

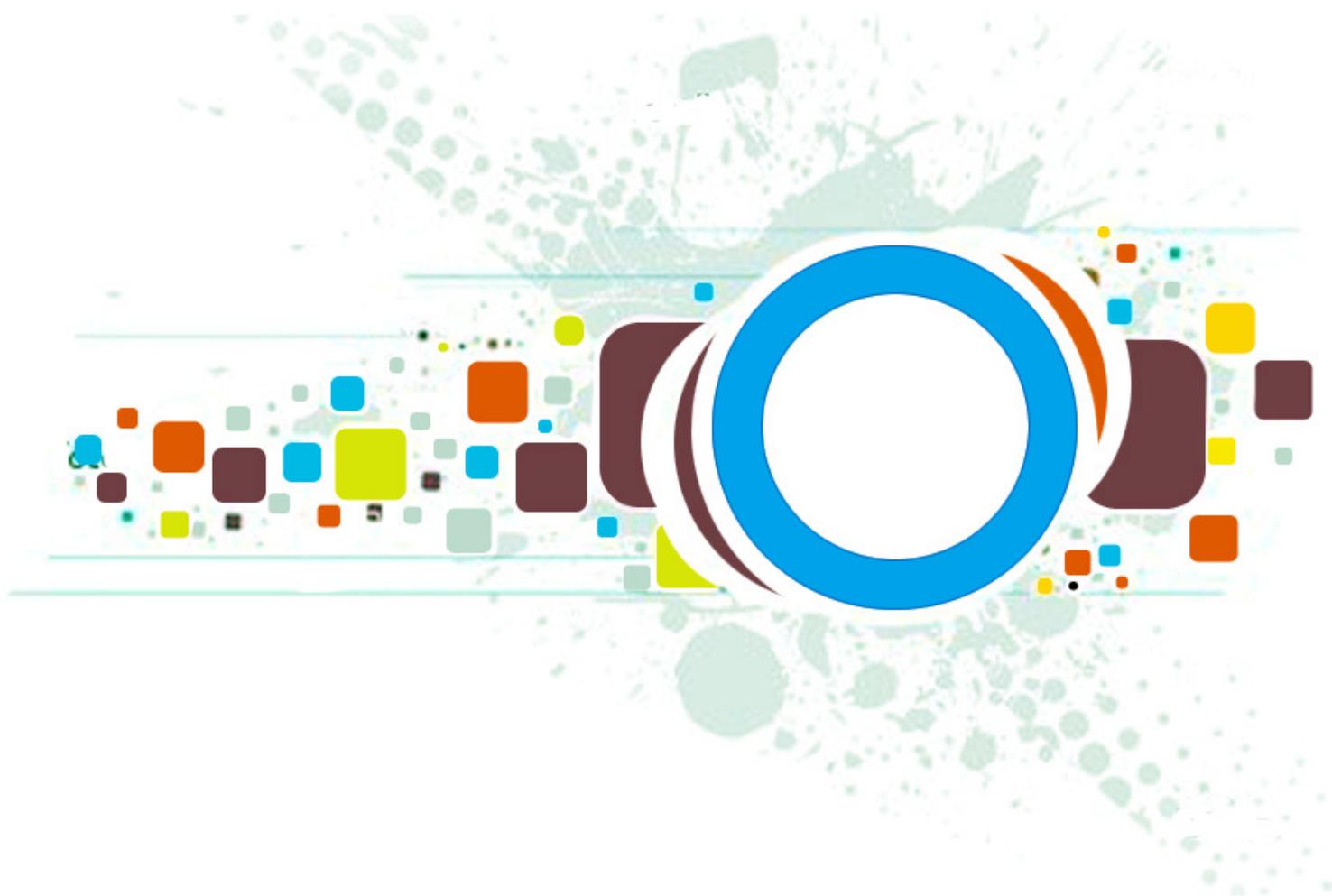
Volume 5 ▪ Issue 5 ▪ December 2011

Editor-in-Chief
Professor Hu, Yu-Chen

INTERNATIONAL JOURNAL OF
IMAGE PROCESSING (IJIP)

ISSN : 1985-2304

Publication Frequency: 6 Issues Per Year



CSC PUBLISHERS
<http://www.cscjournals.org>

INTERNATIONAL JOURNAL OF IMAGE PROCESSING (IJIP)

VOLUME 5, ISSUE 5, 2011

**EDITED BY
DR. NABEEL TAHIR**

ISSN (Online): 1985-2304

International Journal of Image Processing (IJIP) is published both in traditional paper form and in Internet. This journal is published at the website <http://www.cscjournals.org>, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJIP Journal is a part of CSC Publishers

Computer Science Journals

<http://www.cscjournals.org>

INTERNATIONAL JOURNAL OF IMAGE PROCESSING (IJIP)

Book: Volume 5, Issue 5, December 2011

Publishing Date: 15-12- 2011

ISSN (Online): 1985-2304

This work is subjected to copyright. All rights are reserved whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication of parts thereof is permitted only under the provision of the copyright law 1965, in its current version, and permission of use must always be obtained from CSC Publishers.

IJIP Journal is a part of CSC Publishers

<http://www.cscjournals.org>

© IJIP Journal

Published in Malaysia

Typesetting: Camera-ready by author, data conversion by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2011

EDITORIAL PREFACE

The International Journal of Image Processing (IJIP) is an effective medium for interchange of high quality theoretical and applied research in the Image Processing domain from theoretical research to application development. This is the forth issue of volume four of IJIP. The Journal is published bi-monthly, with papers being peer reviewed to high international standards. IJIP emphasizes on efficient and effective image technologies, and provides a central for a deeper understanding in the discipline by encouraging the quantitative comparison and performance evaluation of the emerging components of image processing. IJIP comprehensively cover the system, processing and application aspects of image processing. Some of the important topics are architecture of imaging and vision systems, chemical and spectral sensitization, coding and transmission, generation and display, image processing: coding analysis and recognition, photopolymers, visual inspection etc.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 5, 2011, IJIP appears in more focused issues. Besides normal publications, IJIP intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

IJIP give an opportunity to scientists, researchers, engineers and vendors from different disciplines of image processing to share the ideas, identify problems, investigate relevant issues, share common interests, explore new approaches, and initiate possible collaborative research and system development. This journal is helpful for the researchers and R&D engineers, scientists all those persons who are involve in image processing in any shape.

Highly professional scholars give their efforts, valuable time, expertise and motivation to IJIP as Editorial board members. All submissions are evaluated by the International Editorial Board. The International Editorial Board ensures that significant developments in image processing from around the world are reflected in the IJIP publications.

IJIP editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJIP. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJIP provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members

International Journal of Image Processing (IJIP)

EDITORIAL BOARD

EDITOR-in-CHIEF (EiC)

Professor Hu, Yu-Chen
Providence University (Taiwan)

ASSOCIATE EDITORS (AEiCs)

Professor. Khan M. Iftakharuddin
University of Memphis
United States of America

Assistant Professor M. Emre Celebi
Louisiana State University in Shreveport
United States of America

Assistant Professor Yufang Tracy Bao
Fayetteville State University
United States of America

Professor. Ryszard S. Choras
University of Technology & Life Sciences
Poland

Dr. Huiyu Zhou
Queen's University Belfast
United Kingdom

Professor Yen-Wei Chen
Ritsumeikan University
Japan

Associate Professor Tao Gao
Tianjin University
China

EDITORIAL BOARD MEMBERS (EBMs)

Dr. C. Saravanan
National Institute of Technology, Durgapur West Bengal
India

Dr. Ghassan Adnan Hamid Al-Kindi
Sohar University
Oman

Dr. Cho Siu Yeung David
Nanyang Technological University
Singapore

Dr. E. Sreenivasa Reddy

Vasireddy Venkatadri Institute of Technology
India

Dr. Khalid Mohamed Hosny

Zagazig University
Egypt

Dr. Chin-Feng Lee

Chaoyang University of Technology
Taiwan

Professor Santhosh.P.Mathew

Mahatma Gandhi University
India

Dr Hong (Vicky) Zhao

Univ. of Alberta
Canada

Professor Yongping Zhang

Ningbo University of Technology
China

Assistant Professor Humaira Nisar

University Tunku Abdul Rahman
Malaysia

Dr M.Munir Ahamed Rabbani

Qassim University
India

Dr Yanhui Guo

University of Michigan
United States of America

TABLE OF CONTENTS

Volume 5, Issue 5, December 2011

Pages

- 521 - 541 Analyses of the Watershed Transform
Ramzi Mahmoudi, Mohamed AKIL
- 542 - 551 The Framework of Image Recognition based on Modified Freeman Chain Code
Haswadi Hasan, Habibollah Haron, Siti Zaiton Mohd Hashim
- 552 - 563 A Wavelet Based Automatic Segmentation of Brain Tumor in CT Images Using Optimal
Statistical Texture Features
A.Padma, R. Sukanesh
- 564 – 579 Atmospheric Correction of Remotely Sensed Images in Spatial and Transform Domain
Densities Environments
Priti Tyagi, Udhav Bhosle
- 580 – 589 A Research on Guided Thinning Algorithm and Its Implementation by Using C#
Jia Liang
- 590 - 598 Detection of Diseases on Cotton Leaves and its Possible Diagnosis
Viraj Ashokrao Gulhane, Ajay A. Gurjar
- 599 – 609 Histogram Equalization with Range Offset for Brightness Preserved Image Enhancement
Haidi Ibrahim
- 610 – 622 System of “Analysis of Intersections Paths” for Signature Recognition
Farhad Shamsfakhr

623 - 634 Recognition of Farsi Handwritten Numbers Using the Fuzzy Method
Mansoreh Sharizfاده, Shahpour Alirezaee

Analyses of the Watershed Transform

Ramzi MAHMOUDI

*Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Equipe A3SI
ESIEE Paris - Cité Descartes, BP99, 93162 Noisy Le Grand, France*

mahmoudr@esiee.fr

Mohamed AKIL

*Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Equipe A3SI
ESIEE Paris - Cité Descartes, BP99, 93162 Noisy Le Grand, France*

akilm@esiee.fr

Abstract

In the framework of mathematical morphology, watershed transform (WT) represents a key step in image segmentation procedure. In this paper, we present a thorough analysis of some existing watershed approaches in the discrete case: WT based on flooding, WT based on path-cost minimization, watershed based on topology preservation, WT based on local condition and WT based on minimum spanning forest. For each approach, we present detailed description of processing procedure followed by mathematical foundations and algorithm of reference. Recent publications based on some approaches are also presented and discussed. Our study concludes with a classification of different watershed transform algorithms according to solution uniqueness, topology preservation, prerequisites minima computing and linearity.

Keywords: Watershed Transform, Flooding, Path-cost Minimization, Topology Preservation, Local Condition, Minimum Spanning Forest.

1. INTRODUCTION

The watershed concept began with Maxwell [1] who introduces the theory behind representing physical characteristics of a land by means of lines drawn on a map. He highlights relationships between the numbers of hills, dales and passes which can co-exist on a surface. Subsequently, through the work of Beucher and al. [2], watershed transform was introduced to image segmentation and nowadays it represents one of the basic foundations of image processing [3].

In this framework, the most simplified description of the watershed approach is to consider a grayscale image as a topographic surface: the gray level of a pixel becomes the elevation of a point, the basins and valleys correspond to dark areas, whereas the mountains and crest lines correspond to the light areas. If topographic relief is flooded by water, watersheds will be the divide lines of the attraction's domains of rain falling over the region [4] or sources of water springing from reliefs' peaks. Another synopsis has shown consistency is that topographic surface is immersed in a lake with holes pierced in local minima. Catchment basins will fill up with water starting at these local minima, and, at points where water coming from different basins would meet, dams are built. As a result, the topographic surface is partitioned into different basins separated by dams, called watershed lines. Figure 1 gives a very symbolic description of the mentioned approach. In fact, it shows trends in the use of watershed transform for image processing.

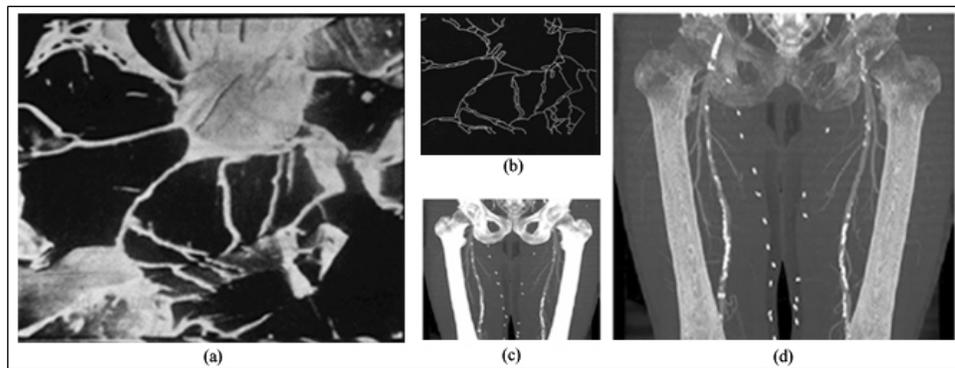


FIGURE 1: (a) Cleavage fractures in steel, (b) contour of (a) obtained truth watershed definition introduced by Beusher and al. [2] in 1979, (c) Maximum intensity projection of original human lower limb (d) Bone tissue removed using mask extended with 3D watershed transform introduced by Straka and al.[5] in2003.

Despite its simplicity, this concept has been formalized in different ways giving rise to several definitions of watershed transform. In the discrete case, which is our main interest in this paper, this problem is amplified since there is no unique definition of the path that the drop of water would follow. This led to a multitude of algorithm to compute watershed transform. Some of these algorithms don't even meet associated watershed definition. We also note that some definitions take the form of algorithm specification which makes the distinction between algorithm specification and implementation very complicated. This problem in literature has been partially resolved in Roerdink and al. [6] ten years ago. Actually authors presented a critical review of several definitions of the watershed transform and the associated sequential algorithms. Even they discuss various issues which often cause confusion in the literature; they don't go further in the classification or comparison of different approaches. They instead focus on parallelization aspect. In other more recent publications, authors tentatively drawn a comparison chart of some watershed transform definition to serve their end goals: showing the relationships that may exist between some discrete definition of watershed [7] or showing that most classical watershed algorithm do not allow the retrieval of some important topological features of the image [8].

The purpose of this paper is to introduce an intensive study of all existing watershed transform in the discrete case: WT based on flooding, WT based on path-cost minimization, watershed based on topology preservation, WT based on local condition and WT based on minimum spanning forest. Indeed, for each approach, we start by giving informal definition, then we present processing procedure followed by mathematical foundations and the algorithm of reference. Recent publications based on some approach are also presented and discussed. Our study concludes with a classification of different algorithms according to criteria of recursion, complexity, basins computing and topology preservation. This paper is organized as follows: in section 2, different approaches to compute watershed are presented. In section 3, we draw a comparison between the various presented algorithms. Finally, we conclude with summary and future work in section 4.

2. WATERSHED TRANSFORMATION

In this section, we propose a comparative study of different approaches to compute the watershed transform in the discrete case. The goal is to identify most suited watershed transform for parallel processing. This study can also be seen as an update of Roerdink work [6] presented ten years ago. Indeed, for each approach, we present processing procedure, mathematical foundations and sequential algorithm. Recent publications based on some approach are also presented and discussed.

2.1 Watershed Based on Flooding

Based on flooding paradigm [9,10,11], the intuitive idea underlying this method comes from geography. Since grayscale image can be seen as topographic surface, the intensity of a pixel can be considered as the altitude of a point. Now, let immerge this surface in still water, with holes created in local minima. Water fills up basins starting at these local minima. As described in algorithm 1, the filling of basins is an iterative process that involves gradually raising the water level from Alt_{min} to Alt_{max} . Algorithm must, for each iteration, fill existing basins (extension regions) and possibly create new basins (new regions). We denote by LR the region list. Dams will be built where waters coming from different basins meet.

Algorithm 1: Flooding watershed process	
1.	for level from Alt_{min} to Alt_{max}
2.	// Action 1 : Extend existing region
3.	foreach ($R \in L_R$) do Growing [R] until level Alt ;
4.	end_for
5.	// Action 2 : Create new region
6.	foreach (Pixel P \in level)
7.	if (Pixel P is not associated to any region R) then
8.	Create new region [R] in L_R ;
9.	Add Pixel P to region [R];
10.	Growing [R] until level ;

For mathematical formulation of the mentioned process, let $f : D \rightarrow \mathbb{N}$ be a digital grey value image, with Alt_{min} and Alt_{max} the minimum and the maximum value of. The threshold set of at level Alt is: $T_{Alt_{min}} = \{p \in D / f(p) \leq Alt\}$ (2.1)

It define a recursion with the gray level Alt increasing from Alt_{min} to Alt_{max} , the basin associated with the minima of f are successively expanded. Let X_{Alt} denote the union of the set of basins computed at level Alt . A connected component of the threshold set T_{Alt+1} at level $Alt+1$ can be either a new minimum or an extension of the basin in X_{Alt} . The geodesic influence zone (IZ) of X_{Alt} within T_{Alt+1} can be computed resulting in an update X_{Alt+1} . Let MIN_h denote the union of all regional minima at altitude Alt . Then we can introduce the following definition:

Definition 2.1 (Flooding watershed) (2.2)

$$\begin{cases} X_{Alt_{min}} = \{p \in D / f(p) = Alt_{min}\} = T_{Alt_{min}} \\ X_{Alt+1} = MIN_{Alt+1} \cup IZ_{T_{ALT+1}}(X_{Alt}) \end{cases}$$

The watershed $Wshed(f)$ of f is the complement of $X_{Alt_{max}}$ in D :

$$Wshed(f) = D / X_{Alt_{max}} \tag{2.3}$$

Vincent and Soille [12] presented an original and efficient implementation (algorithm 2) of the flooding watershed. This implementation uses FIFO queue and it needs two steps:

- (1) Sorting pixels in increasing order of grey values (Alt_{min} ; Alt_{max})
- (2) Flooding process: All nodes with grey level h are first given the initial label. Then those nodes that have labeled neighbors from the previous iteration are inserted in the queue, and from these pixels geodesic influence zones are propagated inside the set of initial pixels.

In their study [6] Roerdink and Meijster have removed two points of inconsistency in the algorithm's recursion. (i) Only pixels with grey value h are masked for flooding (line 13), instead of all non-basin pixels of $(level \leq h)$, as the definition (2.2) would require. This explains why labels of watershed-pixels (line 15) are also propagated with labels of catchment basins. (ii) If a pixel is adjacent to two different basins; it is initially labeled 'wshed'. But it is allowed to be overwritten at the current grey level by another neighbor's label, if that neighbor is part of a basin (lines 35-36).

They also propose some modification to implement the recursion (2.3) exactly. In line 13, all pixels with $im[p] \leq h$ have to be masked, the queue has to be initialized with basin pixels only (drop the disjunct $lab[q] = wshed$ in line 15), the resetting of distances (line 50) has to be done in line 14, and the propagation rules in lines 32-47 have to be slightly changed.

Algorithm 2 : Flooding watershed [Vincent & Soille]	
Data :	Digital grey scale image $G=(D,E,im)$
Result :	Labelled watershed image lab on D
1.	<code>#define INIT -1 //initial value of lab image</code>
2.	<code>#define MASK -2 //initial value of each level</code>
3.	<code>#define WSHED 0 //label of the watershed pixels</code>
4.	<code>#define FICTITIOUS (-1,1) //fictitious pixel $\notin D$</code>
5.	<code>curlab \leftarrow 0 //curlab is the current label</code>
6.	<code>fifo_init(queue)</code>
7.	<code>for all (P\inD) do</code>
8.	<code> lab[p] \leftarrow INIT;</code>
9.	<code> dist[p] \leftarrow 0; //dist is a work image of distances</code>
10.	<code>end_for</code>
11.	<code>SORT pixels in increasing order of grey values (h_{min} , h_{max})</code>
	<i>// starting flooding process</i>
12.	<code>for h = h_{min} to h_{max} do //geodesic SKIZ of level h-1 inside level h</code>
13.	<code> for all (P\inD) with im[p]=h do // mask all pixels at level h</code>
	<i>//these are directly accessible because of the sorting step</i>
14.	<code> lab[p] \leftarrow MASK;</code>
15.	<code> if (p has a neighbour q) with ((lab[p] > 0 or lab[q]=WSHED)) then</code>
	<i>//initialize queue with neighbours at level h of current basins or watersheds</i>
16.	<code> dist[p] \leftarrow 1;</code>
17.	<code> fifo_add(q,queue);</code>
18.	<code> end_if</code>
19.	<code> end_for</code>
20.	<code> curdist \leftarrow 1;</code>
21.	<code> fifo_add(FICTITIOUS,queue);</code>
22.	<code>loop //extend basins</code>
23.	<code> p \leftarrow fifo_remove(queue);</code>
24.	<code> if (p = FICTITIOUS) then</code>
25.	<code> if (fifo_empty(queue)) then</code>
26.	<code> BREAK ;</code>
27.	<code> else (fifo_add(FICTITIOUS,queue) ;</code>
28.	<code> curdist \leftarrow curdis + 1;</code>
29.	<code> p \leftarrow fifo_remove(queue);</code>
30.	<code> end_if</code>
31.	<code> end_if</code>
32.	<code> for all (q\inN_G(p)) do //labelling p by inspecting neighbours</code>
33.	<code> if (dist[q] < curdist) and (lab[q]>0 or lab[q]=WSHED) then</code>

```

34.      //q belongs to an existing basin or to watersheds
35.      if (lab[q]>0) then
36.          if ((lab[p]=MASK) or (lab[p]=WSHED)) then
37.              lab[p] ← lab[q];
38.          else if lab[p] ≠ lab[q] then
39.              lab[p] ← WSHED;
40.          end_if
41.      else if (lab[p]=MASK) then
42.          lab[p] ← WSHED;
43.      end_if
44.      else if ((lab[q]=MASK) and (dist[q]= 0)) then //q is plateau pixel
45.          dist[q] ← curdis + 1;
46.          p ← fifo_add(q, queue);
47.      end_if
48.  end_for
49.  end_loop
50.  //detect and process new minima at level h
51.  for all (p∈D) with (im[p]=h) do
52.      dist[p] ← 0; // reset distance to zero
53.      if (lab[p]=MASK) then // p is inside a new minimum
54.          curlab ← curlab + 1; //create new label
55.          fifo_add(p, queue);
56.          lab[p] ← curlab;
57.          while not (fifo_remove(queue)) do
58.              q ← fifo_remove(queue);
59.              for all (r∈NG(q)) do // inspect neighbours of q
60.                  if (lab[r]=MASK) then
61.                      fifo_add(r, queue);
62.                      lab[r] ← curlab;

```

From the introduction about immersion simulation above, we can see that the level-by-level method during the flood procedure is uniform. Unfortunately, this can cause over-segmentation in several cases. Based on this original simulating immersion, Shengcai and Lixu [13] propose, in 2005, a novel implementation using multi-degree immersion. To our knowledge, it is the last update that can be found in the literature, the proposed implementation resists to over-segmentation problem effectively. It starts by redefining the threshold set of f at level **Alt**. Instead of the original formula (2.1), they propose the following one:

$$T_{Alt} = \{p \in D / f(p) - Diff(p) \leq Alt\} \tag{2.4}$$

With $Diff(p)$ refers to the immersion level when the flooding reaches pixel p . Segmentation results are sensitive to this function. In fact, if $Diff(p) = 0$, (2.4) can be seen as a special case of (2.1). Other extreme case, if $Diff(p)$ reaches its maximum values, all pixels susceptible to be dumped, will be. According to the user requirement, $Diff(p)$ can be even a constant function or a function computed according to the local characteristic of p . In their paper, Shengcai and Lixu define it as introduced in (2.5) with $Neighbor(p)$ is the set of all p neighbors. And $conn$ refers to the predefined connectivity: $\{4,8\}$.

$$Diff(p) = \sum_{q \in Neighbor(p)} \frac{|f(p) - f(q)|}{conn} \tag{2.5}$$

Obtained results through two implementations of the original and the multi-degree watershed shows that multi-degree immersion method resists the over-segmentation problem effectively. Indeed, the number of detected region in an image brain (181*217*181 voxel volume), decreases from 10991, using the old method, to only 35 using the new method. Computation time and consumed memory size are practically the same. More information about implementation can be found in [13].

2.2 Watershed Based on Path-cost Minimization

In this class, there are two possible approaches. The first one associates a pixel to a catchment basin when the topographic distance is strictly minimal to the respective regional minimum. While the second one builds a forest of minimum-path trees, each tree representing a basin. In the following we start by introducing watershed by topographic distance [14] before moving to the watershed by image foresting transform [16,17].

Algorithm 3 : Watershed by topographic distance process

1. **Foreach** (marked area $\in S_M$)
2. insert pixels into priority queue Q ;
3. **end_for**
4. **While** ($Q \neq \phi$)
5. Ex_p = extract pixels with highest priority level;
6. **if** (neighbors of $p \in Ex_p$ have the same label Lab) **then**
7. $Lab_p = Lab$;
8. $Q =$ all non-marked neighbors
9. **end_if**
10. **end_while**

Based on the drop of water principle, the intuitive idea behind topographic watershed approach in the steepest descent path principal [14, 15]. A drop of water falling on a topographic relief flows down, as "quickly" as possible, until it reaches a regional minimum. Let $f : D \rightarrow \mathbb{N}$ be a digital grey value image. Let S_M be the set of markers, pixels where the flooding shall start, are chosen. Each is given a different a label Lab .

Topographic watershed process can be described by algorithm 3. Let us note that priority level when inserting neighbors (line 2) corresponds to the gray level of the pixel. In line 6, only neighbours that have already been labelled are compared. Finally, only neighbors (in line 8) that are not yet in the priority queue are pushed into the priority queue. The watershed lines set are the complement of the set of labeled points.

For mathematical formulation of the mentioned process, we follow here the presentation in [6] which is based on [14]. For the sake of simplicity, we restrict our self to the minimal set of notion that will be useful for our propos. We start by introducing the topographic distance. Let us consider $N_G(p)$ as the set of neighbors of pixel p , and $d(p,q)$ as the distance associated to edge (p,q) . Then the lower slope $LS(p)$ of f at a pixel p can be defined as follow:

$$LS(p) = \text{MAX}_{q \in N_G(p) \cup \{p\}} \left(\frac{f(p) - f(q)}{d(p, q)} \right) \tag{2.6}$$

And the cost for walking from pixel a p to a neighboring pixel q can be defined as:

$$\text{cos}(p, q) = \begin{cases} LS(p).d(p, q) / f(p) > f(q) \\ LS(q).d(p, q) / f(p) < f(q) \\ \frac{1}{2}(LS(p) + LS(q)).d(p, q) / f(p) = f(q) \end{cases} \tag{2.7}$$

The topographical distance between p and q is the minimum of the topographical distances $T_f^\pi(p, q)$ along all paths between p and q : $T_f(p, q) = \text{MIN}_{\pi \in [p \rightarrow q]} T_f^\pi(p, q)$ (2.8)

We recall that the topographical distance along a general path $\pi = (p_0, \dots, q_l)$ is defined as

$$T_f^\pi(p, q) = \sum_i^{l-1} d(p_i, p_{i+1}) \text{cos}t(p_i, p_{i+1}) \tag{2.9}$$

Finally we can define the topographic watershed for a grey value image f , with f^* the lower completion of f . Each pixel which is not in a minimum has a neighbor of lower grey value ($f^* = f_{LC}$).

Let $(m_i)_{i \in I}$ be the collection of minima of f . The basin $CB(m_i)$ of f corresponding to a minimum $(m_i)_{i \in I}$ is defined as a basin of the lower completion of f :

$$CB(m_i) = \left\{ p \in D, j \in I \setminus \{i\} : f^*(m_i) + T_{f^*}(p, m_i) < f^*(m_j) + T_{f^*}(p, m_j) \right\} \tag{2.10}$$

And the watershed is the set of points which do not belong to any catchment basin:

$$\text{Wshed}(f) = D \cap (\cup_{i \in I} CB(m_i))^c \tag{2.11}$$

Several shortest paths algorithms for the watershed transform with respect to topographical distance can be found in the literature but the reference algorithm is that of Fernand Meyer. In the following we present variant of Meyer algorithm with integrate the lower slope of the input image as introduced Roerdink and al. [6].

Algorithm 4 : Watershed by topographical distance [Meyer]

Data : Lower complete image im on a digital grey scale image $G=(D,E)$ with cost
Result : Labelled watershed image lab on D

1. #define WSHED 0 //label of the watershed pixels
2. //Uses distance image $dist$. On output, $div[v]=im[v]$, for all $v \in D$
3. **for all** ($v \in D$) **do** //Initialize
4. $lab[v] \leftarrow 0$;
5. $dist[v] \leftarrow \infty$;
6. **end for**
7. **for all** (local minima m_i) **do**

```

8.  for all ( $v \in m_i$ ) do
9.     $lab[v] \leftarrow i$ ;
10.    $dist[v] \leftarrow im[v]$ ; //Initialize distance with the values of minima
11.  end for
12.  end for
13.   $stable \leftarrow true$ ; //stable is a Boolean variable
14.  repeat
15.   for all pixels  $u$  in forward raster scan order do
16.    propagate( $u$ )
17.   end for
18.   for all pixels  $u$  in backward raster scan order do
19.    propagate( $u$ )
20.   end for
21.  until  $stable$ 
22.  procedure propagate( $u$ )
23.   for all ( $v \in NG(u)$ ) in the future (w.r.t scan order) for  $u$  do
24.    if ( $dist[u] + cost[u, v] < dist[v]$ ) then
25.       $dist[v] \leftarrow dist[u] + cost(u, v)$ 
26.       $lab[v] \leftarrow lab[u]$ 
27.       $stable \leftarrow false$ 
28.    else if  $lab[v] \neq WSHED$  and ( $dist[u] + cost[u, v] = dist[v]$ ) then
29.      if ( $lab[v] \neq lab[u]$ ) then
30.         $lab[v] = WSHED$ 
31.         $stable \leftarrow false$ 

```

The second approach to compute a watershed based on path-cost minimization, as we introduced in the beginning, consists on building a forest of minimum-path trees where each tree represent a basin. This approach is described in the framework of image foresting transform [16]. The IFT defines a minimum-cost path forest in a graph, whose nodes are the image pixels and whose arcs are defined by an adjacency relation between pixels. The cost of a path in this graph is determined by a specific path-cost function, which usually depends on local image properties along the path, such as color or gradient. The roots of the forest are drawn from a given set of seed pixels. For suitable path-cost functions, the IFT assigns one minimum-cost path from the seed set to each pixel, in such a way that the union of those paths is an oriented forest, spanning the whole image. The IFT outputs three attributes for each pixel: its predecessor in the optimum path, the cost of that path, and the corresponding root. Returned solution is usually obtained in linear time and requires a variant of the Dijkstra [18], Moore [19] or Dial's shortest-path algorithm [20].

For mathematical formulation of the IFT-watershed, we start by defining some basic notions of image foresting transform as introduced in [16]. Actually, an image *imgln* can be seen as a pair (J, I) where J refers to a finite set of pixels and I refers to a mapping that assigns to each pixel $(p \in J)$, a pixel value $I(p)$ in some arbitrary value space. Distinct binary relation between pixels of J will define an adjacency relation A . Once the adjacency has been fixed, *imgln* can be interpreted as a directed graph, whose nodes are image pixels and whose arcs are pixel pairs in A .

Before moving to path cost definition, let's remember that a sequence of pixels $\pi = \langle t_1, t_2, \dots, t_k \rangle$ where $(t_i, t_{i+1}) \in A$ for $(1 < i < k - 1)$ constitute a path. In the following we will denote by $org(\pi)$ the

origin t_i of π and by $dst(\pi)$ the destination t_k of π . Now let assume given a function f that assigns to each path π a path cost $f(\pi)$, in some totally ordered set \mathcal{V} of cost values. We introduce the max-arc path-cost function f_{\max} that will be used later. Note that $h(t)$ and $w(s, t)$ are fixed.

$$\begin{aligned}
 f_{\max}(\langle t \rangle) &= h(t) \\
 f_{\max}(\pi \cdot \langle s, t \rangle) &= \max \{ f_{\max}(\pi), w(s, t) \}
 \end{aligned}
 \tag{2.12}$$

For IFT use, a specific function $f^S(\pi)$ can be defined since the search to paths start in a given set $(S \subset J)$ of seed pixels.

$$f^S(\pi) \begin{cases} f(\pi) & \text{if } (org(\pi) \in S) \\ +\infty & \text{otherwise} \end{cases}
 \tag{2.13}$$

Now, we can introduce the spanning forest concept. We remember that a predecessor map is a function P that assigns to each pixel $t \in J$ either some other pixel $s \in J$ or a distinctive marker $\mathbf{M} \notin J$. Thus, a spanning forest (**SF**) can be seen as a predecessor map which contains no cycles.

Definition 2.2 (Spanning forest) (2.14)

For any pixel $t \in J$, a spanning forest P defines a path $P^*(t)$ recursively as $\langle t \rangle$ if $P(t) = \mathbf{M}$ and $P^*(s) \cdot \langle s, t \rangle$ if $P(t) = s \neq \mathbf{M}$, we denote by $P^0(t)$ the initial pixel of $P^*(t)$.

Algorithm 5: IFT Algorithm [Falco and Al.]

Input: $Img = (J, I)$: Image, $A \subset J \times J$: Adjacency relation, f : path-cost function

Output: P : optimum path forest, α, β : two sets of pixels with $\alpha \cup \beta = J$.

1. Set $\alpha \leftarrow \{ \}$, $\beta \leftarrow J$ //Initialize
2. **for all** pixels $t \in J$ **do** //Initialize
3. Set $P(t) \leftarrow \mathbf{M}$
4. **end for**
5. **while** $\beta \neq \emptyset$ **do** //Compute
6. **remove** from β a pixel s such that $f(P^*(s))$ is minimum,
7. **add** s to α
8. **for** each pixel t such that $(s, t) \in A$
9. **if** $f(P^*(s) \cdot \langle s, t \rangle) < f(P^*(t))$ **then**
10. **Set** $P(t) \leftarrow s$;

Falcan and al. algorithm describes IFT computing. Its algorithm is based on Dijkstra's procedure for computing minimum-cost path from a single source in a graph and returns an optimum-path forest P for a seed-restricted cost function f^S or any pixel t with finite cost $f^S(P^*(t))$. Pixels will belong to a tree whose root is a seed pixel.

The IFT-watershed assumes that seeds pixel correspond to regional minima of the image or to markers that can be considered as imposed minima. The max-arc path-cost function f_{\max} is the same as (2.12). We remember that $h(t)$ is a fixed but arbitrary handicap cost for any paths starting at pixel t . We remember also that $w(s, t)$ is the weight of arc $\langle s, t \rangle \in A$, ideally, higher on the object boundaries and lower inside the objects.

There are two usual arc weight functions :

- a) $w_1(s, t) = |J(s) - J(t)|$, where $J(s)$ refers to the intensity of pixel of s . In that case IFT-Watershed is said by dissimilarity.
- b) $w_2(s, t) = G(t)$, where $G(t)$ is the morphological gradient of **imgIn** at pixel t . In that case IFT-Watershed is said on gradient.

Algorithm 6 : IFT-watershed from markers [Lotufo]

Data : I : input image, $wshed$: labeled marker image
Result : $wshed$: watershed catchment basins
Aux : C : cost map, initialized to infinity; $FIFO$: hierarchical FIFO queue

1. **for all pixels** $wshed(p) \neq 0$ **do** //Initialize
2. $C(p) \leftarrow I(p)$
3. Insert p in $FIFO$ with cost $C(p)$
4. **end for**
5. **while** $FIFO$ **do** //Propagation
6. $p \leftarrow$ remove from $FIFO$
7. **for each** $q \in N(p)$
8. **if** $(C(q) > \max \{C(p), I(q)\})$ **then**
9. $C(q) \leftarrow \max \{C(p), I(q)\}$;
10. Insert q in the $FIFO$ with cost $C(q)$;
11. $wshed(p) \leftarrow wshed(q)$

Lotufo and al. [21] introduce the IFT-watershed from markers, algorithm 6, which can be computed by a single IFT where the labeled markers are root pixels. Proposed algorithm use hierarchical FIFO queue (HFQ).

In this case, the root map can be replaced by the label map which corresponds to the catchment basins and the used path-cost function is given by:

$$f_m(\langle p_1, p_1, \dots, p_1 \rangle) = \begin{cases} \max\{I(p_1), I(p_2), \dots, I(p_n)\} & \text{If } p_1 \text{ is a marker pixel.} \\ +\infty & \text{Otherwise} \end{cases} \quad (2.15)$$

Where $p_{i+1} \in N(p_i)$ and $I(p_i)$ is the value of the pixel p_i in the image I .

2.3 Topological Watershed

The original concept behind topological watershed [22] is to define a “topological thinning” that transforms the image while preserving some topological properties, namely the number of connected components of each lower cross-section as we will explain in the following.

Before introducing the topological watershed process, algorithm 7, we define some basic notions. Some of these notions will be resumed in next paragraph for mathematical formulation. Let F be grayscale image and λ be a grey level, the lower cross-section \overline{F}_λ is the set composed of all the points having an altitude strictly lower than λ . A point x is said to be W -destructible for F if its altitude can be lowered by one without changing the number of connected components of \overline{F}_λ , with $k = F(x)$. A map G is called a W -thinning of F if it may be obtained from F by iteratively selecting a W -destructible point and lowering it by one. A topological watershed of F is a W -thinning of F which contains no W -destructible point. The major feature of this transform is to produce a grayscale image.

The following algorithms give a global description of the computing process. Note that this process is repeated on loop until no W -destructible point remains.

<p>Algorithm 7 : Topological watershed process</p> <ol style="list-style-type: none"> 1. For all p in E, check the number of connected components of the lower cross-section at the level of p which are adjacent to p. 2. Lower the value of p by one if this number is exactly one
--

For mathematical formulation, we follow description provided in [11]. We start by defining a simple point in a graph, in a sense which is adapted to the watershed, and then we extend this notion to weighted graphs through the use of lower sections [22].

Coupric and al. define a transform that acts directly on the grayscale image, by lowering some points in such a manner that the connectivity of each lower cross-section \overline{F}_λ is preserved. The regional minima of the result, which have been spread by this transform, can be interpreted as the catchment basins. The formal definition relies on a the following particular notion of simple point:

Definition 2.3 (Simple point) (2.16)

Let $G = (E, \Gamma)$ be a graph and let $X \subset E$.

The point $x \in X$ is simple (for \overline{X}) if the number of connected components of $\overline{X} \cup \{x\}$ equal to the number of connected components of \overline{X} . In other words, x is simple (for \overline{X}) if x is adjacent to exactly one connected component of \overline{X} .

Now, we can define more formerly destructible point, and the topological watershed:

Definition 2.4 (Topological watershed) (2.17)

Let $F \in F(E)$, $x \in E$ and $k = F(x)$. The point x is destructible (for F) if x is simple (for \overline{F}_k). We say that $W \in F(E)$ is a topological watershed of F if W may be derived from F by iteratively lowering destructible points by one until stability (that is, until all points of E being non-destructible for W).

Actually checking whether a point is w -destructible, or not, cannot be done locally if the only available information is the graph (E, Γ) and the function F and a point may also be lowered several times until it is no more w -destructible. Coupric and al. [23] propose a new algorithms

making possible to perform this test on all the vertices of a weighted graph in linear time, and also to check directly how low the W -destructible point may be lowered until it is no more w -destructible, thanks to the component tree which may be built in quasi-linear time. In the following, we introduce Couprie's functions to identify W -destructible point.

<p>Algorithm 8 : Function W-destructible</p> <p><i>Input</i> : $F, C(\bar{F}), \Psi$;</p> <ol style="list-style-type: none"> 1. $V \leftarrow$ Set of element of $C(\bar{F})$ pointed by $\Psi(q)$ for all q in $\Gamma^{-1}(p)$; 2. If $(V = \emptyset)$ then return $[\infty, \emptyset]$; 3. $[k_m, c_m] \leftarrow$ HighestFORK $(C(\bar{F}), V)$; 4. If $[k_m, c_m] = [\infty, \emptyset]$ then return $\min(V)$; 5. If $(k_m \leq F(p))$ then return $[k_m, c_m]$ else return $[\infty, \emptyset]$;
--

Previous algorithm gives correct results with regard to the definition (2.17) and is linear in time complexity with respect to the number of neighbors of p .

