

Title λ-Connectedness and its Application to Image Segmentation, Recognition and Reconstruction

Name Li Chen

This is a digitised version of a dissertation submitted to the University of Bedfordshire.

It is available to view only.

This item is subject to copyright.

λ-CONNECTEDNESS AND ITS APPLICATION TO IMAGE SEGMENTATION, RECOGNITION, AND RECONSTRUCTION

by

LI CHEN

A thesis submitted for the degree of Doctor of Philosophy
of the University of Luton

The Research Centre
University of Luton
The Spires
2 Adelaide Street
Luton
LU1 5DU
UK

July 2001

Abstract

Seismic layer segmentation, oil-gas boundary surfaces recognition, and 3D volume data reconstruction are three important tasks in three-dimensional seismic image processing. Geophysical and geological parameters and properties have been known to exhibit progressive changes in a layer. However, there are also times when sudden changes can occur between two layers. λ-connectedness was proposed to describe such a phenomenon.

Based on graph theory, λ-connectedness describes the relationship among pixels in an image. It is proved that λ-connectedness is an equivalence relation. That is, it can be used to partition an image into different classes and hence can be used to perform image segmentation. Using the random graph theory and λ-connectivity of the image, the length of the path in a λ-connected set can be estimated. In addition to this, the normal λ-connected subsets preserve every path that is λ-connected in the subsets. An $O(nlogn)$ time algorithm is designed for the normal λ-connected segmentation. Techniques developed are used to find objects in 2D/3D seismic images.

Finding the interface between two layers or finding the boundary surfaces of an oil-gas reserve is often asked. This is equivalent to finding out whether a λ-connected set is an interface or surface. The problem that is raised is how to recognize a surface in digital spaces. λ-connectedness is a natural and intuitive way for describing digital surfaces and digital manifolds. Fast algorithms are designed to recognize whether an arbitrary set is a digital surface. Furthermore, the classification theorem of simple surface points is deduced: there are only six classes of simple surface points in 3D digital spaces. Our definition has been proved to be equivalent to Morgenthaler-Rosenfeld's definition of digital surfaces in direct adjacency.

Reconstruction of a surface and data volume is important to the seismic data processing. Given a set of guiding pixels, the problem of generating a λ-connected (subset of image) surface is an inverted problem of λ-connected segmentation. In order to simplify the fitting algorithm, gradual variation, an equivalent concept of λ-connectedness, is used to preserve the continuity of the fitted surface. The key theorem, the necessary and sufficient condition for the gradually varied interpolation, has been mathematically proven. A random gradually varied surface fitting is designed, and other theoretical aspects are investigated. The concepts are used to successfully reconstruct 3D seismic real data volumes.

This thesis proposes λ-connectedness and its applications as applied to seismic data processing. It is used for other problems such as ionogram scaling and object tracking. It has the potential to become a general technique in image processing and computer vision applications.

Concepts and knowledge from several areas in mathematics such as Set Theory, Fuzzy Set Theory, Graph Theory, Numerical Analysis, Topology, Discrete Geometry, Computational Complexity, and Algorithm Design and Analysis have been applied to the work of this thesis.

Acknowledgements

The author wishes to express his many thanks to the University of Luton for the exellent opportunity to study as a Ph.D. student by published work.

Many thanks are due to Dr. Osei Adjei of the Department of Computing and Information Systems, University of Luton, for his excellent supervision of this thesis.

My sincere thanks are also due to my previous professors, Donald H. Cooley and Tom Berkey, all of Utah State University; previous supervisor Ms. Yayen Lu and XiJuin Chui of Research Institute of Geophysical Prospecting for Petroleum in China; my college-classmate, colleague, and professor Jianping Zhang of Utah State University; my previous professor Chong-Zhang Li of Wuhan University; and all other professors and colleagues who co-authored papers related to this thesis with me.

Special thanks also goes to the Chinese Academy of Science, the Chinese 7-5 Key Projects, the Chinese State Department of Mineral and Resource, and Southeast University, Nanjing, China which provided the financial support, for various related projects.

Many thanks to my parents who always encouraged me to pursue my Ph.D. Very special thanks are due to my wife, Lan and my daughter Cera, for their tremendous and continuous support and understanding during the many years of my study and research.

LIST OF CONTENTS

Abstract I

Acknowledgements II

Notation V

List of Figures VII

CONTENTS

CHAPTER 1 Introduction 1

1.1 Background 1

1.2 Research Theme 2

1.3 Overview of the Thesis 4

CHAPTER 2 Background and Literature Review 7

2.1 Simple Graphs and Digital Spaces 7

2.2 Image Segmentation, Object Recognition, and Data Reconstruction 8

2.3 Fast Algorithm Design and Its Relationship to The Thesis 11

CHAPTER 3 Three Real Problems and Their Mathematical Models 14

3.1 Layer Determination and λ-Connectedness 14

3.2 Digital Surface Tracking 17

3.3 Seismic Data Reconstruction and Discrete Data Fitting 20

CHAPTER 4 Theoretical Analysis and Algorithm Design 21

4.1 λ-Connectedness and λ-Connected Segmentation 21

4.2 Digital Surfaces and Manifolds 25

4.3 Gradually Varied Fitting and Data Reconstruction 29

CHAPTER 5 Applications and Experimentation 32

5.1 Reconstruction of Velocity Volume 32

5.2 Ionogram Scaling 33

CHAPTER 6 Comparative Studies 35

6.1 The Contributions of the Thesis 35

6.2 Outline of the Published Work 38

CHAPTER 7 Conclusions and Future Work 43

APPENDIX A Comments on a λ-Connectedness Paper 45

APPENDIX B List of the Works on Which the Application Based 46

APPENDIX C Declaration 48

REFERENCES 49

ATTACHMENT Co-authored Paper Statements and Copies of the Works 53

NOTATION

$p \in V$	p is an element of V.
$U \subset V$	U is a subset of V.
$U \cap V$	The intersection of U and V.
$U \cup V$	The union of U and V.
R^n	n-dimensional real space.
$G = (V, E)$	A graph that has the vertice set V and edge set E.
$\Pi(p, q)$	A simple path from p to q.
Σ_m	The m-dimensional grid space, a special graph.
i_Σ_m	The indirectly adjacent m-dimensional grid space.
$G' \prec G$	G' is a subgraph of G.
$2D$	The two dimensional space or means "two-dimensional."
$3D$	The three dimensional space or means "three-dimensional."
nD-cell	The basic n-dimensional unit in Σ_m.
	An 1D-cell is a line segment. A 2D-cell is a square unit.
n-cell	The same as the nD-cell.
pixel	The picture element that contains the location
	and the value of the element.
ρ	The potential function associated with a Graph.
	For an image, ρ usually represents its intensity.
λ	A real number to indicate the level of λ-connectedness.
$\alpha_\rho(x, y)$	Neighbor-connectivity, i.e. the connectivity between
	two adjacent points x and y with respect to ρ.

$\mu(u,v)$	A measuring function used to define $\alpha_\rho(x,y)$.
$\beta_\rho(\Pi)$	The path-connectivity of path Π with respect to ρ.
$C_\rho(x,y)$	The λ-connectivity between x and y with respect to ρ.
$< G, \rho >$	A graph with its potential function ρ. It usually means a general image in this thesis.
$d(p,q)$	The distance between two points p and q.
$E(u,v)$	A general measure for λ-connectedness.
$CH(S,\Sigma_m)$	The connectivity of an image S in space Σ_m.
$N_{27}(p)$	The 3D neighborhood of a point p in Σ_3.
$S(p)$	p and its 3D neighbors in $N_{27}(p)$.
$A_1, A_2, ..., A_m$	m rational numbers for defining gradually varied functions.
$\|\|f_D\|\|$	The norm of the gradually varied function f_D.
$F_i^k(p)$	An iterativily defined set for gradually varied approximation.

LIST OF FIGURES

Figure 1 Examples of basic unit cells and their connections 9

Figure 2 Examples of waveform data and gray-scale seismic data 15

Figure 3 The six types of regular inner surface points 27

1 Introduction

This chapter presents the objectives of λ-connectedness and its relationship to image processing and discrete geometry. It also provides an overview of the work described in this thesis.

1.1 Background

Applied mathematics plays an important role in modern science and technology. Although applied mathematics focuses on the practical aspects of mathematics, there is no definite distinction between pure mathematics and applied mathematics. Almost every aspect of pure mathematics has applications to the real world, and almost each real problem has its mathematical model. Computer science deals with the nature of computing, algorithms for solving problems, and the structure of computing machines. Finding efficient algorithms for real world problems is at the center of computer science. Signal processing is a joint area of applied mathematics and engineering that deals with data sampling, communication, compression, restoration, and classification, among other aspects.

Digital image processing, a branch of signal processing, has become an important research area in recent years due to the need for improving pictorial information for human interpretation, and for the processing of scene data for autonomous machine perception (Gonzalez and Wood 1993). Statistical methods play a central role in the analysis of images. However, statistical methods are not able to describe single objects in an image. A method called "structural pattern analysis," based on formal language theory, was introduced. It describes the structure of patterns using the grammar of a formal language(Gonzalez and Wood 1993). This method was not very appropriate to analyse an object in three dimensional spaces because of the many different possible structures that could generate large search spaces. In the early 80's, a geometrical and topological method was proposed. This is known today as "the digital geometry and topology" (Rosenfeld 1979, Morgenthaler and Rosenfeld 1982, Herman 1998). In this method, an image is represented as a discrete function on a set of digital/discrete points in a

digital space. Hence, some characteristics of the image can be analyzed using their geometrical and topological properties.

Based on the work of Morgenthaler and Rosenfeld (1982), geophysical layer structures can be described by using digital topology (Chen 1985). That is, digital surfaces can be used to describe the interface between two seismic layers. The concept of the λ-connected search for layers/components of 3D digital spaces was introduced by Chen (1985), and during the last fifteen years has been working in this (and/or related) research area to attempt to build a practical method for processing geophysical images. λ-connectedness is a measure that cannot only determine a "Yes/No" connection between two points or events, but also can determine the degree of the connectedness between them. In other words, λ-connectedness can be viewed as a method similar to that of fuzzy systems (Klir and Yuan 1995). Technical details of the concept are discussed in the following subsections.

1.2 The Research Theme

Seismic prospecting involves placing a small charge of explosives underground to generate artificial earthquakes that are used to record and determine of the structure of the outer part of the earth's crust when searching for petroleum. The seismic waves are picked up by a series of geo-sensors spaced along the line of a traverse. As the travelling waves pass through different kinds of rocks underground, different characteristics are recorded by these geo-sensors.

There are two forms of seismic data. These are the waveform seismic data, such as amplitude profiles; and the gray-scale seismic data, such as velocity sections and porosity sections (Aminzadeh 1988). The goal of seismic data processing is to explore for oil-gas traps (North 1985). Geologists seek to find a location with a high oil-gas probability from which to drill. The decision is carefully made based on geological and geophysical data. The most important geophysical data are the seismic data. Oil-gas traps are of two major types: The first is the structural type formed by a geometric structure such as an anti-formal or a fault structure. To recognize these structures, amplitude waveform profiles (texture images) are usually used.

They are called seismic trace sections. The second is the lithologic type that is related to different rock types such as sandstone and granite. In this case, velocity sections (gray-scale images) are the most important factors. For example, identification of a region with typically low velocity indicates high porosity lithology, and the region is much more likely to store oil and/or gas. Seismic amplitude data, and velocity data, are two major resources for geologists. In the last twenty years, geologists and geophysicists have used pattern recognition and image processing techniques to improve and influence their final decisions (North 1985, Aminzadeh 1988).

By combining discrete geometry with image processing, further improvements in decision-making can be made. Discrete geometry is to study geometric and topological properties of discrete objects, and it includes packing, covering, tiling, convex polytops analysis, and computational geometry (Goodman *et al.* 1997). A sub-category of discrete geometry, namely digital geometry and topology, attempts to solve the theoretical problems of image processing, computer vision, and computer graphics in digital spaces (Herman 1998).

Three important tasks in three-dimensional seismic image processing are (1) seismic layer segmentation, (2) oil-gas boundary surfaces recognition, and (3) 3D volume data reconstruction. Although it is observed that geophysical and geological parameters and properties exhibit progressive changes in a layer on many occasions, there are also times when sudden changes can occur between two layers. In order to describe such a phenomenon, Chen (1985, 1991b) and Chen *et al.*(1994b) proposed the concept of λ-connectedness.

λ-connectedness was introduced to measure partial, incomplete, or fuzzy connection among pixels in a seismic image. A seismic layer search technique was used to determine a set of connected pixels that have similar seismic characteristics. To find an oil-gas trap boundary required the tracking of a closed surface containing oil-gas. A 3D volume data reconstruction method used sample points to build the data volume and preserves the layer interface information. This thesis focuses on how to solve these three problems using digital geometry techniques.

