding and its application for edge detection in color images*, Pattern Recognition Letters **17** (1996), 481–490.
- [46] Adiljan Yimit, Yoshihiro Hagihara, Tasuku Miyoshi, and Yukari Hagihara, *2-d direction histogram based entropic thresholding*, Neurocomputing **120** (2013), 287–297.
- [47] Yudong Zhang and Lenan Wu, *Optimal multi-level thresholding based on maximum tsallis entropy via an artificial bee colony approach*, Entropy **13** (2011), no. 4, 841–859.
- [48] Xiaoli Zhao, Matthew Turk, Wei Li, Kuo chin Lien, and Guozhong Wang, *A multilevel image thresholding segmentation algorithm based on two-dimensional k-l divergence and modified particle swarm optimization*, Applied Soft Computing **48** (2016), no. 1, 151–159.
- [49] Xiulian Zheng, Hong Ye, and Yinggan Tang, *Image bi-level thresholding based on gray level-local variance histogram*, Entropy **19** (2017), no. 5, 1–8.

CURRICULUM VITAE

Surina

EDUCATION

University of Louisville, Louisville, KY

Ph.D. Applied & Industrial Mathematics

Expected August 2018

University of Louisville, Louisville, KY

M.A. Mathematics

May 2014

Beihang University, Beijing, China

M.S. Mathematics

January 2012

Inner Mongolia University, Inner Mongolia, China

B.S. Information and Computer Science

July 2007

EXPERIENCE

University of Louisville, Louisville KY

2012 - 2018

Graduate Teaching Assistant in Mathematics

Primary Instructor for the following courses:

- MATH 205: Calculus I (Spring 2018)
A college-level introduction to the calculus of single-variable functions. Cover functions, limits, continuity, differentiation, integration, and applications.
- MATH 111: Supplemented College Algebra (Summer 2014, Summer 2017)
A general education course for students struggling in mathematics. Reduced class size and extra time are giving to students in this class to help them learn the material. Cover algebraic and rational expressions and factoring; polynomial, rational, exponential, and logarithmic functions, with applications.

- MATH 180: Elements of Calculus (Summer 2015)
A calculus class required for business majors and majors in the life sciences.
Cover limits, derivatives, and integrals; integrals with applications.

Teaching Assistant for the following courses:

- MATH 105: Contemporary Mathematics
- MATH 109: Elementary Statistics
- MATH 111: College Algebra
- MATH 180: Elements of Calculus

Beihang University, Beijing, China

2009 - 2011

Teaching Assistant in Mathematics for the Following Courses:

- Calculus
- Linear Algebra

PUBLICATIONS

Peer-Reviewed Journal Articles

- Surina, Jitao Wu, 2011. “Region-Based Active Contour Model Improving the Signed Pressure Force Function”, *Journal of Image and Graphics, Chinses Academy of Sciences* 16 (12): 2169-2174

Revised and Resubmitted Journal Articles

- Surina Borjigin, Prasanna K. Sahoo, 2018. “Color Image Segmentation based on multi-level Tsallis-Havrda-Charvát entropy and 2D histogram using PSO Algorithms”, *Pattern Recognition*

PRESENTATIONS

- “An Information Theoretic Thresholding Technique Based on Particle Swarm Optimization”, Candidacy Exam, University of Louisville, Louisville, KY, April 2017
- “An Information Theoretic Thresholding Technique Based on Particle Swarm Optimization”, Mathematics Association of America of 100th Kentucky Sectional Meeting, Berea College, Berea, KY, March 2017

- “A Multi-level Color Image Thresholding Based on Two-Dimensional CHT Entropy and Enhanced Particle Swarm Optimization”, 36th Annual Mathematics Symposium, Western Kentucky University, Bowling Green, KY, November 2016
- “Color Image Segmentation Using Information Theory and 2D Histogram of the Image”, Mathematics Association of America of Indiana Sectional Meeting, Purdue University, West Lafayette, IN, October 2016
- “Variational Mumford-Shah Model”, Differential Equations and Applied Math Seminar, University of Louisville, Louisville, KY, April 2016
- “Active Contour Models: Snakes”, Differential Equations and Application Seminar, University of Louisville, Louisville, KY, April 2015

CONFERENCE ATTENDANCE

- Mathematics Association of America of 100th Kentucky Sectional Meeting, Berea College, Berea, KY, March 2017
- 36th Annual Mathematics Symposium, Western Kentucky University, Bowling Green, KY, November 2016
- Mathematics Association of America of Indiana Sectional Meeting, Purdue University, West Lafayette, IN, October 2016
- American Mathematical Society Sectional Meeting, University of Louisville, Louisville, KY, October 2013

GRANTS/ACHIEVEMENTS/AWARDS

- Graduate Student Council Travel Funding (\$350),
School of Interdisciplinary and Graduate Studies,
University of Louisville **2017**
- SOA/CAS Exam P/1 **2015**
- Graduate Dean’s Citation, University of Louisville **2014**
- Guanghai Scholarship, Beihang University **2011**
- Outstanding Master’s Graduate Student, Beihang University **2011**

- Guanghai Scholarship, Inner Mongolia University 2006
- Scholarship as a full-time math major,
Inner Mongolia University 2003 - 2007

SKILLS

- Matlab, Mathematic, Maple, Latex, Visual C/C++, R, Microsoft Word, Microsoft Excel, Microsoft PowerPoint

LANGUAGES

- Mongolian, Mandarin Chinese, English

REFERENCES

- | | | |
|---------------------|--------------|------------------------------|
| • Dr. Steve Xu | 502.852.1524 | ysxu0001@louisville.edu |
| • Dr. Thomas Riedel | 502.852.6826 | thomas.riedel@louisville.edu |
| • Dr. Changbing Hu | 502.852.6219 | changbing.hu@louisville.edu |