Checking whether a point is w -destructible or not, involves the computation of the highest fork of different elements of the set $V(p)$, see algorithm 9. This may require a number of calls to BLCA (Binary lowest common Ancestor) which is quadratic with respect to the cardinality of $V(p)$: every pair of elements of $V(p)$ has to be considered.

<p>Algorithm 9 : Function HighestFORK</p> <p><i>Input</i> : C : a component tree, V : a set of components of C</p> <ol style="list-style-type: none"> 1. $[k_1, c_1] \leftarrow \min(V)$; // let $[k_2, c_2] \dots [k_n, c_n]$ be the other elements of V 2. $k_m \leftarrow k_1$; 3. $c_m \leftarrow c_1$; 4. for i from 2 to n do 5. $[k, c] \leftarrow$ BLCA $(C, [k_i, c_i], [k_m, c_m])$; 6. If $[k, c] \neq [k_i, c_i]$ then $k_m \leftarrow k$; 7. $c_m \leftarrow c$; 8. 9. If $(k_m = k_1)$ then return $[\infty, \emptyset]$ else return $[k_m, c_m]$;

The **HighestFork** algorithm returns the highest fork of the set V , or the indicator $[\infty, \emptyset]$ if there is no highest fork. This algorithm makes $(n-1)$ calls of the BLCA operator, where n is the number of elements in V .

Let C be a component tree, let V be a set of components of C , we denote by $\min(V)$ an element of V which has the minimal altitude. For following algorithm, we assume that C is represented in a convenient manner for BLCA.

Thus, we must propose a criterion for the selection of the remaining W -destructible points, in order to avoid multiple selections of the same point. Couprie and al. introduce the idea to give the greatest priority to a W -destructible point which may be lowered down to the lowest possible value. They prove that an algorithm which uses this strategy never selects the same point twice. A priority queue could be used to select W -destructible points in the appropriate order. Here, we

present their specific linear watershed algorithm which may be used when the grayscale range is small.

Algorithm 10 : Topological watershed [Couprie]	
Data :	$F, C(\bar{F}), \Psi$;
Result :	F ;
1.	For k from k_{\min} to $(k_{\max} - 1)$ do $L_k \leftarrow \emptyset$
2.	For all $(p \in E)$ do
3.	$[i, c] \leftarrow W - Destructible(F, p, C(\bar{F}), \Psi)$
4.	If $(i \neq \infty)$ then
5.	$L_{i-1} \leftarrow L_{i-1} \cup \{p\}$;
6.	$K\{p\} \leftarrow i - 1$;
7.	$H\{p\} \leftarrow$ pointer to $[i, c]$;
8.	end if
9.	end for
10.	For k from k_{\min} to $(k_{\max} - 1)$ do
11.	While $(\exists p \in L_k)$ do
12.	$L_k = L_k \setminus \{p\}$;
13.	If $(K(p) = k)$ then
14.	$F(p) \leftarrow k$;
15.	$\Psi(p) \leftarrow H(p)$;
16.	For all $(q \in \Gamma(p), k < F(q))$ do
17.	$[i, c] \leftarrow W - Destructible(F, q, C(\bar{F}), \Psi)$;
18.	If $(i = \infty)$ then $k(p) \leftarrow \infty$;
19.	Else if $(k(p) \neq (i - 1))$ then
20.	$L_{i-1} \leftarrow L_{i-1} \cup \{q\}$;
21.	$k(p) \leftarrow (i + 1)$;
22.	$H\{q\} \leftarrow$ pointer to $[i, c]$;

2.4 Watershed Transform Based on Local Condition

There is a big similarity between this approach and the drop of water one. Actually basin surface increases in a progressive manner. The local condition of label continuity is iteratively applied along the steepest descent path that reaches the basin minimum. The downhill algorithm, the hill climbing algorithm and the toboggan algorithm are based on this approach. More details of the first two algorithms are given by [24][25] and the toboggan algorithm will be detailed later in this section. Differences between these three algorithms lie in the processing strategy and data structure as shown in [6].

For mathematical formulation, we follow description provided in [7]. In witch Audrier and al. start by presenting the following catchment basin formulation $CB_{LC}(m_i) = \{v \in V, L(v) = L(m_i)\}$, since local condition watershed assigns to each pixel the label of some minimum m_i . Thus watershed can be defined as follow. We recall that the condition $\mathbf{P}_{steepest}(v) \neq \{ \}$ means that v has at least one lower neighbor.

Definition 2.5 (Watershed based on local condition) (2.18)

For any lower complete image L_{CB} , a function L assigning a label to each pixel is called watershed segmentation if:

- a) $L(m_i) \neq L(m_j) \forall i \neq j$, With $\{m_k\}$, the set of minima of L_{LC} .
- b) For each pixel v with $\mathbf{P}_{steepest}(v) \neq \{\}$, $\exists p \in \mathbf{P}_{steepest}(v), L(v) = L(p)$.

As we mentioned earlier, we will introduce the toboggan algorithm [24, 26] as a reference of the local condition watershed approach. Actually this algorithm is referred as a drainage analogy. It seeks to identify the steepest descent from each pixel of the gradient magnitude of the input image to a local minimum of the topographic surface. Then pixels that belong to the same minima are merged by assigning them a unique label. Sets of pixels having the same label will define catchment basins. The resulting watershed regions are divided by a boundary path which will build the watershed lines.

Let consider $G : D \rightarrow R^+$ as a gradient magnitude image, where D is the indexing domain of the image. D can be decomposed into a finite number of disjoint level sets since pixels are sorted in the increasing order. Sets can be denoted by: $D_h = \{p \in D \mid G(p) = h\}$. Lin and al [24] define the following pixels classes: Class C_1 refers to all pixels p in D_h with an altitude strictly greater than the altitude of its lowest neighbor. Class C_2 refers to all pixels p in D_h belonging to a connected component with one or more catchment basin and $p \notin C_1$. Finally, class C_3 refers to all pixels p in D_h belonging to a connected component without any catchment basin. Thus we can give a global description of the computing process.

Algorithm 11 : Toboggan watershed process
<ol style="list-style-type: none"> 1. Records the sliding directions for all $(p \in C_1) \cup (p \in C_2)$ in D <ol style="list-style-type: none"> a. Records the lowest neighbours of all $(p \in C_1)$ in D. b. Grown region from all $(p \in C_1)$ 2. Assign label for all $(p \in C_3)$ 3. Assign label to each unlabeled image by first tobogganing then backtracking using best first search.

Based on this process, authors introduce the following algorithm to compute watershed.

Algorithm 12: Toboggan Algorithm [Lin and al.]
<p>Data : Img : a gradient magnitude image;</p> <p>Result : L : a label image, Q : empty FIFO queue ;</p> <ol style="list-style-type: none"> 1. For all $(p \in D)$ do //Simulation of sliding for all C_1 pixels 2. $h = G(p)$ 3. $h_{MIN} = \min \{G(q), q \in Neighbor(p)\}$ 4. If $(h > h_{MIN})$ then 5. $S = \{(q \mid G(q) = h_{MIN}) \& \& (q \in Neighbor(p))\}$

```

6.      SlidingList( $p$ ) =  $S$ 
7.       $Q \leftarrow p$ 
8.      GrowingDist( $p$ ) = 0
9.      Else if :
10.     SlidingList( $p$ ) =  $\phi$ 
11.     End if
12.     End for
13.     While  $Q \neq \phi$  do //Simulation of keep- sliding for all  $C_2$  pixels
14.      $p \leftarrow Q$ 
15.      $d = \text{GrowingDist}(p) + 1$ 
16.      $h = G(p)$ 
17.     For all ( $q \in \text{Neighbor}(p)$ ) and ( $G(q) = h$ ) do
18.     If (SlidingList( $q$ ) =  $\phi$ ) then
19.         Append ( $p$ ) to SlidingList( $q$ )
20.         GrowingDist( $q$ ) =  $d$ 
21.          $Q \leftarrow q$ ;
22.     Else If (GrowingDist( $q$ ) =  $d$ ) then
23.         Append ( $p$ ) to SlidingList( $q$ )
24.     End if
25.     End while
26.     For all ( $p_0 \in D$ ) and (SlidingList( $p_0$ ) =  $\phi$ ) do // labelling  $C_3$  pixels
27.     If  $L(p_0)$  is not assigned then
28.          $L(p_0) = \text{new\_label}$ 
29.          $h = G(p_0)$ 
30.         While  $Q \neq \phi$  do
31.          $p \leftarrow Q$ 
32.         For all ( $p \in \text{Neighbor}(p)$ ) and ( $G(q) = h$ ) do
33.             If  $L(q)$  is not assigned then
34.                 ( $L(q) = L(p_0)$ )
35.                  $Q \leftarrow p$ 
36.             End if
37.         End for
38.         End while
39.     End if
40. End for
41. For all ( $p \in D$ ) do // Tobogganing – Depth first search
42.     Resolve( $p$ )
43. End For

```

Algorithm 13: Resolve function

```

Input : Pixel site  $p$ 
1.  If  $L(p)$  is not assigned then
2.     $S = \text{SlidingList}(p)$ 
3.    For all ( $q \in S$ ) do

```

```

4.      Resolve(q)
5.      End for
6.      If  $S$  has a unique label  $\alpha$  then
7.           $L(p) = \alpha$ 
8.      Else
9.           $L(p) = RIDGE\_label$ 
10.     End if
11. End if

```

2.5 Watershed Transform Based on Minimum Spanning Forest

The original idea is very close to the second case of the path cost minimization based watershed that consist on building a spanning forest from a graph as we introduced in section 2.2. Actually, the beginning was with Meyer [27] who proposes to compute watershed transform from a weighted neighborhood graph whose nodes are the catchment basins corresponding to the minima of the image. Arcs of the graph, that separate neighbor catchment basins, are weighted by the altitude of the pass between these basins. Extracted minimum spanning forests define partitions that are considered solution of watersheds. It's important to mention that returned solutions are multiple. Authors established also the links between the minimum spanning forest and flooding from marker algorithms. Trough Meyer's bases, Cousty and al. [28] introduce the watershed-cuts and establish the optimality of this approach by showing the equivalence between the watershed-cuts and the separations induced by minimum spanning forest relative to the minima.

For mathematical foundations, we will follow the notations in [28] to present some basic definitions to handle with minimum spanning forest cuts and watershed-cuts.

Let G be graph with $G = (V(G), E(G))$. $V(G)$ is a finite set of vertex of G . Unordered pairs of $V(G)$, called also edge of G , constitute the element of $E(G)$ set. Let denote the set of all maps from E to \square by F and we consider that any maps of F weights the edges of G . Let $F \subseteq F$ and $u \in E(G)$, $F(u)$ will refers to the altitude of u and $M(F)$ will refers the graph whose vertex set and edge set are, respectively, the union of the vertex sets and edge sets of all minima of F .

Let X and Y be two sub-graphs of G . We say that Y is a forest relative to X if Y is an extension of X and for any extension $Z \subset X$ of X , we have $Y=Z$ whenever $V(Z)=V(Y)$.

(i) Y is said a spanning forest relative to X (for G) if Y is a forest relative to X and if $V(Y)=V$. In this case, there exists a unique cut S for Y . It is composed by all edges of G whose extremities are in two distinct components of Y . Since Y is an extension of X , it can be seen that this unique cut S (induced by Y) is also a cut for X .

(ii) Y is said a minimum spanning forest relative to X (for F , in G) if Y is a spanning forest relative to X and if the weight of Y is less than or equal to the weight of any other spanning forest relative to X . In this case, S is considered as a minimum spanning forest cut for X .

Trough these equivalences, Cousty demonstrate that the set $S \subseteq E$ is a minimum spanning forest cut for $M(F)$ if and only if S is a watershed cut of F , that can be computed by any minimum spanning tree algorithm. And he proposes a linear algorithm to compute it using a new 'stream' notion that we will not detail in this section. Only the stream algorithm will be introduced.

Now, before presenting the watershed-cuts algorithm we just recall the definition of the minimal altitude of an edge. Let denote by $F^-(x)$ the map from V to \mathbb{R} such that for any $x \in V$, $F^-(x)$ is the minimal altitude of an edge which contains x . Then a path $\pi = \langle x_0, \dots, x_l \rangle$, is considered as a path of steepest decent for F (in G) if for any $i \in [1, l]$, $F(\{x_{i-1}, x_i\}) = F^-(x_{i-1})$.

In the following, we introduce the watershed-cuts computing algorithm and stream function.

Algorithm 14: Watershed-cuts algorithm [Cousty and al.]	
Data :	(V,E,F) : Edge-weighted graphs;
Result:	Ψ : a flow mapping of F
1.	Foreach $(x \in V)$ do $\psi(x) \leftarrow \text{NO_LABEL}$;
2.	$nb_labs \leftarrow 0$
3.	Foreach $(x \in V)$ such that $(\psi(x)=\text{NO_Label})$ do
4.	$[L, \text{Lab}] \leftarrow \text{Stream}(V,E,F,\psi,x)$;
5.	If $(lab = -1)$ then
6.	$nb_labs ++$
7.	Foreach $(y \in L)$ do $\psi(y) \leftarrow nb_labs$;
8.	Else
9.	Foreach $(y \in L)$ do $\psi(y) \leftarrow labs$;

Algorithm 15: Stream function [Cousty and al.]	
Data :	(V,E,F) : Edge-weighted graphs; Ψ : a label of V ; x : point of V ;
Result :	$[L, lab]$: L is a flow obtained from x (source of L) ; lab is the associated label to an \emptyset flux included in L or (-1) .
1.	$L \leftarrow \{x\}$
2.	$L' \leftarrow \{x\}$ // the set of sources not yet explored of L
3.	While there exists $(y \in L')$ do
4.	$L' \leftarrow L' \setminus \{y\}$;
5.	$breadth_first \leftarrow \text{TRUE}$;
6.	While $(breadth_first)$ and $(\exists \{y,z\} \in E / z \notin L \text{ and } F(\{y,z\}) = F(y))$ do
7.	If $(\Psi(z) \neq \text{No_label})$ then
8.	return $[L, \Psi(z)]$ // exist an \emptyset flow L already labelled
9.	Else if $(F^-(z) < F^-(y))$ then
10.	$L \leftarrow L \cup \{z\}$; // z is the only well of L
11.	$L' \leftarrow \{z\}$; // switch the in-depth exploration first
12.	$breadth_first \leftarrow \text{FALSE}$
13.	Else
14.	$L \leftarrow L \cup \{z\}$; // therefore z is a well of L
15.	$L' \leftarrow L' \cup \{z\}$; // continue exploration in width first
16.	return $[L, -1]$

3. CLASSIFICATION OF WATERSHEDS TRANSFORMS

In this section we will learn from different syntheses present in Roerdink [6] and Audigier [7] works. The following table summarizes some characteristic of introduced watershed transform. Selected criteria are justified by our objective to identify the most suitable algorithm for parallel implementation.

Watershed based on						
	Flooding	Path-cost minimization		Topology	Local condition	MSF
		TD	IFT			
	Vincent & Soille [15]	Meyer [17]	Lotufo [25]	Couprrie [9]	Lin [26]	Cousty [4]
Defined in	Disc. cont. space	Disc. cont. space	Only on discrete space	Disc. cont. space	Disc. cont. space	Only on discrete space
Classified as	Line WT	Line WT	Region WT	Line WT	Region WT	Region WT
Gives unique solution	Yes	Yes	No	No	No	No
Preserve topology	No	No	Yes	Yes	No	Yes
Requires a sorting step	Yes	No	No	No	No	No
Use of hierarchical queue	Yes	Yes	Yes	Yes	Yes	No
Minima computing	Yes	Yes	No	No	-	No
Is linearity	linear	-	linear	Linear*	-	Linear

TABLE 1: Comparison between main watersheds transforms (definitions & algorithms)

The starting point is the definition space; we note that IFT-Watershed and MSF-Watershed definitions are limited to the discrete space while the other watersheds definitions are spread into continue space. IFT-Watershed, MSF-watershed and LC-Watershed form the region based watershed transform family since pixels are assigned to basins. Flooding-Watershed, TD-Watershed and Topological-Watershed form the line based watershed family since some pixels are labeled as watershed. Only Topological-Watershed defines lines that consistently separate basins while Flooding-Watershed and TD-Watershed merely swing between thick and disconnected watershed lines. Through definitions, only Flooding-Watershed and TD-Watershed return unique solution while all other definitions return multiple solutions. Note that set of solutions returned by the IFT-Watershed can be unified by creating litigious zones when solutions differ [7]. All six algorithms, that don't exactly include their definitions, return unique solution but don't preserve the number of connected components of the original input image. Actually, Vincent-Soille, Meyer and Lin's algorithm don't preserve important topological features. Only Lotufo, Couprrie and Cousty's algorithm are correct from this point of view.

Regarding computing process, only Flooding-Watershed needs pixel's sorting while others transforms will pass this costly step. But this does not preclude associated algorithms to use hierarchical structures when implementing. Except Cousty's algorithm that doesn't need any hierarchical queue. Vincent-Soille and Meyer's algorithms impose also a prior minima computation, which is not the case of the others. For complexity, observe that Vincent and Soille algorithm runs in linear with respect to the number N of pixels in the image which is processed. In most current situations of image analysis, where the number of possible values for the priority function is limited and the number of neighbors of a point is small constant, Couprrie's algorithm

runs also in linear time with $O(n + m)$ complexity. Lotufo and Cousty's algorithm run also in linear time. Cousty's algorithm is executed at most $O(|E|)$ times.

4. CONCLUSION

In this paper, we have presented an intensive study of all existing watershed transform in the discrete case: WT based on flooding, WT based on path-cost minimization, watershed based on topology preservation, WT based on local condition and WT based on minimum spanning forest.

The first major contribution in this paper is the global nature of the proposed study. In fact, for each approach, we start by giving informal definition, then we present processing procedure followed by mathematical foundations and the algorithm of reference. Recent publications based on some approach are also presented and discussed. The second contribution concerns algorithms' classification according to criteria of recursion, complexity, basins computing and topology preservation.

In our future work we will present a parallel MSF-Watershed algorithm suitable for shared memory parallel machines based on Jean Cousty stream function already presented.

5. REFERENCES

- [1] J. Maxwell. "On hills and dales." *Philosophical Magazine*, vol. 4/40, pp. 421-427, Dec. 1870.
- [2] S. Beucher and C. Lantuéjoul. "Use of watersheds in contour detection." *International Workshop on Image Processing, Real-Time Edge and Motion Detection/ Estimation*, 1979.
- [3] S. Beucher and F. Meyer. "The morphological approach to segmentation: the watershed transformation." In Dougherty, E., ed.: *Mathematical Morphology in Image Processing*, Marcel Decker, pp. 433-481, 1993.
- [4] J. Serra. "Image Analysis and Mathematical Morphology." Academic Press, New York, 1982.
- [5] M. Straka, A. La Cruz, A. Köchl, M. Šrámek, E. Gröller and D. Fleischmann. "3D Watershed Transform Combined with a Probabilistic Atlas for Medical Image Segmentation." *Journal of Medical Informatics & Technologies*, 2003.
- [6] J.B.T.M. Roerdink and A. Meijster. "The watershed transform: Definitions, algorithms and parallelization strategies." *Fundamenta Informaticae*, Special issue on mathematical morphology vol. 41, pp. 187-228, Jan. 2001.
- [7] R. Audigier and R.A. Lotufo. "Watershed by image foresting transform, tie-zone, and theoretical relationship with other watershed definitions." In *Mathematical Morphology and its Applications to Signal and Image Processing*, pp. 277-288, 2007.
- [8] L. Najman and M. Couprie. "Watershed algorithms and contrast preservation." *Lecture Notes in Computer Science*, vol. 2886, pp. 62-71, 2003.
- [9] P. Soille. "Morphological Image Analysis." Springer-Verlag Berlin and Heidelberg GmbH & Co. K, April 1999.
- [10] E. Dougherty and R. Lotufo. "Hands-on Morphological Image Processing." SPIE Publications, 2003.
- [11] L. Najman, M. Couprie and G. Bertrand. "Watersheds, mosaics and the emergence paradigm." *Discrete Applied Mathematics*, vol. 147, pp. 301-324, April 2005.

- [12] L. Vincent and P. Soille. "Watersheds in digital spaces - An efficient algorithm based on immersion simulations." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 583-598, 1991.
- [13] P. Shengcai and G. Lixu. "A Novel Implementation of Watershed Transform Using Multi-Degree Immersion Simulation." *27th Annual International Conference of the Engineering in Medicine and Biology Society*, pp. 1754 – 1757, 2005.
- [14] F. Meyer. "Topographic distance and watershed lines." *Signal Processing*, vol. 38, pp. 113-125, 1993.
- [15] L. Najman and M. Schmitt. "Watershed of a continuous function." *Signal Processing*, vol. 38, pp. 68-86, 1993.
- [16] A. X. Falcão, J. Stolfi, and R. A. Lotufo. "The Image Foresting Transform: Theory, Algorithms, and Applications." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp.19-29, 2004.
- [17] R. A. Lotufo and A. X. Falcão. "The Ordered Queue and the Optimality of the Watershed Approaches." *Procs. 5th International Symposium on Mathematical Morphology*, pp. 341-350, 2000.
- [18] E.W. Dijkstra. "A Note on Two Problems in Connection with Graphs." *Numerische Mathematik*, vol 1, pp. 269-271, 1959.
- [19] U. Pape. "Implementation and Efficiency of Moore Algorithms for the Shortest Root Problem." *Mathematical Programming*, vol. 7, pp. 212-222, 1974.
- [20] R. B. Dial, F. Glover, D. Karney, and D. Klingman. "A Computational Analysis of Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees." In *Networks*, vol 9, pp. 215-248, 1979.
- [21] R.A. Lotufo, A. X. Falcão and F. A. Zampirolli. "IFT-Watershed from Gray-Scale Marker." *15th Brazilian Symposium on Computer Graphics and Image Processing*, Vol 2, 2002.
- [22] M. Couprie and G. Bertrand. "Topological grayscale watersheds transform." *SPIE Vision Geometry V Proceedings*, vol. 3168, pp. 136-146, 1997.
- [23] M. Couprie, L. Najman and G. Bertrand. "Quasi-linear algorithms for the topological watershed." *Journal of Mathematical Imaging and Vision*, vol. 22, pp. 231-249, 2005.
- [24] Y.C. Lin, Y.P. Tsai, Y.P Hung and Z.C. Shih. "Comparison Between Immersion-Based and Toboggan-Based Watershed Image Segmentation." *15th IEEE Transactions on Image Processing*, vol. 3, pp. 632-640, 2006.
- [25] V.O. Ruiz, J.I.G. Llorente, N. S. Lechon and P.G. Vilda. "An improved watershed algorithm based on efficient computation of shortest paths." *Pattern Recognition*, vol. 40, pp. 1078-1090, 2007.
- [26] E.N. Mortensen and W.A. Barrett. "Toboggan-based intelligent scissors with a four-parameter edge model." In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 452–458, 1999.
- [27] F. Meyer. "Minimum Spanning Forests for Morphological Segmentation." *Proc. Second International Conference on Math. Morphology. and Its Applications to Image Processing*, pp. 77-84, 1994.

- [28] J. Cousty, G. Bertrand, L. Najman and M. Couprie. "Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle." IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 1362-1374, 2009.

The Framework of Image Recognition Based on Modified Freeman Chain Code

Haswadi Hasan

*Faculty of Computer Science and Information System (FSKSM)
Universiti Teknologi Malaysia
Skudai, 81310, Malaysia*

haswadi@utm.my

Habibollah Haron

*Faculty of Computer Science and Information System (FSKSM)
Universiti Teknologi Malaysia
Skudai, 81310, Malaysia*

habib@utm.my

Siti Zaiton Mohd Hashim

*Faculty of Computer Science and Information System (FSKSM)
Universiti Teknologi Malaysia
Skudai, 81310, Malaysia*

sitizaiton@utm.my

Abstract

Image recognition of line drawing involves feature extraction and feature comparison; works on the extraction required the representation of the image to be compared and analysed. Combining these two requirements, a framework that implements a new extraction algorithm of a chain code representation is presented. In addition, new corner detection is presented as pre-processing to the line drawing input in order to derive the chain code. This paper presents a new framework that consists of five steps namely pre-processing and image processing, new corner detection algorithm, chain code generator, feature extraction algorithm, and recognition process. Heuristic approach that is applied in the corner detection algorithm accepts thinned binary image as input and produces a modified thinned binary image containing J characters to represent corners in the image. Using the modified thinned binary image, a new chain code scheme that is based on Freeman chain code is proposed and an algorithm is developed to generate a single chain code series that is representing the line drawing input. The feature extraction algorithm is then extracts the three pre-defined features of the chain code for recognition purpose. The features are corner properties, distance between corners, and angle from a corner to the connected corner. The explanation of steps in the framework is supported with two line drawings. The results show that the framework successfully recognizes line drawing into five categories namely not similar line drawing, and four other categories that are similar but with attributes of rotation angle and scaling ratio.

Keywords: Corner Detection, Chain Code, Line Drawing, Feature Extraction, Recognition

1. INTRODUCTION

Image recognition of line drawing involves comparison and analysis of more than one line drawing against the reference. The recognition includes derivation of features from the input image, therefore, pre-processing, image processing and data representation stages are required to analyze the image in producing the feature of the line drawing. Thinning and corner detection algorithms are among basic steps in image processing while chain code is one of line drawing representation. Combining these steps, a framework is presented that consists of pre-processing, image processing, data representation, feature extraction and finally the recognition process.

This paper is divided into five sections. Section 1 presents an introduction on image recognition and steps involved. Next in Section 2, the framework and steps from previous works on image

recognition are discussed, and then the proposed framework and steps taken are detailed in Section 3. Section 4 examines the experimental result of the framework that is supported with line drawing examples. Finally, conclusion and discussion is presented in Section 5.

2. FRAMEWORK IN IMAGE RECOGNITION

As mentioned earlier, steps in image recognition are including pre-processing, image processing, data representation and feature extraction. This section presents few previous works on image recognition and steps involved. In each step, new algorithm or representation is proposed and they are discussed in this section as motivation and comparison in the development of a new framework in image recognition.

The literature review is divided into three categories namely works about the framework on image recognition and its feature identification and extraction, input of the feature extraction especially the chain code representation, and the steps in image processing of the input image especially corner detection.

The previous works summarizes all works that accepts chain code as its input or data representation, and detecting corner based on chain code. The framework in [1-3] provides basis in identifying features of image recognition of this work. The works by [2-5] show the application of chain code in image recognition and feature extraction. The corner detection in [6] demonstrates the application of chain code in detecting corner while [7] points out how corner detection apply chain code scheme as curvature. Works by [8] proposes new chain code scheme in image retrieval.

Based on these three categories of previous works, it shows that the image recognition and feature extraction can possibly include corner detection as part of pre-processing and image processing step. The study also shows that chain code is relevant scheme and representation in image recognition and feature extraction. They give motivation and ideas on new framework in image recognition particularly for line drawing that combines feature extraction, corner detection and chain code representation.

3. THE PROPOSED FRAMEWORK

3.1 The Framework

This section presents the framework and its steps. Fig. 1 shows the framework diagram. The dotted box in the figure represents the input and output of the process contained in the solid box. There are five steps in the framework namely pre-processing, corner detection, chain code generation, feature extraction, and recognition. First, data preparation and pre-processing involving image processing tools in producing thinned binary image are performed. The derived thinned binary image (TBI) is then altered by the corner detection algorithm to produce a new modified TBI that contains additional J characters representing corner of the image. By defining new chain code scheme, a series of chain code is derived from the modified TBI. This chain code series is then analyzed by the feature extraction algorithm based on three pre-defined features of line drawing namely corner properties, distance of edge between corners, and angle between corners. The extracted features are then used in the recognition process. The following sub-sections explain in detail each step while their experimental results are presented in Section 4.

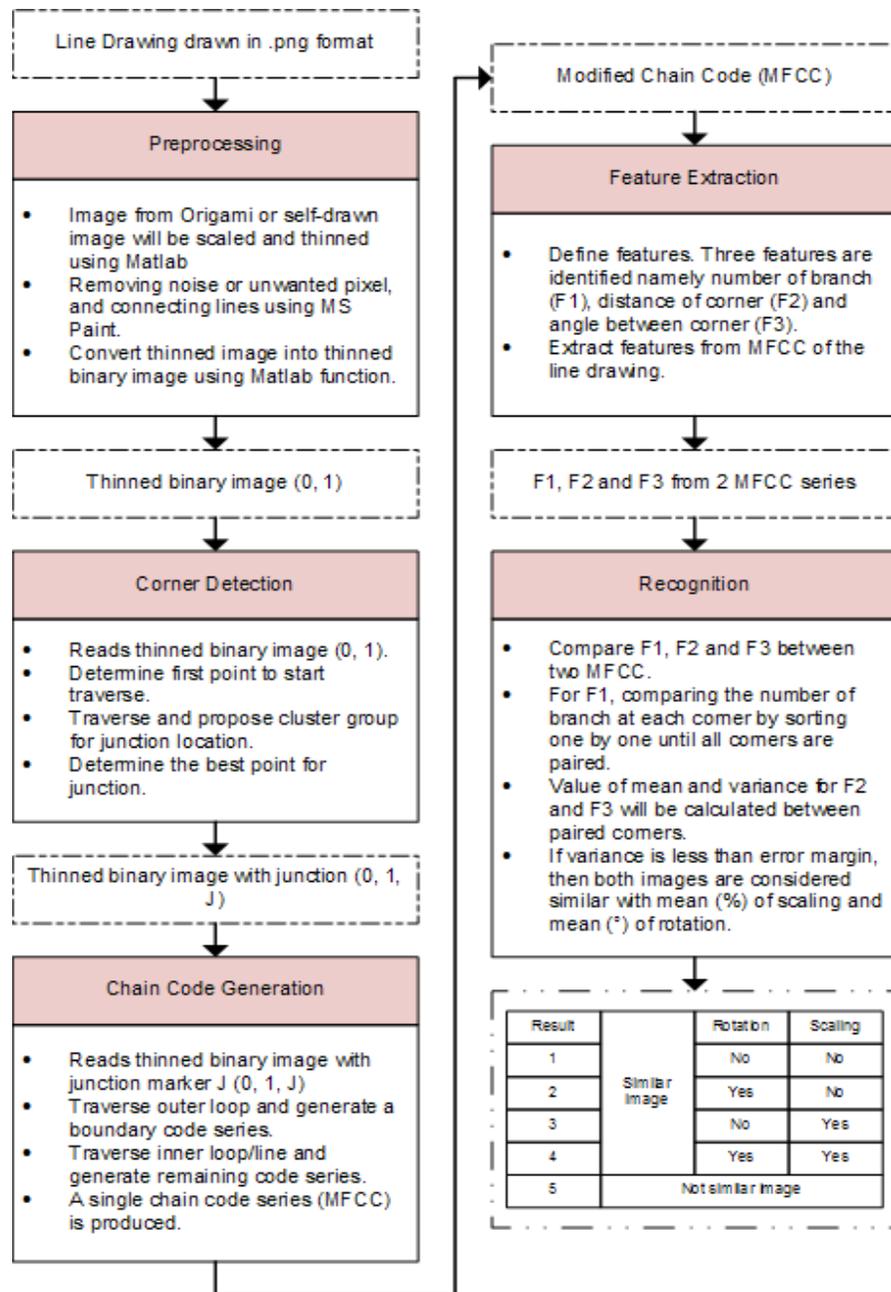


FIGURE 1: The Proposed Line Drawing Recognition Framework.

3.2 Data Definition and Pre-Processing

Defining data set of line drawing involves several assumptions such as the line drawing is a two-dimensional regular line drawing and the source of the line drawing would be from origami world. Pre-processing involves resizing and thinning the image, cleaning or removing noise and unwanted pixel, and reconnecting lines if necessary. This process is performed manually using tools such as Microsoft Paint and simple script based on MATLAB function. Since it is beyond the scope of discussion, they are not discussed in this paper. This step produces a TBI of the line drawing.

3.3 Development of Corner Detection Algorithm

This step reads cleaned TBI and creates a modified TBI with J character in the image. The module involves two phases. First, starting point to traverse is determined. Second, cluster grouping is performed that is labelling the TBI with temporary label 'G', creating vector, generating vector list, and elimination of path and vector. The result of corner list in term of new TBI with J character is produced.

Any gate location is marked with G character and cluster member with + character. Any cluster in image will be grouped so that cluster exit gate which defines start and stop point for branch can be searched. Tracker movement is depends on the number of gates left untouched on current cluster before moving on to another cluster, while the current cluster is always referring to the latest cluster found.

To determine corner position in every cluster, vector creation is executed using cluster gate list that is either a beginning or end of a branch. Since connection between gates is not included in the list, path tracing between gates is inevitable. Combined with the need to include line angle namely inclination or degree of slope as its properties, edge detection is concurrently performed here since it involves scanning for slope changes. A box or window is maintained as reference slope and moved throughout the path trace while the actual branch slope will be calculated from the branch start until current point. When the fluctuation between the slopes has passed over the accepted limit, a corner is declared as found.

Line vectors are created based on cluster gates to find corners located in every cluster by line extrapolation. Every cluster member will be cross-marked its point distance from line vectors attached cluster and the location with lowest value (lowest distance) will be chosen as a corner in the cluster. Newly appointed corner is tested for its connectivity with all gates in corresponding cluster to ensure that corner is enough for the cluster or additional corner is required. Now, all corner locations have been found and will be marked with J character in the line drawing to form a modified TBI.

3.4 Definition of New Chain Code Scheme and Development of Its Generator

This framework section proposed a new modified Freeman chain code scheme (MFCC). The development of this chain code scheme is parallel with the development of the chain code generator. The scheme is based on Freeman chain code but with additional character started from A, B, C...Z not to represent the direction of the pixel but the corner label passed during tracing. The chain code scheme is defined to have classification for outer loop and inner loop of the line drawing: outer loop is derivation of codes in the series of the boundary while inner loop is for remaining inner lines. The MFCC is a single chain code series and the reverse traversal of the code will produce the same source line drawing.

Using TBI with label corner J (0, 1, J) as input, tracking point will be set at the most bottom left pixel of image as starting location. The tracing will start from here and repeatedly tests for current position and neighbouring pixels for next move direction. The unlabelled corner previously marked its location with J, will be assigned a label that follows the latest used label. For starting point, the corner will be assigned 'A' character as its label and the corner is recorded in the chain code. Every corner found during tracing process is recorded in a First In Last Out (FILO) stack for rollback ability when the tracing reaches the end of any path. The boundary of line drawing will be used first for path traversing until the tracking point reaches back the starting point.

After traversing the outer loop, next is extracting chain code for inner lines. Positions for tracker should be at the same starting point as in outer loop counterpart. All remaining paths are traversed and all corner found and direction taken during tracing is recorded. When the tracker reached dead end, where the end should be a corner, corner stack is consulted for previous corner and if there is still corner left, the tracker will be set at that corner position. If there are no more corners left in the stack, then line drawing traversal and MFCC generation is deemed complete.

3.5 Defining the Features and Development of the Extraction Algorithm

This step involves two stages namely defining features to be used in the recognition, and extracting the features of the line drawing based on the generated MFCC. Identifying features of the line drawing is based on the geometric and topological properties of a line drawing. The three features are number of branches at each corner (F1), distance between corners (F2), and angle between corners (F3). Before extracting the features, layer where the corner resides in the line drawing must be determined starting from the boundary layer (outer loop) and moving into the inner loop based on corner linkage.

For F1, two properties is extracted namely the number of branches and the interconnectivity between corners. A table is created to represent these features. For F2 and F3, the calculation of these features is performed by a heuristic approach producing two values namely distance (length) in pixel for each corner, and branch angle at each corner. In MFCC, a branch will be found between 2 corner markers with its distance and angle will be derived based on directional codes defining the branch. Pythagoras theorem will be used to calculate the distance (F2) while the angle (F3) is using tangent formulation. Value of F1, F2 and F3 will be stored in corner property list, where the list is unique for each compared MFCC.

3.6 Recognition Process

Recognition involves comparison between two line drawings and it is based on the values of F1, F2 and F3. All features must be considered matched or accepted so that the recognition session to be declared successful. For F1, comparison of properties for each corner between two line drawings is performed. The list of corner properties with the number of branches at each corner for tested line drawing is rotated by one displacement until the quantity in both lists is matched. Corner labels are also important to be matched, but the pairing is limited to be performed on outer loop corners only since the labelling is in sequential order for this segment.

After F1 analysis is satisfied, analysis for F2 and F3 is performed by calculating the means and variances of both distance and angle. For F2, mean value represents ratio (%) of the scaling process while mean for F3 represents the degree of rotation occurred between two line drawings. Variances for both F2 and F3 are used to measure on how far a set of distance and angle values are spread out among them against a preset limit. Thus for these features, variances are used as rejection/acceptance criterion in the recognition process.

4. EXPERIMENTAL RESULT

Result on two line drawings tested on the framework is presented in this section. The discussion on input and output of four steps in the framework of first line drawing (LD1) namely pre-processing, corner detection, generating the chain code and deriving features are presented. Next, second line drawing (LD2) and its features is presented as input of the fifth step, recognition process. The details of the first four steps for LD2 are not given because of its similarity in steps taken in LD1. After the features of LD1 and LD2 are obtained, the recognition process is conducted and conclusion of the recognition is displayed. The following sub-sections show the input and output of each step in the framework.

4.1 Pre-processing

This step reads input of image as shown in Fig. 2(a), produced temporary thinned image as shown in Fig. 2(b), and lastly output the thinned binary image as shown in Fig. 3(a). The detail of the process can be referred in Section 3.2.

4.2 Corner Detection

This second step reads the thinned binary image (TBI) and produces the thinned binary image with J as shown in Fig. 3(c). The character J indicates corner of the image detected by the algorithm mentioned in Section 3.3. As the intermediate process, the clustering produces a cluster map which shows the cluster member and its gates, marked with + and G, respectively, as depicted in Fig. 3(b).

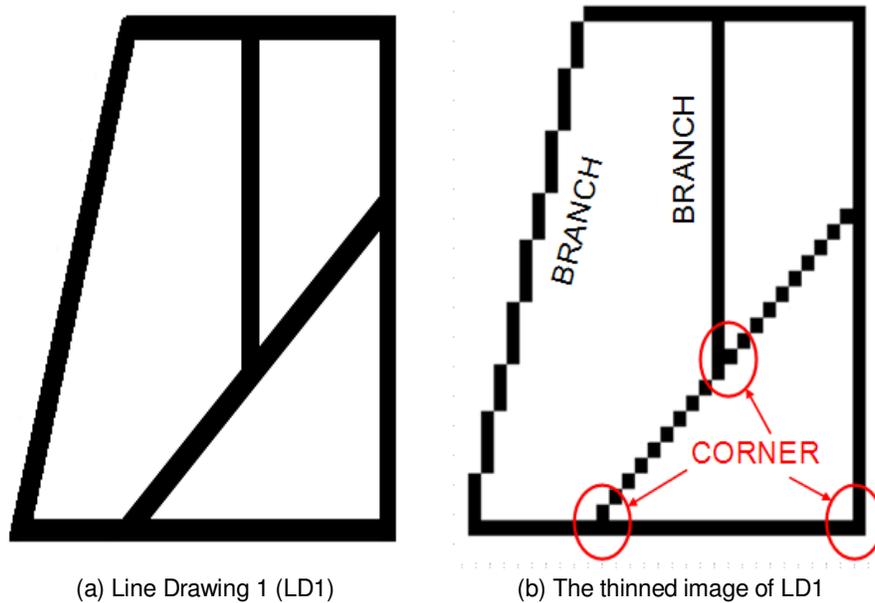


FIGURE 2: The line drawing and its thinned image.