1.3 Overview of the Thesis

This thesis first develops the λ-connected segmentation that is important to solve a class of problems in image processing and computer vision (Chen 1985, Chen *et al.* 1994a). Secondly, the thesis gives a simple and intuitive definition for digital surfaces. It also proves the classification theorem for digital surface points in direct adjacency (Chen and Zhang 1993c, Chen *et al.* 1999a). Thirdly, the thesis proves the theorem of the necessary and sufficient condition for gradually varied interpolation leading to the design of an algorithm to fit discrete surfaces based only on guiding points without the use of fitting formulae such as B-splines or Bezier polynomials (Chen 1990, Chen 1992).

Chapter Two reviews some basic concepts of simple graphs and digital spaces. Based on vertices and edges of the graph, definitions are given in Section 2.1 for simple paths, subgraphs, and the special graph Σ_m, which is the m-dimensional digital space. Section 2.2 introduces the basic concepts of image segmentation such as region growing segmentation and split-and-merge segmentation. Some methods of data reconstruction such as interpolation and fitting techniques are also described in this section. Section 2.3 introduces the concept of computer algorithms. NP-hardness and fast algorithms are described briefly. The relationship between fast algorithm design and the three focused problems of this thesis are also discussed.

Chapter Three has three sections. Each section discusses the mathematical model of one of the three problems related to this thesis and proposes methods that can be used to solve them. Section 3.1 discusses how the seismic layer is determined and describes the reason for introducing the concept of λ-connectedness and also how it is applied. The task is to find some mathematical properties of λ-connected sets and to design algorithms to find them. Section 3.2 focuses on how to recognize and track a digital surface that has potential uses in the boundary description of oil-gas traps. This requires a simple and intuitive definition of digital surfaces and the design of a tracking algorithm for digital surfaces. Section 3.3 discusses how a discrete data fitting technique can be used for seismic data reconstruction. The main problem of reconstruction is fitting the interfaces of layers. A new surface fitting method called gradually

varied fitting is proposed.

In Chapter Four, major results for each one of the problems addressed in Chapter Three are presented. This chapter is the focus of this thesis. In Section 4.1, based on graph theory, λ-connectedness is defined to describe the relationship among pixels in an image. Chen (1991b) and Chen *et al.*(1994b) proved that λ-connectedness is an equivalence relation and that it can be used to partition an image into different classes. Hence, it can be used to perform image segmentation. Using the random graph theory, it is shown that the length of path in a λ-connected set can be estimated (Chen 1991b). In addition to this, the normal λ-connected subsets preserve every path that is λ-connected in the subsets (Chen 1991b, Chen *et al.* 1994b). An $O(nlogn)$ time algorithm is designed for normal λ-connected segmentation. Techniques developed are used to find objects in 2D/3D seismic images.

Section 4.2 covers the work on digital surfaces. In seismic data processing, one of the requirements is to find the interface between two layers, and to find the boundary surfaces of an oil-gas reserve. This is equivalent to finding out whether a λ-connected set is an interface or surface. The problem concerns how to recognize a surface in digital space (Morgenthaler and Rosenfeld 1982). In this section, an intuitive definition of digital surfaces is proposed, and a fast algorithm is designed to recognize whether an arbitrary set is a digital surface. Furthermore, the classification theorem of simple surface points was deduced: there are only six classes of simple surface points in 3D digital spaces (Chen and Zhang 1993c, Chen *et al.* 1999a). It was proven that the proposed definition is equivalent to Morgenthaler-Rosenfeld's definition of digital surfaces in direct adjacency (Morgenthaler and Rosenfeld 1981, Chen *et al.* 1999a). For indirect adjacency, the problem of defining digital surfaces is still open(Chen 1998a).

In Section 4.3, gradually varied surfaces and seismic data reconstruction is discussed. Given a set of guiding pixels, the problem of generating a λ-connected (subset of image) surface is an inverted problem of λ-connected segmentation. In order to simplify the fitting algorithm, gradual variation, an equivalent concept of λ-connectedness that preserves the continuity of

the fitted surface, was used (Chen, 1990, 1991a, 1992, 1994). The key theorem, i.e., the necessary and sufficient condition of the gradually varied interpolation is proven. A random gradually varied surface fitting was designed and other theoretical aspects of the concept were investigated (Chen, 1992, 1994). The concept was used to successfully reconstruct 3D seismic real data volumes (Chen *et al.* 1998).

Chapter Five presents two real applications and discusses experimental results with respect to the work decribed in this thesis. The first application is the reconstruction of velocity volume in seismic data processing, and the other is the ionogram scaling in space science (Chen *et al.* 1994a).

Chapter Six discusses the contribution of this thesis and compares it with the work of other researchers. In addition, explanations of the author's role on multi-authored papers in related references are stated.

Chapter Seven concludes the thesis and discusses some continuing work. Present work in progress relates λ-connectedness to areas of data mining and economic networks (Chen *et al.* 2000a).

2 Background and Literature Review

In this section, some basic concepts used in this thesis are reviewed. These concepts are simple graphs, digital spaces, image segmentation and reconstruction, object recognition, and fast algorithms related to these topics.

2.1 Simple Graphs and Digital Spaces

A graph G consists of two sets V and E, where V is a set of vertices and E is a set of pairs of vertices called edges. An edge is said to be incident with the vertices it joins (Cormen *et al.* 1993). In this thesis, assume that $G = (V, E)$ is an undirected graph which means that if $(a, b) \in E$ then $(b, a) \in E$, or $(a, b) = (b, a)$. Assume that $G = (V, E)$ is a simple graph, meaning every pair of vertices has only one edge which is incident to these two vertices, and there is no $(a, a) \in E$ for any $a \in V$.

If (p, q) is in E, p is said to be adjacent to q. Let $p_0, p_1, ..., p_{n-1}, p_n$ be $n + 1$ vertices in V. If (p_{i-1}, p_i) is in E for all $i = 1, ..., n$, then $\{p_0, p_1, ..., p_{n-1}, p_n\}$ is called a path. If $p_0, p_1, ..., p_{n-1}, p_n$ are distinct vertices, the path is called a simple path.

A simple path $\{p_0, p_1, ..., p_{n-1}, p_n\}$ is closed if (p_0, p_n) is an edge in E. A closed path is also called a cycle. Two vertices p and q are connected if there is a path $\{p_0, p_1, ..., p_{n-1}, p_n\}$ such that $p_0 = p$ and $p_n = q$. G is called connected if every pair of vertices in G is connected. In this thesis, it is always assumed that G is connected .

Let S be a set. If S' is a subset of S, their relationship is denoted by $S' \subseteq S$. If S is not a subset of S', then S' is called a proper-subset of S, denoted by $S' \subset S$. Suppose $G' = (V', E')$ is a graph where $V' \subset V$ and $E' \subset E$ for graph $G = (V, E)$, $G' = (V', E')$ is called a partial-graph of G. If E' consists of all edges in G whose joining vertices are in V', then the partial-graph $G' = (V', E')$ is called a subgraph of G and their relationship is denoted by $G' \preceq G$. If V' is a proper-subset of V, then the relationship is denoted by $G' \prec G$. It is

noted that for a certain subset V' of V, the subgraph G' with vertices V' is uniquely defined.

In this thesis, Σ_m represents a special graph $\Sigma_m = (V, E)$. V contains all integer grid points in the m dimensional Euclidean space. The edge set E of Σ_m is defined as $E = \{(a,b)|a,b \in V \& d(a,b) = 1\}$, where $d(a,b)$ is the distance between a and b. In fact, E contains all pairs of adjacent points. Because a is an m-dimensional vector, $(a,b) \in E$ means that there is only one component i such that $|x_i - y_i| = 1$ and the rest of the components in $a = (x_1, ..., x_m)$ and $b = (y_1, ..., y_m)$ are the same. It is known as direct adjacency. One can define indirect adjacency as $\max_i |x_i - y_i| = 1$. Σ_m is usually called an m-dimensional digital space. The basic discrete geometric element n-dimensional cells (nD-cells, or n-cells) can be defined in such a space, such as 0D-cells (point-cells), 1D-cells (line-cells), and 2D-cells (surface-cells). Some examples of nD-cells are shown in Figure 1.

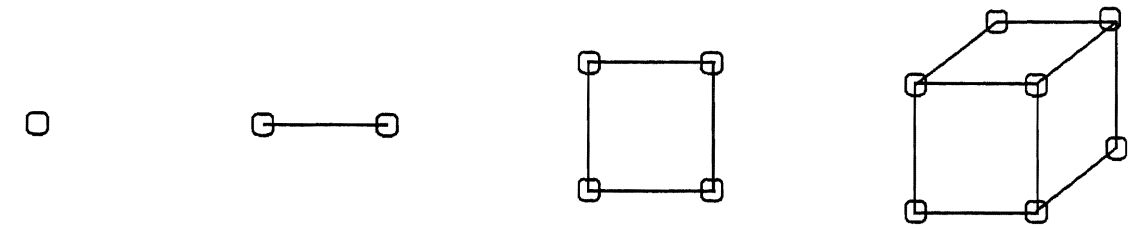
2.2 Image Segmentation, Object Recognition, and Data Reconstruction

Segmentation is a clustering method in image processing (Gonzalez and Wood 1993). There are many types of segmentation, such as: measurement space clustering, region growing, split-and-merge segmentation, and edge detection. Different types of data require different segmentation methods. For gray-scale images such as velocity profiles, one can use region growing or edge detection. For texture images such as waveform profiles or seismic section, rule-based segmentation methods may be preferred.

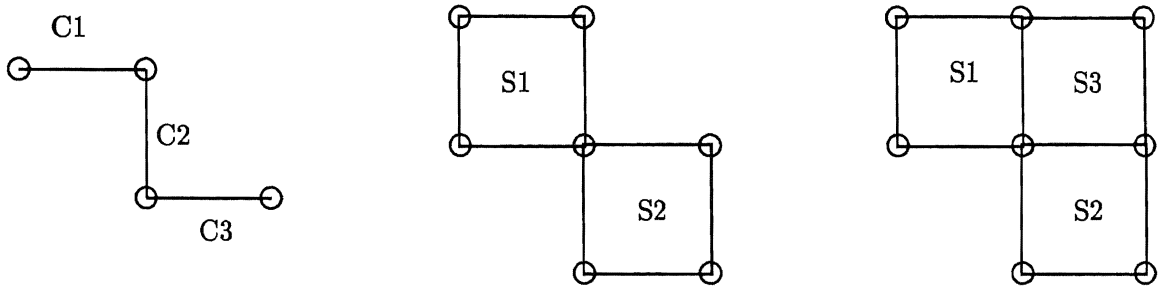
A digital image F is a mapping from a grid-space Σ_m to the real set R (or R^n in general). Let S be a subset of Σ_m. S is said to be uniform if F on S (a sub-image) has properties of uniformity. For instance, a popular uniformity measure is defined as the maximum difference between any pixel value and the mean value of S.

Formally, segmentation is the partitioning of an image into connected subsets, each of which is uniform, but such that no union of adjacent subsets is uniform. The definition of segmentation with respect to a digital image F can be described as follows: If there exists a

non-empty segmentation $F_1, F_2, ..., F_m$, satisfying



(a) 0D-cell (b) 1D-cell (c) 2D-cell (d) 3D-cell



(e) point-connected 1D-cells (f) point-connected 2D-cells (g) line-connected 2D-cells

Figure 1: Examples of basic unit cells and their connections : (a) 0D-cells, (b) 1D-cells, (c) 2D-cells, (d) 3D-cells, (e) point-connected 1D-cells, (f) point-connected 2D-cells, and (f) line-connected 2D-cells.

(1) $F_i \cap F_j = \emptyset$, if $i = 1, .., m, j = 1, ..m, i \neq j$,

(2) $\cup_{i=1,...,m} F_i = F$,

(3) Each F_i is connected,

(4) Each F_i is "uniform",

(5) If F_i and F_j are adjacent, then $F_i \cup F_j$ is not uniform,

then, $\{F_1, F_2, ..., F_m\}$ is called a segmentation of F.

Region growing segmentation: In a gray scale image, the intensity is the uniformity measure. A region (or a segment) in an image may be viewed as a connected group of pixels, all with similar brightness. This method begins with a single pixel, and then by examining neighbours tries to find a maximum sized connected region of similar pixels. Thus, regions can grow from single pixels.

Split-and-Merge Segmentation: A split-and-merge algorithm has some similarity to the region growing technique. In two dimensional image processing, split-and-merge segmentation is based on a quadtree partition of an image and hence it is sometimes called quadtree segmentation. One starts with tree nodes (representing square regions of the image) at some intermediate level of the quadtree. If a square is found to be non-uniform, then it is replaced by its four sub-squares (split). Conversely, if four squares form a uniform square, they are replaced by a single square(merge). This process continues recursively until no further splits or merges are possible.

Object recognition is essential to image processing and computer vision today. Object recognition has many uses in modern image processing. It can be used to find a specific target in an image or to recognize the characteristics of a set of pixels. In sequential image processing, object recognition may be used for tracking a target in different images. Recognizing different objects require different algorithms.

With respect to the work described in this thesis, object recognition refers to the task of recognizing digital curves, surfaces, and solid objects. Given a set of points in Σ_m, decide if the set forms a digital surface. Therefore, the definition of digital surfaces becomes an important issue. An intuitive definition for digital curves and surfaces is presented in Section 4.2.

Reconstruction usually uses inversion and fitting techniques. However in this thesis, only fitting methods are considered. Fitting refers to finding more data points that are not included in the sampling data. Interpolation and approximation are two typical mathematical

techniques used in data fitting. Interpolation requires that fitted points retain the same values at the sample points but approximation does not.

Basically, fitting algorithms are based on mathematical formulas and equations such as B-spline and Bezier polynomials. In some situations, if there are outlier points in the sampled data, such methods may generate faulty results. An advanced technique called robust fitting (or regression) can be used. On the other hand, if the sample data comes from different classes (or categories), fitting may damage the original information. One must separate them into different classes before the fitting. In geophysics data processing, if one must reconstruct a set of data between two points which have the same depth while a fault exists between these two points, fitting will erase the fault. For this reason a more intelligent data fitting technique is needed. An intelligent technique is discussed in Section 5.1.

2.3 Fast Algorithm Design and Its Relationship to The Thesis

An algorithm is a problem solver and consists of a set of finite steps, each of which solves a simple task. In this thesis, algorithms are designed to perform λ-connected segmentation and to recognize objects such as curves and surfaces in two or three dimensional images.

According to the theory of computing, a problem is said to be solvable (feasible) if there exists an algorithm that allows a solution(s) to be found in finite steps in a limited time. However, some problems such as the "Halting Problem" do not always converge to a solution(s). These problems are unsolvable. For some problems, such as the "Traveling Salesman Problem," that finds the shortest path for a salesman who travels all cities in a map, there exists an algorithm that converges to a solution in a limited time. However, the time cost of the algorithm is an exponential growth when the number of cities (i.e. the size of the problem) is increased. For example, if the number of cities is 100, an $O(2^n)$ time algorithm may cost 100 years to obtain a solution. Thus, a theoretically feasible problem may not be practically tractable, meaning that there is a polynomial time algorithm for the problem.

There are two major classes of feasible problems: the polynomial time class and the NP-hard class (Cormen *et al.* 1993). For a polynomial time class problem, a polynomial (time) algorithm can be found to solve the problem using "modern" digital computers. The mathematical model of "modern" digital computers is the deterministic Turing machine. A more powerful model of computers is the non-deterministic Turing machine. If a problem can be solved in a non-deterministic Turing machine in polynomial time, we call the problem an NP-problem. In the NP-problem class, there is a subclass called NP-Complete class if any of which can be solved in polynomial time then all NP-problems are in polynomial time class (Cook's theorem). This is the famous unsolved problem called $P = ?NP$ problem. The traveling salesman problem is an example of one of the NP-Complete problems. An NP-hard problem is defined as being, at least, hard as an NP-complete problem.

A fast algorithm is usually considered to solve a problem fast or in real time. However, there is no mathematical definition for what complexity of the algorithm can be called a fast algorithm. Basically, the complexity of the fast algorithm is varied dependent on people's feeling of a specific problem. For example, as we know that the sorting problem has the $O(nlogn)$ algorithm, then any algorithm whose complexity in the average case is beyond $O(nlogn)$ may not be considered a fast algorithm. On the other hand, an $O(n^3)$ algorithm for linear programming can be called a fast algorithm because the problem has no better algorithm.

The design of algorithms in this thesis is related to all three problems mentioned earlier. An $O(N)$ time algorithm for λ-connected segmentation, where N is the number of points in an image, is presented in Section 4.1. Chen (1997) designed a $O(N)$ time algorithm to extract the boundary of a 2D object using the topological properties of the edges of surfaces and solids with respect to direct adjacency. It is interesting to note that, for indirect adjacency, a general definition for digital surfaces have not been found. Therefore, there is no foundation to deal basically with its search algorithms. The problem has been discussed by Chen (1998a). Although, a definition was given in that paper, it is thought that the recognition problem may be NP-hard. For gradually varied surface fitting in seismic data reconstruction; even through, an $O(n^2)$ algorithm can be designed in a simple way. However, if the domain is separable (

i.e., a Jordan space), some faster algorithms comparable to the $O(n^2)$ time algorithm can be designed (Chen 1991a).

3 Three Real Problems and Their Mathematical Models

Seismic data sets consist of synchronous records of reflected seismic signals registered by a large number of seismic sensors placed along a straight line or in the nodes of a rectangular lattice on the earth's surface (Costain and Coruh 1997). A series of explosions serve as the source of the initial seismic pulse, responses to which are averaged. The vertical time axis forming the resulting two- or three-dimensional picture is identified with depth, the reflected signal under the respective sensor carries information on the local properties of the rock mass at the respective point of the underground medium. In contrast to the above-lying sedimentary cover, the absence of pronounced reflecting surfaces in a crystalline body results in great difficulty inferring the geological information from the basement interval of the seismic picture.

This thesis deals with three of the most important tasks in seismic data processing. These are seismic layer segmentation, oil-gas boundary surfaces recognition, and 3D volume data reconstruction. This chapter describes their mathematical models. The theoretical analysis is presented in Chapter Four.

3.1 Layer Determination and λ-Connectedness

In seismic data processing, researchers usually deal with two types of layers: the geological layer and the seismic layer. The geological layer is formed by sedimentary rocks from one time period to another. The geological layer is usually thin. On the other hand, a seismic layer is distinguished by geophysical properties of sedimentary rocks represented in the form of seismic data in the stratum, velocity and porosity. Since the geological layer can not be identified just using seismic data, the work described in this thesis considers only seismic layer search.

Chen (1985, 1991b) and Chen *et al.* (1994b) proposed λ-connectedness to describe the phenomenon that geophysical and geological parameters and properties exhibit gradual or progressive changes in a layer, but sudden changes frequently occur between two layers. In other words, a λ-connected set can be viewed as a layer, and two layers should be separated

by λ-connectedness.

Seismic data has two forms: waveform seismic data, and gray-scale seismic data. Both waveform data and gray-scale seismic data can be viewed as digital images (See Figure 2). As discussed in Chapter Two, a digital image can be represented as a function from a 2D/3D array to an interval in R, the real number set. Such an array can be viewed as a graph in mathematics if we define: (1) each element in the array as a vertex of the graph, (2) the edge set as consisting of all pairs of elements a, b, denoted by (a, b), where a and b are adjacent in the array.

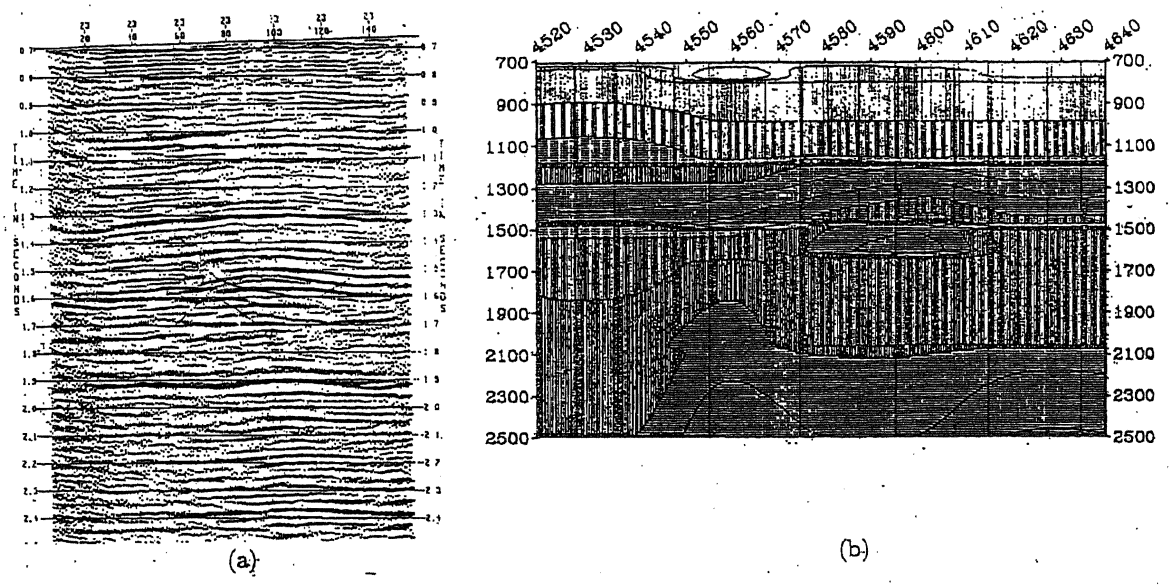


Figure 2: Examples of waveform data and gray-scale seismic data: (a) A waveform seismic section, and (b) a velocity section.

To determine a layer in the stratum is to find a subset of vertices which have similar properties (or values) in the gray scale image. λ-connectedness is defined to describe the relationship among pixels in the image.

λ-connectedness can be defined on an undirected graph $G = (V, E)$ with an associated (potential) function $f : V \to R^m$, where R^m is the m-dimensional real space. Given a measure

$\alpha_\rho(x,y)$ on each pair of adjacent points x, y based on the values $\rho(x), \rho(y)$, we define

$$\alpha_\rho(x,y) = \begin{cases} \mu(\rho(x),\rho(y)) & \text{if } x \text{ and } y \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\mu : R^m \times R^m \to [0,1]$ with $\mu(u,v) = \mu(v,u)$ and $\mu(u,u) = 1$. Note that, for all u, $\mu(u,u) = c$ and $\mu(u,v) \leq c$ where $c \in [0,1]$. α_ρ is used to measure "neighbor-connectivity." Based on α_ρ, λ-connectedness can be defined on $< G, \rho >$ in a general way.

Let $x_1, x_2, ..., x_{n-1}, x_n$ be a simple path. The path-connectivity β of a path $\pi = \pi(x_1, x_n) = \{x_1, x_2, ..., x_n\}$ is defined as

$$\beta_\rho(\pi(x_1,x_n)) = \min\{\alpha_\rho(x_i, x_{i+1}) | i = 1, ..., n-1\} \tag{2}$$

or

$$\beta_\rho(\pi(x_1,x_n)) = \prod\{\alpha_\rho(x_i, x_{i+1}) | i = 1, ..., n-1\} \tag{3}$$

Finally, the degree of connectedness (connectivity) of two vertices x, y with respect to ρ is defined as:

$$C_\rho(x,y) = \max\{\beta(\pi(x,y)) | \pi \text{ is a (simple) path.}\} \tag{4}$$

For a given $\lambda \in [0,1]$, point $p = (x, \rho(x))$ and $q = (y, \rho(y))$ are called λ-connected if $C_\rho(x,y) \geq \lambda$. (One usually avoids saying that x and y are λ-connected since they do not contain the information of ρ.) In image processing, $\rho(x)$ is the intensity of a point x and $p = (x, \rho(x))$ defines a pixel. In this case, one may say two pixels are λ-connected.

If equation (2) applies, λ-connectedness is reflexive , symmetric, and transitive. Thus, it is an equivalence relation. If equation (3) is used, λ-connectedness is reflexive and symmetric. Therefore, it is a similarity relation (Chen 1985, Chen 1991b, Chen *et al.* 1994b).

If $< G, \rho >$ is an image, then this equivalence relation can be used for segmentation, i.e., to partition the image into different objects. Therefore, to find a layer is to find a λ-connected subset of the vertices. A fast algorithm to find such a subset will be discussed in Chapter Four. In order to explain the key concept of this thesis, λ-connectedness, the author presents

an example as follows: Assume that there is a 4×4 small image, all pixels $p_{i,j}$, $i, j = 1, 2, 3, 4$, are given in the following array. For this case, Graph $G = \Sigma_2$ has 16 vertices v_{ij}. $v_{1,1}$ and $v_{2,1}$ are adjacent in direct adjacency as well as $v_{1,1}$ and $v_{1,2}$.

$$\begin{bmatrix} 2.0 & 1.0 & 10. & 8.0 \\ 1.0 & 8.0 & 9.0 & 5.0 \\ 7.0 & 8.0 & 4.0 & 6.0 \\ 7.0 & 4.0 & 6.0 & 8.0 \end{bmatrix} \tag{5}$$

One can see, 10 is the maximum value of all pixels in the image. Let us choose $\mu(p, p') = 1 - |p - p'|/10$ and $\lambda = 0.8$. If the difference of the values of two adjacent pixels is less or equals to 2, then these two pixels are λ-connected. One can easily see that there are three λ-connected components in the image. The second component is shown below:

$$\begin{bmatrix} & & 10. & 8.0 \\ & 8.0 & 9.0 & \\ 7.0 & 8.0 & & \\ 7.0 & & & \end{bmatrix}. \tag{6}$$

The other two components are located at the up-left corner and the low-right corner of the original matrix (5), respectively.