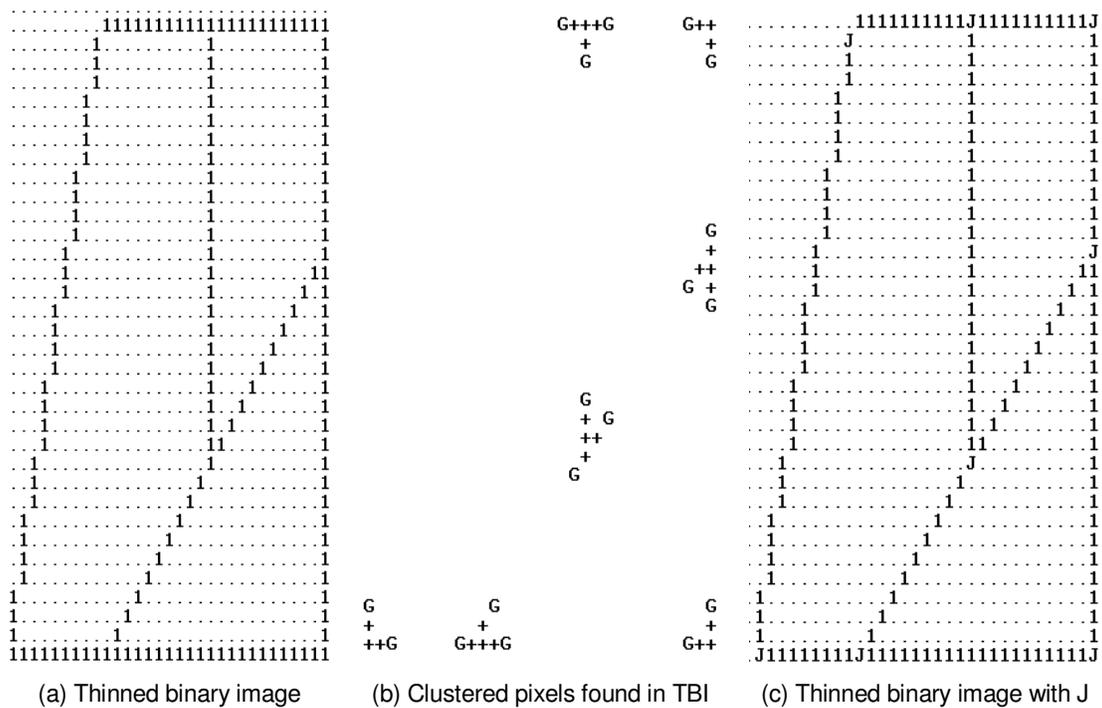


FIGURE 3: The TBI and its modification until final version with J character.

4.3 Chain Code Generation

Third step in the framework reads the TBI with J and produces the modified Freeman Chain Code (MFCC). Temporary binary image with additional code as defined in the MFCC scheme is created by the algorithm. Fig. 4(a) and 4(b) respectively show the temporary TBI created and the MFCC derived from the TBI. The temporary TBI includes the additional corner label A-H of the TBI that is

one of recognition result enumerated in Table 2. The result from recognition also provides angle of rotation and ratio of scaling of LD2 from LD1 based on mean values found during variance calculation in recognition algorithm.

Result		Rotation	Scaling
1	Similar image	No	No
2		Yes	No
3		No	Yes
4		Yes	Yes
5	Not similar image		

TABLE 2: Category of recognition result.

Fig. 5(a) and 5(b) show example of line drawing 1 (LD1) and 2 (LD2), respectively. The features of LD1 are as shown in Table 1 while features of LD2 are shown in Table 3. These tables will be used in recognition process.

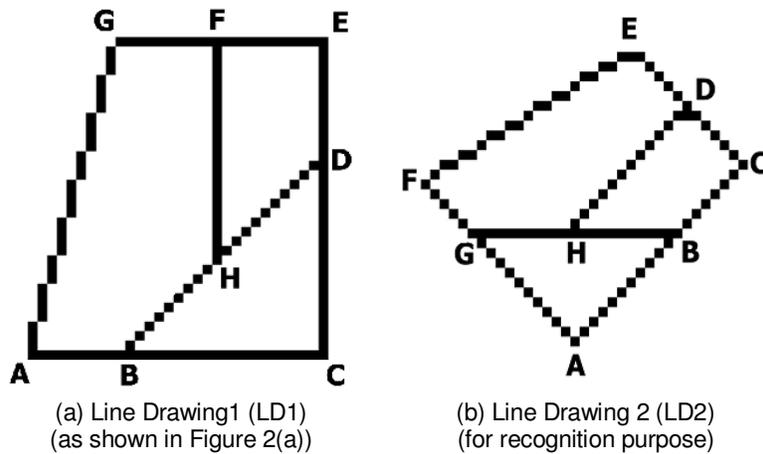


FIGURE 5: The input of recognition process with corners labelled.

Feature 1		Feature 2	Feature 3	Feature 1		Feature 2	Feature 3
Corner Properties		Distance (unit pixel)	Angle (°)	Corner Properties		Distance (unit pixel)	Angle (°)
Current	Target			Current	Target		
A (2)	B	15.56	45.00	E (2)	D	8.60	324.46
	G	15.56	135.00		F	24.70	211.76
B (3)	A	15.56	225.00	F (2)	E	24.70	31.76
	C	9.90	45.00		G	7.07	315.00
	H	12.00	180.00		F	7.07	135.00
C (2)	B	9.90	225.00	G (3)	A	15.56	315.00
	D	8.49	135.00		H	10.00	0.00
D (3)	C	8.49	315.00		H (3)	G	10.00
	E	8.60	144.46	B		12.00	0.00
	H	18.38	225.00	D		18.38	45.00

TABLE 3: Value of features for LD2

Table 4 shows the comparison of features F1, F2 and F3. For F1, the label for outer loop corners will be rotated as well as the sum of branches in order to find the pairing of the corners. The value of Distance Ratio is the scaling ratio between two distance values of F2 in Table 1 and 3. The

Distance Ratio represents the scaling factor of the new line drawing. For F3, the angle difference (Angle Diff.) is calculated based on the values of angle in Table 1 and 3. The values are then normalised. The normalised values indicate the rotation angle of the operation. Table 4 shows the value of mean and variance for F2 and F3. For F2, the value mean 0.768 indicates that the 76.8% scaling had happened for LD2. For F3, the value 44.40 indicates that line drawing 2 also have been rotated 44° with the rotation direction will be conferred from the actual angle difference value since the information is lost when angle normalization is executed. The exact answer would be the LD1 has been resized to 75% and 45°CW rotation to form LD2.

The value of variance is to measure structure lines properties uniformity. This value should be small enough to show that the line drawing 2 is really based on the line drawing 1. To judge how small the variance value must be is by comparing it to generic error margin set so that it must be lower than this value. If error margin is set to 0.1 for F2 and 15 for F3, the comparison can be allowed that LD2 is similar to LD1 after 76% scaled and 45° rotated clockwise. This acceptance is because variance value for F2 is 0.00093 and F3 is 1.863 that is less than error margin set.

F1 (LD1)		F1 (LD2)		F2	F3	
Corner		Corner			Distance Ratio	Angle Diff.
Current	Target	Current	Target	Actual		Normalized
A	B	F	G	0.786	-315.00	45.00
	G		E	0.749	44.20	45.20
B	A	G	F	0.786	45.00	45.00
	C		A	0.741	-315.00	45.00
C	B	A	G	0.741	45.00	45.00
	D		B	0.741	45.00	45.00
D	C	B	A	0.741	45.00	45.00
	E		C	0.825	45.00	45.00
E	D	C	B	0.825	45.00	45.00
	F		D	0.771	45.00	45.00
F	E	D	C	0.771	-315.00	45.00
	G		E	0.779	40.73	40.73
G	F	E	D	0.779	-319.27	40.73
	A		F	0.749	44.20	44.20
H	F	H	D	0.800	45.00	45.00
	D		B	0.771	45.00	45.00
	B		G	0.707	45.00	45.00
Mean				0.768		44.40
Variance				0.00093		1.863

TABLE 4: Feature Comparison Table in Recognition Process

5. DISCUSSION AND CONCLUSION

The results show that the framework successfully detects corner, generates chain code, extracts features, and finally recognizes the line drawing via comparison. The proposed heuristic corner detection algorithm provides a simpler way since it does not involve complex mathematical equation in the calculation. The advantage of the proposed MFCC is that a line drawing can be represented as one single code series. This overcomes problems in representing a line drawing by more than single series of chain code such as proposed by Freeman chain code. The proposed extraction algorithm successfully reads and derives features which is considered simple and yet produced accurate result. The proposed three features is enough for recognition purpose. The advantage of recognition based on the MFCC is storage saving and increasing complexity of input drawing to be compared. Finally, an integrated system which can compare and recognize altered line drawing from its original is presented.

6. ACKNOWLEDGMENT

The authors honourably show appreciation to Universiti Teknologi Malaysia (UTM) and Malaysian Ministry of Higher Education (MoHE) for the support in making this research successful.

7. REFERENCES

- [1] Yueh-Ling Lin and Mao-Jiun J. Wang, "*Automatic Feature Extraction from Front and Side Images*", Industrial Engineering and Engineering Management, 2008. IEEM 2008. p1949, 2008.
- [2] Junding, Sun and Heli, Xu; "*Contour-Shape Recognition and Retrieval Based on Chain Code*", 2009 International Conference on Computational Intelligence and Security, p349 - 352, 2009.
- [3] Yong-Xianga Sun; Cheng-Minga Zhang; Ping-Zenga Liu; Hong-Mei Zhu; "*Shape feature extraction of fruit image based on chain code*", Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, vol3, p1346 - 1349, 2007.
- [4] Chalechale, A.; Naghdy, G.; Premaratne, P.; Moghaddasi, H.; "*Chain-based extraction of line segments to describe images*", 2004 IEEE International Conference on Multimedia and Expo (ICME), Page(s): 355 - 358 Vol.1, 2004.
- [5] Tie-Gen Peng; Ti-Hua Wu; Yong Luo; "*The method based on boundary chain-code for objects recognition and gesture analysis*", Proceedings of the Third International Conference on Machine Learning and Cybernetics, p3700 - 3705 vol.6, 2004.
- [6] Bo Yu; Lei Guo, Xiaoliang Qian and Tianyun Zhao, "*A Corner Detection Algorithm Based on the Difference of FCC*", 2010, International Conference On Computer Design And Applications (ICCD 2010), vol 4, Page(s): V4-226 - V4-229, 2010.
- [7] Nain, N.; Laxmi, V.; Bhadviya, B.; Gopal, A.; "*Corner Detection using Difference Chain Code as Curvature*", Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, p821 - 825, 2007.
- [8] Wang Xiaoling and Xie Kanglin, "*A novel direction chain code-based image retrieval*", Fourth International Conference on Computer and Information Technology (CIT'04), p190 - 193, 2004.

A Wavelet Based Automatic Segmentation of Brain Tumor in CT Images Using Optimal Statistical Texture Features

A.Padma

Research scholar,
Thiyagarajar College of Engineering
Madurai – 625 015, India

giri_padma2000@yahoo.com

Dr.R.Sukanesh

Prof of Electronics and Communication Engineering,
Thiagarajar College of Engineering,
Madurai – 625 015, India.

rshece@tce.edu

Abstract

This paper presents an automated segmentation of brain tumors in computed tomography images (CT) using combination of Wavelet Statistical Texture features (WST) obtained from 2-level Discrete Wavelet Transformed (DWT) low and high frequency sub bands and Wavelet Co-occurrence Texture features (WCT) obtained from two level Discrete Wavelet Transformed (DWT) high frequency sub bands. In the proposed method, the wavelet based optimal texture features that distinguish between the brain tissue, benign tumor and malignant tumor tissue is found. Comparative studies of texture analysis is performed for the proposed combined wavelet based texture analysis method and Spatial Gray Level Dependence Method (SGLDM). Our proposed system consists of four phases i) Discrete Wavelet Decomposition (ii) Feature extraction (iii) Feature selection (iv) Classification and evaluation. The combined Wavelet Statistical Texture feature set (WST) and Wavelet Co-occurrence Texture feature (WCT) sets are derived from normal and tumor regions. Feature selection is performed by Genetic Algorithm (GA). These optimal features are given as input to the PNN classifier to segment the tumor. An Probabilistic Neural Network (PNN) classifier is employed to evaluate the performance of these features and by comparing the classification results of the PNN classifier with the Feed Forward Neural Network classifier (FFNN). The results of the Probabilistic Neural Network, FFNN classifiers for the texture analysis methods are evaluated using Receiver Operating Characteristic (ROC) analysis. The performance of the algorithm is evaluated on a series of brain tumor images. The results illustrate that the proposed method outperforms the existing methods.

Keywords: *Discrete Wavelet Transform(DWT), Genetic Algorithm(GA), Receiver Operating Characteristic(ROC)analysis, Spatial Gray Level Dependence Method (SGLDM), Probabilistic Neural Network (PNN).*

1. INTRODUCTION

In recent years, medical CT Images have been applied in clinical diagnosis widely. That can assist physicians to detect and locate Pathological changes with more accuracy. Computed Tomography images can be distinguished for different tissues according to their different gray levels. The images, if processed appropriately can offer a wealth of information which is significant to assist doctors in medical diagnosis. A lot of research efforts have been directed towards the field of medical image analysis with the aim to assist in diagnosis and clinical studies [1]. Pathologies are clearly identified using automated CAD system [2]. It also helps the radiologist in analyzing the digital images to bring out the possible outcomes of the diseases. The medical images are obtained from different imaging systems such as MRI scan, CT scan, Ultra sound B scan. The computerized tomography has been found to be the most reliable method for

early detection of tumors because this modality is the mostly used in radio therapy planning for two main reasons. The first reason is that scanner images contain anatomical information which offers the possibility to plan the direction and the entry points of radio therapy rays which have to target only the tumor region and to avoid other organs. The second reason is that CT scan images are obtained using rays, which is same principle as radio therapy. This is very important because the intensity of radio therapy rays have been computed from the scanned image. Advantages of using CT include good detection of calcification, hemorrhage and bony detail plus lower cost, short imaging times and widespread availability. The situations include patient who are too large for MRI scanner, claustrophobic patients, patients with metallic or electrical implant and patients unable to remain motionless for the duration of the examination due to age, pain or medical condition. For these reasons, this study aims to explore methods for classifying and segmenting brain CT images. Image segmentation is the process of partitioning a digital image into set of pixels. Accurate, fast and reproducible image segmentation techniques are required in various applications. The results of the segmentation are significant for classification and analysis purposes. The limitations for CT scanning of head images are due to partial volume effects which affect the edges produce low brain tissue contrast and yield different objects within the same range of intensity. All these limitations have made the segmentation more difficult. Therefore, the challenges for automatic segmentation of the CT brain images have many different approaches. The segmentation techniques proposed by Nathali Richarda et al and Zhang et al [3][4] include statistical pattern recognition techniques. Kaiping et al [5] introduced the effective Particle Swarm optimization algorithm to segment the brain images into Cerebro spinal fluid (CSF) and suspicious abnormal regions but without the annotation of the abnormal regions. Dubravko et al and Matesin et al [6] [7] proposed the rule based approach to label the abnormal regions such as calcification, hemorrhage and stroke lesion. Ruthmann.et al [8] proposed to segment Cerobro spinal fluid from computed tomography images using local thresholding technique based on maximum entropy principle. Lunčarić et al proposed [9] to segment CT images into background, skull, brain, ICH, calcifications by using a combination of K means clustering and neural networks. Tong et al proposed[10] to segment CT images into CSF,brain matter and detection of abnormal regions using unsupervised clustering of two stages. Clark et al [11] proposed to segment the brain tumor automatically using knowledge based techniques. From the above literature survey shows that intensity based statistical features are the most straight forward and have been widely used, but due to the complexity of the pathology in human brain and the high quality required by clinical diagnosis, only intensity features cannot achieve acceptable result. In such applications, segmentation based on textural feature methods gives more reliable results. Therefore texture based analysis have been presented for tumor segmentation such as SGLDM method and wavelet based texture features are used and achieve promising results.

Based on the above literature, better classification accuracy can be achieved using wavelet based statistical texture features. In this paper, the authors would like to propose a combination of Wavelet Statistical Texture features (WST) obtained from 2-level Discrete Wavelet Transformed (DWT) low and high frequency sub bands and Wavelet Co-occurrence Texture features (WCT) obtained from two level Discrete Wavelet Transformed (DWT) high frequency sub bands. The extracted texture features are optimized by Genetic Algorithm(GA)[12] for improving the classification accuracy and reducing the overall complexity. The optimal texture features are fed to the PNN[13], FFNN[14] classifiers to classify and segment the tumor region from brain CT images.

2: MATERIALS AND METHODS

Most classification techniques offer intensity based statistical features. The proposed system is divided into 4 phases (i) Discrete Wavelet Decomposition (ii) Feature extraction (iii) Feature selection(iv) Classification and Evaluation. For feature extraction, we discovered two methods which are i) the combination of Wavelet Statistical Texture features (WST) obtained from 2-level Discrete Wavelet Transformed (DWT) low and high frequency sub bands and Wavelet Co-occurrence Texture features (WCT) obtained from two level Discrete Wavelet Transformed (DWT) high frequency sub bands ii) SGLDM method without wavelet transform. Once all the features are extracted, then for feature selection, we use Genetic Algorithm (GA) to select the

optimal statistical texture features. After selecting the optimal texture features, to classify and segment the tumor region from brain CT images using PNN, FFNN classifiers.

2.1 Discrete Wavelet Decomposition

Daubechies wavelet filter of order two is used and found to yield good results in classification and segmentation of tumor from the brain CT images. By applying 2D DWT[15], two level wavelet decomposition of region of interest(ROI) is performed which results in four sub bands. In 2D wavelet decomposition the image is represented by one approximation and three detail images, representing the low and high frequency contents image respectively. The approximation can be further to produce one approximation and three detail images at the next level of decomposition, wavelet decomposition process is shown in Figure 1. LL1, LL2 represent the wavelet approximations at 1st and 2nd level respectively, and are low frequency part of the images. LH1,HL1,HH1,LH2,HL2,HH2 represent the details of horizontal, vertical and diagonal directions at 1st and 2nd level respectively, and are high frequency part of the images.

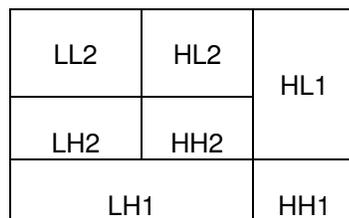


FIGURE 1: Wavelet Image decomposition at 2nd level

Among the high frequency sub bands, the one whose histogram presents the maximum variance is the sub band that represents the most clear appearance of the changes between the different textures. The WST features are extracted from the 2nd level of both low and high frequency sub bands and WCT features are extracted from 2nd level of high frequency sub bands are useful to classify and segment the tumor region from brain CT images.

2.2 Feature Extraction

Texture analysis is a quantitative method that can be used to quantify and detect structural abnormalities in different tissues. As the tissues present in brain are difficult to classify using shape or intensity level of information, the texture feature extraction is founded to be very important for further classification. The purpose of feature extraction is to reduce original data set by measuring certain features that distinguish one region of interest from another. The analysis and characterization of textures present in the medical images can be done by using the combination of Wavelet Statistical Texture features (WST) obtained from 2-level Discrete Wavelet Transformed (DWT) low and high frequency sub bands and Wavelet Co-occurrence Texture features (WCT) obtained from two level Discrete Wavelet Transformed (DWT) high frequency sub bands.

Algorithm for Feature Extraction is as Follows

- Obtain the sub-image blocks, starting from the top left corner.
- Decompose sub-image blocks using 2-D DWT.
- Derive SGLDM or Co-occurrence matrices [16] for two level high frequency sub bands of DWT with 1 for distance and 0,45,90 and 135 degrees for θ and averaged.
- From these co-occurrence matrices, the following nine Haralick texture features [17] called wavelet Co-occurrence Texture features(WCT) are extracted .
- The Wavelet Statistical Texture features (WST) are extracted from 2 level Discrete Wavelet Transformed (DWT) low and high frequency sub bands.
- Combination of the both the WST features and WCT features are used for classification.

Then the feature values of both features are normalized by subtracting minimum value and dividing by maximum value minus minimum value. Maximum and minimum values are calculated based on the training data set. In the data set, if the feature value is less than the minimum value, it is set to minimum value. If the feature value is greater than the maximum value, it is set to maximum value. Normalized feature values are then optimized by feature selection algorithm. Table 1 shows the combination of both WST WCT features extracted using SGLDM method .

Sl.No	Second order WCT features
WST FEATURES	
1	Mean(MN)
2	Standard Deviation(SD)
3	Energy(ENER)
WCT FEATURES	
1	Entropy-ENT (Measure the disorder of an image)
2	Energy- ENE (Measure the textural uniformity)
3	Contrast-CON (Measure the local contrast in an image)
4	Sum Average-SA (Measure the average of the gray level within an image)
5	Variance -VAR (Measure the heterogeneity of an image)
6	Correlation-COR (Measure a correlation of pixel pairs on gray levels)
7	Max probability-MP (Determine the most prominent pixel pair in an image)
8	Inverse Difference Moment - IDM (Measure the homogeneity of an image)
9	Cluster tendency-CT (Measure the grouping of pixels that have similar

TABLE 1: WST and WCT Features extracted using SGLDM method

2.3 Feature Selection

Feature selection is the process of choosing subset of features relevant to particular application and improves classification by searching for the best feature subset, from the fixed set of original features according to a given feature evaluation criterion(ie., classification accuracy). Optimized feature selection reduces data dimensionalities and computational time and increase the classification accuracy. The feature selection problem involves the selection of a subset of features from a total number of features, based on a given optimization criterion. T denotes the subset of selected features and V denotes the set of remaining features. So, $S = T \cup V$ at any time. $J(T)$ denotes a function evaluating the performance of T. J depends on the particular application. Here $J(T)$ denotes the classification performance of classifying and segmenting tumor region from brain CT images using the set of features in T. In this work, genetic algorithm (GA) technique is used.

Genetic Algorithm

We consider the standard GA to begin by randomly creating its initial population. Solutions are combined via a crossover operator to produce offspring, thus expanding the current population of solutions. The individuals in the population are then evaluated via a fitness function, and the less fit individuals are eliminated to return the population to its original size. The process of crossover, evaluation, and selection is repeated for a predetermined number of generations or until a satisfactory solution has been found. A mutation operator is generally applied to each generation in order to increase variation. In the feature selection formulation of the genetic algorithm ,individuals are composed of chromosomes: a 1 in bit position indicates that feature should be selected; 0 indicates this feature should not be selected. As an example the chromosome 00101000 means the 3rd and 5th features are selected. That is the chromosome represents $T=\{3,5\}$ and $V=\{1,2,4,6,7,8\}$. Fitness function for given chromosome T is defined as

$$\text{Fitness}(T) = J(T) - \text{penalty}(T) \quad (1)$$

where T is the corresponding feature subset, and $\text{penalty}(T) = w * (|T| - d)$ with a penalty coefficient w . The size value d is taken as a constraint and a penalty is imposed on chromosomes breaking this constraint. The chromosome selection for the next generation is done on the basis of fitness. The fitness value decide whether the chromosome is good or bad in a population. The selection mechanism should ensure that fitter chromosomes have a higher probability survival. So, the design adopts the rank-based roulette-wheel selection scheme. If the mutated chromosome is superior to both parents, it replaces the similar parent. If it is in between the two parents, it replaces the inferior parent; otherwise, the most inferior chromosome in the population is replaced. The selected optimal feature set based on the test data set is used to train the PNN, FFNN classifiers to classify and segment the tumor region from brain CT images. Table 2 shows the best features selected using Genetic Algorithm(GA) during the execution.

SI-NO	Feature set	Classification accuracy
1	IDM,ENT, ENE, VAR,CON,ENER	96%
2	IDM,CON,ENE, MP, VAR,SD	96%
3	ENT,IDM,VAR, IDM, CT,ENER	97%
4	IDM,ENT, CT, ENE,CON,ENER	97%
5	CON, IDM,VAR,ENT,ENE,MN	96%
6	ENT,SA, IDM, ENE,VAR,ENER	97%
7	VAR,ENT, ENE, SA,IDM,ENER	97%
8	ENT,CON,ENE,VAR,IDM,ENER	97%
9	IDM,ENT, CT,CON, VAR,MN	96%
10	ENE,ENT, MP,CON, COR,SD	96.5%

TABLE 2: Feature set generated by Genetic algorithm

The WCT texture features like energy (ENE), entropy (ENT), variance (VAR), inverse difference moment (IDM) and WST features like energy are present in most of the feature set generated by GA. The features such as CT,CON,SA,MP,COR,SD,MN which are least significant. The classification accuracy of 97% is obtained with five of the available 39 features using GA. Therefore this minimum number of features are possible to classify and segment the tumor region from brain CT images.

2.4 PNN Classifier

Classification is the process where a given test sample is assigned a class on the basis of knowledge gained by the classifier during training. Probabilistic Neural Networks (PNNs) is a widely used classification methodology, because of its simplicity, robustness to noise, fast training speed, fast online speed, no local minima issues and training samples can be added or removed without extensive retraining. Their main task is the classification of unknown feature vectors into predefined classes, where the Probability Density Function (PDF) of each class is estimated by kernel functions. PNNs are supervised neural network models, closely related to the Bayes classification rule and Parzen nonparametric probability density function estimation theory. Their training procedure consists of a single pass over all training patterns, thereby rendering PNNs faster to train, compared to the Feed forward Neural Networks (FFNNs).

The PNN architecture is composed of many interconnected processing units or neurons organized in successive layers: input layer, pattern layer, summation layer, decision layer or output layer. The input layer unit does not perform any computation and simply distributes the input to the neurons in the pattern layer. On receiving a pattern X from the input layer, the neuron X_{ij} of the pattern layer computes its output. The output of pattern layer is represented by

$$Y_{ij}(X) = \frac{e^{-|x-x_{ij}|^2 / \sigma^2}}{\sigma^2} \quad i, j = 1, 2, \dots, n \quad (2)$$

n represents number of training sets, σ is the smoothing parameter. The summation layer neurons compute the pattern X being classified into G_i by summing and averaging the output of all neurons that belong to the same class.

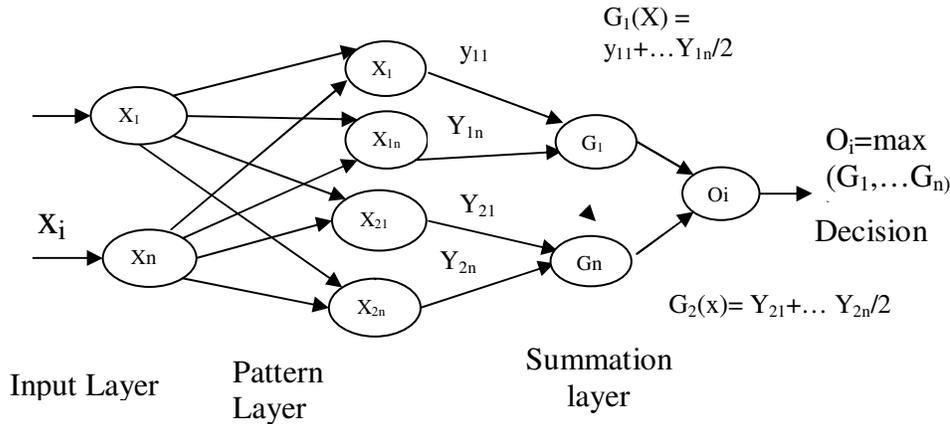


FIGURE 2: Architecture of probabilistic neural network

The output of summation layer is represented by

$$G_i(X) = \frac{1}{n_i} \sum_{k=1}^{n_i} e^{-|x-x_{ik}|^2 / \sigma^2} \quad i = 1 \dots n, X = X_1, X_2, \dots, X_n \quad (3)$$

where n_i denotes the total number of training sets or patterns in class G_i . If the a priori probabilities for each class are the same, and the losses associated with making an incorrect decision for each class are the same, the decision layer unit classifies the pattern X in accordance with Bayes's decision rule based on the output of all the summation layer neurons:

$$O_i(X) = \max(G_i(X)), \quad i = 1 \dots n \quad X = X_1, X_2, \dots, X_n \quad (4)$$

where $O_i(X)$ denotes the estimated class of the pattern X and n is the total number of classes in the training sets or patterns.

2.4.1 Segmentation of Tumor Region

Segmentation is important in selecting the sub band of the image to be decomposed. The segmentation is done by the combination of WST features are extracted from 2 level discrete wavelet transformed low and high frequency sub band and WCT features are extracted from two level discrete wavelet transformed high frequency sub bands as given in the feature extraction algorithm and the optimal texture feature set is selected by GA based on the classification performance of PNN, FFNN classifiers. From the experiments conducted for feature selection, it is found that the optimal feature set which gives good classification performance are WST features like energy and the second order WCT features like energy entropy, variance, inverse difference moment. The four WCT texture features from high frequency sub bands and one WST texture feature from low and high frequency sub bands forms the feature set. These feature set is given as input to the PNN, FFNN classifiers to segment the tumor region from

brain CT images. Efficiency or accuracy of the classifiers for each texture analysis method is analyzed based on the error rate. This error rate can be described by the terms true and false positive and true and false negative as follows:

True Positive (TP) : The test result is positive in the presence of the clinical abnormality.

True Negative (TN): The test result is negative in the absence of the clinical abnormality.

False Positive (FP): The test result is positive in the absence of the clinical abnormality.

False Negative(FN): The test result is negative in the presence of the clinical abnormality. Based on the above terms, to construct the table called contingency table.

Actual group	Predicted group	
	Normal	Abnormal
Normal	TN	FP
Abnormal	FN	FP

TABLE 3: Contingency table of classifier performance.

$$\text{Sensitivity} = TP / (TP + FN)$$

$$\text{Specificity} = TN / (FP + TN)$$

$$\text{Accuracy} = (TP+TN)/(TP + TN + FN + FP)$$

Sensitivity measures the ability of the method to identify abnormal cases. Specificity measures the ability of the method to identify normal cases. Correct classification rate or accuracy is the proportion of correct classifications to the total number of classification tests. The PNN, FFNN classifiers were tested by using leave one out cross validation method. Leave one out cross validation can be used as a method to estimate the classifier performance in unbiased manner. Here each step, one data set is left out and the classifier is trained using the rest and the classifier is applied to the left out data set. This procedure is repeated such that each data set is left out once. In our application, to evaluate the classification accuracy of the classifiers, the 3,5, 10 fold cross validation is done on the data set collected from 100 images. Classification accuracy is calculated by taking the average number of all the correct classifications. Other statistical method known as Receiver Operating Characteristics (ROC) analysis [18] is also used to analyze the experimental results of the classifiers for each texture analysis method. In this method, the data set is divided randomly into 5 groups of 100 images with 50 benign tumor and 50 malignant tumor images. Each group consists of 10 benign and 10 malignant tumor images. Sensitivity and specificity values are recorded for each group and the ROC curve is drawn and analyzed. Depending on the training set, each group will have a different threshold value for determining true positive and true negative cases. The ROC curve is a graphical representation of sensitivity versus specificity as a threshold parameter is varied. The Area Under ROC Curve (AUC) has been calculated and that AUC value is used to determine the overall classification accuracy of a specific classifier. The larger the area (the higher AUC value) means higher the classification performance. In this research, the ROC analysis and classification accuracy are used to measure the performance of the classifiers based on the different texture analysis methods.

3. RESULTS

Our proposed method is implemented on real human brain CT dataset with the two different types of tumor images of 30 patients .We have tested our system on segmentation for two types of benign, malignant tumor images. The output of the image is compared with the ground truth (target). Ground truth was obtained from the boundary drawings of the radiologist. The segmentation results of 2 different slices from two different patients with benign and malignant tumor images are as shown in Figure 3. From the obtained results, this segmentation algorithm

is effectively workable on the acquired CT images. The input data set consists of 100 images: 50 images are benign tumor images, 50 images are malignant tumor images. For each texture analysis methods, input data set is partitioned into training and test sets which are classified using PNN, FFNN classifiers. This section describes the wavelet based texture analysis method of segmenting tumor region from brain CT images.

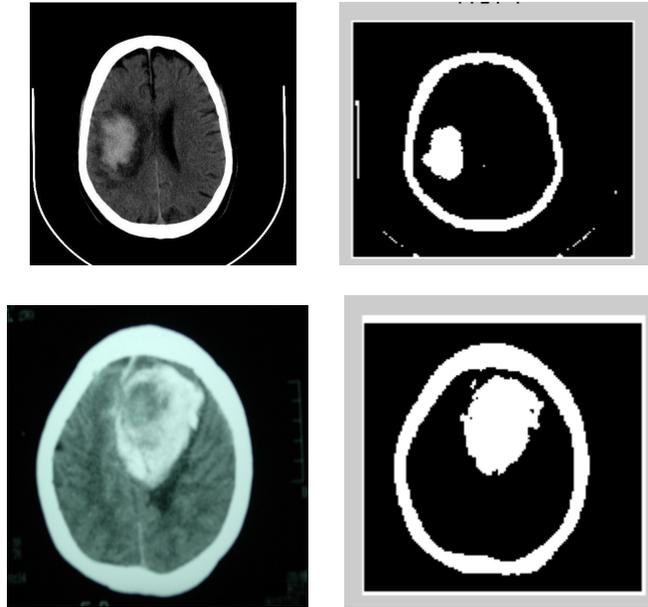


FIGURE: 3 Segmentation of tumor for sample images

There is an outstanding issue associated with the PNN concerning network structure determination (i.e) determining the network size, the number of pattern layer neurons as well as the value of the smoothing parameter. The PNN (trained on 16 extracted features) implemented for classification of the brain CT images had 16 neurons in the input layer for giving 16 extracted features (WST features 1×4 and WCT feature $4 \times 3 = 12$, i.e $4 + 12 = 16$) as inputs, 50 pattern layer neurons as training set contains 50 feature vectors and one summation layer neuron corresponding to one segmented tumor region, and one decision layer neuron to make a one class Bayesian decision. The objective is to select representative pattern layer neurons from the training samples. The output of a summation layer neuron becomes a linear combination of the outputs of pattern layer neurons. Subsequently, an orthogonal algorithm was used to select pattern layer neurons. The smoothing parameter sigma was determined based on the minimum misclassification rate computed from the partial evaluation data set. The minimum misclassification rate was attained at sigma equal to 0.04.

Feature selection is carried out using GA. There are 9 WCT features are extracted from two level discrete wavelet high frequency sub bands and 3 WST features are extracted from 2 level wavelet transformed low and high frequency sub bands. So totally $9 \times 3 + 3 \times 4 = 39$ features are extracted. The next step is to determine the relevance of each selected feature to the process

Parameter used	Wavelet domain	Gray level domain
TP	49	47
FN	48	47
TP	2	4
FN	1	2
Sensitivity in %	98%	95.9%
Specificity in %	96%	92.15%
Accuracy in %	97%	94%

TABLE 4: Classification performances of the PNN classifier for 100 images

of segmenting the tumor region. During the evaluation process by using GA, some features may be selected many times as the number of generation increases. If the feature was selected more times in the feature set that feature was given as more important in the feature selection. The number of times the WCT features selected were energy, entropy, variance and inverse difference moment and WST feature was energy. The parameter set for the GA algorithm is as follows: Population size is 30; Cross Over probability is 1.0; Mutation rate is 0.1; Penalty coefficient is 0.5 and stopping condition is 100 generations.

Parameter used	Wavelet domain	Gray level domain
TP	48	47
FN	47	46
TP	3	4
FN	2	3
Sensitivity in %	96%	94%
Specificity in %	94%	92%
Accuracy in %	95%	93%

TABLE 5: Classification performances of FFNN classifier for 100 images

A comparative study of the classification accuracy was performed for both combined wavelet based texture analysis method and Spatial Gray Level Dependence Matrix method. To justify the choice of using wavelet domain for the same data set, without applying wavelet transform in the gray level domain, the performance of the PNN classifier is as shown in Table 4 and the performance of the FFNN classifier is as shown in Table 5. The accuracy of the PNN, FFNN classifiers in the wavelet domain are 97%,95% and in the gray level domain are 94%, 93% respectively. From the statistical analysis, PNN and FFNN classifiers in wavelet domain have better accuracy compared to the classifiers in gray level domain.

Cross validation method	Classification accuracy of PNN classifier	Classification accuracy of FFNN classifier
3 fold	96%	94%
5 fold	96.6%	94.6%
10 fold	97%	95%

TABLE 6: Classification results of classifiers with different cross validation methods

Table 6 shows the classification performances of the classifiers with different cross validation methods. The accuracy of PNN with 3,5,10 fold cross validation methods are 96%,96.6%,97% and the accuracy of FFNN with 3,5,10 fold cross validation methods are 94%,94.6%,95% respectively for the same training and testing data sets. Results show that, we get good

classification accuracy for the 10 fold cross validation method in both PNN, FFNN classifiers. A comparison is made between the PNN and FFNN classifiers with different texture analysis methods using the performance criteria as the area under the Roc value for detecting and segmenting the tumor region. The results of comparisons are represented using Roc analysis graph as shown in Figure 4 and Figure 5. The AUC values of PNN, FFNN classifier in wavelet domain are 0.974,0.938 and in gray level domain are 0.936,0.928 respectively. From the results, it is observed that the performance of the classifiers in wavelet method is better both quantitatively and qualitatively compared to the classifiers in SGLDM method.

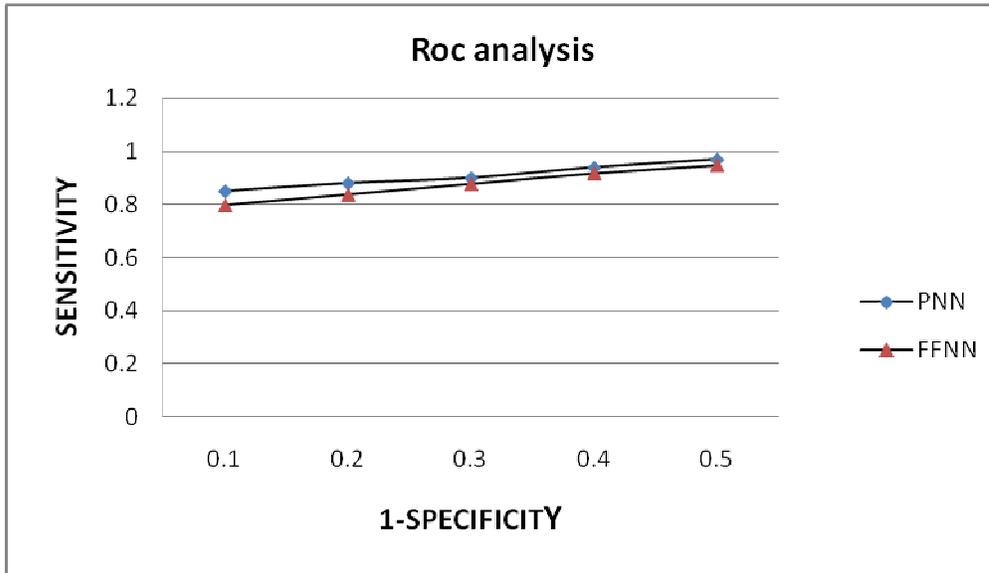


FIGURE 4 : Roc analysis curve of classifiers in wavelet domain

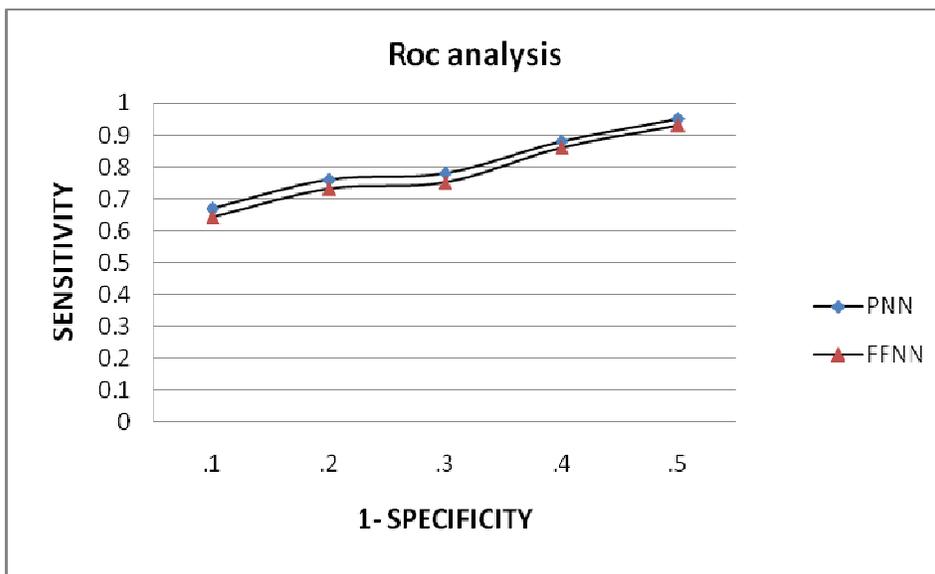


FIGURE 5: ROC analysis curve of classifiers in gray level domain

Table 7 shows the classification performances of our proposed combined technique and SGLDM method. The classification accuracy of our proposed method is 97% is high while

compared with the SGLDM method. The proposed system which may be valuable especially in cases of small region of benign, malignant brain tumor images.

SI-NO	Technique	Classification accuracy
1	WT+SGLDM+GA+ PNN	97%
2	SGLDM+GA+PNN	94%
3	WT+SGLDM+GA+FFNN	95%
4	SGLDM+GA+FFNN	93%

TABLE 7: Classification accuracy of the proposed technique

4. DISCUSSION AND CONCLUSION

As a conclusion, we have presented a method for combined WST and WCT based texture feature extraction method and selecting the optimal texture features using GA, and evaluated the PNN, FFNN classifiers to segment the tumor region. The algorithm has been designed based on the concept of different types of brain soft tissues have different textural features. This method effectively works well for segmentation of tumor region with high sensitivity, specificity and accuracy. The results show that the segmentation by the combined feature extraction method yields better results compared to the SGLDM method without using wavelet transform based on the PNN, FFNN classifiers. It is found that this method gives favorable result with accuracy percentage of above 97% for the images that are being considered. This would be highly useful as a diagnostic tool for radiologists in the automated segmentation of tumor region from brain CT images.

The goal of this work is to compare the classification performance of the PNN classifier using combined wavelet feature extraction method and SGLDM method. Hence it is concluded that the neural network supported by conventional image processing operations can be effectively used for segmentation of tumor region from CT images. Use of large data bases is expected to improve the system robustness and ensure the repeatability of the resulted performance. The automation procedure proposed in this work using a PNN enables proper abnormal tumor region detection and segmentation there by saving time and reducing the complexity involved. The proposed system may be particularly useful in small tumor regions, where segmentation of tumor in these two types of brain tumor images is radio logically difficult. The work can be extended to get 100% segmentation accuracy by using other classifiers such as Radial basis function neural network with Particle Swarm Optimization as a future work. The developed segmentation system is expected to provide valuable diagnosis for the physicians.

5. REFERENCES

- [1] Duncan J.S.,Ayache N, "Medical Image Analysis Progress Over two decade and challenges ahead", *IEEE Trans on PAMI* ,Vol 22,pp. 85 – 106,2000.
- [2] Tourassi G D, "Journey towards computer aided Diagnosis – Role of Image Texture Analysis", *Radiology* , Vol 213,pp.317 – 320,1999.
- [3] Nathalii Richards, Michael Dujata, Catherine Garbay, "Distributed Markovian segmentation : Application to MR brain scans", *Journal of Pattern Recognition* , Vol 40,pp. 3467 – 3478,2007.
- [4] Zhang Y, Brady M, Smith S, "Segmentation of Brain MR images through Hidden Markov Random field Model and the expectation maximization algorithm", *IEEE Transactions on Medical Imaging* , Vol 20,pp. 45-57,2001.

- [5] Kaiping Wei, Bin He, Tao Zhang, Xianjun Zhen, "A Novel Method for Segmentation of CT Head Images", *1st International conference on Bioinformatics and Biomedical Engineering; Wuhan: IEEE Explore*; p 717 – 720, 2007 July 6-8.
- [6] Dubravko Cosic ,Sven Loncaric, "Rule based labeling of CT head image"; *6th conference on Artificial Intelligence in Medicine, Europe: Springer* ; p.453 – 456, 1997 March 23-26.
- [7] Matesn Milan, Loncaric Sven, Petravic Damir, " Rule based approach to stroke lesion analysis from CT brain Images", *2nd International Symposium on Image and Signal Processing and Analysis; Pula, Croatia : IEEE Explore* ; p.219 – 2231, , 2001 June 19-21.
- [8] Ruthmann,V.E, Jayce E.M, Reo D.E, Eckardit M.J, "Fully Automated segmentation of cerebrospinal fluid in computed tomography ", *Psychiatry Research : Neuro Imaging* , Vol 50 ,pp. 101 – 119,1993.
- [9] Loncaric S and Kova Cevic D, "A Method for segmentation of CT head images", *Lecture Notes on Computer Science* ,Vol 1311,pp.1388 – 305,1997.
- [10] Tong Hau Lee, Mohammad Faizal, Ahmad Fauzi and Ryoichi Komiya, " Segmentation of CT brain images using unsupervised clusterings", *Journal of Visualization*, Vol 12,pp.131-138,2009.
- [11] Clark M C., Hall L O., Goldgof D B., Velthuzien R., Murtagh F R., and Silbiger M S., "Automatic tumor segmentation using knowledge based techniques", *IEEE Transactions on Medical Imaging*, Vol 17,pp.187-192,1998.
- [12] Frank Z. Brill., Donald E. Brown., and Worthy N. Martin, "Fast Genetic Selection of features for Neural Network Classifiers", *IEEE Trans. Neural Networks*, Vol 3,pp. 324-328,1993.
- [13] Xiukun Yang., Finger print Smear Detection based on sub band Feature representation. *Eurasip Journal on Advances in Signal processing* , Vol 5,pp. 325-339,2011.
- [14] Fausett L, "Fundamentals of Neural Networks: Architectures Algorithms and Applications", *Englewood cliffs, NJ: Prentice Hall International*, p. 289-293,1994.
- [15] Van G., Wouver P.,Scheunders and D.Van Dyck, "Statistical texture characterization from discrete wavelet representation", *IEEE Trans. Image processing*,Vol 8,pp.592-598,1999.
- [16] Haddon J F, Boyce J F, "Co-occurrence Matrices for Image analysis", *IEE Electronic and Communications Engineering Journal* , Vol 5,pp. 71 – 83,1993.
- [17] Haralick R M, Shanmugam K and Dinstein I, "Texture features for Image classification", *IEEE Transaction on System, Man, Cybernetics* ,Vol 3,pp. 610 – 621,1973.
- [18] Tom Fawcett , "An introduction to ROC analysis", *Pattern Recognition Letters* , Vol 27,pp. 861-874,2006.

Atmospheric Correction of Remotely Sensed Images in Spatial and Transform Domain

Priti Tyagi

*Associate Professor
PVPP College of Engineering
Mumbai University India*

tpriti15@rediffmail.com

Dr. Udhav Bhosle

*Principal
Rajiv Gandhi Institute of Technology
Mumbai University India*

udhavbhosle@gmail.com

Abstract

Remotely sensed data is an effective source of information for monitoring changes in land use and land cover. However remotely sensed images are often degraded due to atmospheric effects or physical limitations. Atmospheric correction minimizes or removes the atmospheric influences that are added to the pure signal of target and to extract more accurate information. The atmospheric correction is often considered critical pre-processing step to achieve full spectral information from every pixel especially with hyperspectral and multispectral data. In this paper, multispectral atmospheric correction approaches that require no ancillary data are presented in spatial domain and transform domain. We propose atmospheric correction using linear regression model based on the wavelet transform and Fourier transform. They are tested on Landsat image consisting of 7 multispectral bands and their performance is evaluated using visual and statistical measures. The application of the atmospheric correction methods for vegetation analyses using Normalized Difference Vegetation Index is also presented in this paper.

Keywords: Atmospheric Correction, Multispectral, Spatial Domain, Transform Domain, Vegetation Analyses.

1. INTRODUCTION

The atmosphere influences the amount of electromagnetic energy that is sensed by the detectors of an imaging system and these effects are wavelength dependent. This is particularly true for imaging systems such as Landsat Multispectral Scanner (MSS) and Thematic Mapper (TM) that record data in the visible & near infrared parts of the spectrum. When electromagnetic radiation travels through the atmosphere, it may be absorbed or scattered by the constituent particles of the atmosphere. Atmospheric absorption affects mainly the visible and infrared bands. It reduces the solar radiance within the absorption bands of the atmospheric gases. Atmospheric scattering is important only in the visible and near infrared regions. Scattering of radiation by the constituent gases and aerosols in the atmosphere causes degradation of the remotely sensed images. Most noticeably, the solar radiation scattered by the atmosphere towards the sensor without first reaching the ground produces a hazy appearance of the image. This effect is particularly severe in the blue end of the visible spectrum due to the stronger Rayleigh scattering for shorter wavelength radiation. Atmospheric absorption has multiplicative effect & atmospheric scattering has additive effect on the data.

Several different Atmospheric Scattering or haze removal techniques have been developed for use with digital remotely sensed data. Most of the methods use various atmospheric transmission models, in situ field data, or require specific targets to be present in the image. [7] [8] [9]. Major limitation with these sophisticated techniques is that they require information other than digital Image data. [e.g., path radiance and (or) atmospheric transmission at several locations within the image area collected during satellite's overflight. Ideally, a method that uses in situ or ground

truth information is most accurate in terms of correcting for atmospheric haze effects. However most of the users work with remotely sensed data that has already been collected. Most of the time only data available is the image itself. Image-based radiometric correction methods are simple and effective as they require no ancillary data to estimate the path radiance and sensor offset terms. Hence we propose image based atmospheric correction methods which do not require any information about the camera, Image acquisition and imaging conditions. Eight methods wherein, three methods (Spatial domain) namely Standard Dark Object Subtraction Technique, Improved Dark Object Subtraction Technique and Linear Regression Method, along with five methods in transform domain namely Wavelet Thresholding, Homomorphic filtering, DCT, Wavelet Regression and Fourier Regression are presented in this paper along with performance evaluation of each method. Results are assessed statistically and compared with each other.

The rest of this paper is organized as follows: Section 2 describes Image based atmospheric correction methods in spatial domain wherein Section 3 describes proposed Transform Domain methods. Section 4 gives details of application of the atmospheric correction methods for vegetation analyses using NDVI method. In section 5 we present the results based on visual and statistical measures. Finally section 6 concludes the paper.

2 : IMAGE BASED ATMOSPHERIC CORRECTION METHODS (SPATIAL DOMAIN METHODS)

2.1 Simple Dark Object Subtraction Method

Dark object subtraction Technique removes the effects of scattering from the image data. It requires only the information contained in the digital image data. It derives the corrected DN (Digital Number) values solely from the digital data with no outside information[3].

Dark-object subtraction (DOS) is a widely used method of reducing haze within an image and is done for each band individually. It is assumed that there are pixels within each band of a multispectral image that have very low or no reflectance on the ground, and that the difference between the brightness value of these pixels and zero is due to haze. This per-band estimated difference is subtracted from each band of the image. Most dark object subtraction technique assumes that there is a high probability that there are at least a few pixels within an image which should be black (0% Reflectance) [2]. This assumption is made because in a single band there are large number of pixels (Landsat MSS single band images- over 7 million pixels and Landsat TM single band images- over 45 million pixels). Thus there are some shadows due to topography or clouds in the image where pixels should be completely dark. Ideally, the imaging system should not detect any radiance at these shadow locations and a DN of zero should be assigned to them. However because of atmospheric scattering, the imaging system records a non zero DN value at these supposedly dark shadowed pixel locations. This represents the DN value that must be subtracted from the particular spectral band to remove first order scattering component.

Haze DN value is directly selected from the DN frequency Histogram of a digital Image. A different constant is used for each spectral Band with a different set of constants used from image to image. Histogram of given spectral Bands, particularly in the visible spectrum will offset towards higher DN values by some amount due to scattering. There is usually very sharp increase in the number of pixels at some non zero DN or Gray Level X. This DN value is the amount of Haze in that particular Band. Haze DN value is subtracted from the respective spectral Band.

2.2 : Improved Dark Object Subtraction Technique

The technique of Dark object haze correction is further improved in this method. [2][3] DN value selected for haze removal using standard dark object subtraction technique may not conform to a realistic relative atmospheric scattering model. Hence problems are encountered in the analysis stage if the digital multispectral image data is haze corrected using the standard Dark object subtraction technique. This lack of conformity may cause the data to be

overcorrected in some or all the spectral bands and the relationship between the bands will not be corrected.

In improved dark object subtraction method, we select a starting band dark object subtraction haze value using the Histogram of one of the spectral bands. Relative scattering model that best represents atmospheric conditions at the time of data collection is then selected. The amplitude of the starting Haze can be used as a guide to identify the type of atmospheric condition that existed during data collection (i.e., very clear, clear, moderate, hazy, very hazy). The selected relative scattering model is then used to predict the haze values for the other spectral bands from the starting haze values.

Two well known relative scattering models are the Rayleigh and Mie models [12]. The haze correction values used by dark object subtraction technique should be computed using a relative scattering model which ensures that the haze values represent true atmospheric scattering possibilities. One possible set of relative scattering model is given in Table(1)

Atmospheric Conditions	Relative Scattering Model
Very Clear	λ^{-4}
Clear	λ^{-2}
Moderate	λ^{-1}
Hazy	$\lambda^{-0.7}$
Very Hazy	$\lambda^{-0.5}$

TABLE 1 : Relative Scattering Model [2]

The histogram method is used to identify initial or starting haze value for one band and then a relative scattering model is used to predict the haze values of other bands. These values are then used to do dark object correction. The relative scattering model is used to predict the haze values for the spectral bands being used, given the haze value of one band. The relative scattering model is not used to compute the path radiance values from scratch.

The spectral Bandwidth of the individual bands affects the amount of radiance detected. TM Bands 1, 2, 3 are affected most by scattering [2]. Using the above relative scattering model haze values are calculated.

This haze value is used as a guide to help select the relative scattering model that best represent the atmospheric conditions. Table 2 shows the multiplication factors needed to compute or predict the haze values for Landsat TM nonthermal bands when TM band 1 is selected as the starting haze value [2].

The haze values for other bands are calculated using the following Table 2.

TM Bands	Average Wavelength λ^{-4}	Very Clear λ^{-4}	Clear λ^{-2}	Moderate λ^{-1}	Hazy $\lambda^{-0.7}$	Very Hazy $\lambda^{-0.5}$
1	0.485	1.000	1.000	1.000	1.000	1.000
2	0.560	0.563	0.750	0.866	0.905	0.930
3	0.660	0.292	0.540	0.735	0.807	0.857
4	0.830	0.117	0.342	0.584	0.687	0.765
5	1.650	0.075	0.086	0.294	0.424	0.542
7	2.215	0.002	0.048	0.219	0.345	0.468

TABLE 2 : Multiplication factors to produce Haze values.[2]

2.3 Regression Line Method

Linear Regression Model

The regression intersection method of minimizing the effect of the atmosphere provides absolute results from the image data without the use of ancillary data. The method does not require any information or assumptions about the scene, atmospheric conditions or sensor calibrations.

The method generally involves calculation of regression lines for a number of surface materials of contrasting spectral properties. The regression line method (RLM) determines a 'best fit' line for multispectral plots of pixels within homogenous cover types. Ideally, the intersection of lines must represent a point of zero ground reflectance since this is the only point at which radiometric values of two spectrally different materials can be safe. If no atmospheric scattering has taken place, the intersection of the line would be expected to pass through the origin. The slope of the plot is proportional to the ratio of the reflective material. However, the lines will, in reality, intersect the x and y axis producing two offset values. These brightness values represent the amount of bias caused by atmospheric scattering. Crippen (1987) recommends the collection of a series of training areas resulting in many regression lines intersecting in two dimensional spaces at the same point using training sets to represent homogeneous land cover types [10]. The relative values generated by regression method tend to be more reliable. However, many new high-resolution satellites provide data which is spectrally and spatially different from Landsat derived data.

General Linear Models

The GLM (General Linear Model) relates a set of independent variables (X_1 through X_p) to a set of dependent variables (Y_1 through Y_q). Two special cases of the GLM recorded in literature are bivariate regression and multivariate regression.

Bivariate Regression: Two Parameter (β_0 and β_1) Model

If there is only one X and only one Y, then the GLM simplifies to the simple bivariate linear correlation/regression. The least squares criterion is applied to reduce the squared deviations between observed Y and predicted \hat{Y} to the smallest value possible for a linear model.

Let x_1, x_2, \dots, x_n be specific settings of the predictor variable. Let y_1, y_2, \dots, y_n be the corresponding values of the response variable. Assume that Y_i is the observed value of a random variable Y_i , which depends on x according to the following model:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad (i = 1, 2, \dots, n)$$

Here ε_i is the random error with $E(\varepsilon_i) = 0$ and $\text{Var}(\varepsilon_i) = \sigma^2$.

Thus, $E(Y_i) = \mu_i = \beta_0 + \beta_1 x_i$ (true regression line)

The x_i 's usually are assumed to be fixed (not random variables).

We need to find the line, i.e., values of β_0 and β_1 that minimizes the sum of the squared deviations:

$$Q = \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]^2$$

Solve for values of β_0 and β_1 for which $\frac{\delta Q}{\delta \beta_0} = 0$ and $\frac{\delta Q}{\delta \beta_1} = 0$

Finding Regression coefficients

$$\frac{\delta Q}{\delta \beta_0} = -2 \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]$$

$$\frac{\delta Q}{\delta \beta_1} = -2 \sum_{i=1}^n x_i [y_i - (\beta_0 + \beta_1 x_i)]$$

The normal equations are given by

$$n\beta_0 + \beta_1 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$\beta_0 \sum_{i=1}^n x_i + \beta_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

Finding solution to Normal Equations we get

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$$

The coefficient " β_0 " is the Y-intercept, and " β_1 " is the slope, the average amount of change in Y per unit change in X.

Algorithm for Regression Line Method

Regression line method (RLM) suggested here uses one-independent variable regression model described in previous section for estimation of path radiance. The band values, for which correction coefficient is to be determined, regressed against higher spectral bands over homogeneous area. The band to be corrected is plotted on y axis and estimated y intercept is considered as correction coefficient as it is assumed that it equals zero-ground radiance. Threshold for mask to select homogeneous area is determined using histogram of band 5.

- TM Band 5 data is corrected by Improved dark object subtraction method.
- TM Band 4 values are corrected using RLM where Band 5 values are repeated on the independent variable's axis.
- RLM is again applied to correct Band 3, Band 2 and Band 1, using band 5 as independent variable.

3 Transform Domain Methods

3.1 Wavelet Thresholding Method

As a consequence of atmosphere on remotely sensed images, the images are corrupted by blur and noise. Here, we assume that the image degradation can be described by a linear space-invariant blurring operator and additive Gaussian noise.

To remove atmospheric effects without excessive smoothing of important details, a denoising algorithm needs to be *spatially adaptive*. The wavelet representation, due to its sparsity, edge detection and multiresolution properties, naturally facilitates such spatially adaptive noise filtering.

Discrete Wavelet Transform

The discrete wavelet analysis is a two channel digital filter bank (composed of the lowpass and the highpass filters), iterated on the lowpass output. The lowpass filtering yields an *approximation* of a signal (at a given scale), while the highpass (more precisely, bandpass) filtering yields the *details* that constitute the difference between the two successive approximations. A family of wavelets is then associated with the bandpass, and a family of scaling functions with the lowpass filters. Mallat has introduced a fast, pyramidal filter bank algorithm [Mallat89b] for computing the coefficients of the orthogonal wavelet representation; later it was generalized for the biorthogonal case. This algorithm, is in literature usually referred to as the discrete wavelet transform (DWT).

$$\frac{1}{\sqrt{2}} \varphi\left(\frac{x}{2}\right) = \sum_{k \in \mathbb{Z}} h_k \varphi(x - k)$$

$$\frac{1}{\sqrt{2}} \psi\left(\frac{x}{2}\right) = \sum_{k \in \mathbb{Z}} g_k \varphi(x - k)$$

First equation is called the *dilation equation*, *two-scale equation* or the *scaling equation*, while the second one is referred to as the *wavelet equation*. The sequences **h** and **g** can be interpreted as discrete filters.

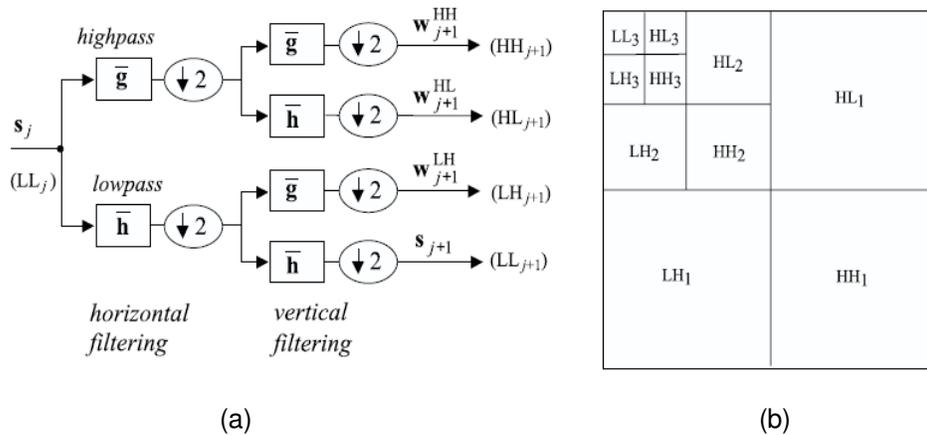


FIGURE 1 : Two dimensional DWT. Decomposition step (a) and the usual organization of the subbands (b).

The fast algorithm is a straightforward extension of the one in this Section, where the filter banks are applied successively to the rows and to the columns of an image. A decomposition step is shown in Fig. 1(a), and a usual representation of the frequency subbands in Fig. 1(b). The DWT of an image yields fairly well decorrelated wavelet coefficients. Large-magnitude coefficients tend to occur near each other within subbands, and also at the same relative spatial locations in subbands at adjacent scales and orientations, in [Simoncelli99]. The positions of the large wavelet coefficients indicate image edges, i.e., the DWT has an *edge detection* property.

Denosing by wavelet thresholding

Wavelet thresholding is a popular approach for denoising due to its simplicity. In its most basic form, this technique operates in the orthogonal wavelet domain, where each coefficient is *thresholded* by comparing against a threshold; if the coefficient is smaller than the threshold it is set to zero, otherwise, it is kept or modified. A systematic theory was developed mainly by Donoho and Johnstone [Donoho92a]-[Donoho95b]. They have shown that various wavelet thresholding schemes for denoising have near optimal properties in the minimax sense and perform well in simulation studies of one dimensional curve estimation.

Hard and soft thresholding

Two standard thresholding policies are: *hard-thresholding*, (“keep or kill”), and *soft-thresholding* (“shrink or kill”). In both cases, the coefficients that are below a certain threshold are set to zero. In hardthresholding, the remaining coefficients are left unchanged

$$T^{hard}(w) = \begin{cases} 0, & \text{if } |w| \leq T \\ w, & \text{if } |w| > T \end{cases}$$

In soft thresholding, the magnitudes of the coefficients above threshold are reduced by an amount equal to the value of the threshold

$$T^{soft}(w) = \begin{cases} 0, & \text{if } |w| \leq T \\ \text{sgn}(w)(|w| - T), & \text{if } |w| > T \end{cases}$$

Most methods for estimating the threshold assume AWGN noise and an orthogonal wavelet transform. Among those, well known is the *universal threshold* of Donoho and Johnstone [Donoho92a]

$$T_{univ} = \sigma_n \sqrt{2 \log(n)}$$

where σ_n is the estimate of the standard deviation of additive white noise and n is the total number of the wavelet coefficients in a given detail image.. At different resolution scales, the threshold

differs only in the constant factor that is related to the number of the coefficients in a given subband.

ALGORITHM

- 1) Decompose multispectral Landsat TM Image into seven Bands.
- 2) Consider band 1. Apply discrete wavelet transform to band 1 of the multispectral image.
- 3) Perform soft thresholding by applying threshold $t_n = \sigma \sqrt{2\log(n)}$ to the decomposed band 1 in the multispectral image.
- 4) Find Inverse DWT of the thresholded image.
- 5) Repeat steps (2) to (6) for other TM Bands.
- 6) Concatenate band TM2, TM3, TM4 to see the output image.

3.2 Wavelet Regression

Let ϕ and ψ be, respectively, a father and mother wavelet [13] that generate the following complete orthonormal set in $L^2[0, 1]$:

$$\phi_{j_0,k}(x) = 2^{\frac{j_0}{2}} \phi(2^{j_0}x - k),$$

$$\psi_{j,k}(x) = 2^{\frac{j}{2}} \psi(2^jx - k),$$

for integers $j \geq j_0$ and k , where j_0 is fixed. Any function $f \in L^2[0, 1]$ may be expanded as

$$f(x) = \sum_{k=0}^{2^{j_0}-1} \alpha_k \phi_{j_0,k}(x) + \sum_{i=j_0}^{\infty} \sum_{k=0}^{2^i-1} \beta_{i,k} \psi_{i,k}(x)$$

Where $\alpha_k = \int f \phi_{j_0,k}$ and $\beta_{i,k} = \int f \psi_{i,k}$ and $f(x)$ denote the projection of f onto the span of the first basis elements.

For fixed j , we call $\beta_j = \{\beta_{j,k} : k=0, \dots, 2^j - 1\}$ the *resolution-j coefficients*. [13]

Consider non parametric regression problem where we have observations at 2^n regularly spaced points x_i of some unknown function f subject to noise

$$Y_i = f(x_i) + \sigma \epsilon_i, \quad i = 1, \dots, n$$

where $f \in L^2[0, 1]$, $x_i = i/n$ and ϵ_i are iid standard Normals. The goal is to estimate f under squared error loss. The standard wavelet based approaches to the estimation of f proceed by taking Discrete Wavelet Transform of the data Y_i , processing the resulting coefficients to remove noise and then transforming back to obtain the estimate.

The underlying notion behind wavelet methods is that the unknown function has an economical wavelet expression in that f is or is well approximated by a function with a relatively small proportion of nonzero wavelet coefficients.

We assume that $n = 2^{j_1}$ for some integer j_1 .

Empirical wavelet coefficients are given by

$$\tilde{\alpha}_k = \sum_{i=1}^n Y_i \int_{\frac{i-1}{n}}^{\frac{i}{n}} \phi_{j_0,k}(x) dx \approx \frac{1}{n} \sum_{i=1}^n \phi_{j_0,k}(x_i) Y_i \approx \alpha_k + \frac{\sigma}{\sqrt{n}} Z_k$$

$$\tilde{\beta}_{j,k} = \sum_{i=1}^n Y_i \int_{\frac{i-1}{n}}^{\frac{i}{n}} \psi_{j,k}(x) dx \approx \frac{1}{n} \sum_{i=1}^n \psi_{j,k}(x_i) Y_i \approx \beta_{j,k} + \frac{\sigma}{\sqrt{n}} Z_{j,k}$$

where the Z_k and $Z_{j,k}$ are iid standard Normals. We consider soft thresholding estimation.

Algorithm

- 1) Haze correct Band 5 of the Landsat TM image using Dark object subtraction techniques.
- 2) Decompose Haze corrected band 5 using wavelet Transform.

- 3) Perform soft thresholding by applying threshold $t_n = \sigma \sqrt{2\log(n)}$ to the decomposed band 5 in the multispectral image.
- 4) Apply discrete wavelet transform to band 1 of the multispectral image.
- 5) Threshold decomposed band 1 using soft thresholding.
- 6) Regress decomposed (DWT) and thresholded band1 to the multiresolution band 5 data obtained in step (3) using Linear least squares Regression Equation.
- 7) Find Inverse DWT of the Regressed image.
- 8) Repeat steps (4) to (7) for other TM Bands.
- 9) Concatenate band TM2, TM3, TM4 to see the output image.

3.3 Homomorphic Filtering

Illumination results from the lighting conditions present when the image is captured, and can change when lighting conditions change. Reflectance results from the way the objects in the image reflect light, and is determined by the intrinsic properties of the object itself, which (we can safely assume in this theoretical analysis) does not change. We can further argue that illumination varies slowly in space (slow spatial changes \leftrightarrow low spatial frequency) while reflectance can change abruptly (high spatial frequencies). For our given problem of eliminating atmospheric effects due to the change in lighting conditions, we would like to enhance the reflectance while reducing the contribution of illumination, hence, we need to somehow separate the illumination and reflectance components and then high pass the resulting image in frequency domain. *Homomorphic filtering* [7], [8] is a frequency domain filtering process that does just that.

The high pass filter normally used in this procedure is the Butterworth filter [7] defined as:

$$H(u, v) = \frac{1}{1 + \left[\frac{D_0}{D(u, v)} \right]^{2n}}$$

where n defines the order of the filter. D_0 is the cutoff distance from the center and $D(u, v)$ is given by:

$$D(u, v) = \left[\left(u - \frac{M}{2} \right)^2 + \left(v - \frac{N}{2} \right)^2 \right]$$

where M and N are the number of rows and columns of the original image.

ALGORITHM

- 1) Decompose multispectral Landsat TM Image into seven Bands.
- 2) Consider band 1. Take Log of the image.
- 3) Find FFT of the respective logarithmically transformed image.
- 4) Multiply it with Gaussian High Pass filter Transfer function.
- 5) Find Inverse FFT of the Filtered image.
- 6) Find Anti logarithm of the filtered image to get the final output image.
- 7) Repeat steps (2) to (6) for other TM Bands.
- 8) Concatenate band TM2, TM3, TM4 to see the output image.

3.4 Fourier Regression

The purpose of Fourier transform is to break down the image into its scale components, which are defined to be sinusoidal waves with varying amplitudes, frequencies and directions. The Fourier Transform thus provides details of the frequency of each of the scale components of the image and the proportion of information associated with each frequency component

ALGORITHM

- 1) Decompose multispectral Landsat TM Image into seven Bands.
- 2) Haze correct Band 5 using Improved Dark Object subtraction Method.

- 3) Find FFT of the Haze corrected Image.
- 4) Multiply it with Butterworth High Pass filter Transfer function.
- 5) Consider band 1. Find FFT of the image.
- 6) Multiply it with High Pass filter Transfer function.
- 7) Regress the image obtained in step (6) against the image data obtained in step(4) using Linear regression equations.
- 8) Find Inverse FFT of the Regressed band 1 image.
- 9) Repeat steps (5) to (8) for other TM Bands.
- 10) Concatenate band TM2, TM3, TM4 (false color composite) to see the output image.

3.5 Atmospheric Correction of Multispectral Data using DCT

A discrete Cosine Transform (**DCT**) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies.

The DCT does a better job of concentrating energy into lower order coefficients for image data. For most images, after transformation the majority of signal energy is carried by just a few of the low order DCT coefficients. These coefficients can be more finely quantized than the higher order coefficients. Many higher order coefficients may be quantized to 0. Formulae for DCT and inverse DCT are as given below:

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$$f(i, j) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v) F(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

Which can be written in matrix form, where the rows of [T] are the DCT basis vectors, as:

$$[f]_{N \times N} = [T^T]_{N \times N} [F]_{N \times N} [T]_{N \times N}$$

$$[F]_{N \times N} = [T]_{N \times N} [f]_{N \times N} [T^T]_{N \times N}$$

In the formulas, F(u,v) is the two-dimensional NxN DCT. u,v,x,y = 0,1,2,...N-1. x,y are spatial coordinates in the sample domain. u, v are frequency coordinates in the transform domain.

C(u), C(v) = 1/(square root (2)) for u, v = 0.

C(u), C(v) = 1 otherwise.

ALGORITHM

- 1) Decompose multispectral Landsat TM Image into seven Bands.
- 2) Consider band 1. Find DCT of the image.
- 3) DCT coefficient at zero frequency is made zero.
- 4) Find Inverse DCT of the image.
- 5) Repeat steps (2) to (6) for other TM Bands.
- 6) Concatenate band TM2, TM3, TM4 to see the output image.

4 : NDVI METHOD FOR VEGETATION ANALYSES

One of the applications of the atmospheric correction methods discussed in this paper is Vegetation analysis. Satellite Image processing can be increasingly used in examination of land use and land cover change. NDVI can be very useful in generation of land use/land cover classification. Ratio indices such as the normalized difference vegetation index (NDVI), of *Rouse et al. (1973)*, use various ratios of red and near-infrared bands to determine presence of vegetation. One of the functions of this method is to detect vegetation and plant refreshments and is capable of monitoring those levels in different ages.

NDVI represents the amount of green vegetation and is calculated from reflected red and near infra red light.

$$NDVI = (NIR - red) / (NIR + red)$$

NDVI ranges are usually between -1 to 1. Water, snow and clouds or any other nonvegetated scene is represented by negative number. Low positive number near zero indicates rock and bare soil, which reflect near infra red and red at the same level. Increasingly positive number indicates greener vegetation [6]. However, the NDVI is also influenced by sun angle changes and are affected by soil background to the point that they are as sensitive to soil darkening as to vegetation development [4].

5. RESULTS

The image used is a multispectral Landsat 7 ETM+ image of San Francisco acquired on March 03, 2000. The seven bands of the image is shown in fig (2)

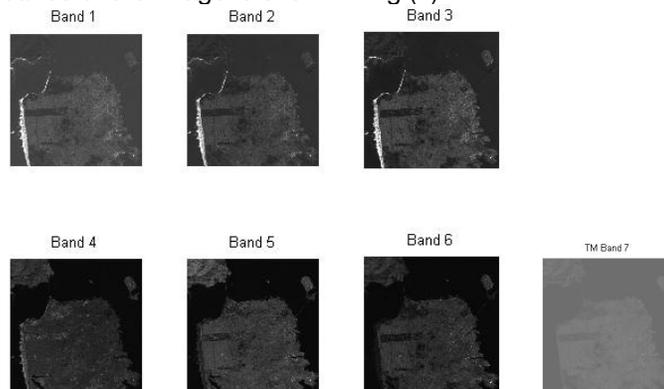


FIGURE 2: Landsat TM Bands 1-7

Eight entirely image based radiometric correction models are implemented. These methods are based solely on the digital image and do not require in situ field measurements during the satellite overflight. Atmospheric scattering is important only in the visible and near infrared regions. Hence TM band 1, 2, 3 and 4 of the multispectral image is considered for the analyses.

The multispectral image was radiometrically corrected by eight different methods, i.e. Simple Dark object subtraction, Improved Dark object subtraction, Regression Line method, Wavelet Thresholding, Wavelet Regression, Homomorphic filtering, Fourier regression and using DCT.

The DN frequency Histogram of the spectral Bands 1, 2, 3, 4 of the Landsat TM image is as shown in Figure (3).

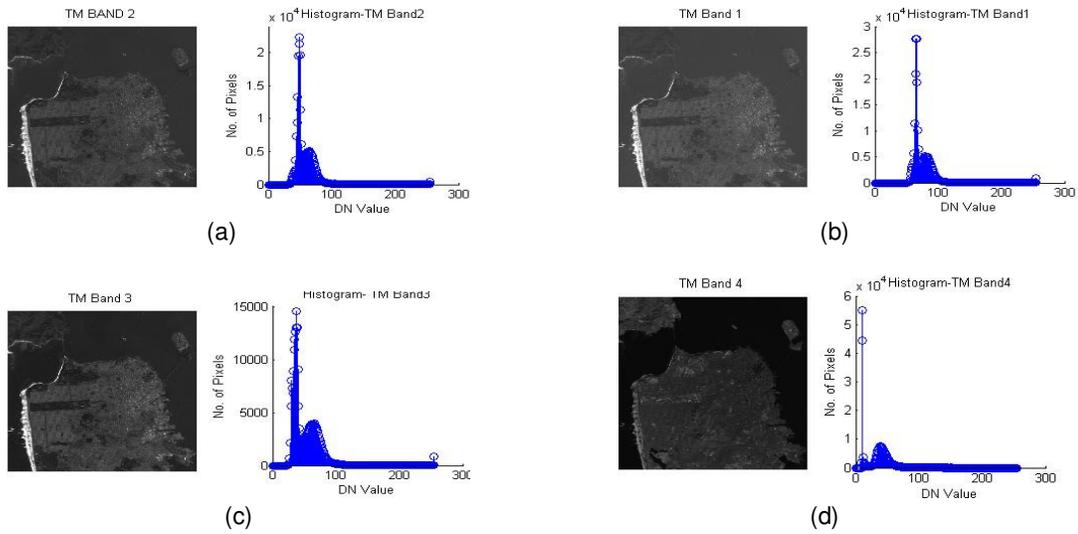


FIGURE 3: Landsat TM Bands and its corresponding Histogram (a) TM1, (b) TM 2, (c) TM 3, (d) TM 4

In Simple Dark Object Subtraction Method DN Haze value selected from the DN frequency histogram of an image is shown in Table (3)

TM 1	TM 2	TM 3	TM 4
54	35	25	8

TABLE 3 : Haze DN values

This Haze DN value is subtracted from the respective spectral band. The original and the corrected images (composite band 2, 3 and 4) obtained after this is shown in fig (5a) and fig (5b) respectively.

This relative normalization method assumes that the effects of haze are distributed evenly across the entire image, which may or may not be the case. This is a good initial adjustment, but there may be problems analyzing the data unless one of five atmospheric scattering models (scaled from very clear to very hazy) is chosen in addition to a dark-object haze value.

In Improved Dark object Subtraction Method, the starting haze value selected from TM Band 1 using histogram method is 54. The starting DN haze value must not overpredict the values for other bands.

The predicted haze value for TM bands 1, 2, 3, and 4 are shown below.

Haze DN Value			
TM 1	TM 2	TM 3	TM 4
54	40.5	29.16	18.468

TABLE 4: Haze DN values

These values were generated using the clear relative scattering model factors shown in Table 2. The image obtained after haze correction is shown in Figure (5c).

As seen from DN haze values (Table 4 and Table 5) TM Band1 is most affected by atmospheric scattering and the value gradually decreases with the bands. Hence we consider that TM band 5

can be used as a reference image to validate the results based on visual analyses as no reference image is available in Absolute Radiometric Correction.

The original and the corrected version of the multispectral image using RLM method is shown in Fig (5d).

Wavelet decomposed and reconstructed images are shown in Figure (4)

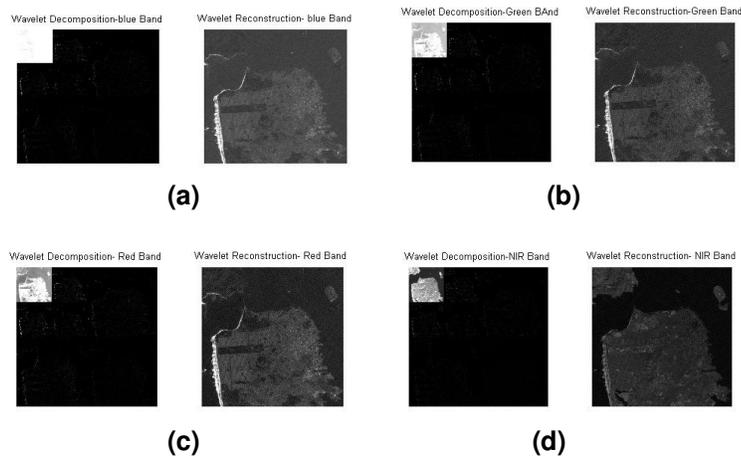


FIGURE 4: Wavelet decomposition and Reconstruction after thresholding (a)TM1, (b)TM2 (c)TM3 (4) TM4

Accuracy Analyses of the Image Based Atmospheric Correction Results

Visual Analyses

Comparing the visual appearance of multi spectral imagery is the most common method for testing the fidelity of atmospheric correction techniques. Although visual distinction between images is useful for large differences between images, it is highly prone to subjectivity when the differences are more subtle.

Figure (5a) shows the color composite made with Landsat TM bands 4, 3 and 2. The composite in Figure (5b) to (5i) shows the results of applying atmospheric correction methods discussed in this paper. The corrected image obtained after atmospheric correction using spatial domain methods are shown in (5b)-(5d) and using Wavelet Thresholding, Homomorphic filtering, DCT, Wavelet Regression and Fourier Regression method is shown in figure (5e), (5f), (5g), (5h) and (5i) respectively. As seen from figure (5) and (6) Regression method in spatial as well as transform domain gives the best results. Homomorphic filtering method gives better results in case of urban area.

Statistical Analyses

Another method is to compare the Root Mean Square Error (RMSE) and Peak Signal to Noise Ratio (PSNR) between images (Table 5 and Table 6 shown below). This method can determine discrete differences if care is taken to ensure that the data used in the RMSE calculations have not experienced change, otherwise that change is incorporated as error.