3.2 Digital Surface Tracking

Digital surface tracking is a very general and practical task in image processing. Many applications can be found in medical imaging, and 3D computer vision. In geophysics, finding the interface between two layers or the boundary surfaces of an oil-gas reserve is often asked. This is equivalent to finding out whether a λ-connected set is an interface or surface. For example, when a λ-connected set is found, one usually wants to ask if the set is a curve, surface, or a 3D object (3D manifold). In geology and geophysics, a curve could be a fault line, a surface could

be an interface, and a 3D object could be an oil-gas trap. The problem that is often raised concerns how to recognize a surface in digital space (Morgenthaler and Rosenfeld 1982).

In continuous spaces, a surface can be defined as a Hausdorff space where each point has an open neighborhood that is homeomorphic to an open Euclidean 2D-space. However, in a discrete space, there exist three fundamental problems. The first relates to what defines a discrete surface. The second problem concerns the properties of continuous surfaces that are required to be preserved in discrete spaces. The third and the most important problem relates to how the properties of discrete surfaces are applied to image processing and computer vision.

In the early 80s, two kinds of digital surfaces were studied by Artzy *et al.* (1981), and Morgenthaler and Rosenfeld (1982), respectively. The motivation of Artzy *et al.* (1981) was to find the boundary surface of a solid object in a 3D image and to develop a fast algorithm to track the surface. In general, the boundary surface of a solid object is a surface (or union of several surfaces), but a surface may not be the boundary of a 3D object. Morgenthaler and Rosenfeld defined a surface mathematically in Σ_3, a space containing all integer grid points. Based on their definition, a surface recognition algorithm was developed in a formal way. However, their definition of digital surfaces is not only difficult to understand, but also difficult to expand to higher dimensional digital spaces.

Digital objects such as curves and surfaces are constructed by basic elements. Examples of these elements, shown in Figure 1, are 0D-cells (point-cells), 1D-cells (line-cells), 2D-cells (surface-cells), and 3D-cells. In order to give a mathematical definition of digital surfaces, Chen and Zhang (1992, 1993a, 1993b) investigated the relationship among these types of cells. The author presents some of the observations as follows:

A point $p = (x, y, z)$ in Σ_3 is a 0D-cell. A pair of points $p = (x_1, x_2, x_3)$ and $q = (y_1, y_2, y_3)$ in Σ_3 forms a line-cell if p and q are (directly) adjacent points, i.e.,

$$d(p, q) = \sum_{i=1}^{3} |x_i - y_i| = 1. \tag{7}$$

Where $d(p,q)$ is the distance between p and q. Therefore, a line-cell can be denoted as a set of p and q, $\{p\} \cup \{q\}$ or $\{p,q\}$.

A surface-cell is a set of four points that form a unit square parallel to coordinate planes. So a surface-cell is the union of two line-cells $\{p,p'\}$ and $\{q,q'\}$. The set of $\{p,p',q,q'\}$ are four distinct points.

A 3-cell is a unit cube, which includes eight points. A 3-cell is the union of two surface-cells. In Figure 1 (d), surface-cells $\{p_1,p_2,p_3,p_4\}$ and $\{q_1,q_2,q_3,q_4\}$ forms a 3D-cell. It is required that $\{p_1,p_2,p_3,p_4\} \cap \{q_1,q_2,q_3,q_4\} = \emptyset$ and for all $i = 1,2,3,4$, $d(p_i,q_i) = 1$.

An important concept of parallel-moves is defined below:

Definition 1 A line-cell $\{q,q'\}$ is called a parallel-move of line-cell $\{p,p'\}$ if $\{p,p'\} \cap \{q,q'\} = \emptyset$, and $d(p,q) = 1$ and $d(p',q') = 1$.

In the same way, one can define parallel-moves for point-cells, surface-cells, etc. In Σ_3, each line-cell has four parallel-moves, and a line-cell and each of its four parallel-moves form a surface-cell. In fact, a point-cell and one of its parallel-moves form a line-cell, and a surface-cell and one of its parallel-moves form a 3D-cell. This technique philosophically matches the idea of "curves made by dots and surfaces made by curves" (Chen, 1997). Furthermore, without making use of continuous space, a recursive definition for k-dimensional unit cells using parallel-moves in direct adjacency was given by Chen and Zhang (1993a, 1993b) and Chen (1998b).

Based on line-cells, surface-cells, and 3D-cells, an intuitive definition of surfaces was proposed. An surface tracking algorithm was designed by parallel-moving a line-cell (Chen and Zhang 1993b). Chen (1997) discussed more general cases, which are only based on simple graphs. The relationship between the raster space and the point space can be found in Chen (1998b). The definition and algorithm of digital surfaces and digital manifolds are discussed in Chapter Four.

3.3 Seismic Data Reconstruction and Discrete Data Fitting

Since the cost of field data collection is prohibitive, seismic prospecting often uses a data fitting method to reconstruct unknown data points. In seismic prospecting, a vertical line of recorded signals is called a common data point (CDP) trace. Sometimes, a CDP trace is inserted to fill a bad trace in a 2D waveform profile. In other situations, data fitting is needed in building a 3D data volume from 2D data sections.

In 3D data reconstruction, surface fitting is an important task. Unlike automobile frame shape construction, seismic data sometimes does not require a higher degree of smoothness. In some cases, a reasonable "continuity" is enough to describe the interface of two seismic layers. On the other hand, the standard surface fitting algorithm assumes that the resulting data satisfies a mathematical formula such as a polynomial (e.g., Bezier Polynomial and B-Spline). One can see that this is not always the case. In addition, the goal here is to get a set of discrete points to fill the data array. After fitting, one needs to calculate the actual data point based on the mathematical formula with the known coefficients. That is to say, it is necessary to determine the value of the unknown discrete points.

The standard surface fitting algorithm computes from "discrete" to "continuous" and then back to "discrete." It is very natural to ask if there is a discrete fitting algorithm to complete the task without employing a continuous mathematical formula. Formally, given a set of guiding pixels, the problem is to generate a λ-connected (subset of image) surface.

This is an inverted problem of λ-connected segmentation. In order to simplify the fitting algorithm, gradual variation, an equivalent concept of λ-connectedness that preserves the continuity of the fitted surface is used (Chen, 1990, 1991a, 1992, 1994). The key theorem of gradual variation (i.e., the necessary and sufficient condition of the gradually varied interpolation) is proven (Chen 1990). The design of the random gradually varied surface fitting and other theoretical aspects were investigated by Chen (1992, 1994). These concepts are used to successfully reconstruct 3D seismic real data volumes in Chen, Cooley, and Zhang(1998). A detail description of the volume reconstruction is discussed in Chapter Five.

4 Theoretical Analysis and Algorithm Design

This chapter is devoted to the theoretical analysis and the design of algorithms with respect to the work discussed in Chapter Three. First, the basic theorem and the algorithm for λ-connected segmentation are presented. Then, the definition of digital surfaces is introduced, and the surface tracking algorithm is discussed. The author also presents the classification theorem of simple surface points. Lastly, various results of gradually varied fitting and data reconstruction are presented.

4.1 λ-Connectedness and λ-Connected Segmentation

As discussed in Chapter Three, a digital image F is a mapping from a grid-space Σ_m to the real set R. Two points, p and q, in Σ_m are adjacent if the distance between these two points is 1. Two points, $p, q \in S$ and $S \subset \Sigma$, are connected if there is a path that consists of adjacent points that link p and q. S is connected if each pair of points in S are connected.

Theorem 1 (Chen 1985, Chen 1991b, Chen *et al.* 1994b) λ-connectedness is an equivalence relation if equation (2) in Chapter 3 is used (i.e., it can be used for classification). λ-connectedness is a similarity relation if equation (3) is used.

A λ-connected segmentation partitions P into one or more λ-connected components using the λ-connected equivalence relation if equation (2) applies (Chen 1985, Chen 1991b, Chen *et al.* 1994b). Generally, for any function $E : S \times S \rightarrow R$ in the set S possessing the following properties (Chen *et al.* 1994a):

(i) $E(u,u) \leq c$, c is a constant,

(ii) $E(u,v) = E(v,u)$,

(iii) $E(u,v) \geq \min\{E(u,w), E(w,v)\}$,

the λ-connectedness of (S,E) is an equivalence relation. λ-connectedness is a general dynamic

connectedness relation. λ-connectedness deals with the degree of connectivity between two points in an image, so it can be viewed as a kind of fuzzy connectedness. There are many functions satisfying the three conditions (i), (ii), and (iii) above, such as :

$$E(u,v) = \begin{cases} 1 - \frac{\|u-v\|}{\|u\|+\|v\|} & \text{if} \|u\| + \|v\| \neq 0 \\ 1 & \text{otherwise .} \end{cases} \tag{8}$$

Since a region growing technique can be manipulated by Tarjan's breadth-first search algorithm (Cormen *et al.* 1993), we can develop a fast algorithm for λ-connected segmentation. Such an algorithm only requires a linear time complexity. If X is a 2D/3D array and Z is the real set, we can see that λ-connected segmentation can be easily used to segment gray-scale images, such as velocity sections in seismic data or to find an object by using the intensity in an image. Other useful results include the following:

Corollary 1 (Chen 1991b, Chen *et al.* 1994b) If $\lambda_1 > \lambda_2$, then λ_1-connected segmentation is a refinement of λ_2-connected segmentation.

Let S be a subset of Σ_m. For an arbitrary path Π in S, we have $C_p(\Pi) \geq \lambda$. S is said to be λ-normal-connected.

Lemma 1 (Chen *et al.* 1994b) $S \subset \Sigma_m$ is a λ-normal-connected set with respect to f, if and only if any two adjacent points p and p' have $C_a(p,p') \geq \lambda$.

Lemma 2 (Chen *et al.* 1994b) λ-normal-connectedness is not an equivalent relation.

Even though the λ-normal-connectedness is not an equivalence relation, the split-and-merge technique in pattern recognition can be used to obtain the normal λ-connected subsets.

Let $f : \Sigma_m \to [0,1]^n$ be a fuzzy sub-fiber and $E(u,v)$ and $C_a(x,y)$ be given. For an arbitrary $\lambda \in [0,1]$, the following algorithm can be used to find out all λ-connected parts in Σ_N.

Algorithm A (Chen 1991b) Breadth-First-Search technique (See Cormen *et al.* 1993) for λ-connectedness.

Step 1: Let p_0 be a point in a set Σ. Set

$L(p_0) \leftarrow *$ and $STACK \leftarrow STACK \cup p_0$

i.e., labeling p_0 and p_0 is sent a stack $STACK$.

Step 2: If STACK is empty, go to Step 4; otherwise,

$p_0 \leftarrow STACK$ (top of $STACK$). Then,

$L(p_0) \leftarrow 0$.

Step 3: For each p linking p_0 , if

$L(p) \neq 0, L(p) \neq *$, and $C(p, p_0) \geq \lambda$, then

$STACK \leftarrow STACK \cup p$ and $L(p) \leftarrow *$. Then, go to Step 2.

Step 4: Stop. $S = \{p : L(p) = 0\}$ is one λ-connect ed part.

Lemma 3 Algorithm A is time optimal, and its time complexity is $O(|n \cdot \Sigma_m|)$.

Algorithm B(Chen *et al.* 1994b) Split-and-Merge technique (see Gonzales and Wood, 1993) for λ-normal-connectedness. Here one is concerned with $(2, n)$-fuzzy sub-fibers.

Step 1: Let Σ_2 be a $(2^l \times 2^l)$ array. $L \leftarrow \Sigma_2$, where L be a set.

Step 2: If region L is already homogeneous, i.e., L is λ-normal-connected set,

$NOMERGE \leftarrow NOMERGE \cup L$ and go to step 3; otherwise,

divide L into four subsquares, L_{11},L_{12},L_{21}, and L_{22};

then $NOSPLIT \leftarrow NOSPLIT \cup \{L_{11}, L_{12}, L_{21}, L_{22}\}$.

Step 3: If there exists four regions that are four sons of some region Ls in $NOMERGE$, namely L_{11},L_{12},L_{21}, and L_{22}, they will be conquered by a merging approach. It tests any two λ-normal-connected subsets that are neighbors , and belong to different regions in L_{11},L_{12},L_{21}, and L_{22}. Merge these subsets if their union is a λ-normal-connected subset of L. $NOMERGE \leftarrow NOMERGE - \{L_{11}, L_{12}, L_{21}, L_{22}\}$. Otherwise, if there are no such four regions; then stop when $NOSPLIT$ is empty, choose a region from $NOSPLIT$, send it to L, and go to Step 2.

Theorem 2 There is an $O(n|\Sigma_2| + |\Sigma_2|log|\Sigma_2|)$ time split-and-merge algorithm for normal λ-connected segmentation on $(2, n)$-fuzzy sub-fiber.