Method	TM1	TM2	TM3	TM4
DOS	55.9981	34.9994	24.9999	8
Improved DOS	53.9999	39.96	28.9835	15.373
Regression	78.0928	61.7631	58.076	35.5385
FFT	74.3782	82.8634	92.1803	50.4442
DCT	3.0925	2.2263	1.7110	0.9554
Wavelet Thresholding	12.6758	12.9239	16.223	11.0508
Wavelet Regression	44.5731	27.09	21.0541	10.7571
Fourier Regression	47.6508	31.2915	26.505	3.8348

TABLE 5: ROOT MEAN SQUARE ERROR (RMSE)

Method	TM1	TM2	TM3	TM4
DOS	13.1673	17.2496	20.172	30.069
Improved DOS	13.4829	16.0983	18.8878	24.3956
Regression	10.2786	12.3162	12.8509	17.1168
FFT	10.701	9.7636	7.210	14.074
DCT	39.324	41.032	43.46	41.037
Wavelet Thresholding	26.071	25.902	23.92	27.262
Wavelet Regression	15.1493	19.4746	21.6641	27.4969
Fourier Regression	14.5694	18.2223	19.638	36.456

TABLE 6: Peak Signal to Noise Ratio (PSNR)

As seen from Table 5 RMSE is maximum in TM Band1 indicating maximum change and consistently decreasing from TM1 to TM4 in spatial domain methods (DOS, Improved DOS and Regression). All three atmospheric correction techniques vastly improved radiometric consistency from the original image. RMSE is more in regression method indicating maximum change as compared to other two methods. Regression method yielded the best results. Improved DOS performed slightly better than DOS method. PSNR is minimum in TM1 and maximum in TM4.

In case of atmospheric correction using homomorphic filtering and wavelet thresholding TM2 and TM3 undergoes more change as compared to TM1 and TM4 undergoing minimum change.

Other Image quality measures [14] like Normalized Cross correlation (NK), Normalized Absolute error (change in this case) (NAE) and Normalized Mean Square Error is also found for the implemented methods (Table 7).

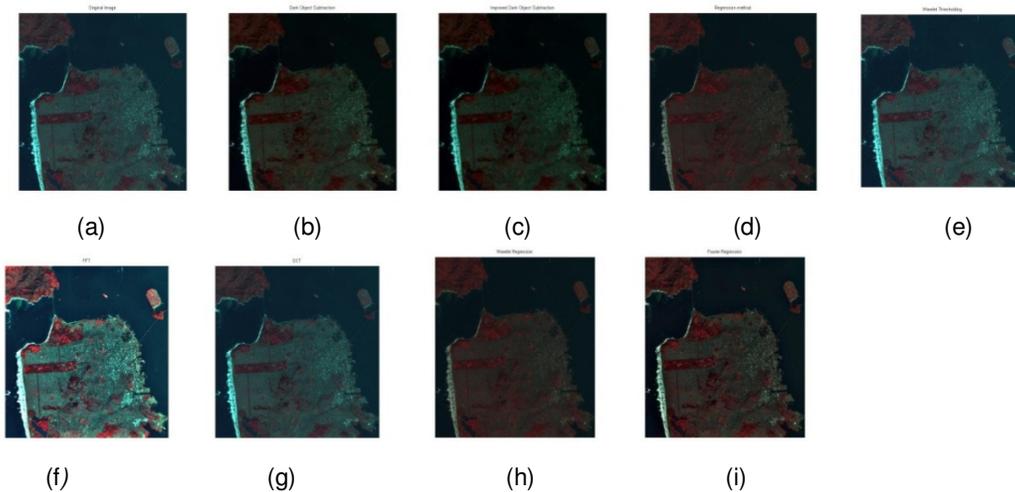


FIGURE 6: Atmospheric Correction in Spatial Domain (Composite band 2, 3&4). (a) Original Image, (b) DOS method (c) Improved DOS method (d) Regression Method, (e) Wavelet Thresholding (f) FFT, (g) DCT, (h) Wavelet Regression, (i) Fourier Regression

Method	NK	NAE	NMSE
DOS	0.435	0.6275	0.4014
Improved DOS	0.4094	0.6651	0.3139
Regression	0.4062	0.5995	0.3971
FFT	0.584	0.4236	0.1372
DCT	0.99	0.022	0.076
Wavelet Thresholding	0.657	0.45	0.2674
Wavelet Regression	0.6949	0.3851	0.1523
Fourier Regression	0.7086	0.383	0.1998

TABLE 7: Image Quality Measures

Another method used for testing the fidelity of atmospheric correction techniques is vegetation analyses using NDVI method.

Vegetation analyses using NDVI method is done before and after applying different atmospheric correction techniques. Figure (7) shows the images with vegetation analyses done before and after applying atmospheric correction techniques in spatial domain. After applying atmospheric correction methods we see that more vegetation area can be noticed. Comparing figure (7a) with figure (7b), (7c), (7d), (7e), (7f), (7g), (7h), (7i) we see that Regression method gives the best results in both the domains.

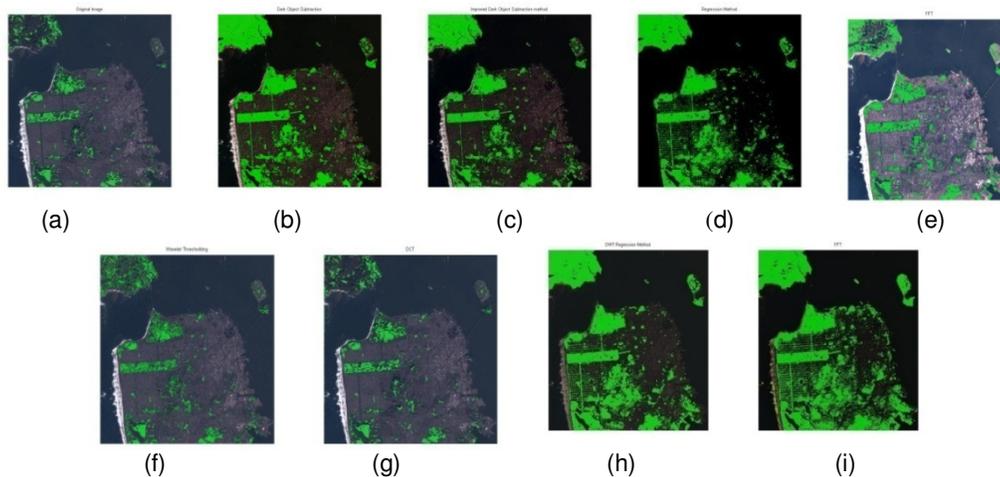


FIGURE 7: Vegetation Analysis (a) before and (b), (c), (d) (e), (f), (g), (h), (i) after atmospheric correction in spatial domain. (a) Original image before atmospheric correction, (b)DOS method, (c)Improved DOS, (d)Regression method. (e) FFT, (f) Wavelet Thresholding, (g) DCT

6 CONCLUSIONS

Dark Object Subtraction Correction assumes a constant DN Haze value throughout the image, which is often not the case. However, it does accomplish first order correction. DN values selected using this model does not conform to a realistic relative atmospheric scattering model. This lack of conformity may cause the data to be overcorrected in some or all of the spectral bands and the spectral relationship between the bands will not be correct.

In improved dark object subtraction technique the relative scattering models used were power law models where the power used was based on the amplitude of stating haze value. DN values used confirm to some realistic relative scattering model so that the haze values will be wavelength dependent and correlated with each other.

The Regression Line methods determine regression lines for multispectral plot of pixels. Regression method has an advantage over DOS method. Dark object subtraction method is based on the assumption that somewhere in the image is a pixel with zero illumination or zero reflectivity such that its radiometric value equals only the contribution of additive components. One problem with this method is that such pixels do not exist in many images, or it may not be confidently known whether they exist. Regression analysis methods do not require such dark pixels. If the pixels are off-scale (saturated at zero), Regression methods can provide reliable results as compared to DOS.

In terms of radiometric consistency DCT produced poor results. Wavelet Regression and Fourier Regression gives better results as compared other methods. The performance of classical FFT method and Wavelet Thresholding method were of medium quality. It may be suitable for urban area classification but not for studies analyzing vegetation. Regression methods in Spatial as well as transform domain gives effective results.

Vegetation analysis done after applying Atmospheric correction techniques gives improved results. More vegetation area can be observed after correcting the image using Fourier and Wavelet Regression.

As seen from the images as well as statistical measures, Wavelet Regression and Fourier regression gives better results in terms of atmospheric correction and vegetation analyses. However, these methods correct only atmospheric scattering effect. It is simple and easy to implement. These methods can be further assessed using Image classification and change detection algorithms.

7 REFERENCES

- [1] Bhosle, U. V., Pudale S., “Multivariate regression method for Radiometric correction of High resolution of Satellite data”, *International Conference on Signal Processing, Instrumentation and Control*, VIT, Pune.
- [2] Chavez, P.S. (1988). “An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data”. *Remote Sensing of Environment*, Vol. 24, pp. 459-479.
- [3] Chavez, P.S. (1988). “Image Based atmospheric corrections- Revisited and Improved.”, *Photogrammetric Engineering & remote sensing*, Vol 62, No. 9, September 1996
- [4] Huete, A. R. and Jackson, R. D., 1987, “Suitability of spectral indices for evaluating vegetation characteristics on arid rangelands”. *Remote Sensing of Environment*, 23, pp. 213-232.
- [5] Janzen, D. T., Fredeen A. L. and Wheate, R. D., “Radiometric Correction techniques and accuracy assessment for Landsat TM data in remote forested regions”. *Can. J. Remote Sensing*, Vol 32, No. 5, pp. 330-340, 2006
- [6] Lillesand, T. M. and Kiefer, R. W. (1994), “Remote sensing and image linter predation”, *John Wiley and sons Press*
- [7] Rahman, H., and G. Dedieu. 1994. ”SMAC: a simplified method for the atmospheric correction of satellite measurements in the solar spectrum”. *International Journal of Remote Sensing*, 15(1):123-143.
- [8] Richter, R. 1990. “A fast atmospheric correction algorithm applied to Landsat TM images”. *International Journal of Remote Sensing*, 11(1):159-166.
- [9] Richter, R. 1996. “A spatially adaptive fast atmospheric correction algorithm”. *International Journal of Remote Sensing*, 17(6):1201-1214
- [10] Robert E. Crippen, “Regression intersection method of adjusting Image data for band ratioing”, *Int. J. Remote Sensing*, 1987, Vol 8, no. 2, 137-155
- [11] Shunlin Liang, Hongliang Fang, and Mingzhen Chen, “Atmospheric Correction of Landsat ETM+ Land Surface Imagery—Part I: Methods”, *IEEE Transactions On Geoscience And Remote Sensing*, vol. 39, no. 11, November 2001, 2490-2498
- [12] Slater, P. N., Doyle, F. J., Fritz, N. L. and Welch R. (1983), “Photographic systems for remote sensing”, *American Society of Photogrammetry Second Edition of manual of Remote Sensing*, Vol. 1, Chap6. pp. 231-291
- [13] Christopher R. Genovese and Larry Wasserman, “Confidence sets for nonparametric wavelet regression”. *The Annals of Statistics*, 2005, Vol. 33, No. 2, 698–729.
- [14] Ahmet M., Eskicioglu and Paul S. Fisher, “Image Quality Measures and their performance”, *IEEE Transactions on Communications*, Vol. 43, No. 12, Dec 1995

A Research on Guided Thinning Algorithm and Its Implementation by Using C#

Jia Liang

*School of Information Science & Engineering
Chang Zhou University
Chang Zhou, 213164, China*

Sanctifier.jia@yahoo.com.cn

Abstract

After the advent of C# in year 2000, it's gradually and widely applied in commercial software developments on various Microsoft platforms. To seamlessly merge image processing algorithms with the technical trend, one efficient means is implementing algorithms directly by using C#. Since there hasn't any way to perform pixel-level operations in the development environment named Visual Studio which integrates C#, a class encapsulating image processing algorithms is developed, and guided thinning algorithm is implemented as a method of the class. The amelioration of the algorithm is made through employing extension methods, manipulating pointers and modifying the condition of executing thinning. Under these conditions, the skeletonization is smoothly implemented and the resulting data is visualized as a skeletonized binary image.

Keywords: Skeletonization, Guided Thinning Algorithm, C#.

1. INTRODUCTION

The skeletonization of binary image is an efficient means assisting image processing algorithms in various applications such as the object detection. In the intermediate-level vision, Hough transformation is one main means to detect objects. For the real-time detections, the performance of a certain Hough transformation-based algorithm is mainly determined by the amount of input data[1][2], i.e., the number of edge pixels, and skeletonization shrinks the detected edges to unit-width edges. This minimizes the amount of input data[1].

Usually, the skeletonization employs the morphological methods which are drastically subject to the size of structure element(also called mask)[3], thus the results may not well retain the general shapes of objects. An alternative solution has been provided by Davies[1]. The distance function is introduced to guide the thinning algorithm which generates unit-width homotopic skeletons[2], but there are isolated points scattering throughout the resulting image when the input image is taken from the real world. Such points provide poor information about the general shape and even add some difficulty to the following analysis. Therefore, the isolated points should be removed and this is achieved by introducing an additional condition of thinning in the algorithm.

To merge with the technical trend, the algorithm is implemented on Windows platforms by using IDE(Integrated Development Environment) named VS2008(Visual Studio 2008). Among all programming languages supported by VS2008, C# is chosen as the actual programming language on account of its powerful features provided directly by Windows producer Microsoft.

During the implementation, several problems arises one after another. Some difficult problems are how to implement logical operations of numbers belonging to the value type Double and how to save computations in subroutines of the thinning algorithm. The former is solved by employing extension methods which are features of C# and the latter is handled by interchanging pointers of image matrices. These may be the best solutions in this particular situation. Finally, all routines are integrated into the class ImageProcessing.

With the help of class ImageProcessing, an application software is successfully developed and the data of skeletonized binary image is obtained and visualized.

2. METHODS

The guided thinning algorithm mainly consists of two parts. The first is finding the set of local maxima of distance and the second is thinning objects based on the set. Finding local maxima involves manipulation of the distance function, and thinning is actually composed by stripping points in four directions in the image space and finally removing spurs generated by the stripping.

The whole procedure is shown in the following activity diagram of Unified Modeling Language(UML)[4][5].

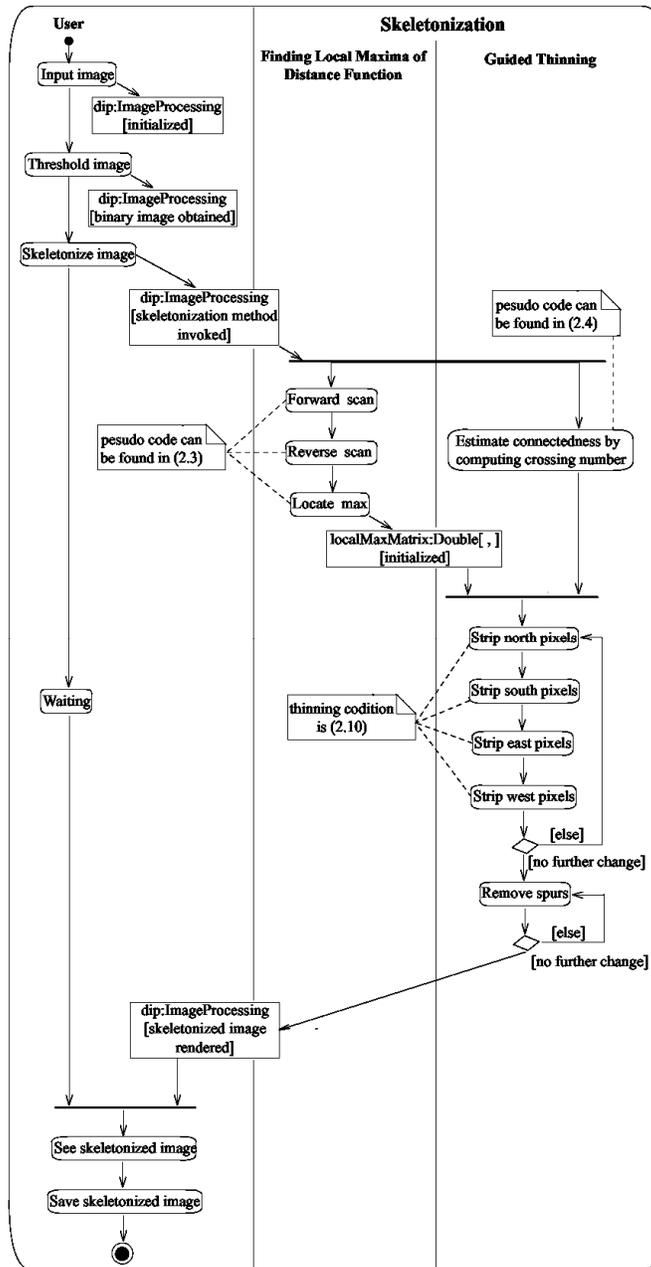


FIGURE 1: The activity diagram of skeletonization..

2.1 Finding Local Maxima

The concept of distance function is quite simple, it just requires every pixel is labeled by an integer representing the distance from the background. This can be summarized as following[2]:

$$\forall p \in X, \quad \text{dist}(p) = \min\{ n \in N, p \text{ not in } (X \bullet nB) \} \tag{2.1}$$

For each pixel p in the object X , its distance is given by $\text{dist}(p)$. Operator \bullet , notations B and N respectively denote the erosion operation, the object and an integer which represents the maximal number of the erosion operations shrinking B to single points. Therefore, $\text{dist}(p)$ assigns the integer i to p when p is removed by the i th erosion.

The distance is measured by using two raster scans, the forward and reverse scans. For making the explanation more comprehensible, every pixel in a 3-by-3 mask is denoted by one of notations $A0, A1, \dots, A8$ as following:

$$\begin{bmatrix} A4 & A3 & A2 \\ A5 & A0 & A1 \\ A6 & A7 & A8 \end{bmatrix} \tag{2.2}$$

For the forward scan, the mask regularly moves from left to right and top to bottom in an image and stops when the central notation $A0$ is of non-zero value to find the minimum of $A2, A3, A4$ and $A5$, then changes the value of central pixel to the sum of the minimum and 1 and move to next pixel. The scan is sequential which implies the current value-modified pixel is involved in the sequential value modification of the adjacent pixel next to the current one. Hence, there is only one image space for simultaneously reading data to process and storing the resultant data, unlike the usual parallel algorithm which requires two independent congruent image spaces to separate the two procedures for keeping the input data unchanged during processing. After the scan, because the procedure just checks $A2, A3, A4$ and $A5$, the integers as distance markers increase gradually from left to right and up to down in the inner area of an object.

If the left image of the following figure is the input binary image, then the right image is the visualized processed data.

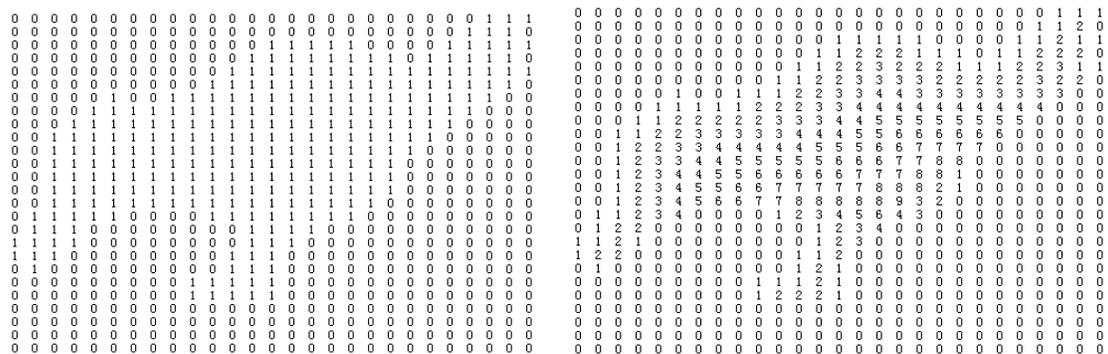


FIGURE 2: Thinning process 1.

Obviously, pixels in the lower right area of the object are labeled by wrong integers. This can be fixed by applying a reverse scan to it. Reverse raster scan works exactly as the forward scan except it moves the mask from right to left and bottom to top, i.e., in an inverse direction with respect to the forward scan. The result is visualized as the left part of the following figure:

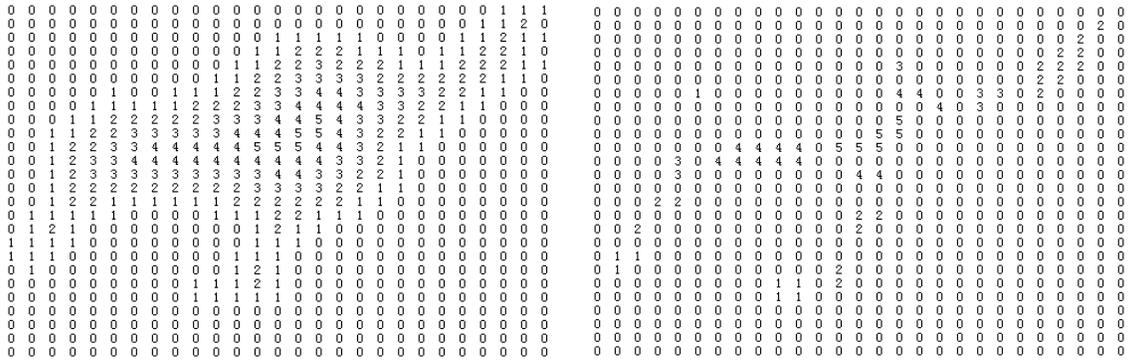


FIGURE 3: Thinning process 2.

After the distances of every pixel are measured by these two scans, finding local maxima can eventually be performed by using a parallel subroutine. It checks the neighborhood of the central pixel in a 3-by-3 mask in one image space, finds the maximum and compares it with the value of the center. If the value is larger than the maximum, then the value is recorded at the corresponding coordinate of a separate congruent space, otherwise 0 is recorded. This generates a result shown in the right part of fig.3.

The general procedure of two raster scans and finding maxima can be summarized as the following pseudo code:

$$\begin{aligned}
 &\text{forward scan : if}(A0 > 0) \quad A0 = \min(A2, A3, A4, A5) + 1; \\
 &\text{reverse scan : if}(A0 > 0) \quad A0 = \min(A6, A7, A8, A1) + 1; \\
 &\text{locate max : if}(A0 > 0 \ \&\& A0 > \max(A1, A2, A3, A4, A5, A6, A7, A8) \\
 &\qquad\qquad\qquad B0 = A0; \quad \text{else} \quad B0 = 0;
 \end{aligned}
 \tag{2.3}$$

Note the direction of moving mask in curly brackets of reverse scan is the inverse direction of forward scan and locate max, and the raster scans are sequential and locate max is parallel.

2.2 Guided Thinning

2.2.1. Crossing Number

Once the maxima are found and recorded, it's possible to design and implement guided thinning algorithm. The size of the mask is assumed to be 3-by-3 as (2.2) in finding maxima. In such a small mask, the crossing number χ (chi)[1] is employed to judge connectedness of A0.

The χ is obtained by summing the times of value changing during the travel which starts from an arbitrary pixel in the neighborhood of A0 clockwise or anticlockwise and ends when the same pixel is visited again. Under this definition, the integer χ can't be odd. To correctly compute χ , three logical operators are necessary, i.e., $\&\&$ (AND) operator, $!=$ (NOT EQUAL) operator and $!$ (NOT) operator, and the connectedness criterion must be 8-connectedness.

Firstly, the value changing is checked along four diagonal directions, namely, A1 to A3, A3 to A5, A5 to A7 and A7 to A1. For each diagonal, if two values of Ai's are different, χ is then added by 1 to denote one time of value changing.

Secondly, four corners are checked, e.g., A3, A4 and A5 form the upper left corner. For each corner, if the case is that the corner pixel is 1 and the other two are 0, then χ is added by 2. This case can't be detected by checking diagonals. These two checks complement each other and there's no intersection between them. In pseudo code, this can be expressed as:

$$\begin{aligned}
 \chi = & (A1 \neq A3) + (A3 \neq A5) + (A5 \neq A7) + (A7 \neq A1) + \\
 & 2 * \{ (!A1 \&\& A2 \&\& !A3) + (!A3 \&\& A4 \&\& A5) + \\
 & (!A5 \&\& A6 \&\& !A7) + (!A7 \&\& A8 \&\& !A1) \}.
 \end{aligned} \tag{2.4}$$

Clearly, the larger the χ is, the more intense the connectedness of the central pixel is. The marginal value of χ separates the cases the central pixel is removed or preserved is 2 inasmuch as such pixels are located on the edge of an object and appropriate to be removed. There are two cases causing ambiguity when χ equals 2, they are exemplified and shown in the following figure.

$$\begin{array}{ccc}
 0 & 0 & 0 \\
 0 & 1 & 0 \\
 0 & 0 & 1
 \end{array}
 \quad
 \begin{array}{ccc}
 0 & 0 & 0 \\
 0 & 1 & 1 \\
 0 & 1 & 1
 \end{array}
 \tag{2.5}$$

The left matrix of (2.5) represents an end point of a unit-width line which possibly denotes a part of the final skeleton, so the central point in this case can't be removed. Therefore, an additional condition of thinning has to be made, i.e., the sum of values of neighborhood doesn't equal 1. The conditions of a common thinning are summarized as following:

$$(\chi == 2) \&\& (\text{sumOfNeighborhood} \neq 1) \tag{2.6}$$

2.2.2. Extension Methods for Computing Crossing Number

Only the logical operator $\&\&$ (AND) shown in (2.4) is valid for Byte values and the \neq (NOT EQUAL) operator and $!$ (NOT) operator both return Boolean values. This complicates programming, especially when the computation of converting data type of matrix is expected to be saved. One of the possible solutions is using extension methods[6] which extend the existing class by adding new methods. In this paper, all matrices involved in calculation are assumed to be of Double[,] for programming convenience and saving computation of converting data types. The extension methods of Double are organized as a static class to meet the requirement of C#, and the actual code is shown in the following figure:

```

public static class doubleExtension
{
    public static double NOTEQUAL(this double num1, double num2)
    {
        if (num1 != num2) return 1.0;
        else return 0.0;
    }

    public static double NOT(this double num)
    {
        if (num != 0) return 0.0;
        else return 1.0;
    }

    public static double AND(this double num1, double num2, double num3)
    {
        if (num1 == 0 | num2 == 0 | num3 == 0) return 0.0;
        else return 1.0;
    }
}

```

FIGURE 4: Code of Extension methods.

The code of χ is then simplified as shown in the following:

```

double[,]thinningMatrix = (double[,])sourceMatrix.Clone();
int height = matrix.GetLength(0);
int width = matrix.GetLength(1);
int upperLeft = 0;
int upperMiddle = 1;
int upperRight = 2;
int left = width;
int center = width + 1;
int right = width + 2;
int lowerLeft = 2 * width;
int lowerMiddle = 2 * width + 1;
int lowerRight = 2 * width + 2;
fixed (double* pointer3 = thinningMatrix){
chi = pointer3[left].NOTEQUAL(pointer3[upperMiddle]) +
    pointer3[left].NOTEQUAL(pointer3[lowerMiddle]) +
    pointer3[right].NOTEQUAL(pointer3[upperMiddle]) +
    pointer3[right].NOTEQUAL(pointer3[lowerMiddle]) +
    2 * (pointer3[upperRight].AND(pointer3[right].NOT(), pointer3[upperMiddle].NOT()) +
        pointer3[upperLeft].AND(pointer3[upperMiddle].NOT(), pointer3[left].NOT()) +
        pointer3[lowerLeft].AND(pointer3[left].NOT(), pointer3[lowerMiddle].NOT()) +
        pointer3[lowerRight].AND(pointer3[lowerMiddle].NOT(), pointer3[right].NOT()));
}

```

FIGURE 5: Code of χ .

As shown in fig.5, extension methods AND(), NOT() and NOTEQUAL() are envisaged as the methods of Double which can be directly invoked by instances of Double, i.e., Double values.

2.2.3. Thinning Based on Local Maxima

The clusters of maxima shown in the left part of fig.3 consist of pixels in the central area of an object. These pixels are of very high possibility to be parts of the skeleton and they represent the general shape of the object in some degree. Hence, these pixels shouldn't be removed during the thinning procedure.

As the raster scans, thinning is sequential. The sequentiality introduces a new problem that is the final skeleton would be biased towards the bottom of the image inasmuch as the direction of regular moving the mask causes pixels in the upper left area of an object are continuously removed until the mask reaches the lower right edge of the object. Thus, the removal of pixels shouldn't be executed once, it should be executed in several times. For each time, the pixels in the circumambient ring of the area of the object are removed if the condition is satisfied. This can be summarized as the following formula[2]:

$$X \otimes \{B_{(i)}\} = (((X \otimes B_{(1)}) \otimes B_{(2)}) \dots \otimes B_{(n)}) \tag{2.7}$$

The notation $\{B_{(i)}\}$ denotes the Golay alphabet[7] and operator \otimes denotes the stripping operation. Here n is 4, and $B_{(1)}$, $B_{(2)}$, $B_{(3)}$, $B_{(4)}$ are shown in the following figure:

$$\begin{bmatrix} * & 0 & * \\ * & 1 & * \\ * & 1 & * \end{bmatrix}
 \begin{bmatrix} * & 1 & * \\ * & 1 & * \\ * & 0 & * \end{bmatrix}
 \begin{bmatrix} * & * & * \\ 0 & 1 & 1 \\ * & * & * \end{bmatrix}
 \begin{bmatrix} * & * & * \\ 1 & 1 & 0 \\ * & * & * \end{bmatrix}
 \tag{2.8}$$

Notation * denotes the pixel whose value can be 0 or 1. Actually, $B_{(1)}$ represents north pixels. The corresponding pseudo code is as following.

```

do { strip north pixels;
    strip south pixels;
    strip east pixels;
    strip west pixels }
until no further change

```

Combining this detection of directional pixels and (2.6), the condition for stripping directional pixels is obtained. That is:

$$(A0 > 0) \ \&\& \ (A0 \notin \text{local maxima}) \ \&\& \ (A0 \in \text{directional pixels}) \ \&\& \ (\text{chi} = 2) \ \&\& \ (\text{sum of neighborhood} \neq 1) \tag{2.10}$$

The result of applying (2.9) and (2.10) is shown in the left part of the fig.6.

Obviously, the object is not completely skeletonized, e.g., the upper right corner of the skeleton is 2-pixel wide. This is because these non-unit wide parts are maxima shown in the right part of the fig.3 which can't be removed according to the condition (2.10). In addition, the algorithm will generate unexpected spurs. Thus, a final sequential thinning with a modified condition (2.10) is necessary. The condition is:

$$(A0 > 0) \ \&\& \ (\text{chi} = 2) \ \&\& \ (\text{sum of neighborhood} \neq 1). \tag{2.11}$$

The final result is shown in the right part of the following figure.

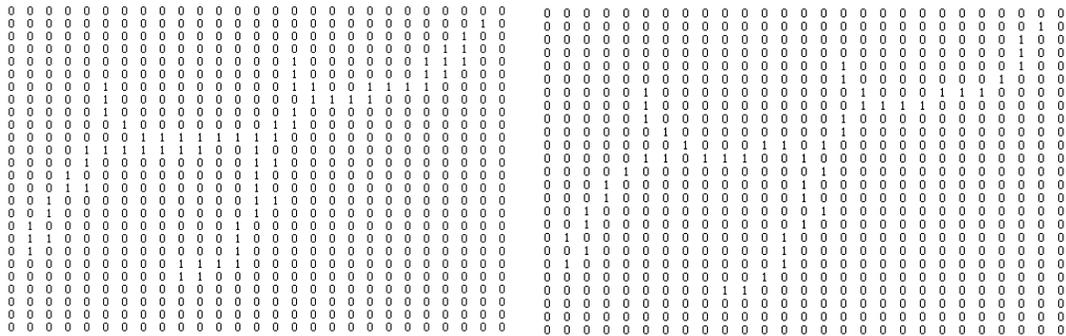


FIGURE 6: Thinning process 3.

This result is quite successful. The general shape of the object is preserved and there are no broken lines based on 8-connectedness criterion.

2.2.4. Amelioration of The Guided Thinning Algorithm

A detail of (2.9) should be noticed, that is how the four stripping subroutines are combined. For a stripping subroutine, there should be at least two image spaces to store the input data generated by the previous subroutine and the output data processed by the current subroutine. A possible solution may be that keeping one space constantly store the input data and the other only store the output data. This is problematic on account of that another image space has to be introduced to complete the interchange of the two spaces.

A much better solution is just interchanging the pointers of two spaces before a stripping subroutine is triggered. Assuming a pointer ptr1 points to a copy of the original binary image and another pointer ptr2 points to a congruent image space, the matrix with ptr1 is fed into the subroutine “strip north pixels” and the processed data is stored in the matrix with ptr2. For the next subroutine “strip south pixels”, the input data should be the matrix with ptr2 and the processed data can be stored in the matrix with ptr1. For now, the data of original binary image storing in the matrix with ptr1 is useless for the further processing in (2.9). There are four such interchanges in one loop.

Although the interchange can't change the nature of (2.9), i.e., it roughly is $O(n^2)$ where n denotes one dimension of the image space, it saves $8*n^2$ times of value assignments and a storage of an image space.

For the ideal binary image as fig.2, this result is satisfactory. But for real images, isolated points scatter throughout the whole image. These points provide little information about the skeleton and may make the following analysis difficult. Thus, they should be eliminated. This is achieved by modifying the condition (2.11) of the final spur-removing subroutine. The condition is:

$$(A0 > 0) \& \& (chi == 2 || chi == 0) \& \& (sum\ of\ neighborho\ od != 1). \quad (2.12)$$

3. RESULTS

The application software[8] was executed in Windows XP Professional SP3 on a laptop with Intel Core(TM)2 Duo 1.6 GHz CPU. The left of the following figure is a screenshot of the application when a color image whose size is 800×600 had been loaded and the right is the visualized result of applying the Sobel operator.

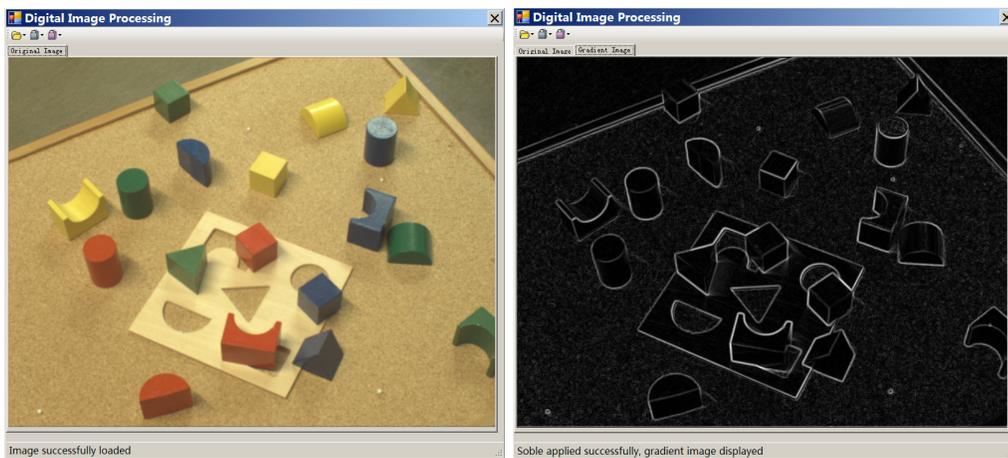


FIGURE 7: A loaded image and its gradient image.

The left of the following figure is the skeletonized binary image generated by the algorithm with (2.11) and the right is generated by the algorithm with (2.12).

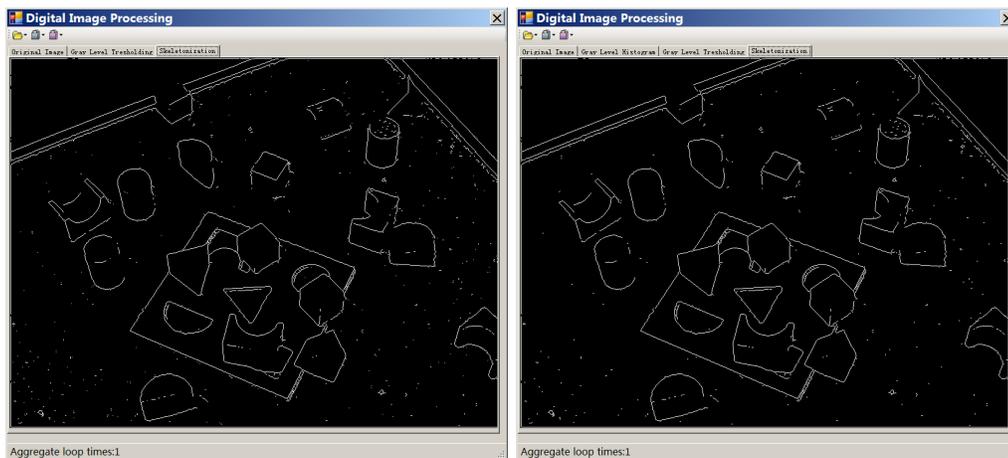


FIGURE 8: Binary image and its skeletonized version.

Obviously, white points scattering in the left lower corner of the left figure of fig.9 vanish in the right figure of fig.9. Another example is shown in the following figures.

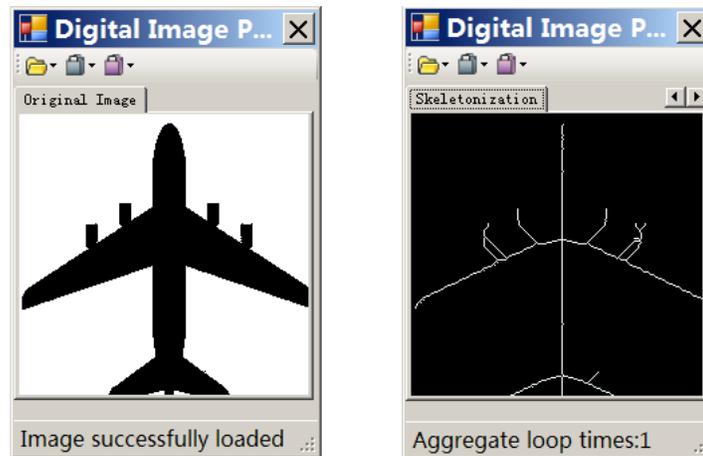


FIGURE 9: The silhouette of a plane and its skeletonized version.

4. CONCLUSION

For practical purpose, the ameliorated guided thinning algorithm is implemented on Windows platforms by using IDE named VS2008. Among all programming languages supported by VS2008, C# is chosen as the actual programming language on account of its powerful features provided directly by Windows producer Microsoft. Since the VS2008 lacks means of performing pixel-level operations, a class named ImageProcessing is developed, and the guided thinning algorithm is encapsulated as a method by the class. The problems arising in the development are overcome by employing extension methods, interchange of pointers and modifying the thinning condition. The programming procedure is thus simplified, and both the computation and storage are saved to the full extent, but the nature of the algorithm is not deeply improved. Hence the performance and spur removal can be further ameliorated. This may require a larger mask and subtle design of the thinning conditions.

5. REFERENCES

- [1] E. R. Davies, Machine Vision: Theory, Algorithms, Practicalities 3rd ed. San Fransisco, CA: Morgan Kaufmann, 2005.
- [2] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis, and Machine Vision 3rd ed. CT: 2008.
- [3] R. C. Gonzalez, R. E. Woods, Digital Image Processing 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2007.
- [4] I. Jacobson, G. Booch, and J. Rumbaugh, Unified Modeling Language User Guide 2nd ed. Boston, MA: Addison-Wesley, 2005.
- [5] J. Rumbaugh, I. Jacobson, and G. Booch, Unified Modeling Language Reference Manual 2nd ed. Boston, MA: Addison-Wesley, 2010.
- [6] J. Albahari, B. Albahari, C# 4.0 In A Nutshell 4th ed. Sebastopol, CA: O'Reilly, 2010.
- [7] J. Serra, Image Analysis and Mathematical Morphology, London: Academic Press, 1982.

- [8] L. Jia, Y. Sun, M. Wang, Y. Gu, "A Research on Implementation of Image Scattergram by Using C#", 2011 International Conference on System Design and Data Processing(ICSDDP 2011), IEEE press, February. 2011, pp. 353- 355.