Corollary 2(Chen *et al.* 1994b) There is an $O(nm|\Sigma_m| + |\Sigma_m|log|\Sigma_m|)$ (or $O(n2^m|\Sigma_m| + |\Sigma_m|log|\Sigma_m|)$ for indirectly adjacent) time split-and-merge algorithm for normal λ-connected segmentation on (m, n)-fuzzy sub-fiber.

In a region-growing image segmentation, the length of a connected (or λ-connected) path is an important factor for the segmentation result. In order to estimate the performance of λ-connected segmentation, the connectivity of an image S on Σ_m is introduced as follows:

$$CH(S, \Sigma_m) = \frac{1}{2} \sum_{d(q,r)=1} C(q, r). \tag{9}$$

If image S is viewed as a random graph with edge selection probability p (Bollobas 1985), it is proved,

Theorem 3(Chen 1991b) If $CH(S, \Sigma_m) \leq \frac{1}{2}\Sigma_m$, then

(1) if $k > (m+1)\frac{ln(n)}{-ln(1+\frac{n-2}{n}mp)}$, then almost all images on Σ_m do not have a k-length path,

(2) if $k \leq (m-1)\frac{ln(n)}{-ln(mp)}$, then almost all images on Σ_m do have a k-length path,

where n is the length of each side of m dimensional array Σ_m, and p is the selection probability of each edge in the graph $S = (\Sigma_m, E_p)$, edge E contains only the adjacent points pairs.

4.2 Digital Surfaces and Manifolds

This section discusses the formal definition of digital surfaces and its generalization to digital manifolds. A surface tracking algorithm is also briefly designed in this section.

Section 3.2 introduced the concept of parallel-moves. This section uses the concept to define the digital surface. Two line-cells (or two surface-cells) are point-adjacent if they share a point. Two surface-cells are line-adjacent if they share a line-cell, and so forth.

Two line-cells are point-connected if there is a path of line-cells such that these two cells are two end elements of the path that is point-adjacent. Similarly, one can define the line-connectedness of two surface-cells. Figure 1 (e), (f), and (g) show three examples for point-connected 1D-cells, point-connected 2D-cells, and line-connected 2D-cells, respectively. A new definition of digital surfaces is introduced below:

Definition 2 (Chen and Zhang, 1993a, 1993b) A (point-)connected set $S \subset \Sigma_3$ is a parallel-move based digital surface if

1) each line-cell of S has at least 1 and at most 2 parallel-moves in S,

2) any two surface-cells of S are line-connected in S, and

3) S does not contain any 3D-cells.

Definition 3(Chen and Zhang, 1993a, 1993b) p is an inner point in surface S if every line-cell containing p has exactly 2 parallel-moves in S. p is a boundary point in S if there exists a line-cell containing p that has only one parallel-move in S. The set of all boundary points is denoted by ∂S.

Definition 4(Chen and Zhang, 1993a, 1993b) Let S be a connected subset of Σ_3. Assume

$p \in S$ and $S(p) = S \cap (N_{27}(p) \cup \{p\})$. p is called a regular surface point of S if:

(1) each line-cell in S containing p has at least 1 and at most 2 parallel-moves in $S(p)$,

(2) any two surface-cells containing p in S are line-connected in $S(p)$, and

(3) $S(p)$ does not contain any 3D-cells.

p is said to be a regular inner surface point if p is a regular surface point and each line-cell containing p has exactly two parallel-moves in $S(p)$.

Theorem 4 (Chen *et al.* 1999a) A simple surface point defined by Morgenthaler-Rosenfeld (1982) of a set $S \subset \Sigma_3$ is a regular inner surface point, and vice versa.

Theorem 5 (Chen and Zhang 1993c, Chen *et al.* 1999a) There are exactly six types of regular inner surface points.

The six types of regular inner surface points is shown in Figure 3. Chen and Zhang (1993c) and Latecki (1995, 1998) contained more detail discussions about this aspect.

The definition of digital manifolds is similar to that of digital surface. Two points $p = (x_1, x_2, ..., x_m)$ and $q = (y_1, y_2, ..., y_m)$ are two directly adjacent points in Σ_m if $d_D(p,q) = \sum_{i=1}^{m} |x_i - y_i| = 1$. p and q are adjacent points if $d(p,q) = \max_{1 \le i \le m} |x_i - y_i| = 1$. Only direct adjacency is connected in this section.

Generally, a k-cell (k-cube) has exactly 2^k points, $p_1, ..., p_{2^k}$, where each of $p_1, ..., p_{2^k}$ has m components, denoted by $p_i = (x_1^{(p_i)}, ..., x_m^{(p_i)})$. There are $m - k$ components that do not change value for all $p_1, ..., p_{2^k}$. For p_i and $p_j, i \neq j$, $1 \le \sum_{t=1}^{m} |x_t^{(p_i)} - x_t^{(p_j)}| \le k$, and $\max_{t=1}^{m} |x_t^{(p_i)} - x_t^{(p_j)}| = 1$.

The above definition is not intuitive for i-cells. A recursive form using parallel-moves is given in Definition 5. Chen and Zhang (1993a) presented a similar definition for i-cells.

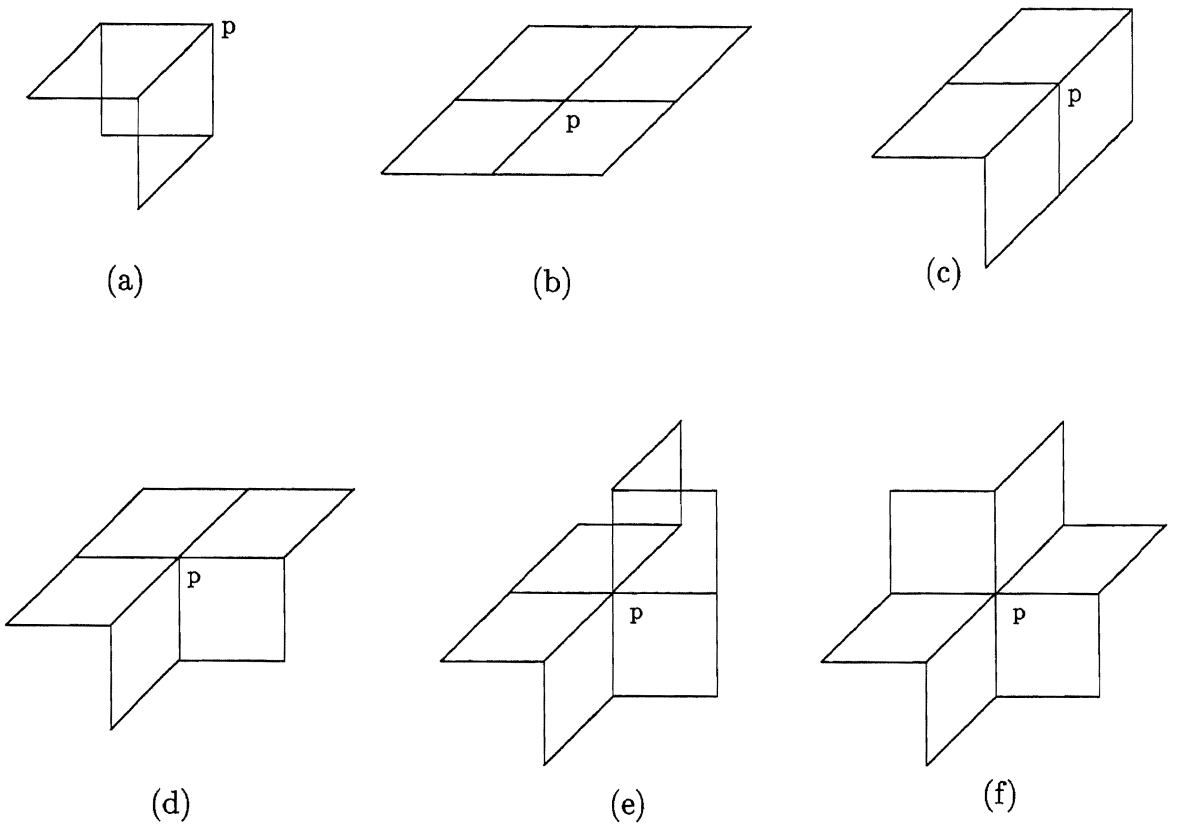


Figure 3: The six types of regular inner surface points.

Definition 5 Let A be a subset of Σ_m. A' is called a parallel-move of A if (1) $|A| = |A'|$, (2) $A \cap A' = \phi$, and (3) there exists a bijection mapping $f : A \to A'$ such that $d(a, f(a)) = 1$ for all $a \in A$. In addition, a and b are adjacent in A if and only if $f(a)$ and $f(b)$ are adjacent in A'.

If an i-cell A' is a parallel-move of an i-cell A, then $\{A, A'\}$ is a $(i+1)$-cell.

Two cells u and v are point-adjacent if they share a point, line-adjacent if they share a line-cell, and k-adjacent if they share a k-cell. Cells u and v are said to be k-connected in S if there exists a path $u_0 = u, u_1, ..., u_n = v$ in S, such that u_i and u_{i+1} are k-adjacent, $i = 0, ..., n-1$.

Definition 6 A connected subset M of Σ_m is an n-dimensional digital manifold if any

point $p \in M$ is included in some n-cell of M, and (1) any two n-cells are $(n-1)$-connected in M, (2) every $(n-1)$-cell in M has only one or two parallel-moves in M, and (3) M does not contain any $(n+1)$-cell.

Some algorithms are designed to decide if a set is a digital surface, to track the boundary surface of a solid object, and also to decide if the set contains non-orientable sub-surfaces (Chen and Zhang 1993b, Chen 1997, Chen 1998a). The author presents the tracking algorithm in this section briefly as follows.

Algorithm C (Chen and Zhang, 1993b) A digital surface tracking algorithm: given a solid object D in 3D space Σ_3, track the boundary of the object ∂D. To solve the problem, the algorithm needs to start at a boundary point of the object then to search all line-adjacent 2D-cells on the boundary surface. An algorithm described in Herman (1998) was based on single point movement.

Step 1: Find a boundary point p of the object D. Then find an adjacent boundary point of p, denoted by q. $\{p, q\}$ forms a line-cell, denoted by B_{pq}.

Step 2: Find a parallel-move of B_{pq}, B_{rs} so that point r, s are on ∂D (just need to decide if r or s has an adjacent point whose value is 0, i.e., this point is outside of D). $\{B_{pq}, B_{rs}\}$ forms a 2D-cell, denoted by S_{pqrs}. Store four line-cells of S_{pqrs} into a queue Q. Mark all four line-cells.

Step 3: Pop up a line-cell in queue Q and denote it as B_{pq}. Find an unmarked parallel-moves of B_{pq}, namely B_{rs} on ∂D. Push all unmarked line-cells of $\{B_{pq}, B_{rs}\}$ in Q.

Step 4: Repeat Step 3 until Q is empty. All marked line-cells indicates a component of the boundary surface. (If the object is simply connected (Newman 1954), then the component is the boundary surface.)

4.3 Gradually Varied Fitting and Data Reconstruction

Seismic signals are often registered by a large number of seismic sensors placed in the nodes of a rectangular lattice on the earth surface (Costain and Coruh 1997). Data is only available on the rectangular lattice. If one needs an entire data volume, then data reconstruction must be performed. For instance, if four edge-line data of a layer is known, then the surface on the rectangle bounded by the four edges can be fitted. One can assume that the data on the edge is "continuous" or λ-connected. How then is the surface fitted? Mathematically, one can choose Coons surfaces or B-spline methods. Such techniques assume the guiding data satisfies a continuous equation. Chen (1990) proposed a solution that allows discrete outputs to be computed directly from discrete inputs without any assumption of a continuous equation.

In order to make the mathematical concept simpler to deal with, this thesis proposes gradual variation instead of λ-connection (Chen 1990). Let $A_1, A_2, ..., A_m$ be the m rational numbers, and $A_1 < A_2 < ... < A_m$. Assume f is a function from Σ_2 to $\{A_1, A_2, ..., A_m\}$. Let p, q be two adjacent points in Σ_2. f is said to be gradually varied on p and q, if $f(p) = A_i$ implies $f(q) = A_{i-1}$, A_i, or A_{i+1}. f is said to be gradually varied if f is gradually varied on any pair of adjacent points p, q in Σ_2. If $f(p) = A_i$ and $f(q) = A_j$, the level-difference between p and q is $|i - j|$. Hence, it is easy to see that gradual variation is a type of λ-connection.

Let D be a connected subset in Σ and $J \subset D$. If given $f_J : J \to \{A_1, A_2, ..., A_m\}$, is there an extension of f_J, $f_D : D \to \{A_1, A_2, ..., A_m\}$ such that for all $p \in J$, $f_J(p) = f_D(p)$? This is called the gradually varied interpolation problem. The key theorem is shown below.

Theorem 6 (Chen, 1990, 1992) The necessary and sufficient condition under which there exists the gradually varied interpolation is that for any two points p and q in J, the length of the shortest path between p and q in D is not less than the level-difference between p and q.

More generally, the constructive theorem was proved by Chen (1994): Let i_Σ_m be an indirectly adjacent grid space. Given a subset J of D and a mapping $f_J : J \to i_\Sigma_m$, if the distance of any two points p and q in J is not less than the distance of $f_J(p)$ and $f_J(q)$ in i_Σ_m,

then there exists an extension mapping f of f_J, such that the distance of any two points p and q in D is not less than the distance of $f(p)$ and $f(q)$ in i_Σ_m. Such f is called a gradually varied surface. In other words, any digital manifold (graph) can normally immerse an arbitrary i_Σ_m. Chen (1994) also has shown that any digital manifold (or graph) can normally immerse an arbitrary tree T.

An envelop theorem, a uniqueness theorem, and an extension theorem preserving the norm are also obtained. These results can be used to design specific gradually varied surfaces.

Theorem 7(Chen 1994) If J is a non-empty subset of D and a mapping $f_J : J \to \Sigma_1$ has a gradual varying extension on D, then there are two gradually varied functions S_1 and S_2 on D, such that every gradually varied function S is between S_1 and S_2. In other words, for any $x \in D$, $S_1(x) \geq S(x) \geq S_2(x)$.

Suppose that p and p' in J satisfy $d(p,p) = d(f(p), f(p'))$, then the valuations of each point in the shortest path between p and p' are unique. This is called a non-abundant path.

Theorem 8 (Chen, 1992, 1994) Let J be a non-empty subset of D and a mapping $f_J : J \to \Sigma_1$ has a gradual varying extension on D, then there is only a unique gradually varied function on D if and only if D is covered by non-abundant shortest paths whose two end points are in J.

Let f_D be a gradually varied functional on D, the length of the longest non-abundant path is called the norm of f_D , which is denoted by $||f_D||$. $||f_D||$ has some strongly geometric meanings.

Theorem 9 (Chen 1994) Let f_J be a function defined on a non-empty subset J of D. If for some $n > 0$, then:

1) for all p and p' in J, $d(p,p') \geq d(f_J(p), f_J(p'))$;

2) if $d(f_J(p), f_J(p')) = k \cdot n + i$, $0 \leq i < n$ then

a) $d(p,p') \geq k \cdot n + (k-1)$, $i = 0$,

b) $d(p,p') \geq k \cdot n + k + i$, $i \neq 0$,

then there exists a gradually varied extension f_D of f_J with $||f_D|| \leq n$.

If there is no gradually varied interpolation for J, one can still perform a gradually varied approximation, practically according to the following theorem.

Without loss of generality, assume $A_i = i$ for convenience. Let us denote

(1)$F_i(p) = \{f_J(p) \pm t | t = 0, 1, ..., i\}$,

(2)$F_i(p)/F_i(q) = \{x | x \in F_i(p)$ and there is $y \in F_i(q)$ such that $d(p,q) \geq d(x,y)\}$,

(3)$F_i^k(p) = \cap_{q \in J} F_i^{k-1}(p)/F_i^{k-1}(q)$,

where $F_i^0(p) = F_i(p)$.

Theorem 10 (Chen, 1992, 1994) (1) For every p and k, $F_i^k(p) \subset F_i^{k-1}(p)$. (2) If there exists a k, for all $p \in J$, $F_i^k(p) = F_i^{k+1}(p)$. Thus, for all $N > k$, $F_i^k(p) = F_i^N(p)$.

Theorem 11 (Chen, 1992, 1994) If $\exists k \forall p \in J(F_i^{k+1}(p) = F_i^k(p) \neq \emptyset)$, then for every $p \in J$, let $g_J(p) = inf\{F_i^k(p)\}$. Then, g_J satisfies that for any $p, q \in J$, $d(p,q) \geq d(g(p), g(q))$, and $\forall_{p \in J}(d(g(p), f(p)) \leq i)$.

Algorithm D(Chen 1994) The algorithm for the uniform approximation can be designed as follows:

Step 1: Initially, let $i = 1$, and compute $F_i(p)$, $F_i(p)/F_i(q)$, and $F_i^k(p)$.

Step 2: If the condition of theorem 11 is satisfied. If it does not hold, $i \leftarrow i + 1$ and go to Step 1.

Step 3: Let $g_J(p) = inf\{F_i^k(p)\}$ for each $p \in J$.

Step 4: Get a gradually varied extension f_D of g_J.

5 Applications and Experimentation

The methods described in previous chapters are applied to two practical problems in this thesis. The first application is the reconstruction of velocity volume in seismic data processing, and the other is the ionogram scaling in space science.

5.1 Reconstruction of Velocity Volume

The goal for the first application is to reconstruct a 41 x 20 x 201 interval velocity array (Chen *et al.*, 1998). In seismic data processing, each vertical data trace is called a CDP trace. Assuming there are m CDP traces with known interval velocity in the edge of a region of ground surface, then the distance between each CDP point can vary. Eight CDP traces and 201 sample points along each of the traces were acquired from the Sichuan province region in China. The data was subjected to the following procedures:

Step 1: Velocity layer classification: In this step, λ-connected segmentation was applied to the 8 CDP traces. A total of six λ-connected components that indicates five interfaces of layers were obtained.

Step 2: Boundary curve fitting and interface fitting: The goal of this step was to fit or find every interface. The program first fits the boundary of the surface to a curve such as a rational function. Then the software used gradually varied surface fitting algorithm to obtain the layer interfaces.

Step 3: Surface Lofting: A lofted surface was obtained by the interpolation of two interface surfaces. In this application, the software program arrived at 171 lofted surfaces. Each lofted surface was treated as a small layer.

Step 4: Gradually varied fill on lofted surfaces: A lofted surface gave the geometric locations in which each point should be assigned a "continuous" velocity value.

Step 5: Final filling: Since some points were not in the lofted surfaces, one simply assigned a value to these points using 1-D linear interpolation.

5.2 Ionogram Scaling

This subsection discusses an application of the λ-segmentation technique to search and examine the frequency-height trace in ionograms acquired by a vertical incidence HF sounder (ionosonde) (Chen *et al.* 1994a). Since each trace is represented by a set of irregular points in the ionogram image, a dynamic genetic algorithm is developed to find the optimal trace. This problem is also referred to as automatic ionogram scaling. The scaling of ionospheric parameters is a laborious task because of the complex structures which often appear in ionograms.

The following procedure is designed to arrive at the solution:

Step 1: Fuzzification and amplitude-based filter: In order to use the λ-segmentation technique, first a fuzzification for the ionogram is made. It is a straightforward method, which converts the number of points in a small region into a meaningful number.

Step 2: λ-connected segmentation: The purpose of applying λ-connected segmentation is to partition the height-frequency image into several inter-connected components. Any two parts are disconnected in the sense of λ-connectedness.

Step 3: Curve search in a component: In order to find all physically realizable curves, all possible curves in a component must be searched. It is known that all components consist of curves, although it is not known which one of them is the most physically relevant curve. Before applying the curve-search algorithm, it is necessary to build a point-successor list which represents the point (pixel) and its successors in a component. The algorithm can also eliminate some curves that are not relevant to our solution. For convenience, the curves in the components are called sub-curves.

Step 4: Genetic algorithms applied to search optimal traces: Merge two or more sub-

curves of different components into a single curve. Once each of the sub-curves in the entire image is found, all possible curves can be enumerated and evaluated. However, due to the computational overhead, it may not be possible to do so. Genetic algorithms have many advantages with respect to the optimal search in real-time computing. A genetic algorithm is used to get a near optimal answer.

In addition to Step 4, when applying a genetic algorithm, the curve is represented as a string. Each element in the string is a number representing a subcurve in a component. The curve generated by some sub-curves must have no overlap. The fitness function is used to evaluate whether a string is a viable curve. Since the target of this project is to find the longest and smoothest curve, the fitness function was simply defined as the length of the curve. According to the properties of ionograms, the smoothness of a curve is a desirable property, required for further processing.

6 Comparative Studies

The aim of this chapter is to present some personal remarks on the contributions of the research and to explain the author's role on multi-authored papers in related references.

6.1 The Contributions of the Thesis

The work presented in this thesis demonstrates three contributions to image processing and/or discrete geometry. The first contribution is the development of the λ-connected segmentation: A significant method for the solution of a class of problems as encountered in seismic velocity layer analysis (Chen 1985, Chen *et al.* 1994a). This work extends Rosenfeld's fuzzy segmentation. λ-connectedness provides a general idea to many problems. Based on λ-connectedness, a searching technique can be developed to solve ionogram scaling (Chen *et al.* 1994a, Tsai and Berkey 2000). λ-connectedness can also be applied to social science too. See Chen, Adjei and Cooley (2000).

Fuzzy connectedness and fuzzy segmentation method were first proposed by Rosenfeld (1979, 1984). It is still an active area of research (Wendling *et al.* 1997, Herman 1998). λ-connectedness differs from Rosenfeld's fuzzy connectedness because the latter does not have the property of transitivity. Therefore, Rosenfeld's fuzzy connectedness is not an equivalence relation. Rosenfeld's fuzzy geometry defines an image as a 2D fuzzy set. λ-connectedness emphases the fuzziness of the connection of pixels but not the pixel itself.

λ-connected segmentation is more natural and powerful than some threshold (clipping level) methods in terms of image segmentation. λ-connected segmentation is a type of region-growing segmentation (Gonzalez and Wood 1993, Chen 1991b, Chen *et al.* 1994b). The work described in this thesis has provided a general theoretical foundation and practical algorithms to λ-connectedness. In theory, it has been proven that the λ-connectedness is an equivalence relation and designed the $O(n)$ time search algorithm (Chen, 1985, 1991b). Chen, Cheng, and Zhang (1994b) expanded the λ-connectedness to the multiple-range case and developed an

$O(nlog(n))$ time algorithm for normal λ-connected segmentation. In practice, λ-connectedness was implemented and used in seismic layer segmentation and ionogram scaling applications. Therefore, λ-connectedness not only has beneficial mathematical properties but also has other practical advantages.

The paper by Chen, Cheng, and Zhang (1994b) was reviewed by German Math Abstracts (also see Appendix A). This paper was also selected by Klir and Yuan (1995) as one of the significant papers in their book "Fuzzy Sets and Fuzzy Logic." In addition, Tsai, Berkey, and Liu (1996) proposed three special formulas for λ-connectedness in ionogram segmentation.

The second contribution of the thesis is that it proposes a simple and intuitive definition for digital surfaces as given in Definition 2 of Section 4.2. Accordingly, the classification theorem for digital surface points in direct adjacency was proven (Chen and Zhang 1993c, Chen *et al.* 1999a). Based on the definition, it is easy to develop tracking and decision algorithms for general surfaces. The definition is not only able to describe closed digital surfaces but also able to deal with open surfaces unlike the first mathematical definition of digital surfaces that is only applied to closed surfaces (Morgenthaler and Rosenfeld 1982).

A λ-connected set can be found by Algorithm A in section 4.1. If this λ-connected set is the boundary of an oil-gas trap or an interface of two layers, then it is important to know or determine if the λ-connected set is a surface in a digital space. Furthermore, if the set is not a perfect surface, to extract the surface information from the λ-connected set is necessary. Because Morgenthaler and Rosenfeld (1982) only considered closed surfaces, a definition for general digital surfaces is needed for real applications. A simple and intuitive definition for digital surfaces and digital manifolds was introduced by Chen and Zhang (1993a, 1993b). Based on this definition a digital surface or the boundary surface of a solid object can be identified or tracked by Algorithm C in Section 4.2 (Chen and Zhang 1993b, Chen 1997). This algorithm uses line-cells as the search element that may eliminate some redundant search points comparable to the tracking algorithm in Herman(1998).

Based on the definition of digital surfaces given by this thesis, the classification theorem

of simple surface points in direct adjacency was discovered for 3D digital spaces (Chen and Zhang 1993c, Chen *et al.* 1999a). The theorem states that there are exactly six kinds of simple surface points in direct adjacency. This classification theorem is important to computer vision because it has a direct bearing on the computation of structures and topological properties of 3D objects. In other words, any point satisfying Morgenthaler and Rosenfeld's definition must be one of the six types of surface-point in direct adjacency. Francon (1995) discussed the use of five of the six types of points to generate a digital surface. Subsequently, Latecki (1995, 1998) found that if an image is well-composed in 3D, then its boundary is a union of such six types of surface-points. In addition, the definition proposed by Chen and Zhang (1993a, 1993b) is equivalent to Morgenthaler-Rosenfeld's definition (1982) of digital surfaces in direct adjacency in closed surface cases (Chen *et al.* 1999a). For indirect adjacency, the definition of digital surfaces is still under investigation (Chen, 1998a, 1998b).