Detection of Diseases on Cotton Leaves and Its Possible Diagnosis

Mr. Viraj A. Gulhane

*Department of Electronics and Telecommunication
S.G.B. Amravati University
Amravati, 444603, India*

virajgulhane@live.com

Dr. Ajay A. Gurjar

*Department of Electronics and Telecommunication
S.G.B. Amravati University
Amravati, 444603, India*

prof_gurjar1928@rediffmail.com

Abstract

In a research of identifying and diagnosing cotton disease, the pattern of disease is important part in that, various features of the images are extracted viz. the colour of actual infected image, there are so many diseases occurred on the cotton leaf so the leaf colour for different diseases is also different, also there are various other features related to shape of image, also there are different shape of holes are present on the leaf of the image, generally the leaf of infected image have elliptical shape of holes, so calculating the major and minor axis is the major task. The features could be extracted using self organizing feature map together with a back-propagation neural network is used to recognize colour of image. This information is used to segment cotton leaf pixels within the image, now image which is under consideration is well analyzed and depending upon this software perform further analysis based on the nature of this image.

Keywords: Image Processing Application in Agriculture Science, Coding, Analysis and Recognition, Biomedical Image Processing.

1. INTRODUCTION

The world textile industries are being ruled by "King Cotton". The antiquity of cotton has been traced to the fourth millennium BC. For over three thousand years (1500 BC to 1700 AD), India was recognized as cradle of cotton industry[1]. India thus enjoys the distinction of being the earliest country in the world to domesticate cotton and utilize its fiber to manufacture fabric. India is India accounts for approximately 25 per cent of world's cotton area and 16 per cent of total cotton production. Maharashtra is the important cotton growing state in India with 31.33 lack hector area and production of 62.00 lack bales (2008-09). The 2nd largest producer of cotton in the world. About 3 million farmers are engaged in cotton cultivation in the state mostly in backward region of Marathwada and Vidarbha[1].

In Vidarbha region, cotton is the most important cash crop grown on an area of 13.00 lacks hectares with production of 27 lack bales of cotton (2008-09). Disease on the cotton is the main problem that decreases the productivity of the cotton.

The main source for the disease is the leaf of the cotton plant. About 80 to 90 % of disease on the cotton plant is on its leaves. So for that our study of interest is the leaf of the cotton tree rather than whole cotton plant the cotton leaf is mainly suffered from diseases like fungus, Foliar leaf spot of cotton, Alternaria leaf spot of cotton. The machine vision system now a day is normally consists of computer, digital camera and application software. Various kinds of algorithms are integrated in the application software. Image analysis is one important method that helps segment image into objects and background. One of the key steps in image analysis is feature

detection [8]. Study of diseases on the cotton leaf can robustly studied by the image processing toolbox and also the diagnosis by using MATLAB helps us to suggest necessary remedy for that disease arises on the leaf of cotton plant.

We know that perception of the human eye is not so much stronger that he can differ minute variation in the infected part of image because that minute variation pattern of colour can be a different disease present on the leaf of cotton.

Our software can provide the exactly differentiate the variation of colour present on these leaves and depending upon that variation the further compare with database stored image features related to the colour.

2. REVIEW OF PRIOR FEATURE TECHNIQUE

Various papers are suggesting to diagnosis the cotton leaves using various approach suggesting the various implementation ways as illustrated and discussed below. In the research of identifying and diagnosing cotton disease using computer vision intellectually in the agriculture, feature selection is a key question in pattern recognition and affects the design and performance of the classifier. In previous paper [9], the fuzzy feature selection approach fuzzy curves (FC) and surfaces (FS) - is proposed to select features of cotton disease leaves image. In order to get best information for diagnosing and identifying, a subset of independent significant features is identified exploiting the fuzzy feature selection approach. Firstly, utilize FC to automatically and quickly isolate a small set of significant features from the set of original features according to their significance and eliminate spurious features; then, use FS to get rid of the features dependent on the significant features. This approach reduces the dimensionality of the feature space so that lead to a simplified classification scheme appropriate for practical classification applications. The results show that the effectiveness of features selected by the FC and FS method is much better than that selected by human randomly or other methods. Also another approach is used to diagnosis the grape leaf disease identification or diagnosis, i.e. paper explaining the grape leaf disease detection from colour imaginary using hybrid intelligent system, in that automatic plant disease diagnosis using multiple artificial intelligent techniques. The system can diagnose plant leaf disease without maintaining any expertise once the system is trained. Mainly, the cotton leaves disease is focused in this work. The proposed system consists of three main parts (a) cotton leaf colour segmentation (b) cotton leaves disease segmentation and (c) analysis and classification of diseases.

Segmentation algorithms fall into two general classes, based on whether they searching for discontinuities or similarities. Algorithms focusing on locating discontinuities in the data are primarily edge-based, while algorithms concerned with locating adjacent pixels based on similarities are primarily region-based.

Threshold techniques, a major category of algorithms, can fall into either class. In addition to these two major classes, there are also a number of general subcategories. For instance, algorithms either process colour or gray-scale data, operate on either an individual pixel basis (global) or a neighbourhood of pixels (local), and may use different window sizes or different colour representations [10]. For example survey of segmentation algorithms [11]. Cheng discussed the major segmentation approaches for segmenting monochrome images: histogram threshold, characteristic feature clustering, edge detection, region-based methods, fuzzy techniques, neural networks training method.

The cotton leaf disease segmentation is performed using modified self organizing feature map with genetic algorithms for optimization and support vector machines for classification. Finally, the resulting segmented image is filtered by Gabor wavelet which allows the system to analyze leaf disease colour features more efficient. The support vector machines are then again applied to classify types of grape leaf diseases. Similar idea can be extracted from to grape leaf disease diagnosis system [7] and applicable to cotton leaves diagnosis system.

3. DISEASES ON LEAVES OF COTTON

The diseases on the cotton leaves are classified as

- a) Bacterial disease: e.g. Bacterial Blight, Crown Gall, Lint Degradation.
- b) Fungal diseases: e.g. Anthracnose, Leaf Spot.
- c) Viral disease: e.g. Leaf Curl, Leaf Crumple, Leaf Roll.
- d) Diseases Due To insects: e.g. White flies, Leaf insects.

Out of the above types of disease these are dramatically affect the leaf of cotton plant and its leaves. We go through the selective type of diseases on the cotton leaves. And further we discuss the ANN image segmentation method to detect the diseases on cotton plant by scanning of cotton leaves through our portable dedicated scanner.

Various diseases are found on the cotton plant out of this we discuss the disease some of the major diseases which are often found on the leaves of cotton, that are viz.

3.1 Foliar leaf spot on cotton



FIGURE 1



FIGURE 2

As shown in above figures the, the disease is known as foliar disease arises due to potassium deficiency [2],[3],[4]. The early stage of this disease is as shown in fig 1, now if the more spots of this disease results into the final stage of this plant where the plant leaf is get fall so it is called as Foliar disease of the cotton plant as shown in fig 2. The leaf is having multiple no of spots which clearly denotes more potassium deficiency in the plant.

3.2 Curl Gemini virus



FIGURE 3



FIGURE 4



FIGURE 5

Cotton leaf curl Gemini virus (CLCuV) causes a major disease of cotton in Asia and Africa [2],[3],[4]. Leaves of infected cotton curl upward Fig 3. and bear leaf-like enations on the underside along with vein thickening Fig 4. Plants infected early in the season are stunted and yield is reduced drastically. Severe epidemics of CLCuV have occurred in Pakistan in the past few years, with yield losses as high as 100% in fields where infection occurred early in the growing season. Another cotton Gemini virus, cotton leaf crumple virus (CLCrV), occurs in Arizona, California, and Mexico. CLCrV symptoms are distinguishable from CLCuV symptoms in that infected leaves curl downward accompanied by interveinal hypertrophy and foliar mosaic Fig 5, both CLCrV and CLCuV infect dicotyledonous plants and are whitefly-transmitted (Brown et al., 1983; Mansor et al., 1993). Previous studies (Brown and Nelson, 1984; 1987; Hameed et al., 1994; Mansor et al., 1993) suggested that they belong to the subgroup III Gemini viruses. However, little information is available on the relationship of these two viruses with each other and with other subgroup III Gemini viruses.

3.3 Bacterial Blight



FIGURE 6

Xanthomonas campestris pv. *Malvacearum* Bacterial blight starts out as angular leaf spot with a red to brown border [2],[3],[4]. The angular appearance is due to restriction of the lesion by fine veins of the cotton leaf. Spots on infected leaves may spread along the major leaf veins as disease progresses, leaf petioles as shown in Fig 6. The angular leaf spot, results in premature defoliation and stems may become infected resulting in premature defoliation.

3.4 Cerco Spora-leaf Spot Cerco Spora



FIGURE 7

The disease affects older leaves of mature plants. The spots are round or irregular in shape yellowish brown, with purple, dark brown or blackish borders and white centers affected leaves become pale in colour and finally fall off [2],[3],[4] as shown in Fig 7.

3.5 Alternaria Leaf Spot-alternaria Macro Spora

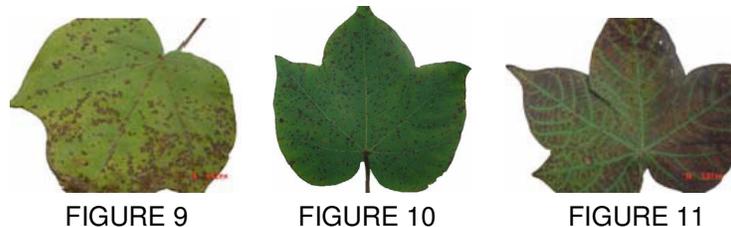


FIGURE 8

As shown in Fig 8, small, pale to brown, round or irregular spots measuring 0.5 - 3 mm in diameter and cracked centers appears on the affected leaves of the plant. Affected leaves become dry and fall off [3],[4],[5],[6]. The disease may cause cankers on the stem. The infection spreads to the bolls and finally falls off.

4. FEATURES OF LEAF IMAGE

When crops suffer from many diseases, batches (spots) often happen on leaves [7]. Leaf spots are considered the important units indicating the existence disease and regarded as indicator of crops disease. In order to classify disease leaf samples category, a set of spot features for classification and detection of the different disease leaves, as in Fig 9 ,10,11 are investigated[9].



Spot features are extracted from image using the appropriate image processing method. These features are very important for the colour and morphology of the leaf spots and they provide critical information about its visual representation. The features correspond to colour characteristics are the mean and variance of the gray level of the red, green and blue channel of the spots; and other features correspond to morphological and geometrical characteristics of the spots. By using segmentation technique it is easy for us to extract the features of disease leaf of the image.

There is main feature related to colour of leaf image i.e. infected part of the particular disease leaf image is having the variations in its RGB values, means that variations is certain, i.e. that variation of RGB values i.e. combined RGB value is not repeated with another diseased leaf image RGB values.

The extraction of the features and image disease classification during this steps is as shown in the following fig. 12. There are various features of the leaf of image, thresholding, sobel features, canny features etc.

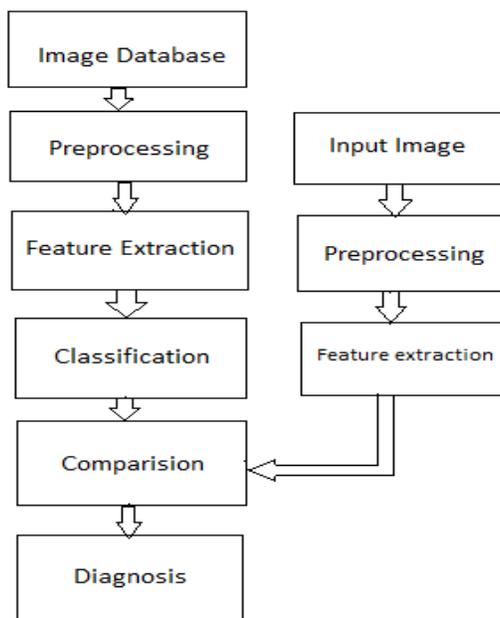


FIGURE 12: Overview of Diagnosis system using feature extraction

Here in the feature extraction, we have to classify our features in accordance with the various diseases presents on the leaf. As the diseases changes features are also changes and lastly actual image compare and final result should be available.

5. COLOR IMAGE SEGMENTATION

It is one of the colour image feature extraction technique, this technique is called as colour image segmentation [8],[9]. Using this technique it is easy for us to extract the various features of diseased leaf of cotton image. as shown in fig 13 below.

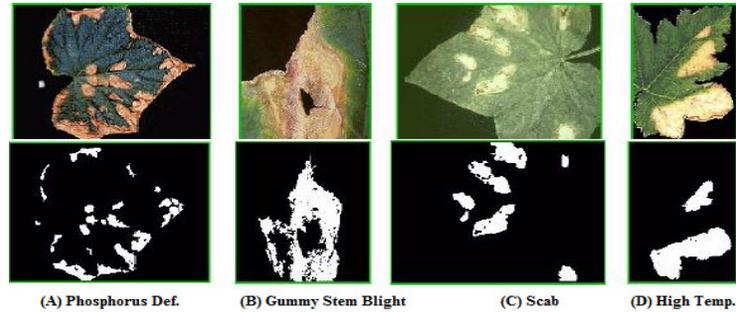


FIGURE 13

Now by formulating overall cotton leaf diagnosis system as shown in Fig 14

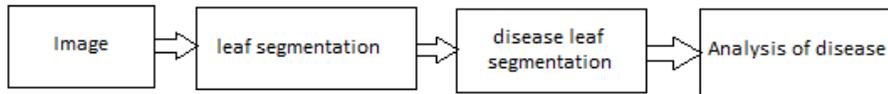


FIGURE 14: Cotton leaf diagnosis system

In this part the input image is enhanced by using anisotropic-diffusion technique to preserve the information of extracted pixels before extracting cotton leaf colour from background and B components from HIS and LAB colour space, respectively, are use to reduce effect of illumination. The resulting colour pixels are clustered by the unsupervised SOFM network to obtained group of colour in the image. The back propagation neural network is then applied to extract cotton leaf colour from diseased part of image. The implementation can be shown in fig 15.

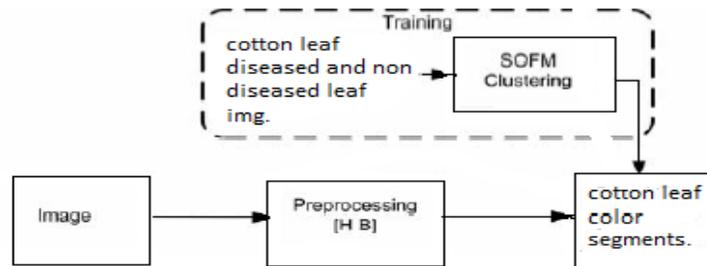


FIGURE 15: Cotton leaf colour extraction system diagram

5.1 Cotton Leaf Disease Colour Extraction

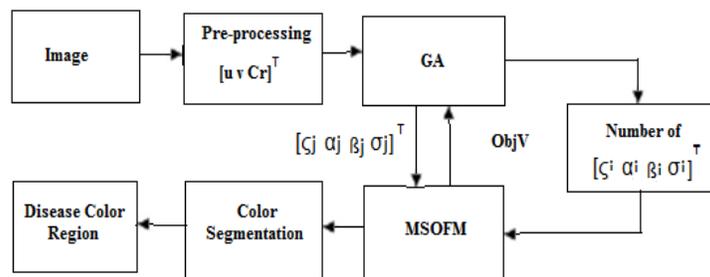


FIGURE 16: Cotton leaf disease colour extraction

The image background is additionally processed to remove the edge pixels in order to preserve the actual affected pixels as many as possible. In additions A,U, and Cr components from LAB, UVL and YCbCr colour space, respectively are applied for affected leaf colour extraction with the

purpose of less illumination effects. The remaining colour pixels are then extracted for cotton leaf disease colour by using modified self organization feature map, the clustering process does not require any training nor predefined no of colour groups[7]. This network is also adjustable allowing similarity of each colour group to be fine tuned. The cotton leaf disease colour extraction system is shown in the fig. 16.

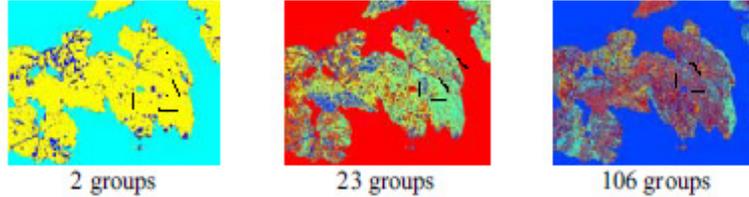


FIGURE 17: Example of different no of colour group from MSOFM

The cotton leaves disease colour are segmented corresponding to number of group of colour, the information from both disease and non-disease pixels are used for training with support vector machines(SVSM) for cotton leaf disease segmentation as shown in fig.17 using this SVSM technique perform better segmentation for cotton leaves[8]. We consider SVMs are trained using only 20 cotton leaves disease samples and 25 non disease samples. The resulting segmented pixels are then processed for classification of cotton leaves disease.

Now before the classification process, some irrelevant pixels is eliminated by method of convolution and thresholding applied. Here H and S components (from HIS colour space) and Cr component (from YCbCr colour space) are used to extract salient colour features of diseased cotton leaf. Now by using the Gabor filter we can separate different disease[12] appearance features. Now all information from this technique is used to examining affected pixels within the image.

In order to obtain optimum no of colour group Genetic Algorithm (GA) is applied developed to search for optimal parameters above fig. 18 shows the required result.

5.2 Equations

$$d_{wx} = \frac{1}{N_w} \sum_{g=1}^{N_g} \frac{1}{N_g} \sum_{i=1}^{N_g} \sqrt{(x_{g1} - w_{g1})^2 + (x_{g2} - w_{g2})^2} \tag{1}$$

$$d_{ww} = \frac{1}{N_d} \sum_{j=1}^d \sum_{i=j+1}^{N_w} \sqrt{\sum_{k=1}^n (w_{ik} - w_{jk})^2} \tag{2}$$

$$ObjV = d_{wx} + \frac{N_w}{d_{ww}} \tag{3}$$

- Where,
- X=colour component of input pixel
- W=weight vector
- N_w=Number of weight vector
- N_g=Number of colour pixels within the weight vector
- N_d=Number of measures between weight vectors
- n=Number of input x
- d_{wx}=distance between input and weight vector
- d_{ww}=Distance between weight vector
- ObjV=Objective value.

With the help of above equations[7] the model of cotton leaf disease detection can be formulated that equations suggests the model of the SVM(support Vector Machine) which is used for clustering of cotton leaves pixels using GA(Genetic Algorithm).

5.3 Detection of disease

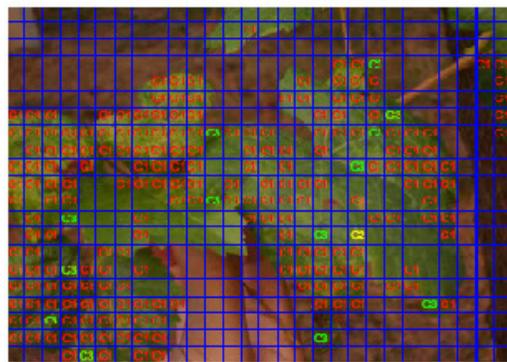
Various disease like Scab, Rust etc mainly, and their detection in accordance of map and feature extraction techniques [7]. We have following results as shown in table 1

Diagnosis	Disease Type: Number of sub images		
	Scab	Rust	No disease
Scab	16,690	2,245	619
Rust	2,059	17,130	900
False Detection	755	599	0
Correction	84.5	83.5	92.5
Average	86.83		
Detection	90.5		

TABLE 1: Results



(A) Example of rust disease detection



(B) Example of scab disease detection

FIGURE 18: Example of cotton leaf diagnosis result.

Results obtained (as shown in TABLE 1) can be provided by means of method of feature extraction as shown in fig.16 [7]. Where show two disease detection (A) Rust disease detection and at another hand (B) Scarab disease detection [7] mainly as shown in fig. 18.

6. CONCLUSION

Using the colour image segmentation method to exact intensity pattern to various diseases accordingly it is then possible to analyze the n no of cotton diseases and it works very efficiently. Here there is more scope to reduce the various errors which will be occurred during the simulation, that can be minimize as the more no of input is provided accordingly. That is because of training feature of ANN approach which will not available with fuzzy method. ANN method is providing 85 to 91% of exact disease detection depending upon the quality of image provided by the portable scanner and the training. More train network leads to a very efficient diagnosis of the diseases on the cotton leaf.

7. REFERENCES

- [1] <http://www.pdkv.ac.in/CottonUnit.php>(cotton area in viderbha)
- [2] http://www.ikisan.com/Crop%20Specific/Eng/links/mh_seeds.shtml
- [3] <http://www.ces.ncsu.edu/depts/pp/notes/Cotton/cdin1/cdin1.htm>
- [4] Management of seeding diseases of Cotton by Thomas Isakeit, Associate professor and extension plant pathologist Texas A&M University college station.
- [5] W.C. Schnathorst and P.M. Halisky "Potentially serious cotton disease, angular leaf spot established in California"
- [6] Detection of Citrus Greening Using Microscopic Imaging by Dae G. Kim
- [7] A.Meunkaewjinda, P. Kumsawat, K.Attakitmongcol And A.Sirikaew" Grape leaf disease detection from colour imaginary using Hybrid intelligent system. "proceedings of ECTI-CON 2008.
- [8] Jiazhi Pan,Young He. "Recognition of plants by leaves digital image and neural network" IEEE proceedings on 2008 International Conference on Computer Science and Software Engineering.
- [9] Yan Cheng Zhang, Han Ping Mao, Bo Hu, Ming Xili "features selection of Cotton disease leaves image based on fuzzy feature selection techniques" IEEE Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China, 2-4 Nov. 2007.
- [10] Funck J.W., Zhong Y., Butler D.A., Brunner C.C., ForrerJ.B.,2003." Image segmentation algorithms applied to wood defect detection." Computers and Electronics in Agri 41(13).
- [11] Fu, K.S., Mui, J.K., 1981." A survey on image segmentation. Pattern Recognition" 13(1)
- [12] Liu C and Wechsler H, "Gabor feature based classification using the enhanced fisher discriminant model for face recognition," IEEE Transactions on image processing, vol. 11, pp46,2002

Histogram Equalization with Range Offset for Brightness Preserved Image Enhancement

Haidi Ibrahim

*School of Electrical and Electronic Engineering,
Engineering Campus, Universiti Sains Malaysia,
14300 Nibong Tebal, Penang,
Malaysia*

haidi_ibrahim@ieee.org

Abstract

In this paper, a simple modification to Global Histogram Equalization (GHE), a well known digital image enhancement method, has been proposed. This proposed method known as Histogram Equalization with Range Offset (HERO) is divided into two stages. In its first stage, an intensity mapping function is constructed by using the cumulative density function of the input image, similar to GHE. Then, during the second stage, an offset for the intensity mapping function will be determined to maintain the mean brightness of the image, which is a crucial criterion for digital image enhancement in consumer electronic products. Comparison with some of the current histogram equalization based enhancement methods shows that HERO successfully preserves the mean brightness and give good enhancement to the image.

Keywords: Digital Image Processing, Image Contrast Enhancement, Histogram Equalization, Brightness Preserving Enhancement

1. INTRODUCTION

Recent trends show that the usage of visual information has becoming more and more prominent in our daily life. In addition to television, camera, camcorder, and personal computer, many high-tech electronic products, such as hand-phone, or even refrigerator, nowadays are being equipped with capabilities to display digital images. Unfortunately, the input images that are provided to (or captured by) these devices are sometimes not really in good contrast. Therefore, a process known as digital image enhancement is normally required to increase the quality of these low contrast images [1].

One of the commonly used digital image enhancement techniques is the Global Histogram Equalization (GHE). This method stretches the dynamic range of the histogram and produces an overall contrast enhancement in the image [2]. GHE is popular because it is effective, simple, and easy to be implemented. GHE uses the Cumulative Density Function (CDF) of the input image as its intensity mapping function. GHE's intensity mapping function can be considered as a scaled version of CDF [3].

Although there are several advantages of GHE, this method is not recommended to be used directly in consumer electronic products. This is because there are several unwanted effects associated with GHE, such as the saturation artifact and washed out appearance [4]. Therefore, in 1997, Kim suggested a simple rule to overcome this problem. This rule requires digital image enhancement methods, which are used in consumer electronic products, to preserve the mean brightness of the original image in the enhanced image [5].

This rule has grabbed attentions of many researchers, and as a consequence, several enhancement methods have been proposed to fulfill this requirement. Several of them are; Brightness Preserving Bi-Histogram Equalization (BBHE) [5], Multipeak Histogram Equalization (MHE) [6], Dualistic Sub-Image Histogram Equalization (DSIHE) [7], Minimum Mean Brightness Error Bi-Histogram Equalization (MMBEBHE) [8,9], Recursive Mean-Separate Histogram

Equalization (RMSHE) [9,10], Recursive Sub-image Histogram Equalization (RSHE) [11], Brightness Preserving Dynamic Histogram Equalization (BPDHE) [4,12], Brightness Preserving Weight Clustering Histogram Equalization [13], Bi-Histogram Equalization with a Plateau Limit (BHEPL) [14], and Simple Histogram Modification Scheme (SHMS) [3].

Methods such as BBHE, DSIHE, and MMBEBHE, divide the input histogram into two sections, using one intensity value as their separating point. This separating point is selected based on some criteria, depending on the enhancement method. Although these methods are simple, they can preserve the mean brightness only to a certain extent. For example, BBHE is able to maintain the brightness if and only if the input histogram has a quasi-symmetrical distribution around its separating point [15]. Other methods, such as MHE, RMSHE, RSHE, and BPDHE, divide the input histogram into more than two sections. Although some of these methods are good in preserving the mean brightness, not much enhancement could be obtained due to these separating points. Furthermore, these methods are relatively requiring more computational power in order to select the separating point properly.

In this paper, a new histogram equalization based image enhancement method is proposed. Because this method uses an offset value to maintain the mean brightness, this method is named as Histogram Equalization with Range Offset (HERO). Interestingly, unlike most of the histogram equalization based methods, HERO does not require any histogram portioning process to preserve the mean brightness.

The remainder of this paper is organized as follows. Section 2 describes the basic equations related to GHE, in order to familiarize the reader with the framework of this research. Next, Section 3 presents the algorithm of HERO. Experimental results obtained from this work are presented in Section 4. Finally, Section 5 concludes the findings.

2. GLOBAL HISTOGRAM EQUALIZATION (GHE)

By taking $\mathbf{X}=\{X(i, j)\}$ as the input image with L discrete gray levels denoted by $\{X_0, X_1, \dots, X_{L-1}\}$, and $X(i, j)$ presents the intensity of the image at spatial location (i, j) with condition $X(i, j) \in \{X_0, X_1, \dots, X_{L-1}\}$, the histogram h is defined as:

$$h(X_k) = n_k, \quad \text{for } k = 0, 1, \dots, L-1 \quad (1)$$

where X_k is the k -th gray level and n_k is the number of times the gray level X_k appears in the image. Histogram h presents the frequency of occurrence of the gray levels in the image.

The Probability Density Function (PDF) is defined as the normalized h with respect to the total number of pixels contained in \mathbf{X} . If the size of \mathbf{X} is $M \times N$ pixels, PDF for intensity X_k , $p(X_k)$, is defined as:

$$p(X_k) = h(X_k) / (M \times N) \quad \text{for } k = 0, 1, \dots, L-1 \quad (2)$$

From Eq. (2), CDF for intensity X_k , $c(X_k)$, is given as:

$$c(X_k) = \sum_{j=0}^k p(X_j) \quad \text{for } k = 0, 1, \dots, L-1 \quad (3)$$

GHE enhances \mathbf{X} by using CDF as its transformation function. This transformation function, $f(X_k)$, is defined as:

$$f(X_k) = (L-1) \times c(X_k) \quad \text{for } k = 0, 1, \dots, L-1 \quad (4)$$

Then, the output image produced by GHE, $\mathbf{Y}=\{Y(i, j)\}$, is given by equation (5).

$$\mathbf{Y} = f(\mathbf{X}) = \{f(X(i, j)) \mid \forall X(i, j) \in \mathbf{X}\} \quad (5)$$

Although GHE successfully increases the contrast in the image, this method does not put any constrain in preserving the mean brightness. Therefore, a simple modification to GHE is introduced in the next section.

3. HISTOGRAM EQUALIZATION WITH RANGE OFFSET (HERO)

The implementation of Histogram Equalization with Range Offset (HERO) is divided into two stages. The main aim of the first stage is to improve the image contrast. On the other hand, the main aim of the second stage is to restore the mean brightness into the output image. Yet, it is worth noting that at the end of the first stage, only the transformation function is created, and no output image is produced. The processes involved in these two stages are explained in the following two subsections.

3.1. STAGE 1: Contrast Enhancement

The procedure in this stage is exactly the same to GHE as described in Section 2. First, the histogram h is constructed from the original input image \mathbf{X} by using Eq. (1). Then, PDF and CDF of \mathbf{X} are found by using Eq. (2) and Eq. (3), respectively. Finally, the transformation function $f(X_k)$ is calculated by using Eq. (4) to become the base function for the contrast enhancement process for HERO.

3.2. STAGE 2: Brightness Restoration

In order to restore the mean brightness effectively, the total brightness value of the input image, B_X , is calculated by using the following equation.

$$B_X = \sum_{x=0}^{x=L-1} x \times h(x) \tag{6}$$

Another total brightness value, B_Y , also been calculated. This value estimates the total brightness value for the output image. This is calculated by using the transformation function obtained in Stage 1.

$$B_Y = \sum_{x=0}^{x=L-1} f(x) \times h(x) \tag{7}$$

Next, the total brightness difference between the input image and the expected image, ΔB , is calculated.

$$\Delta B = B_Y - B_X \tag{8}$$

A positive ΔB value indicates that the expected output image will be brighter than \mathbf{X} . Conversely, a negative ΔB value shows that the expected output image will be darker than \mathbf{X} .

The idea of brightness restoration technique employed in HERO is depicted in Fig. 1. When the output image is expected to be brighter than the original image (i.e. indicated by a positive ΔB value), the expected histogram h_Y will be translated to the left side in order to reduce the image brightness. On the other hand, when the output image is expected to be darker than the original image (i.e. indicated by a negative ΔB value), the histogram will move to the right side in order to increase the image brightness. In HERO, the shape of the histogram obtained by Eq. (4) is maintained in order to keep the overall contrast in the output image.

The movement of the histogram as shown by Fig. 1 can be obtained by introducing a range offset value, X_{off} . This offset value is used to modify Eq. (4) to create a new intensity transformation function, g , which will be used to create the output image.

$$g(X_k) = f(X_k) + X_{\text{off}} \quad \text{for} \quad k = 0, 1, \dots, L-1 \tag{9}$$

For the case of positive ΔB , X_{off} needs to be a negative integer in order to move the histogram to the left side. On the contrary, for the case of negative ΔB , X_{off} is presented by a positive integer, and thus the histogram can move to the right side. The intensity values of mapping function g are clipped at 0 and $L-1$.

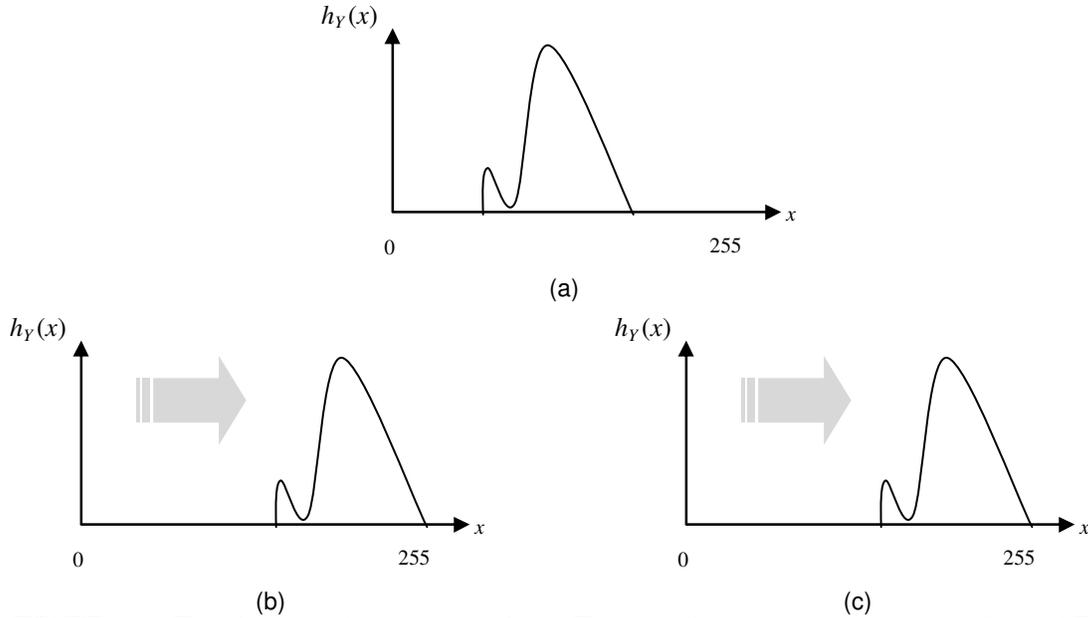


FIGURE 1: (a) The initial expected histogram, h_Y . (b) The correction applied when ΔB is positive. (c) The correction applied when ΔB is negative.

The value of X_{off} in HERO is determined iteratively, starting from $X_{\text{off}} = 0$. For every increment, or decrement in the value of X_{off} , the brightness correction value, ΔC , is calculated. Ideally, X_{off} in Eq. (9) will be tuned to obtain the following condition:

$$\Delta C = \left(\sum_{x=0}^{x=L-1} x \times h(x) \right) - \left(\sum_{x=0}^{x=L-1} g(x) \times h(x) \right) \approx 0 \tag{10}$$

However, the condition stated by Eq. (10) is frequently impossible to be achieved due to quantization applied in digital images. Therefore, one constrain is imposed to HERO to help the process of finding the suitable X_{off} value, effectively. For the positive ΔB , the iteration for the offset finding process will stop when the updated X_{off} value produces ΔC which is equal or less than zero. In opposite, if ΔB is negative, the iteration will stop when the updated X_{off} value produces ΔC which is equal or greater than zero.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of HERO, four standard grayscale images of size 512x512 pixels, as shown in Fig. 2, have been used. These images are eight-bit depth images, and therefore $L-1 = 255$. Fig. 2(a) has been chosen because this image has a lot of small details. Fig. 2(b) is selected because it contains multiple objects. Fig. 2(c) is used because the object of interest in this image (i.e. the sailboat) is relatively small as compared with the total image size. Fig. 2(d) is utilized to present an input image with low contrast.

To benchmark the performance of HERO as regards to some state-of-the-art enhancement methods, ten other histogram equalization enhancement techniques have been implemented in this work. They are GHE, BBHE, DSIHE, MMBEHE, MHE, RMSHE, RSHE, BPDHE, BHEPL and SHMS. These methods present a wide range of brightness preserved histogram equalization methods. For the implementation of RMSHE and RSHE, recursive level of two (i.e. $r=2$) is chosen. With this parameter setting, both RMSHE and RSHE will divide the histogram into four sub-histograms.

With the intention to evaluate the effectiveness of these contrast enhancement methods in keeping the input mean brightness inside their output images, following the same framework as other researchers in this field, a quantitative measure known as Absolute Mean Brightness Error (AMBE) has been used in this work. If b_x presents the mean brightness of the input image, and b_y presents the mean brightness of the output image, this measure is defined as:

$$AMBE = |b_x - b_y| \tag{11}$$

Thus, a good enhancement method that is able to preserve the mean brightness will give a small value of AMBE.

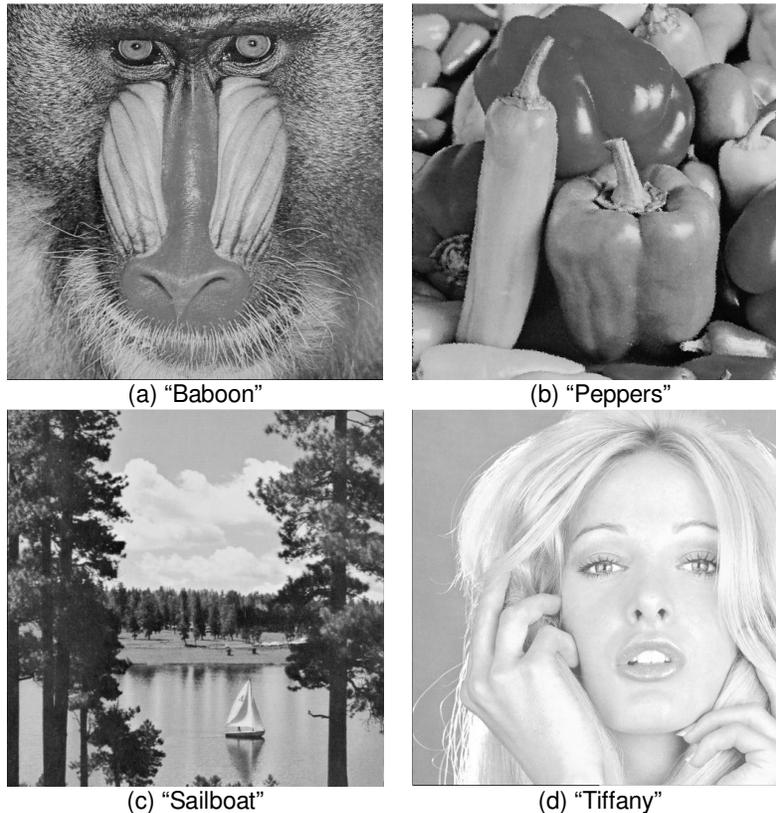


FIGURE 2: The four test images used in this work for the evaluation purpose.

Contrast Enhancement Method	Average AMBE
GHE	23.64
BBHE [5]	9.52
DSIHE [7]	11.01
MMBEBHE [8,9]	1.50
MHE [6]	12.10
RMSHE [9,10]	6.72
RSHE [11]	6.49
BPDHE [4,12]	0.84
BHEPL [14]	8.96
SHMS [3]	23.60
HERO (the proposed method)	3.43

TABLE 1: The average AMBE value taken from four test images.