On this topic, three papers (Chen and Zhang 1993b, 1993c; Chen *et al.* 1999a) were reviewed by Mathematical Review, Computer and Control Abstract, and Engineering Index. Two papers of Chen and Zhang (1993a, 1993b) were cited by Latecki (1995, 1998).

The third and the most important contribution of this thesis, in the author's view, is the discovery of a discrete surface fitting method (Chen 1990) that was based on the proof of the theorem of the necessary and sufficient condition for gradually varied interpolation. An algorithm to fit such a discrete surface that is based only on their guiding points was designed. This approach does not require fitting formulae such as B-splines or Bezier polynomials (Chen 1990, Chen 1992).

This work provides a mathematical solution or an algorithm to fill a λ-connected region. Since 1989, the concept of "gradual variation" was proposed to simplify λ-connectedness. This concept subsequently led to the proving of the key theorem: the necessary and sufficient condition of the gradual varied fill (Chen 1990). The results from this work were later reviewed by German Math Abstract (index number 712.68085, 1991). Pawlak (1998), the founder of "Rough Set Theory," cited the paper of Chen (1990). Similar to the gradually varied fuction,

Rosenfeld (1986) used a concept called the "continuous" digital function for image processing. Recently, Pawlak (1998) studied rough functions and rough calculus, and Nakamura and Rosenfeld (1997) proposed digital calculus. Although, these concepts are all based on the discretely "continuous" functions, similar to the gradually varied functions, the filling theorem, i.e. Theorem 6 in Section 4.3, was the first non-trivial result.

For the application of seismic data processing, the author led an application group in China to apply λ-connectedness and other AI techniques to find oil-gas reserves between 1986-1990 (Chen and Lu 1990). The related software developed by the author's group was one of the first generation of software in the field of AI application of geological data processing in China. The software found an industrial gas reserve in a region in the west of China. Zhao (1998) mentioned the author's work in his textbook "Introduction to Computing Science."

6.2 Outline of the Published Work

The submitted paper list of this thesis includes nine only-author papers and nine first-author papers. All of these papers are unrelated to any other degree program. (The author's B.S. degree was obtained in 1982 in the field of combinatorial algorithm. The author's M.S. degree thesis was on fuzzy neural networks (Chen 1995, Chen *et al.* 1999b).)

The only-author papers are Chen (1985), Chen (1990), Chen (1991a), Chen (1991b), Chen (1992), Chen (1994), Chen (1997), Chen (1998a), and Chen(1998b). The multi-authors papers are Chen and Zhang (1993a), Chen and Zhang (1993b), Chen and Zhang (1993c), Chen, Berkey, and Johnson (1994), Chen, Cheng, and Zhang (1994), Chen, Zhang, and Cooley (1996) , Chen and Berkey *et al.* (1998a), Chen, Cooley, and Zhang (1998b), and Chen, Cooley, and Zhang (1999a).

The following summaries the contributions made by the author. The paper of Chen (1985) was a theoretical paper that first proposed λ-connectedness.

The paper of Chen (1990) was a theoretical paper that established the concept of grad-

ual variation and proved the theorem of necessary and sufficient condition of gradually varied fill. The paper was reviewed by Zentralblatt Math (1991) (i.e., German Math Abstract) with reference number 712.68085 and Engineering Index (EI1005534), and it was indexed by Mathematical Review with index number 1071838. This paper was cited by Pawlak (1998).

The paper of Chen (1991a) was the continuation of Chen (1990). It gave some theoretical analysis and algorithms for gradually varied interpolation. The main objective of the paper was to design fast algorithms for gradually varied interpolations in Jordan domains.

The paper of Chen (1991b) considered both theoretical and practical aspects of λ-connectedness. This paper was divided into three parts. The first part detailed the λ-connected segmentation and the design of an optimal time algorithm for image segmentation. The second part improved the well-known split-and-merge image segmentation. These two parts were contributed by the author alone. The third part was to compare the λ-connected segmentation and the split-and-merge segmentation in seismic data segmentation. Lan Zhang, Yen-Feng Yin, and Dong-Bin Zhou completed the implementation of λ-connected segmentation. Lan Zhang implemented the split-and-merge segmentation. The author was the project leader. The paper was cited by Tsai *et al.* (1996) and Tsai and Berkey (2000).

The paper of Chen (1992) was the continuation of the paper Chen (1990). The paper concentrated on a random algorithm for gradually varied fitting when the guiding points do not satisfy the necessary and sufficient condition as presented in Theorem 6 in Section 4.3. The paper was written by the author alone. Professor Yao-Ting Zhang helped the author to revise the paper.

The paper of Chen (1994) was the continuation of the paper Chen (1990, 1991a, 1992). This paper dealt with many aspects of gradual variation such as the general constructive theorem, the envelop theorem, the uniqueness theorem, and the extension theorem. These theorems are presented again in Section 4.3. It was reviewed by Engineering Index (EI2024377).

The paper of Chen (1997) was the continuation of the previous papers on digital surfaces

by Chen and Zhang (1993a, 1993b, 1993c). A general definition of digital manifolds, and some algorithms for discrete surface recognition, were proposed by Chen (1997). The general definition let to the use of quad-tree representation of a surface-cell. Based on that, the author designed an algorithm to extract all boundary points of a region (Chen 1997).

The paper of Chen (1998a) discussed the well-known Morgenthaler-Rosenfeld digital surfaces (Morgenthaler and Rosenfeld, 1982) and showed the definition given by Morgenthaler and Rosenfeld (1982) is not complete. The paper also proposed a new definition of digital surfaces in indirect adjacency.

The paper of Chen (1998b) analyzed the relationship of point digital spaces (used in this thesis) and raster spaces (used in Herman (1998)) and stated that these two spaces are dual-spaces.

The paper of Chen and Zhang (1993a) proposed a definition of digital manifolds based on parallel-moves. The definition was contributed by Chen. Zhang was the project leader. The project was funded by Utah State University. Zhang revised the original paper for the publication.

The paper of Chen and Zhang (1993b) designed several fast algorithms for surface tracking and decision. The author designed the first phase of the algorithms. Dr. Zhang made the final version for the publication based on the author's first manuscript. It was reviewed by Mathematical Review 95g:68119 and Engineering Index (EI1810389). The paper was cited by Latecki (1995, 1998).

The paper of Chen and Zhang (1993c) found the classification of simple surface points and presented a topological property for simple closed surfaces. The author found these two results. Dr. Zhang verified these results. It was reviewed by Mathematical Review 95g:68120 and Engineering Index (EI1810390). The paper was also cited by Latecki (1995, 1998).

The paper of Chen, Berkey, and Johnson (1994) contained three applications using λ-connected search in geophysical data processing. First application was for seismic cross-well

data. The research work was done in Dr. Johnson's group at University of Utah. Dr. Johnson provided the inversion data and the author implemented the search algorithm. The second application refered to surface reconstruction that was made by the author. The third application was for ionogram scaling. The research work was done in Dr. Berkey's group at Utah State University. The research was funded by National Science Foundation of USA. The author designed and implemented a program for the ionogram scaling under the supervision of Dr. Berkey.

The paper of Chen, Cheng, and Zhang (1994) was the continuation of Chen (1985, 1991b). This paper made three contributions. It extended the λ-connectedness into multi-range images and proved that the complexity of the normal λ-connected segmentation is $O(nlogn)$. Finally, it applied the method to search lithology layers. The author discussed the original ideas of extending the λ-connectedness into multi-range images to Drs. Cheng and Zhang. Dr. Cheng selected the direction for this paper on the topic. The author wrote the first proof of the theorem on the normal λ-connected segmentation and implemented the lithology layer search program. The paper was revised twice with major changes on the normal λ-connected segmentation part. It was reviewed by Zentralblatt Math 817.68137 and Engineering Index (EI1987378). The paper was cited by Tsai, Berkey, and Liu (1996) and listed by Klir and Yuan (1995).

The paper of Chen, Zhang and Cooley (1996) gave some examples of a regular polygon that can be used to partition a plane. Chen wrote the first manuscript. Dr. Zhang and Dr. Cooley revised it.

The paper of Chen *et al.* (1998a) was designed to be used to solve many curve search problems. However, the work was not completed. Most of the work of this paper was done by the author.

The paper of Chen, Cooley, and Zhang (1998b) presented a method for intelligent data fitting in seismic data processing (Section 5.1). Some of the work was done by the author's research group in 1986-1990. The author was the principal investigator for the project and

designed the method of intelligent data fitting.

The paper of Chen, Cooley, and Zhang (1999) gave a complete proof of the equivalence between Morgenthaler-Rosenfeld digital surfaces and the regular digital surface defined by Chen and Zhang (1993b). This paper mathematically verified that the surface definition by Chen and Zhang (1993b) is valid. The author worked out the first proof and wrote the first manuscript. Dr. Cooley did most of the editing on the paper. It was reviewed by Mathematical Review 2000a:68158 and Engineering Index (EI4593944). It is in the process of Zentralblatt Math pre991.05360.

7 Conclusions and Future Work

This thesis introduced the method of λ-connectedness for solving three important problems in seismic image processing– (1) Seismic layer search, (2) oil-gas boundary surfaces recognition, and (3) 3D volume data reconstruction.

Based on graph theory, the λ-connectedness method was developed systematically. λ-connectedness is used to represent the relationship among pixels in an image. Any smoothly connected region in a digital image can be described as a λ-connected subset or component.

This thesis first proved that the λ-connectedness is a mathematical equivalence relation. Then, this thesis developed the method of λ-connected search for image segmentation, that can be applied to seismic layer determination. Several theoretical and practical results were obtained, such as $O(n)$ time search algorithm, normal λ-connected segmentation, and λ-connectivity analysis of the image using the random graph theory.

This thesis also proposes a simple and intuitive definition of digital surfaces. It allows us to describe mathematically what special set of pixels forms a digital surface. A surface tracking algorithm was also designed to extract surface points from a solid object. This research can be applied to find the interface between two layers, or the boundary surfaces of an oil-gas reservoir. This thesis proposes a method of searching a λ-connected interface or boundary that allows the extraction of a digital surface from it. A by-product, the classification theorem of digital surface points, was used to help with the understanding of digital geometry and topology.

This thesis also proposes a purely discrete surface fitting technique called gradually varied fitting. Based on the concept of λ-connectedness, the gradual variation was defined to describe the continuity in a subset of an image. The theorem of the necessary and sufficient condition for the gradually varied interpolation was proven. A random gradually varied surface fitting was designed and applied to fit interfaces of seismic layers. Since seismic 3D data reconstruction can be manipulated by individual layer data reconstruction, an example to reconstruct a 3D velocity volume is provided in this thesis.

In the future, λ-connectedness can be developed for use on a region or a grouped set of pixels instead of a single pixel. In this case, after selecting the partition (group of pixels), one can still apply the λ-connected segmentation to the region search. The main advantage of using small regions rather than single pixels is that the noise is reduced.

Work is still continuing on λ-connectedness and related methods. The author and other researchers are working on three related areas. Based on previous work, Chen, Adjei, and Cooley (2000) wrote a methodology paper to enlarge the scope of the λ-connectedness method. Adjei and Chen(2000) are working on the relationship between λ-connectedness and rough set theory. In that paper, they apply the λ-connectedness method to data mining. Chen, Adjei, and Lu (2000) also recently worked on the reconstruction of surfaces and oil-gas prediction.

The author plans to continue his work on some other practical issues, such as the use of λ-connected digital surfaces to evaluate the oil-gas trap volume and also plans to extend the idea of λ-connectedness to texture image processing.

Recently, the author introduced the method of λ-connected segmentation to the NASA Ames Research Center for Bioinformatics (USA) for a possible application in 3D human brain image processing. The center showed their considerable interest on implementing λ-connected segmentation in their existing software system. The author was invited to visit the center "for further discussion and potential collaboration." The author also received a travel award ($1500) from North Dakota NASA EPSCoR through NASA grant #NCC5-399 for this event.

Appendix A Comments on a λ-Connectedness Paper

Zentralblatt MATH 817.68137

Chen, Li; Cheng, Heng-Da; Zhang, Jianping

Fuzzy subfiber and its application to seismic lithology classification.(English) [J] Inf. Sci., Appl. 1, No.2, 77-95 (1994). [ISSN 1069-0115]

Let Σ_N be an N-dimensional discrete space. A fuzzy (N, n) subfilter is defined as a mapping $f : \Sigma_N \to [0, 1]^n$. In particular for $N = 2$ and $n = 1$, the concept reduces to a 2D fuzzy set (relation). Basic operations on fuzzy subfibers are defined and resulting structures investigated. The key idea of an λ-connectedness is introduced in the context of image segmentation viz. an identification of pixels of similar characteristics. Some λ-connected segmentation algorithms are proposed and analyzed with respect to their time complexity. A detailed application of fuzzy subfibers to lithology classification in oil-gas exploration is also presented. [W.Pedrycz (Winnipeg)]

Appendix B List of the Works on Which the Application Based

1 Chen, L. (1985) "Three-dimensional fuzzy digital topology and its applications(I)," Geophysical Prospecting for Petroleum, 24(2), pp 86-89.