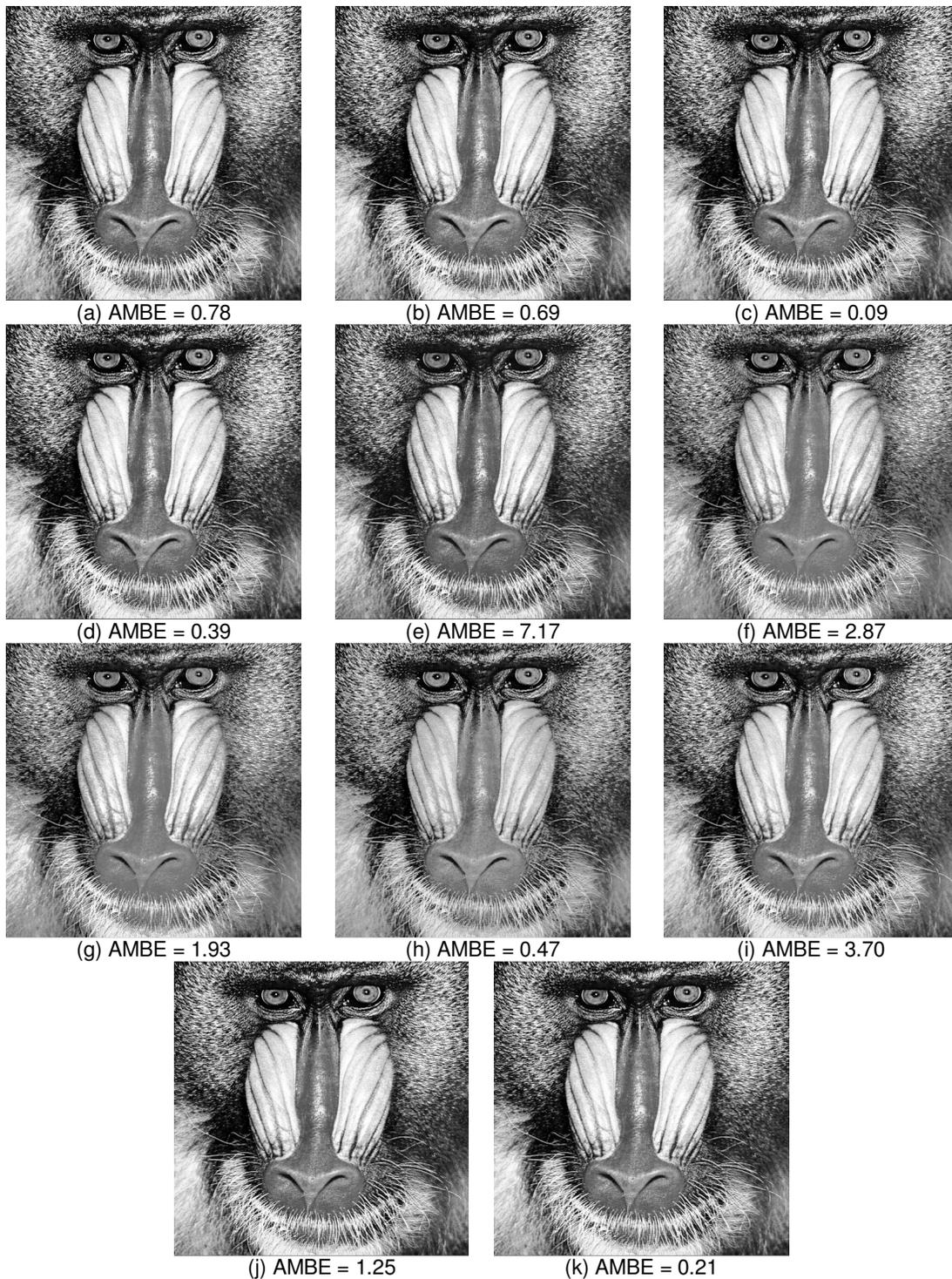


FIGURE 3: The enhanced version of “Baboon” by using (a) GHE, (b) BBHE, (c) DSIHE, (d) MMBEBHE, (e) MHE, (f) RMSHE, (g) RSHE, (h) BPDHE, (i) BHEPL, (j) SHMS, and (k) HERO (i.e. the proposed method)

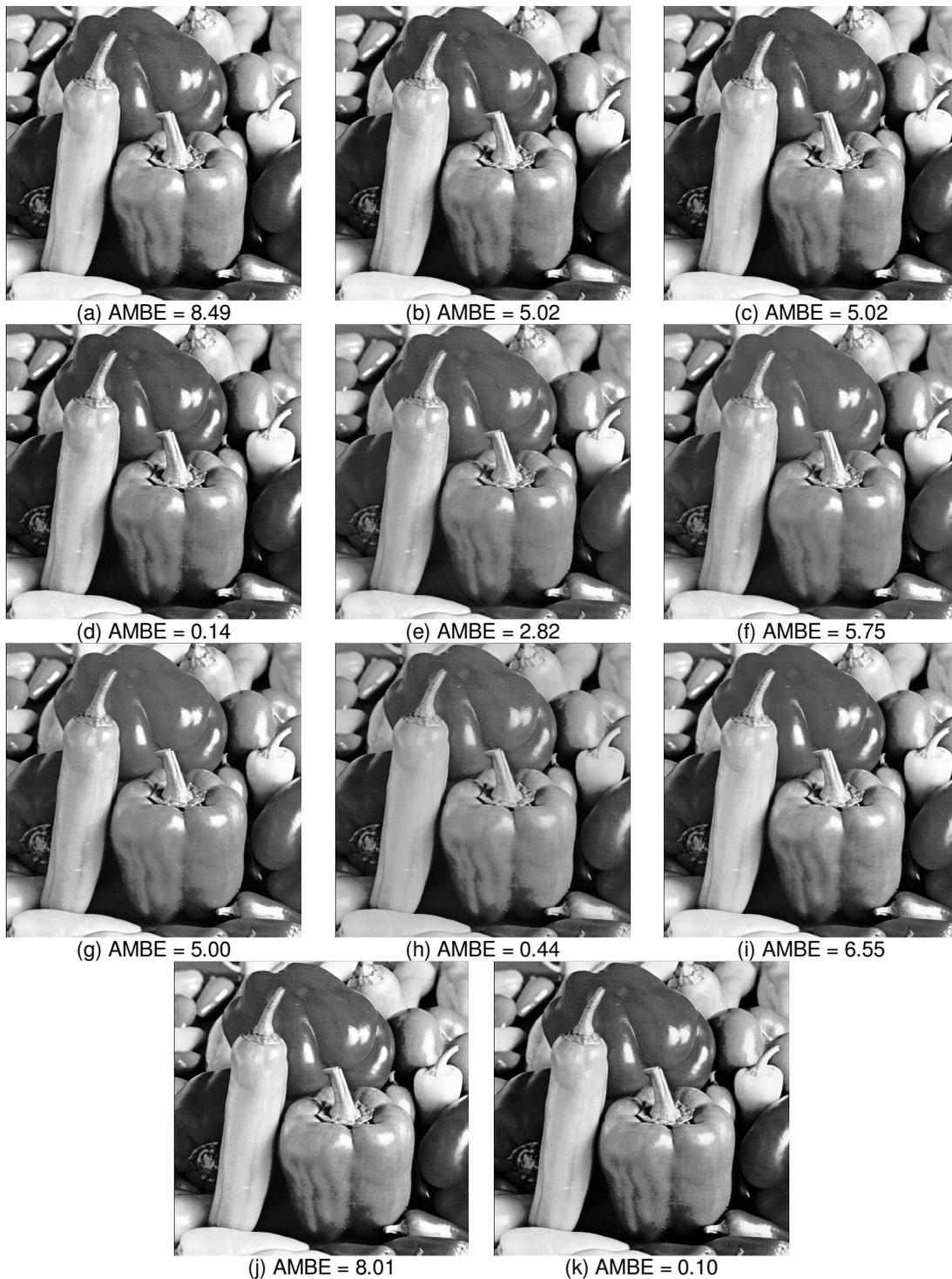


FIGURE 4: The enhanced versions of “Peppers” by using (a) GHE, (b) BBHE, (c) DSIHE, (d) MMBEBHE, (e) MHE, (f) RMSHE, (g) RSHE, (h) BPDHE, (i) BHEPL, (j) SHMS, and (k) HERO (i.e. the proposed method).

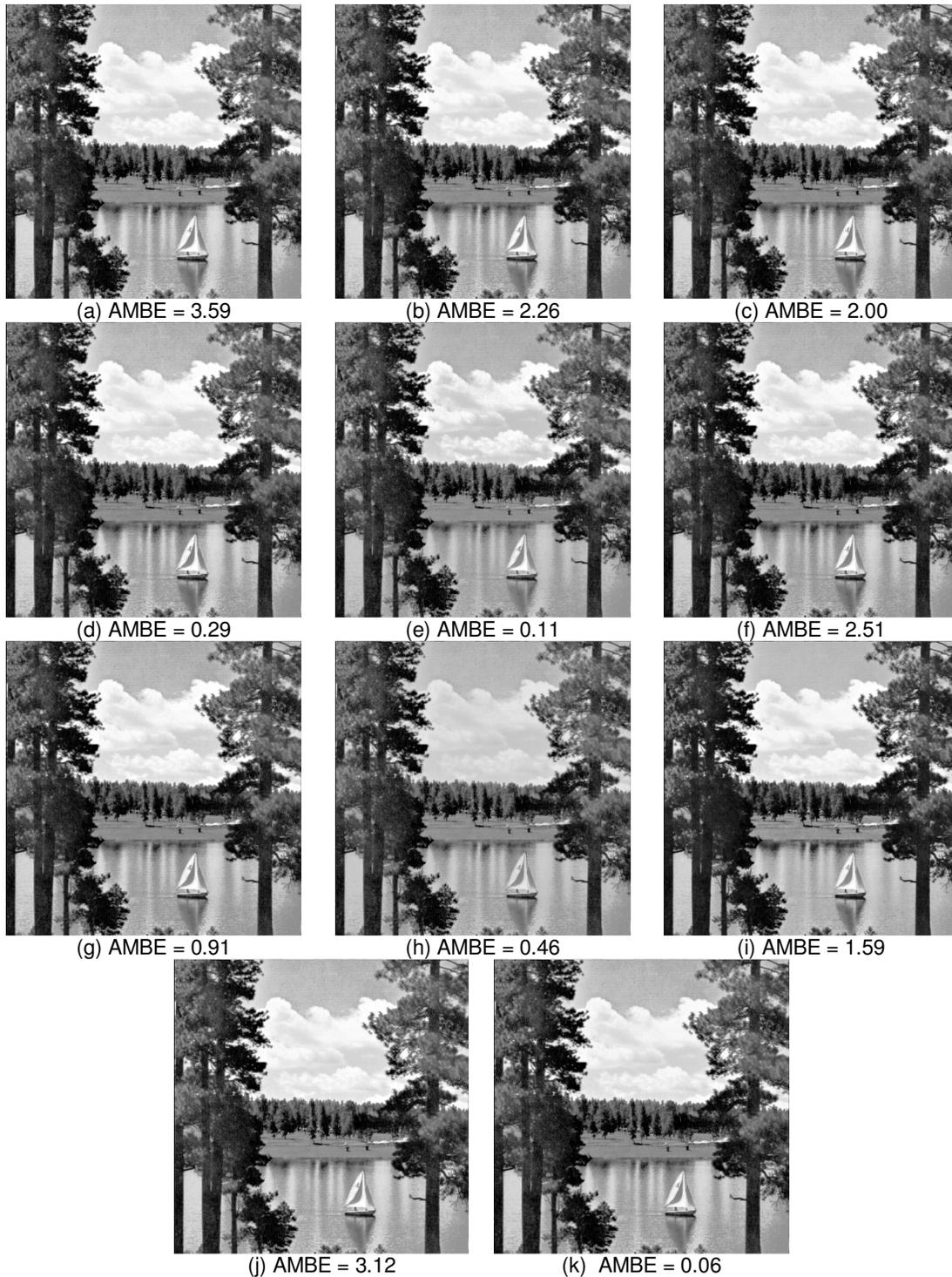


FIGURE 5: The enhanced versions of “Sailboat” by using (a) GHE, (b) BBHE, (c) DSIHE, (d) MMBEBHE, (e) MHE, (f) RMSHE, (g) RSHE, (h) BPDHE, (i) BHEPL, (j) SHMS, and (k) HERO (i.e. the proposed method).

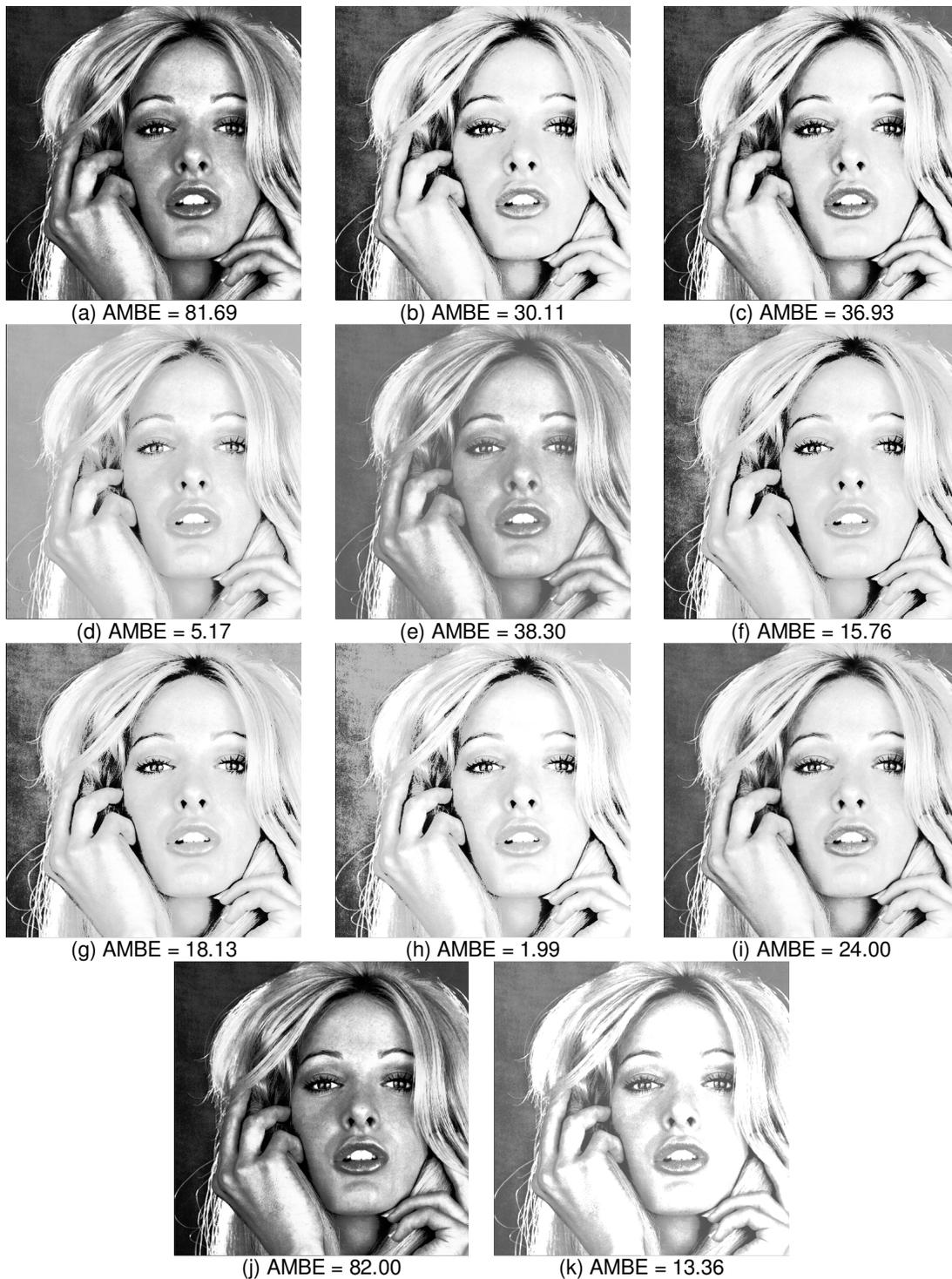


FIGURE 6: The enhanced versions of “Tiffany” by using (a) GHE, (b) BBHE, (c) DSIHE, (d) MMBEBHE, (e) MHE, (f) RMSHE, (g) RSHE, (h) BPDHE, (i) BHEPL, (j) SHMS, and (k) HERO (i.e. the proposed method).

The contrast enhanced version of the images shown in Fig. 2, obtained by using several histogram equalization based methods, including HERO, are presented in Fig. 3 to Fig. 6. As shown by these figures, all enhancement methods implemented in this research successfully increase the contrast in the image. In order to evaluate the performance of these histogram equalization based methods in terms of preserving the mean brightness, the average AMBE values obtained from these four test images are tabulated in Table 1.

Table 1 shows that in average, GHE method produces the highest AMBE value. This is not surprising as GHE does not put any constrain in preserving the brightness. On the other hand, the method that produces the lowest average AMBE value is the BPDHE. This is also not surprising, as this method is relatively more complex and requires more processing time as compared with other methods. The method that produces the second lowest average AMBE value is the MMBEBHE. Yet, similar to BPDHE, MMBEBHE requires histogram partitioning. The proposed method, HERO, has the third lowest average AMBE value. However, despite of this fact, it is worth nothing that as shown in Fig. 3 to Fig. 6, HERO produced the lowest AMBE value for two test images, namely "Peppers" and "Sailboat". In addition to this, excluding "Tiffany", HERO produced AMBE values less than one, indicates that the mean brightness of the input image is almost similar to the mean brightness of the output image produced by this method.

5. CONCLUSION

In this paper, an improved version of histogram equalization method has been proposed. By using a range offset, the proposed method successfully maintains the mean brightness of the image. This method is simple, and unlike other brightness preserving histogram equalization based methods, the proposed method does not require any histogram partitioning.

ACKNOWLEDGMENT

This work was supported in part by the Universiti Sains Malaysia's Short Term Research Grant with account number 304/PELECT/60311013 and by Incentive Grant (Postgraduate Students) with project number 1001/PELECT/8022006.

REFERENCES

- [1] Rafael C. Gonzalez, and Richard E. Woods, "Digital Image Processing", 2nd Edition, Prentice Hall, 2002.
- [2] Haidi Ibrahim, and Nicholas Sia Pik Kong, "Image sharpening using sub-regions histogram equalization", *IEEE Trans. Consumer Electronics*, vol. 55, no. 2, pp. 891-895, May 2009.
- [3] Yen-Ching Chang, and Chun-Ming Chang, "A simple histogram modification scheme for contrast enhancement", *IEEE Trans. Consumer Electronics*, vol. 56, no. 2, pp. 737-742, May 2010.
- [4] Haidi Ibrahim, and Nicholas Sia Pik Kong, "Brightness preserving dynamic histogram equalization for image contrast enhancement", *IEEE Trans. Consumer Electronics*, vol. 53, no. 4, pp. 1752-1758, November 2007.
- [5] Yeong-Taeg Kim, "Contrast enhancement using brightness preserving bi-histogram equalization", *IEEE Trans. Consumer Electronics*, vol. 43, no. 1, pp. 1-8, February 1997.
- [6] K. Wongsritong, K. Kittayaruasiriwat, F. Cheevasuvit, K. Dejhan, and A. Soomboonkaew, "Contrast enhancement using multippeak histogram equalization with brightness preserving", *The 1998 IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 455-458, Chiangmai, Thailand, November 2008.

- [7] Yu Wang, Qian Chen, and Baeomin Zhang, "Image enhancement based on equal area dualistic sub image histogram equalization method", *IEEE Trans. Consumer Electronics*, vol. 45, no. 1, pp. 68-75, February 1999.
- [8] Soong-Der Chen, and Abd. Rahman Ramli, "Minimum mean brightness error bi-histogram equalization in contrast enhancement", *IEEE Trans. Consumer Electronics*, vol. 49, no. 4, pp. 1310-1319, November 2003.
- [9] Soong-Der Chen, and Abd. Rahman Ramli, "Preserving brightness in histogram equalization based contrast enhancement techniques", *Digital Signal Processing*, vol. 14, no. 5, pp. 413-428, September 2004.
- [10] Soong-Der Chen, and Abd Rahman Ramli, "Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation", *IEEE Trans. Consumer Electronics*, vol. 49, no. 4, pp. 1301-1309, November 2003.
- [11] K. S. Sim, C. P. Tso, and Y. Y. Tan, "Recursive sub-image histogram equalization applied to gray scale images", *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1209-1221, July 2007.
- [12] Nicholas Sia Pik Kong, and Haidi Ibrahim, "Color image enhancement using brightness preserving dynamic histogram equalization", *IEEE Trans. Consumer Electronics*, vol. 54, no. 4, pp. 1962-1968, November 2008.
- [13] Nyamlkhagva Sengee, and Heung Kook Choi, "Brightness preserving weight clustering histogram equalization", *IEEE Trans. Consumer Electronics*, vol. 54, no. 3, pp. 1329-1337, August 2008.
- [14] Chen Hee Ooi, Nicholas Sia Pik Kong, and Haidi Ibrahim, "Bi-histogram equalization with a plateau limit for digital image enhancement", *IEEE Trans. on Consumer Electronics*, vol. 54, no. 3, pp. 2072-2080, November 2009.
- [15] Chao Wang, and Zhongfu Ye. "Brightness preserving histogram equalization with maximum entropy: a variational perspective", *IEEE Trans. Consumer Electronics*, vol. 51, no. 4, pp. 1326-1334, November 2005.

System of “Analysis of Intersections Paths” for Signature Recognition

Farhad Shamsfakhr

*Faculty of Engineering/Computer Engineering Department
Islamic Azad University/Hamedan Branch
Hamedan, 65138-734, Iran*

farhad_sm@ymail.com

Abstract

In today's world, the electronic city which is the offspring of the development of the information world, paves the way for a round-the-clock interaction among computers and networks. The planners of these electronic cities are mostly concerned about the accuracy and security of the exchanged information. In order to elevate security and raise speed and accuracy in the reviewing of network performance and the dependable identification of persons involved in electronic operations, recognizing the accuracy of the electronic signature is deemed absolutely essential. In this article, a system named "Analysis of Intersections" has been utilized for the accurate recognition of the electronic signature. Of important features of this system are the utilization of simple data structures such as array, stack, and list and determination of the sensitivity level for recognizing the accuracy of the signature by setting an error percentage for the size and recognition of the shape. An accuracy recognition test was performed on 15 samples of 150 types of signatures using "Analysis of Intersections". Findings indicated that this system showed an accurate recognition of 2,220 out of 2,250 signatures, indicating an applicability of 98.66 percent.

Keywords: Analysis of Intersections, Intersection, Threshold, Rotational Routing Algorithm, Intersection Recognition Samples, Adaptation of Paths.

1. INTRODUCTION

With the ever increasing advancement of the information technology over the recent years, electronic operations have gained momentum. Meanwhile, dissemination of personal and organizational information over the insecure worldwide web and the easy access of individuals and organizations to internet resources raise concerns regarding the unauthorized access of strangers to users personal information and the breaching of privacies. To the end of safeguarding the privacy of their users, planners of electronic cities have taken measures to define an identity for their users. This could happen in the form of defining a username and a password for the users or identifying unique characteristics of individuals such as fingerprint, face recognition, etc. Question is which method is the fastest, most precise, and most cost-effective of all? In fact, in order to increase the accuracy and precision of information on the one hand and the speed of electronic operation completion on the other, it is advisable that a sensible, optimum, intelligent, and easy-to-use method be created. A proper option for securing the entrance of users into the web is the accurate and intelligent identification of individuals signatures. Using intelligent systems, not only the speed of signature recognition but also its accuracy are augmented which is crucial when performing important monetary, information, and security operations. [2] The method introduced in this article concerns offline electronic signatures. In electronic signatures, there is no noise or halo of colors. In this study, we attempt to analyze the signature image as a collection of ways and introduce the main algorithms as follows: 1) way-finding algorithm shown as "Around _ perceive (x,y)" and "Way _ Finder (x,y,z)" functions; 2) intersection recognition algorithm shown as "Inter _ Section (x,y)" function; 3) end-checking algorithm shown as "sensing _ opr _ End _ checker"; and 4) a unique algorithm for comparing and adapting samples with other algorithms that are explained later. We refer to this system as "Analysis of Intersections". [3] By using electronic signatures, we do not need to take pre-processing operations such as elimination

of noises. In some systems, however, the image of the signature has a halo of colors instead of the main signature color. In such a case, a pre-processing operation is needed to transform the halo of colors to only one color (e.g. black). After receiving the signature in the form of a bitmap file, we separate the signature from the background; change the color of the background to white and color of the signature to black.



FIGURE 1: shows the main image of the signature.



FIGURE 2: the pre-processed image.

The program starts with retrieving "Image _ Analysis ()" function. This function navigates the array of the image to find the first pixel of the signature image (black pixel). After finding the first pixel, "Around _ Perceive (x,y)" function is retrieved. The task of this function is to search the environment around the received pixel in order to find the path it is going to navigate. It finds the right path, navigates it, and continues this action until it reaches the first intersection. The design of this function does not lead to registration of repeated paths and being trapped in loops known as "wrong paths". We take the pixel signified in the following figure to be the first pixel of the signature image which has been found by "Image _ Analysis ()" function.

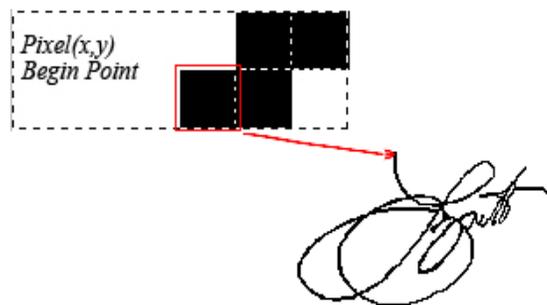


FIGURE 3: shows the magnified image of the signature.

After receiving x and y of the first pixel as the input parameters of "Around _ Perceive (x,y)" function, this function reviews 8 pixels around the current pixel in order to find one pixel to continue the path. This way the path is navigated and this continues until the first intersection in the signature image is reached. Further explanations in this regard are given in the Appendix. By this function, after finding one pixel as the found path, we must study the position of the pixel in every stage in order to reach the intersection. To this end, we retrieve "Inter _ Section" function. This function tries to receive the path pixel and decide whether it is in the intersection or not. If yes, the intersection is found and "Around _ Perceive" function has completed its job successfully. If not, "Around _ Perceive" is retrieved through the same pixel.

2. INTERSECTION

[4] Intersection is defined as the pixel where at least 4 different paths meet. In Inter _ Section function, various figures of the intersection, known as "intersection recognition samples" exist. [1] Comparing the position of the path pixel (the input pixel of Inter _ Section function) with the pixels around indicates whether it can be candidate as the intersection center or not.

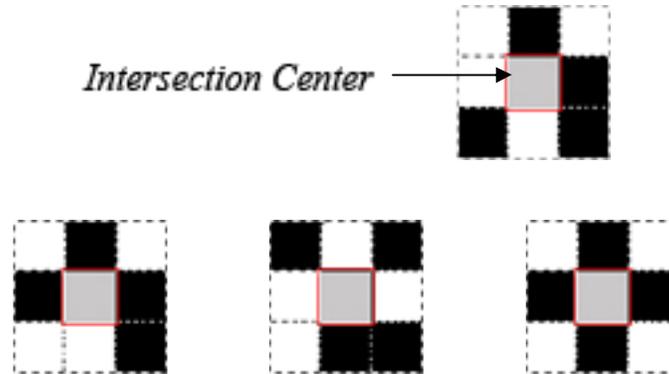


FIGURE 4: shows some of the intersection recognition samples.

2.1 Intersection Recognition

[5] After a pixel is candidate as the centre of intersection, the function decides whether this pixel is a definite center or just candidate as one. Pixel A in Figure 5 is just a candidate center of intersection as it is not the intersection of at least 4 distinct paths and is just a path pixel. In Figure 6, however, pixel A is a definite center as it is the intersection of at least 4 distinct paths. If a pixel is the intersection of 3 distinct paths, that pixel is known as the center of a three-way junction which after being added a virtual path, may become a definite intersection center. To decide whether an intersection center is definite or not, "Inter _ Section" function uses an algorithm known as Rotational Routing algorithm described fully in the Appendix.

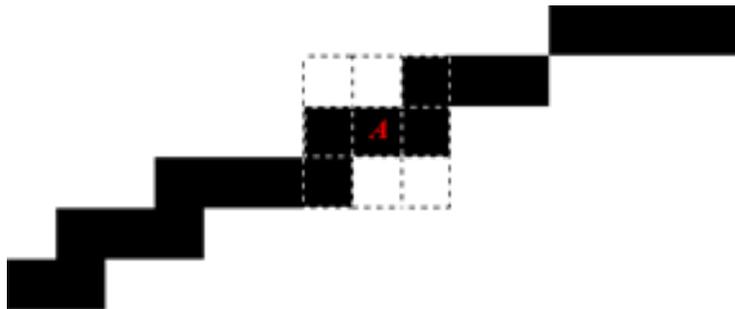


FIGURE 5: shows the candidate intersection center.

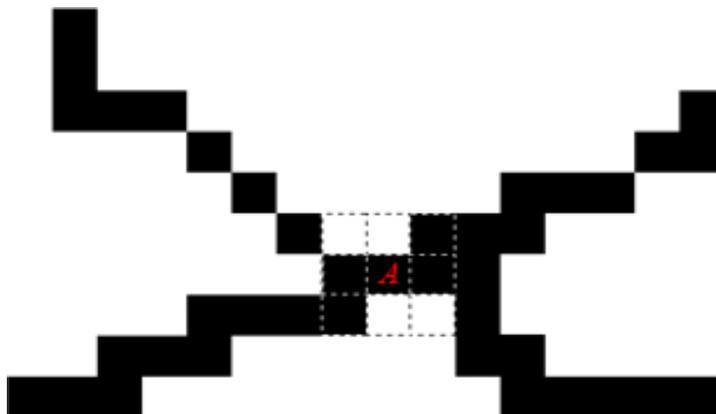


FIGURE 6: depicts the definite one.

After recognizing the intersection, all paths ending to the intersection are registered in the array temporarily, with the direction of every path and then the paths are navigated. Navigating these paths is done by "Way _ Finder (x,y,z)" function. As said before, every path has a direction. In order to optimize searches, the method of navigating each path is based on the direction of that path. Therefore, for navigating two paths with two different directions, we use two different methods, described fully in the Appendix. Navigating the path continues to the end of the path or until meeting an intersection. If the path is not repeated, it is registered in the array. "Way _ Finder (x,y,z)" function reviews the position of the pixel in every stage of navigation in order to reach the end of the path. To find out whether this path is terminated or not, "sensing _ opr _ End _ checker" function is used. This function receives a pixel from "Way _ Finder" function and decides whether the current path has ended or not. If the path has not ended, the said function will retrieve "Way _ Finder" function again. Therefore, all the paths ending to the first intersection in the signature are registered in the array. Then, all the paths of this intersection are eliminated from the signature image and all the stages of the operation are repeated until no non-navigated path remains in the signature image. If a path does not end to an intersection, it will be called a separated path. Such paths are eliminated without being registered in the array. Figure 8 shows the signature analysis stages of Figure 7. In every stage of signature processing, navigated paths of an intersection are shown. The signature image after one stage of processing is also depicted. The red path is a virtual path that we add to three-way junctions so that they will be turned to intersections.



FIGURE 7: shows a signature sample.

<i>path 1</i>	<i>path 2</i>	<i>path 3</i>	<i>path 4</i>	<i>signature afther process</i>
			<i>none</i>	
			<i>none</i>	
		<i>none</i>	<i>none</i>	
		<i>none</i>	<i>none</i>	

FIGURE 8: shows the stages of processing the paths of that signature along with processed paths in every stage and the image of the signature after being processed in each stage.

3. Adaptation of Paths

After storing all pixels in every path in the array, we have to use these paths to identify the signature. In order to identify the signature, first "Path Information Table" must be formed for each of the paths of the signature. This Table includes: 1) starting point: containing the features of the first pixel of every path; 2) ending point: containing the features of the last pixel of every path; 3) x range: containing the range of x's in every path; 4) y range $Dx=Max(x)-Min(x)$: containing the range of y's in every path; and 5) the whole path $Dy=Max(y)-Min(y)$: containing the number of pixels in every path. "Path _ Receipt _ Data _ Process" function creates the Path Information Table, fully described in the Appendix. After such table is created for all paths, we have to decide on the shape, direction, size & the slope measurement of every path based on the information contained in this table as follows: first a Measure is considered for every signature. Measure is a unit that measures the length of the path. For instance, if the length of a path is 30 pixels and its measure is 3, the length of this path will be 10 measures.

$$\text{Measure} = \text{Path Length} / n$$

By using Measure, size will no more be a limiting factor; therefore, any size signature can easily be recognized. We consider that Harry Hilton signed his signature in two different sizes.



FIGURE 9: shows a path of his first signature.

3.1) length path = 180 pixel , Measure = $180 \setminus 10=18$, length path= $180\setminus 18=10$ unite and



FIGURE 10: a path of his second signature.

3.2) length path = 78 pixel , Measure = $78 \setminus 10=7.8$, length path= $78\setminus 7.8=10$ unite The lengths of both paths are 10 units.

Threshold

Threshold is the amount we use as the error limit. If in the rectangular in Figure 11 the threshold is 4, the rectangular will be complete if we ignore 4 pixels (error limit=4).



FIGURE 11: the threshold is 4, the rectangular will be complete if we ignore 4 pixels (error limit=4).

In this article, we calculate $\text{Span_Limit} = \text{Path Length} / 7$. A different calculation method can also be used. The smaller the Span_Limit , we will identify shapes with smaller error compared with real shapes. The bigger the Span_Limit , bigger adaptation errors are ignored. The slope of each path are calculated as follows:

$$Y=mx+b \quad , \quad m = dy/dx = (y_2-y_1)/(x_2-x_1)$$

We apply our own definition for the shape, direction, and size of images. For example, Figure 12 in the signature is called a big quasi-rectangular which is shown in Figure 12.



FIGURE 12: big quasi-rectangular.

Our definition of shapes and directions is given in the Appendix.

The shape, direction, slope and size of a path are referred to as the features of a path. In order to identify the accuracy of the signature, the following operation is taken: The number of paths in signature A is compared with the number of paths in signature B. If they are equal, the features of every path in one signature are compared with those of the corresponding path in the other signature. In case features of the paths in both signatures are identical, the accuracy of the signature is vindicated. In order to improve the identification performance, a certain percentage of size differences must be ignored. For instance, if the features of every single path in one signature is exactly identical with those of the corresponding path in the other signature, but there is discrepancy in the size or the length of paths by 1 or 2 units, we may consider the length of

paths as identical after ignoring 2 units of discrepancy. An accuracy recognition test was performed on 15 samples of 150 types of signatures using "Analysis of Intersections". Findings indicated that this system showed an accurate recognition of 2,220 out of total 2,250 signatures, indicating an applicability of 98.66 percent.

Authors	Method	Results
Ammar, M. 1991	Distance Threshold	85.94%
Ammar et al., 1990	Distance Statistics	88.15%
Quek & Zhou, 2002	Neuro-Fuzzy Network	96%

TABLE 2: Comparative Analysis of different off-line signature verification systems.

4. REFERENCES

- [1] S. E. Umbaugh, Computer Imaging Digital Image Analysis and Processing, CRC Press, 2005
- [2] M.Savov, G.Gluhchev, Automated Signature Detection from Hand Movement, "International Conference on Computer Systems and Technologies _ CompSysTech" 2004
- [3] A.Zimmer, L.Luan Ling, A Hybrid On/Off Line Handwritten Signature Verification System, "Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003) " 0-7695-1960-1/03 17.00 C 2003 IEEE
- [4] A recognition algorithm for the intersection graphs of paths in trees . Computer Science Division, Department of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv, Tel-Aviv, Israel Received 10 March 1977;
- [5] A recognition algorithm for the intersection graphs of directed paths in directed trees . Department of Computer Science, University of Illinois, Urbana, Ill. 61801, USA

Annex

1. "Around Perceive (X,Y)" function

The task of this function is to search around the pixel received by itself to find the right route at its own front and to keep sensing the same until it comes to the first intersection . this effect the eight pixels around the current pixel are surveyed and the first black pixel so found shall be considered as the conducted pixel of continuation of the route. The adjacent pixels are surveyed line by line beginning from the first line and the first column to the last line and the last column in the following way.

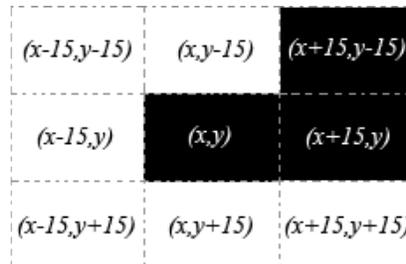


FIGURE 1: pixels around the current pixel(* Now suppose that each Pixel measures 15x15 in dimensions.)

```

For j = y - 15 To y + 15 Step 15
For i = x - 15 To x + 15 Step 15
  If i <> x Or j <> y Then
    color = pic3.Point(i, j)
    If color = 0 Then
      {
        .
        .
        .
      }

```

The current pixel that is the input of the around perceive (X,Y) function with in each phase is refereed to as parent pixel and the candidate pixels of the same phase are culled child pixels. In case a candidate pixel is taken as a route, its parent pixel shall be reserved in an array named pixels Memory. After the candidate pixel has been discovered within each phase, it is this surveyed from the viewpoint of being a repeated pixel where we may face two alternative circumstances; 1st: there is a minimum of one (1) unrepeated candidate pixel. Absence of candidate pixel in the array of pixels _Memory would indicate that the route has been discovered and that it is the route of candidate pixel. Then, it is considered if its parent pixel exists in the array parent pixel shall be reserved in the array, it is reserved in the array 2nd. All candidate pixels are repeated pixels. In case all candidate pixels are present in pixels Memory array, this means that the route has not been discovered in which case a stack shall be used. With this state also the parent pixel shall be reserved in the array provided that it is not a repeated pixel. Lack of discovery of a route indicates that the route we have covered to this point is wrong and that we should go back to take another way. Pointer points at the end of the array pixels Memory. Therefore, we are to reduce the pointer by one unit per phase and consider as new candidate pixels the child pixels pointed at by the pointer continuing this to the point where a route appears. We convert the array of pixels Memory to stack in fact by using the pointer and retrieve the function by the values available in the stack in the following method.

Around_Perceive(Pixels_memory(pointer).x, Pixels_memory(pointer).y)

2. Intersection Identification Function

How many patterns are required to identify the intersection? Is it possible for us to identify intersection candidate in the form of signature by using these patterns? It may appear initially that to improve intersection identification performance we are to create all combinations of N black pixels in M-1 locations and in other words we should have 2520 different intersection identification patterns:

$$C(8,4) = \frac{8!}{4!(8-4)!} = 2520$$

It is evident that it shall be extremely time- consuming to create this number of patterns and it is a useless effort. We don't need to do this; there is no requirement of taking as a pattern any four of eight combination. We take as a model only the states more frequently occurring in the intersection of a signature and those states do not exceed 40 in number. Whereas one compares to intersection patterns any singles pixel adjacent to the input pixel of the function inter section, then any pixel conforming to none of our patterns shall be left and shall give its place to an adjacent pixel in an order of comparison. As an instance let us suppose that the pattern 1 has not been defined in the function; that pattern 2 has been defined in the functional and that we find the following form in the signature. It is evident that we cannot identify the position of pixel A in the intersection be cause we have not defined pattern 1 in the function. However, me may identify the position of pixel B in the intersection as pattern 2 has been defined in the function.

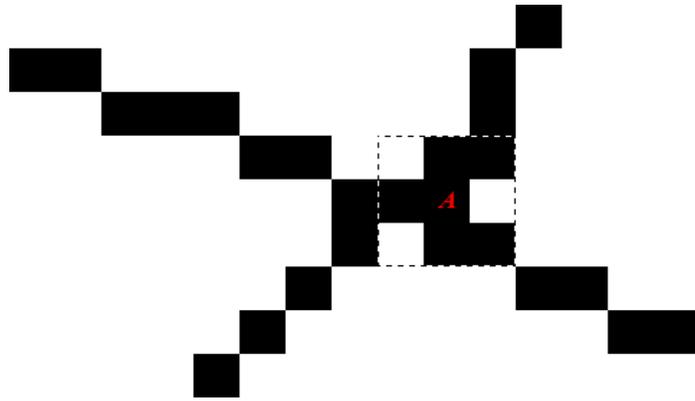


FIGURE 2: position of pixel A in the intersection

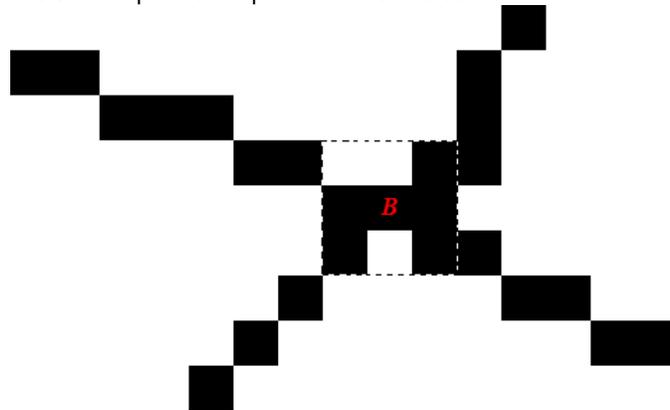


FIGURE 3: position of pixel B in the intersection

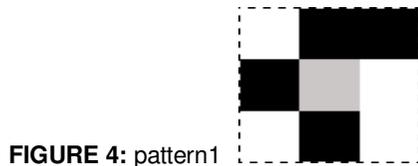


FIGURE 4: pattern1



FIGURE 5: pattern2

As observed, the intersection was finally identified with the definition of one pattern.

3. ROTATIONAL ROUTING

To determine if a pixel can be the center of a definite intersection, the function inter section takes use of a method which we call Rotational routing. Because of congeries of pixels at the intersections,

For recognizing definite intersection, the distincgtion action of the directions would be perform in the place in which, the route is released of the pixel congestion.

all the possible routes on the eight sides of the intersection are counted clockwise after the intersection center candidate has been identified within the intentional routing method. Keep in mind that the starting point of the aforesaid routes shall be considered to be at a clearance by three (3) pixels from the intersection center so that a separation in made from the point of congeries of pixels at the intersection center. The rotational routing method at a depth of 3 is considered here. However, a rotational routing at a depth of 4 may be used as an alternative with the advantages of bringing greater flexibility and preventing errors.

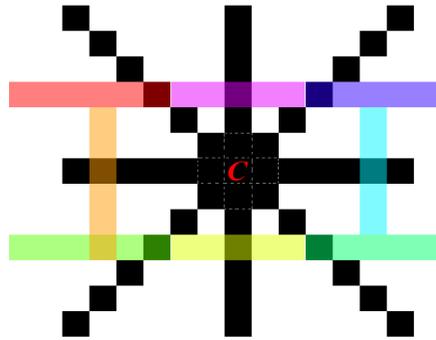


FIGURE 6: Rotational Routing at a depth of 3.

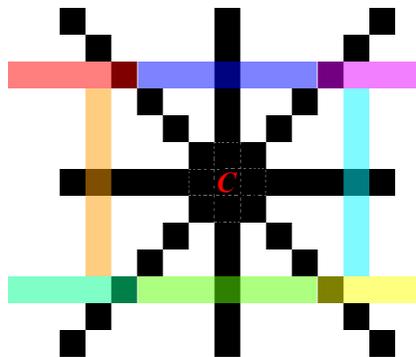


FIGURE 7: Rotational Routing at a depth of 4.

The following figure illustrates method of functioning of rotational routing at a depth of 3.

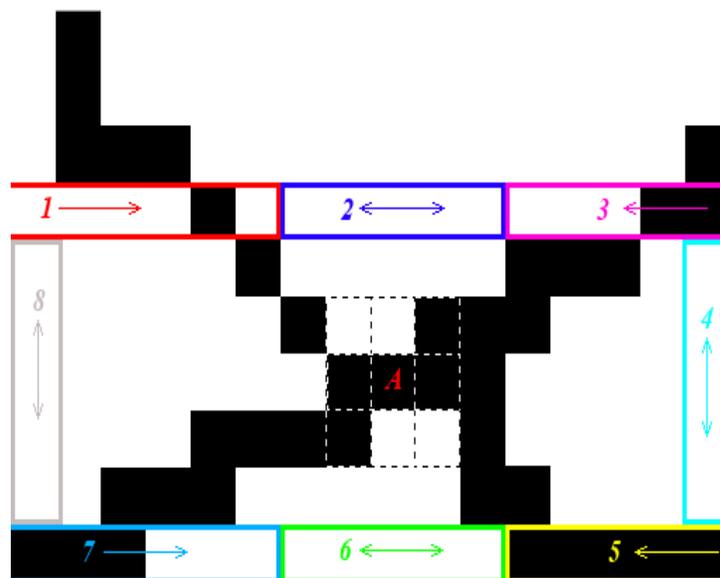


FIGURE 8: functioning of rotational routing at a depth of 3.

(X,Y) is intersection center coordinate of the pixel dimensions are 15x5 in this example). A is the center of intersection. The searches 1,2,3,5,6 and 7 are made at 3 pixels from the intersection center (x-45) or (y-45). The searches 4 and 8 are made at an optional clearance from intersection center. However, the clearance shouldn't be so little that it is included within the area of congeries of pixels and not so large that make the length of the route so discovered excessively small. The search is in every case made at the beginning of the search area and continued to the end. The search is terminated upon reaching the first pixel. The pixel so encountered is recorded in the array as the first pixel of the discovered route. Each intersection has a maximum of eight routes and hence eight searches shall be required to discover the eight routes. The search no 1 is dedicated to the discovery of the route No.1 (northwestern route). The length of search route includes the distance between an optional clearance, say X-225 as an instances to specified clearance e.g, X-45. The search No. 2 is dedicated to the discovery of the route No.2 (northern route). The length of the search route includes to distance between X-30 to X+30. search No.3 is dedicated to the route No.3 (northeastern route). The length of search route is the distance between an optional clearance, say X+225 to the specified clearance X+45. the search no 4 is dedicated to the discovery of route No.4 (eastern route). The length of search route includes to distance between Y-30 and Y+30 The search no 5 is dedicated to the route No.5 (southeastern route). The length of the search route is the distance between an optional charana say X+225 to the known clearance X+45. The search No. 6 is dedicated to the route No.6 (southern route). The length of search route covers the distance from X-30 to x+30. The search No. 7 is dedicated to discovery of route No. 7 (southwestern route) the length of search route is the distance from an optional clearance, say X-225 to the specified clearance X-45. Search No. 8 is dedicated to the discovery of route No. 8 (western route). The length of search route covers the distance from Y-30 to Y+30. Therefore, all the routes, their respective numbers and their directions are reserved in an array namely Inter Section Ways Array by using rotational routing algorithm. In case number of the routes is equal to exceeds 4, then the intersection candidate pixel is an absolute intersection and otherwise the function Around_Perceive is retrieved by using a candidate pixel and the remaining part of the route is measured until the intersection appears again.

4. OPTIMIZATION OF SEARCHES

Any pixel to be searched by Way_Finder function for the continuation of the route, should have a specific search method of its own. Uniform order of searches dedicated to all roles shall result in an increase in number of comparisons, emergence of errors and a drop in processing rate of the program. Suppose that current pixel is the pixel A and the purpose is to find pixel B for the continuation of the route.

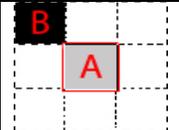
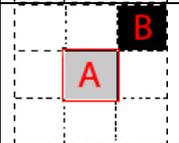
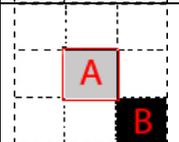
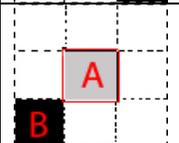
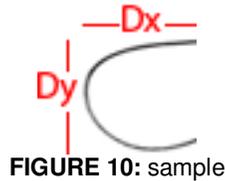
	For j = y - 15 To y + 15 Step 15 For i = x - 15 To x + 15 Step 15
	For j = y - 15 To y + 15 Step 15 For i = x + 15 To x - 15 Step -15
	For j = y + 15 To y - 15 Step -15 For i = x - 15 To x + 15 Step 15
	For j = y + 15 To y - 15 Step -15 For i = x + 15 To x - 15 Step -15

FIGURE 9: every route along with the proper search orders (dimensions of each pixel measure 15x15)

5. RULES OF FORM & ROUTE DIRECTION IDENTIFICATION

The following rules are those of form and route direction identification. As observed , form and route identification rule , name the name of route form, an example of the same and the direction of the respective route have all been given in each line.



Law	Shape	Sample	Direction
1.1) If $Abs(DX - DY) < (Span_Limit)$	Half square		horizontal
1.2) If $((DX - DY) \geq (Span_Limit)) \& (DX \leq 2 * DY)$	Half rectangular		horizontal
1.3) If $(DX > Span_Limit) \& ((DX + Span_Limit) \leq (DY))$	small Half square		horizontal
1.4) If $(DX > 2 * DY)$	big Half rectangular		horizontal

TABLE 1: Horizontal forms Identification rules. (1.) If $Abs(y_n - y_1) > Span_Limit \& Abs(x_n - x_1) \leq Span_Limit$

In case name of the aforesaid rules applies to a form, then the following rules shall be surveyed:

Law	Shape	Sample	Direction
2.1) If $Abs(DX - DY) < (Span_Limit)$	Half square		vertical
2.2) If $((DY - DX) \geq (Span_Limit)) \& (DY \leq 2 * DX)$	Half rectangular		vertical
2.3) If $(DY > Span_Limit) \& ((DY + Span_Limit) \leq (DX))$	small Half square		vertical
2.4) If $(DY > 2 * DX)$	big Half rectangular		vertical

TABLE 2: Vertical forms identification rules .(2.) If $Abs(y_n - y_1) \leq Span_Limit \& Abs(x_n - x_1) > Span_Limit$

In case none of the afore rules applies to a form, then the following rules shall be considered.

Law	shape	The Sample	Direction
3) If $(DY \leq Span_Limit)$	Horizontal line		horizontal
4) If $(DX \leq Span_Limit)$	Vertical line		vertical

TABLE 3: Secondary rules.

Recognition of Farsi Handwritten Numbers Using the Fuzzy Method

Mansoreh Sharifzadeh
*Faculty of Engineering
University of Hormozgan
Bandar Abbas, 74318, Iran*

sharifzade@yahoo.com

Shahpour Alirezaee
*Faculty of engineering
Razi University
Kermanshah, 74555, Iran*

Sh_alirezaee@yahoo.com

Abstract

There are wide varieties of handwritten characters which differ not only from person to person but also from the state of mood of the same person. Nevertheless humans are trained to extract the specific features characterizing a symbol. This paper aims to introduce fuzziness in the definition of the proposed pattern features, which provides the enhancement to the handwritten character information to be stored. Some novel shape features in fuzzy linguistic domain are proposed. The experimental results indicate 95 % accuracy on recognition of Farsi numbers over the selected database.

Keywords: Character Recognition, Farsi Handwriting Numbers, Fuzzy Method, Geometrical Features.

1. INTRODUCTION

The objective of the Handwritten Character Recognition is the recognition of data that describe handwritten objects. On-line handwritten recognition deals with a time ordered sequence of data for different persons from various backgrounds e.g. nationality, education or profession. Development of such a system is only possible if the character prototypes are defined with a flexible rule base including only handwriting style independent features. The primary goal of our methodology is to represent the handwriting information into a minimum number of meaningful features. These features are subsequently stored into a compact database.

Many methodologies have been developed for the recognition of handwritten characters. Some of the more recent methodologies include the use of Bayesian inference (Cheung and Yeung, 2002), Neural Networks (Koerich and Leydier, 2002; Lee and Kim, 1995), Fuzzy Logic (Malaviya and Peters, 1995) and Genetic Algorithms (Kim and Kim, 2000). When compared with other description methods, fuzzy description is probably the most efficient in terms of computational usage. Due to this reason, fuzzy logic is an appropriate method for online character recognition [1],[2]. The proposed method achieves the flexibility in recognizing varied handwritings by using linguistic fuzzy rules.

The human visual sense is selectively activated in response to curved lines and other geometrical characteristics [2]. These features themselves contain certain vagueness in terms of their definition. The human recognition system is most accurate in grasping the typical geometrical features of handwritten characters while ignoring the vagueness. To compensate these inherent geometrical shape distortions existing in the different handwriting styles we have applied the fuzzy set theory [8]. By use of fuzzy technique we integrate the existing vagueness into membership function of basic structural features. These possibilities are then estimated and further processed with fuzzy aggregation techniques.

This paper has been organized as follows. First, a quick glance on fuzzy linguistic modeling is given and some definition and general consideration related to shape recognition are presented. Shape recognition using the fuzzy feature will be described in the fourth section. In this section we will have a complete presentation on the fuzzy features for shape recognition. Then the proposed method will be presented in the fifth section. Experimental results will be discussed in the sixth section. Finally the conclusion of the presented method and the related performance will be discussed in the last section.

2. THEORY OF FUZZY LINGUISTIC MODELING

The linguistic variable is the core of the fuzzy modeling technique. Through the introduction of linguistic descriptors, phenomena which are too complex or too ill defined to be susceptible to be described by exact quantitative terms can be defined with ease. Therefore a linguistic variable can be either regarded as a variable whose value belongs to a fuzzy set or as a variable whose values are defined by linguistic terms.

Each linguistic term has a meaning associated to it by a fuzzy set. Membership functions of a given linguistic term A can be formulated such as the members of universal set X fall within a specified range $[0,1]$ and indicate the membership grade $m_A(x)$ of these elements in question. Definition of a Fuzzy linguistic is as follows:

"A linguistic variable x is identified by its name and characterized by a term set $T(F(x))$, where $F(x)$ denotes the fuzzy membership function."

For example, if a linguistic variable x_{hp} , which stands for "handwriting property x " is taken as a general example, its term set could be represented as $T(F(x_{hp})) = \{ \text{Zero, Very Very Low, Very Low, Low, Medium, High, Very High, Very Very High, Excellent} \}$

Definition of the fuzzy sets can be categorized into the discrete and continues values. In case of discrete values numerical vectors are used for assignments of grades of memberships. The second method, for continuous values, which is accomplished with the help of membership assignment in functional form. Typically used membership functions are S- or P-shaped, bell-shaped, triangular and trapezoidal function. The most widely used fuzzy membership function is the triangular membership function, whose shapes are variations of a triangle.

3. COMBINING FEATURES WITH FUZZY AGGREGATION

Fuzzy aggregation mechanism aims to find an overall measure of certain fuzzy information from an uncertain and imprecise information data. The selection of meaningful features from a given set of handwriting data is dependent on the possible relationships among these feature categories. The meaningful feature designates the features with a good discrimination factor. To make a hierarchical structure a two phase aggregation scheme has been developed which is based on the union and weighted generalized mean aggregation connectives. The wide ranges of these connectives provide flexibility in finding the most suitable output feature set.

4. SHAPE RECOGNITION

The shape is considered as the primal geometrical property for extracting pattern recognition features. The daily life objects are mostly recognized because of their specific shapes. The perception of shapes can be viewed as a collective cognition of the properties like size, form, symmetry and orientation. Thus a fuzzy approach to shape analysis incorporating imprecise concepts merits consideration.

The definition of a shape in terms of linguistic description like "curve is long / round/ thin" etc. is easily related by the humans to known patterns. But it is quite vague to define the shape of the objects by these descriptors as there is an obvious incompatibility with numerical processing methods. A closer look at these descriptors show that geometrical, positional and global features provide some insight about the shape itself provided they can be related to linguistic attributes or

terms. For example “curve” or “circle” or “line” have a well-defined geometric description but the definition of terms like “long”, “thin” “round” are more difficult to be expressed by formulas without the semantic power offered by the fuzzy set theory. Thus if we take again the circle example the shape description “the circle is round” is a fuzzy syntactic rule where circle is the linguistic variable and round is one of the possible linguistic terms associated to the circle with a membership defined for e.g. by Eq. 8. Our proposed approach collects the shape information through the extraction of geometric, positional and global features and expresses this information in the set theoretic fuzzy manner.

In the following subsections the computation and the properties of these features are described. While the global features characterize the pattern as a whole, e.g. aspect-ratio, the geometric and positional features describe local aspects for each identified sub-pattern. The extraction of predefined sub-pattern is done through a rough segmentation step in which abrupt changes in shape are used as discriminator.

To find or identify certain shape features we have first to define our universe of discourse. In our case this is the smallest rectangle in which the unknown character fits. This rectangle is defined by four parameters:

4.1 Positional Features

The positional features determine the relative position of the global or geometrical feature in the given rectangular window and are defined in the universe of discourse [0,1]. The two dimensional universe of discourse is divided into six linguistic terms: {Left, Centre, Right} and {Top, Middle, Bottom}. These are associated to the fuzzy linguistic variable or feature Vertical Position (VP) and Horizontal Position (HP) and express the relative position of a point, region, or segment to the centroid (xm,ym) of the analysed character.

These linguistic terms can be combined creating additional linguistic terms, e.g. “circle to the left-centre” and “line right-top-centre” which could identify a “d”. Just by interchanging the linguistic term “left” with “right”, “circle to the right-centre” and “line left-top-centre” the renewed description would correspond to a “b”.

4.2 Global Features

The global features describe the character as a whole and can be used for the pre-classification of the character. These shape features represent the global approximation of the shape. One method of shape classification relay on global features, e.g. area, compactness. These shape features are the global approximation of the shape. In other words information about the shape from all portions combines to form the global description.

The basic idea is to follow human visual modeling and to reproduce geometrical features in comprehensive terms. In this paper we have used the Start-point and End-point as follows.

Start-point and End-point

The starting point and the ending point of the character can be described related to the defined universe of discourse:

$$\mu_{S_Y} = \frac{y_0 - y_{\min}}{y_{\max} - y_{\min}} \quad . \quad . \quad !$$

$$\mu_{S_X} = \frac{x_0 - x_{\min}}{x_{\max} - x_{\min}} \quad . \quad \square$$

$$\mu_{E_Y} = \frac{y_N - y_{\min}}{y_{\max} - y_{\min}} \quad . \quad \square$$

$$\mu_{E-X} = \frac{x_N - x_{\min}}{x_{\max} - x_{\min}}$$

. □

where $(x_0, y_0), (x_N, y_N)$ are the coordinates of the start and end point of the handwriting profile and $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ are the borders of the universe of discourse. The values of μ_{S-X}, μ_{S-Y} are scaled to fit into a range of: [0:Left, 1:Right] and [0:Bottom, 1:Top] respectively. Fig.1 presents this feature.

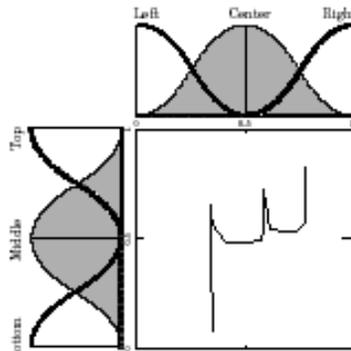


FIGURE1: Start-point and End-point feature

4.3 Geometric Features

We defined two classes of local geometric features: straight-line and curved-line. A segment description based on these two feature classes could be e.g. {class Arc, feature C-like} or {class Straight Line, feature Vertical Line,}.

The first step is the classification of the unknown segment to one of these geometric classes. This process is done through a fuzzy measure of “arced-ness” and “straight-ness”. As each class has several features (subclasses) the next step is to measure the belonging of the unknown segment to all subclasses. Due to the “fuzzy” nature of handwritten characters several solution are possible. The defined features (subclasses) form several fuzzy sets which are related thus covering the whole space of possible pattern distortions. We have applied two kinds of the geometrical features, measure of arced-ness and straight-ness, as follows.

Measure of arced-ness and straight-ness

The shape of a given segmented sub-patterns can be classified to the class of “straight line” or “arced line”. Pal and Majumder showed that the measure of straight-ness of a segment is determined by fitting a straight line with the minimum least squares error. Similarly in a given segment, ratio of the distance between end points to its total arc length shows its arced-ness. In other words if the distance between the end points is nearly equal to the total arc length, it is possibly a straight line. Otherwise if the arc length is much greater than the distance between the end-points then it is very likely a curve. Here it is assumed that the curve is monotonous. If we consider the measure of arced-ness and the measure of straight-ness to be two complementary fuzzy linguistic terms of the fuzzy linguistic variable shape and their definition for a segment is the relative deviation in the arc-length from the length of a straight line joining its end-points, then these two measures can be expressed as follows:

$$\mu_{Arc}(S_j) = \left(1 - \frac{d_{P_{j0}P_{jk_j}}}{k_j - 1} \sum_{k=0}^{k_j-1} d_{P_{jk}P_{j(k+1)}}\right)^\beta \quad \square$$

$$\mu_{Straightness}(S_j) = \left(\frac{d_{P_{j0}P_{jk_j}}}{k_j - 1} \sum_{k=0}^{k_j-1} d_{P_{jk}P_{j(k+1)}}\right)^\beta \quad \square$$

where β is a real positive number. This fuzziness factor β introduces a compression or expansion of the defined fuzzy membership function enables to adapt the relation to the respective operating range. Based on experimental test results the suitable value of β is chosen 0.5 for the arced-ness and 1 for straight-ness. The results indicate, for Farsi numbers, 0.3 for the arced-ness and 1.4 for straight-ness achieve the best recognition rate.

4.3.1 Class of Fuzzy Straight Lines

If a segment is identified as a straight line then the orientation or the angle of inclination of this segment distinguishes it further into one of the following features: vertical line, horizontal line, positive slant, and negative slant. The corresponding membership function as $\Lambda(\varphi, B, C)$ (Eq. 6) where φ represents the angle of orientation of the straight line, the bandwidth b is 90o and at c the membership value is maximum which is unity.

Vertical line: VL

A vertical line (|) has an ideal orientation of 90o or 270o. Therefore the fuzzy linguistic term *vertical line* is defined as a triangular membership function $\square(\square; b, c)$ by the following equation:

$$\mu_{VL}(\varphi) = \text{MAX}(\Lambda(\varphi, 90^\circ, 90^\circ), \Lambda(\varphi, 90^\circ, 270^\circ)) \quad \square$$

Horizontal line: HL

A horizontal line () has an ideal orientation of 0o, 180o or 360o. Therefore a fuzzy *horizontal line* is defined by a triangular membership function by the following equation.

$$\mu_{HL}(\varphi) = \text{MAX}(\text{MAX}(\Lambda(\varphi, 90^\circ, 0^\circ), \Lambda(\varphi, 90^\circ, 180^\circ)), \Lambda(\varphi, 90^\circ, 360^\circ)) \quad \square$$

Positive Slant: PS

A positive slant (/) has an ideal orientation of 45o or 225o. Therefore a *fuzzy positive slant* is defined as a triangular membership function by the following equation:

$$\mu_{PS}(\varphi) = \text{MAX}(\Lambda(\varphi, 90^\circ, 45^\circ), \Lambda(\varphi, 90^\circ, 225^\circ)) \quad \square$$

Negative Slant: NS

A negative slant (\) has an ideal orientation of 135o or 315o. Therefore a *fuzzy negative slant* is defined as a triangular membership function:

$$\mu_{NS}(\varphi) = \text{MAX}(\Lambda(\varphi, 90^\circ, 135^\circ), \Lambda(\varphi, 90^\circ, 315^\circ)) \quad \square$$

4.3.2 Class of Fuzzy Curved Lines

Due to the wide range of possible shapes and forms, distinguishing curved lines is a more complex task. From the large number of possible curved shape features we selected the shapes which are frequent in handwriting patterns. We divided these curves into four categories namely Circles, S or Z types of curves, loops and open arcs. While the first two classes correspond each to a linguistic term describing the shape, loops and open circles are described by several primitive linguistic terms dependent on their crossing point or the position of their starting and ending point.

Most curves in handwritten characters are wholly or partially monotone convex. Depending on the missing part and the length of the arc we can define them as: vertical curves (E,.), horizontal curves (C,E), hockey and walking sticks. The distinction of these shape categories is accomplished by using the angle of rotation, the angle of slope joining the end points of the segment, measure of arced-ness, the relative length and the area covered by the segment. All of these measures are relative and normalized in the fuzzy domain of discourse [0,1], and can be combined with the fuzzy aggregation operators.

The output of this evaluation can classify a segment to several features and by this associate several possible meanings to its shape. The decision for one of these classes can be done at a later stage by adding contextual information or linking it to global or segment related features. In case of unavailability of contextual relation, maximum grade of membership is considered as the best choice. In the following we present the structure of the feature primitives and the corresponding fuzzy membership function.

Simple curve features

First we introduce some primitive features which through the fuzzy aggregation process form features of the defined curve categories. The category of the horizontal (C,E) and vertical curves (E,.) can be described by the convexity. In case of a vertical curve (E,.), a vertical line joins the end points of the curve and in case of a horizontal curve(C,E), a horizontal line joins the end points of this curve.

Vertical curve (VC) The line joining the end-points of vertical type of curves (e.g. E,.) has typically an inclination of 90o or 270o. Therefore a fuzzy vertical curve is defined as a triangular membership function $L(j;b,c)$: Based on the definition of the vertical and horizontal curve additional information about the direction of the convexity of the curve we can identify the shape to one of the defined categories.

Vertical Curve (VC)

The line joining the end-points of vertical type of curves (e.g. □□) has typically an inclination of 90o or 270o. Therefore a fuzzy vertical curve is defined as a triangular membership function $\square(\square;b,c)$:

$$\mu_{VC}(\varphi) = \text{MAX} \left(\Lambda(\varphi, 180^\circ, 90^\circ), \Lambda(\varphi, 180^\circ, 270^\circ) \right) \quad . \text{!!}$$

Horizontal Curve (HC)

The line joining the end-points of horizontal type of curves (e.g. □□) has typically an inclination of 0o,180o or 360o. Therefore a fuzzy horizontal curve is defined as a triangular membership function $\square(\square;b,c)$:

$$\mu_{HC}(\varphi) = \text{MAX} \left(\text{MAX} \left(\Lambda(\varphi, 180^\circ, 0^\circ), \Lambda(\varphi, 180^\circ, 180^\circ), \Lambda(\varphi, 180^\circ, 360^\circ) \right) \right) \quad . \text{!□}$$

Based on the definition of the vertical and horizontal curve additional information about the direction of the convexity of the curve we can identify the shape to one of the defined categories.

C-like Curve (CL)

To distinguish a C-like curve from a D-like curve, both vertical curves, we use the qualitative statement of the left or right convexity direction. In a curve which is of vertical type (VC), if the global minimum of horizontal projections x_{min} is relatively much lower than the weighted average of x projections, w_S, w_E , of its endpoints, x_S, x_E then it is very likely to be a C-like curve. is the binary function which possess the truth values over the whole segment regarding the point position. That means, if a segment point is on the left hand side of the median of the end point x -projections, then this binary function is equal to 1, else it is 0. The summation function is then normalized to the universe of fuzzy discourse [0,1].

$$\mu_{CL} = \min \left(1, \frac{\sum_{i=0}^n l_{x_i}}{n} \right)$$

where $l_{x_i} = \begin{cases} 1, & \text{if } (x_i < (x_S + x_E) / 2) \\ 0, & \text{else} \end{cases}$

. □

D-like Curve (DL)

Symmetric to the given definition for the C-like curve, we define the D-like curve. In a curve which is of type vertical (VC), if the global maximum of horizontal projections x_{max} is relatively much higher than a weighted average of x projections of its end-points x_S, x_E , then it is very likely to be a D-like curve. This statement is represented by the summation of a binary function , which possess the truth values over the whole segment regarding the point position. That means, if a segment point is on the right hand side of the median of the end point x -projections, then this binary function is equal to 1, else it is 0. The summation function is then normalized to the universe of fuzzy discourse [0,1].

$$\mu_{DL} = \min \left(1, \frac{\sum_{i=0}^n r_{x_i}}{n} \right)$$

where $r_{x_i} = \begin{cases} 1, & \text{if } (x_i > (x_S + x_E) / 2) \\ 0, & \text{else} \end{cases}$

. □

A-like curve (AL)

Same as for the two vertical curves C- and D-like, the two horizontal curves A- and U-like differ in the direction of their upward or downward convexity. While for C- and D-like the x projection was of great importance for A-like curves the y projection and their distribution is the main identification factor. In a curve which is of type horizontal (HC), if the global maximum of vertical projections y_{max} is relatively much higher than the weighted average (w_S, w_E) of y projections of its end points (y_S, y_E), then it is very likely to be an A-like curve. As for the above presented features, A-like can be identified with help of the binary function . The segment points have to be above the median of the end point y -projections, to be counted. The summation function is then normalized to the universe of fuzzy discourse [0,1].

$$\mu_{AL} = \min \left(1, \frac{\sum_{i=0}^n a_{y_i}}{n} \right)$$

where $a_{y_i} = \begin{cases} 1, & \text{if } (y_i > (y_S + y_E)/2) \\ 0, & \text{else} \end{cases}$. □

U-like Curve (UC)

In a curve which is of type horizontal (HC), if the global minimum of the vertical projections y_{min} is relatively much lower than the weighted average (w_S, w_E) of y projections of its end-points (y_S, y_E), then it is very likely to be a U-like curve. This is represented by the summation of a binary function . That means, if a segment point is below the median of the end point y-projections, then this binary function is equal to 1, else it is 0. The summation function is then normalized to the universe of fuzzy discourse [0,1].

$$\mu_{UL} = \min \left(1, \frac{\sum_{i=0}^n b_{y_i}}{n} \right)$$

where $b_{y_i} = \begin{cases} 1, & \text{if } (y_i < (y_S + y_E)/2) \\ 0, & \text{else} \end{cases}$. □

5. PROPOSED METHOD

The main idea of the proposed method consists five blocks as follows:

1. Number Segmentation
2. Feature extraction
3. Fuzzy feature combination
4. Training and recognition

Figure 2 shows the diagram of the proposed method which will be presented in more details in the next sections.

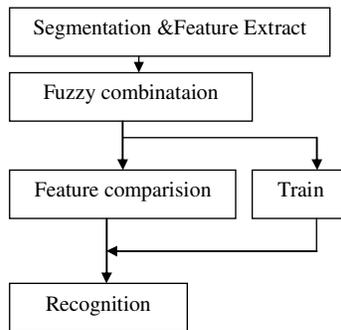


FIGURE 2: Diagram of the proposed method

5.1 Number Segmentation

In this stage the characters should be segmented and bounded by a box. We have applied the traditional bounding box methods and the results of this stage will send to the feature extraction block.

5.2 Feature Extraction

Feature extraction methods can be classified to structural and statistical features. In this paper we have extracted the structural or shape-based features. We have extracted several temporal and geometrical features and describe them with the Fuzzy method which will be described in more details.

5.3 Fuzzy Aggregation Feature Combination

The main objective of the fuzzy aggregation mechanism is to find an overall measure of certain fuzzy information from an uncertain and imprecise information data. The selection of meaningful features from a given set of handwriting data is dependent on the possible relationships among these feature categories. The meaningful feature designates the features with a good discrimination factor. To make a hierarchical structure a two phase aggregation scheme has been developed which is based on the union and weighted generalized mean aggregation connectives. The wide range of these connectives provides flexibility in finding the most suitable output feature set.

In this stage the local extracted features are aggregated to create a proper feature set. These feature sets are uniquely describe a symbol. A selected subset of Fuzzy aggregated and combined extracted features can form a unique feature for each symbol.

In the proposed method first we find the measure of arced-ness and straight-ness then these features are combined with the vertical line(VL), Horizontal line(HL), Positive Slant(PS) and Negative Slant(NS) features and the maximum value will be kept. Then the curve shaped features are combined and the maximum value is extracted. Finally the maximum of two combined values are considered as the shape feature and the maximum fuzzy value of that will be stored.

As the above statements, local extracted features will be combined in an efficient manner to achieve the best recognition rate. These features should uniquely represent each symbol. In this paper the memdanian product is applied to combine the features. This value is achieved by multiplication of straight line measure with each of line features. In the same manner for curves each feature is extracted from multiplication of arc measure with any extracted curve features as follows:

$$\begin{aligned} \mu_{AggLine1} &= \mu_{Straightness} * \mu_{VL} && \cdot \square \\ \mu_{AggLine2} &= \mu_{Straightness} * \mu_{HL} && \cdot \square \\ \mu_{AggLine3} &= \mu_{Straightness} * \mu_{PS} && \cdot \square \\ \mu_{AggLine4} &= \mu_{Straightness} * \mu_{NS} && \cdot \square \\ \mu_{AggArc1} &= \mu_{ArcHC} * \mu_{CL} && \cdot \square \\ \mu_{AggArc2} &= \mu_{ArcVC} * \mu_{UL} && \cdot \square \\ \mu_{AggArc3} &= \mu_{ArcVC} * \mu_{AL} && \cdot \square \\ \mu_{AggArc4} &= \mu_{ArcHC} * \mu_{DL} && \cdot \square \end{aligned}$$

In the next stage

$$\mu_{AggArc}(i) = \mu_{aggArc}(i) * \mu_{Arc}$$

□□

5.4 Training

In this stage for preparing the database, we have selected 40 people with different knowledge and ages. For this reason we have asked them to write numbers zero to nine several times. These people have written the farsi numbers "zero" to "nine" and the system automatically segmented and extracted the related features. If the sample existed then the number is recognized else maximum of curveness and linearity, the related fuzzy value, segment numbers and temporal features are used to train as a new value and is stored.

5.5 Recognition

In the recognition stage, first percent of the stored numbers in the database and the under recognition number are obtained for recognition, extracted feature values of the tested number are compared with the database value and the number is recognized. In this stage, each number is represented based on the structural features. If these features matched with one of the numbers in the database then the number is recognized. For maximizing the recognition rate in the farsi numbers, we have selected nine linguistic variables as follows:

{Zero, VeryVeryLow, VeryLow, Low, Medium, High, VeryHigh, VeryVeryHigh, Excellent}

The membership functions of these linguistic terms can be seen in Fig.3 . In this stage for recognizing the number "seven" and "eight", we have used the temporal features as well as the geometrical features. Table 1 shows a sample of fuzzy terms for recognition of numbers.

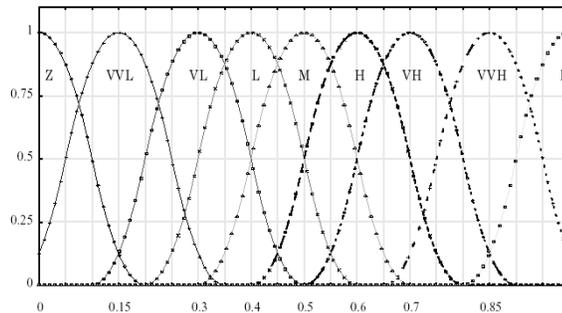


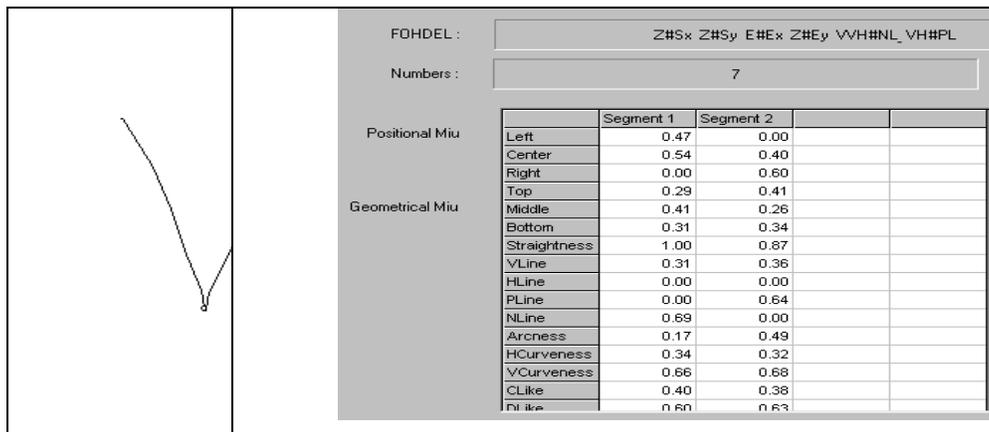
FIGURE 3: The membership functions of these linguistic terms

TABLE 1: A sample of fuzzy terms for recognition of numbers.

Number	Fuzzy term	Term description
Two	VVH#UL_T, VVH#VL_L	A u-type curve in the upper part, A "T" with the highest curvness, A vertical line at the left of "L" with the highest linearity
Three	VVH#UL,VH#UL_C, E#VL_L	A u-type curve in the upper part, A u-type curve in the center of "C" with the highest curvness, A vertical line at the left of "L" with the high linearity
Seven	E#Ex, Z#Ey, VVH#NL, VH#PL	A negative slope line(NL) with the high linearity, A positive slope line(PL) with the high linearity, EX and EY represent the start and end points in the xy coordination.

6. EXPERIMENTAL RESULTS

For testing the proposed method, we have prepared a database. We have asked 50 people to write Farsi numbers 10 times with restriction and 10 times without any restriction. The proposed method indicates around 95% recognition rate over the restricted database and 74% over the non-restricted sample set. Figure 4 shows a sample of the program execution.



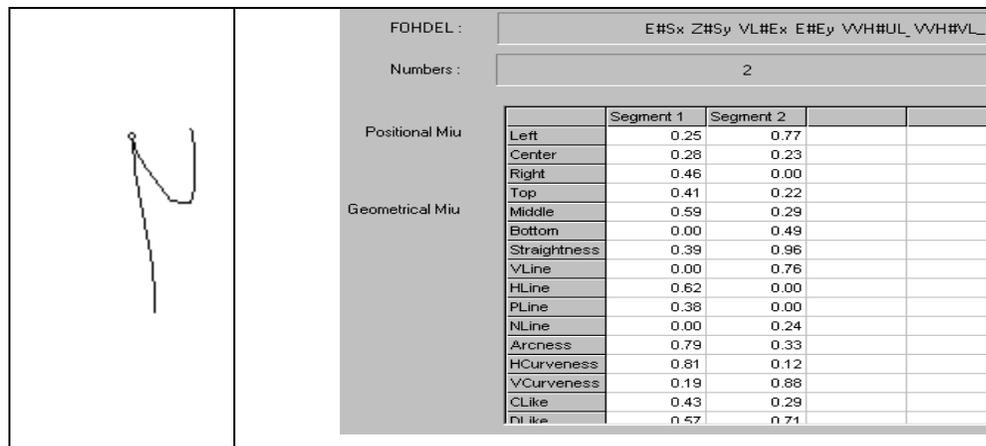


FIGURE 4: A sample of the program execution

7. CONCLUSION

In this paper we introduced fuzziness in the definition of the proposed pattern features for Farsi numbers, which provided the enhancement to the handwritten character information to be stored. The proposed method was applied on the related database for the Farsi handwritten numbers. The experimental results indicate 95% accuracy on recognition of Farsi numbers over the selected database.

8. REFERENCES

- [1] A.Malaviya, L. Peters "Fuzzy Feature Description of Handwriting Patterns" German National Research Center for Information Technology Schloß Birlinghoven, 53754, St. Augustin Pattern Recognition [Volume 30, Issue 10](#), 1997,
- [2] Koerich A.L., Leydier Y., "A Hybrid Large Vocabulary Word Recognition System using Neural Networks with Hidden Markov Models", IWFHR 2002,
- [3] Lee S., Kim Y., "A new type of recurrent neural network for handwritten characterrecognition", Third International Conference on Document Analysis and Recognition (Volume 1), 1995.
- [4] Malaviya A. and Peters L., "Extracting meaningful handwriting features with fuzzyaggregation method", 3rd International conference on document analysis and recognition,ICDAR`95, pp. 841-844, Montreal, Canada, 1995.
- [5] R.Ranawana, V.Palade, G.Bandana "An Efficient Fuzzy Method for Handwritten Character Recognition" LNAI 3214, pp. 698–707, 2004.
- [6] Cheung K., Yeung D., "Bidirectional Deformable Matching with Application toHandwritten Character Extraction", IEEE Transactions on Pattern Analysis and MachineIntelligence, Vol. 24, No. 8, August 2002
- [7] Kim G., Kim S., "Feature Selection Using Genetic Algorithms for Handwritten CharacterRecognition", Proceeding of the Sixth ACM SIGKDD International Conference onKnowledge Discovery and Data Mining, 2002
- [8] C.Liu, S.Jaeger,M.Nakagawa "Online Recognition of Chinese Characters:The State-of-the-Art" IEEE transaction onpattern analysis and machine intelligence,VOL. 26, NO. 2, 2004

INSTRUCTIONS TO CONTRIBUTORS

The *International Journal of Image Processing (IJIP)* aims to be an effective forum for interchange of high quality theoretical and applied research in the Image Processing domain from basic research to application development. It emphasizes on efficient and effective image technologies, and provides a central forum for a deeper understanding in the discipline by encouraging the quantitative comparison and performance evaluation of the emerging components of image processing.

We welcome scientists, researchers, engineers and vendors from different disciplines to exchange ideas, identify problems, investigate relevant issues, share common interests, explore new approaches, and initiate possible collaborative research and system development.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJIP.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 5, 2011, IJIP appears in more focused issues. Besides normal publications, IJIP intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

LIST OF TOPICS

The realm of International Journal of Image Processing (IJIP) extends, but not limited, to the following:

- Architecture of imaging and vision systems
- Character and handwritten text recognition
- Chemistry of photosensitive materials
- Coding and transmission
- Color imaging
- Data fusion from multiple sensor inputs
- Document image understanding
- Holography
- Image capturing, databases
- Image processing applications
- Image representation, sensing
- Implementation and architectures
- Materials for electro-photography
- New visual services over ATM/packet network
- Object modeling and knowledge acquisition
- Photographic emulsions
- Prepress and printing technologies
- Remote image sensing
- Autonomous vehicles
- Chemical and spectral sensitization
- Coating technologies
- Cognitive aspects of image understanding
- Communication of visual data
- Display and printing
- Generation and display
- Image analysis and interpretation
- Image generation, manipulation, permanence
- Image processing: coding analysis and recognition
- Imaging systems and image scanning
- Latent image
- Network architecture for real-time video transport
- Non-impact printing technologies
- Photoconductors
- Photopolymers
- Protocols for packet video
- Retrieval and multimedia

- Storage and transmission

- Video coding algorithms and technologies for ATM/p

CALL FOR PAPERS

Volume: 6 - Issue: 2 - April 2012

i. Paper Submission: January 31, 2012

ii. Author Notification: March 15, 2012

iii. Issue Publication: April 2012

CONTACT INFORMATION

Computer Science Journals Sdn Bhd

B-5-8 Plaza Mont Kiara, Mont Kiara

50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6207 1607

006 03 2782 6991

Fax: 006 03 6207 1697

Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2011
COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6207 1607
006 03 2782 6991

FAX: 006 03 6207 1697
EMAIL: cscpress@cscjournals.org