2 Chen, L. (1990) "The necessary and sufficient condition and the efficient algorithms for gradually varied fill," Chinese Science Bulletin, 35(10), pp 870-873. (It's Chinese version can be found in Kexue Tongbao, Vol 34(1989), 1588-1590.)

3 Chen, L. (1991a) "The properties and the algorithms for gradually varied fill," Chinese J. Computers, 14(3), pp 161-169.

4 Chen, L. (1991b) "The lambda-connected segmentation and the optimal algorithm for split-and-merge segmentation," Chinese J. Computers 14(5), pp 321-331.

5 Chen, L. (1992) "Random gradually varied surface fitting," Chinese Science Bulletin, 37(16), pp 1325-1329.

6 Chen, L. (1994) "Gradually varied surface and its optimal uniform approximation," *IS&TSPIE* Symposium on Electronic Imaging, SPIE Proc. Vol. 2182, pp 300-307.

7 Chen, L. (1997) "Generalized discrete object tracking algorithms and implementations," Vision Geometry VI, Proc. SPIE Vol. 3168, pp 184-195.

8 Chen, L. (1998a) "(alpha, beta)-type digital surfaces and general digital surfaces," R. Melter, A. Wu, and L. Latecki, Vision Geometry VII, SPIE Proc. 3454, pp 28-39.

9 Chen, L. (1998b) "Point spaces and raster spaces in digital geometry and topology," R. Melter, A. Wu, and L. Latecki, Vision Geometry VII, SPIE Proc. 3454, pp 145-155.

10 Chen, L., Berkey, F.T., and Johnson, S.A. (1994a) "The application of a fuzzy object search technique to geophysical data processing," Proc. of *IS&TSPIE* Symposium on Electronic Imaging, SPIE Proc. Vol. 2180, 300-309.

11 Chen, L., Berkey, F.T., Cooley, D. H. , He, Y., Zhang, J-P., and Zhang, L. (1998a) "Intelligent curve tracking algorithms and implementations," Sadjadi, F., ed, Automatic Target Recognition VIII, SPIE Proc. 3371, pp 74-82.

12 Chen, L., Cheng, H., and Zhang, J. (1994b) " Fuzzy subfiber and its application to seismic lithology classification," Information Science: Applications, Vol 1, No 2, pp 77-95.

13 Chen, L., Cooley, H., and Zhang, J. (1999a) "The equivalence between two definitions of digital surfaces," Information Sciences, Vol 115, No 1, pp 201-220.

14 Chen, L., Cooley, H., and Zhang, L. (1998b) "An Intelligent data fitting technique for 3D velocity reconstruction," Application and Science of Computational Intelligence, Proc SPIE 3390, 1998, pp 103-112.

15 Chen, L. and Zhang, J. (1993a) "Digital manifolds: An intuitive definition and some properties," Proc. Second ACM Symp. on Solid Modeling and Applications, pp 459-460.

16 Chen, L. and Zhang, J. (1993b) "Digital surface definition and the fast algorithms for decision and tracking," Vision Geometry II, SPIE Proc. Vol. 2060, pp 169-178.

17 Chen, L. and Zhang, J. (1993c) "Classification of simple digital surface points and a global theorem for simple closed surfaces," Vision Geometry II, SPIE Proc. Vol. 2060, pp 179-188.

18 Chen, L., Zhang, J., Cooley, H., (1996) " Regular polygons and their application to digital curves," Vision Geometry V, SPIE Proc. Vol 2826, pp 133-139.

Appendix C Declaration

I declare that this thesis is my own unaided work. It is being submitted for the degree of Doctor of Philosophy at the University of Luton. It has not been submitted before for any degree or examination in any other University

Li Chen 

31 day of March, 2001.

References

Adjei, O. and Chen, L. (2000) "λ-connected searching for rough sets and data mining," Manuscript.

Aho, A., Hopcroft, J. E., and Ullman, J. D. (1974) "The Design and Analysis of Computer Algorithms," Addison-Wesley.

Aminzadeh, F. (1988) "Pattern Recognition and Image Processing (Handbook of Geophysical Exploration/Section 1, Seismic Exploration, Volume 20)," Pergamon Press.

Artzy, E., Frieder, G. and Herman, G. (1981) "The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm," Computer Vision Graphics Image Process, Vol 15 , pp. 1-24.

Bollobas,B.(1985) "Random Graphs," Academic Press.

Chen, L. (1985) "Three-dimensional fuzzy digital topology and its applications(I)," Geophysical Prospecting for Petroleum, 24(2), pp 86-89.

Chen, L. (1990) "The necessary and sufficient condition and the efficient algorithms for gradually varied fill," Chinese Science Bulletin, 35(10), pp 870-873. (It's Chinese version can be found in Kexue Tongbao, Vol 34(1989), 1588-1590.)

Chen, L. (1991a) "The properties and the algorithms for gradually varied fill," Chinese J. Computers, 14(3), pp 161-169.

Chen, L. (1991b) "The lambda-connected segmentation and the optimal algorithm for split-and-merge segmentation," Chinese J. Computers 14(5), pp 321-331.

Chen, L. (1992) "Random gradually varied surface fitting," Chinese Science Bulletin, 37(16), pp 1325-1329.

Chen, L. (1994) "Gradually varied surface and its optimal uniform approximation," *IS&TSPIE* Symposium on Electronic Imaging, SPIE Proc. Vol. 2182, pp 300-307.

Chen, L. (1995) " Possibility-based fuzzy neural networks and their application to image processing," MS Thesis of Computer Science, Utah State University;

Chen, L. (1997) "Generalized discrete object tracking algorithms and implementations," Vision Geometry VI, Proc. SPIE Vol. 3168, pp 184-195.

Chen, L. (1998a) "(alpha, beta)-type digital surfaces and general digital surfaces," R. Melter, A. Wu, and L. Latecki, Vision Geometry VII, SPIE Proc. 3454, pp 28-39.

Chen, L. (1998b) "Point spaces and raster spaces in digital geometry and topology," R. Melter, A. Wu, and L. Latecki, Vision Geometry VII, SPIE Proc. 3454, pp 145-155.

Chen, L., Adjei, O., and Cooley, D.H. (2000a) "λ-Connectedness : Method and Application," Proceedings of IEEE conference on System, Man, and Cybernetics 2000, pp 1157-1562.

Chen, L., Adjei, O., and Lu, Y. (2000b) "Digital surface reconstruction and oil-gas trap prediction," L. Latecki ed, Vision Geometry VX, SPIE Proc. 4117, pp 176-185.

Chen, L., Berkey, F.T., and Johnson, S.A. (1994a) "The application of a fuzzy object search technique to geophysical data processing," Proc. of $IS\&TSPIE$ Symposium on Electronic Imaging, SPIE Proc. Vol. 2180, 300-309.

Chen, L., Berkey, F.T., Cooley, D. H. , He, Y., Zhang, J-P., and Zhang, L. (1998a) "Intelligent curve tracking algorithms and implementations," Sadjadi, F., ed, Automatic Target Recognition VIII, SPIE Proc. 3371, pp 74-82.

Chen, L., Cheng, H., and Zhang, J. (1994b) " Fuzzy subfiber and its application to seismic lithology classification," Information Science: Applications, Vol 1, No 2, pp 77-95.

Chen, L., Cooley, H., and Zhang, J. (1999a) "The equivalence between two definitions of digital surfaces," Information Sciences, Vol 115, No 1, pp 201-220.

Chen, L., Cooley, D. H., and Zhang, J. (1999b) "Possibility-based fuzzy neural networks and their application to image processing," IEEE Trans. on Systems, Man, and Cybernetics (B), Vol 29(1), pp 119-126.

Chen, L. and Lu, Y. *et al.* (1990) "Study of Fuzzy Recognition Methods for Geophysical Prospecting Data," The Report of the 7.5 State Key Projects of China, Index number 75-54-02-08-12.

Chen, L., Cooley, H., and Zhang, L. (1998b) "An Intelligent data fitting technique for 3D velocity reconstruction," Application and Science of Computational Intelligence, Proc SPIE 3390, 1998, pp 103-112.

Chen, L. and Zhang, J. (1993a) "Digital manifolds: An intuitive definition and some properties," Proc. Second ACM Symp. on Solid Modeling and Applications, pp 459-460.

Chen, L. and Zhang, J. (1993b) "Digital surface definition and the fast algorithms for decision and tracking," Vision Geometry II, SPIE Proc. Vol. 2060, pp 169-178.

Chen, L. and Zhang, J. (1993c) "Classification of simple digital surface points and a global theorem for simple closed surfaces," Vision Geometry II, SPIE Proc. Vol. 2060, pp 179-188.

Chen, L., Zhang, J., Cooley, H., (1996) " Regular polygons and their application to digital curves," Vision Geometry V, SPIE Proc. Vol 2826, pp 133-139.

Cormen, T.H., Leiserson, C.E., and Rivest, R.L. (1993) "Introduction to Algorithms," MIT Press.

Costain, J.K. and Coruh, C. (1997) "Basic Seismic Theory (Handbook of Geophysical Exploration, Vol 1)," Pergamon Press.

Francon, J. (1995) "Discrete combinatorial surfaces, Graphical Models and Image Processing, Vol 57, 20-26.

Gonzalez, R.C. and Wood, R. (1993) "Digital Image Processing," Addison-Wesley, Reading, MA, 1993.

Goodman, J.E. and O'Rourke J. eds. (1997) " Handbook of Discrete and Computational Geometry," CRC Press, New York, 1997.

Herman, G.T. (1998) "Geometry of Digital Spaces," Birkhauser, Boston, 1998.

Klir, G.J. and Yuan, B. (1995) "Fuzzy Sets and Fuzzy Logic: Theory and Applications," Prentice Hall.

Latecki, L.J.(1995) "3D well-composed pictures," Proceedings of SPIE's Conference on Vision Geometry, San Diego, California, 196-203. or (1997) Graphical Models and Image Processing Vol 59, 164-172.

Latecki, L.J.(1998) "Discrete Representation of Spatial Objects in Computer Vision," Kluwer Academic Publishers.

Morgenthaler, D.G. and Rosenfeld, A. (1982) "Surfaces in three-dimensional images," Information and Control, vol 51, pp. 227-247.

Nakamura, A. and Rosenfeld, A. (1997) "Digital calculus," Information Sciences, Vol 98, pp 63-98.

Newman, M. (1954) "Elements of the Topology of Plane Sets of Points," Cambridge Press, London.

North, F.K. (1985) "Petroleum Geology," Allen & Unwin Inc., Boston.

Pawlak, Z. (1999) "Rough sets, rough functions and rough calculus," in S.K. Pal and A. Skowron ed, *Rough Fuzzy Hybridization*, Springer-Verlag, 99-109.

Rosenfeld, A. (1979) "Fuzzy digital topology," Information and Control, Vol 40, 76-87.

Rosenfeld, A. (1984) "Fuzzy geometry of image subsets," Pattern Recognition letter, Vol 2, 311-317.

Rosenfeld, A. (1986) " 'Continuous' functions on digital pictures, " Pattern Recognition Letters, vol 4, 177-184.

Tsai, L., Berkey, F.T., and Liu, J.Y. (1996) "Automatic ionogram trace identification using fuzzy classification techniques," Computer Aided Processing of Ionograms and Ionosonde Records, France, pp 45-50.

Tsai, L., Berkey, F.T., (2000) "Ionogram analysis using fuzzy segmentation and connectedness techniques," Radio Science, vol 35, No, 5, pp 1173-1186.

Wendling, L., Paries, A., and Desachy, J. (1997) "Image splitting into trees of fuzzy regions applied to pattern recognition," International Journal of Intelligent Data Analysis, vol.2, No 1.

Zhao, Z. (1998) "Introduction to Computing Science," Science Press of China, 1998.

Attachment

Co-authored Paper Statements and Copies of the Works

The research time period of the related projects on λ-connected search is from 1984-present. The author started the research when he was an assistant engineer at the Research Institute of Geophysical Prospecting for Petroleum in Nanjing, China in 1984. Some related projects were supported by the Chinese State 7.5 Key Projects Program from 1986 to 1991.

The author started the research on digital surfaces and manifolds in 1991, and continues to work on it to the present day. This research took place in Utah, USA. The related project was partially supported by Utah State University.

The author studied gradually varied functions during 1989-1994. This research took place in Wuhan University, China. The related project was mainly supported by the Chinese Academy of Science.

Co-authored Paper Information Sheet*
For

Chen, L., Berkey, F., Cooley, D., He, Y., Zhang, J., Zhang, L., ``Intelligent curve tracking algorithms and implementations,'' SPIE Conference on Automatic Target Recognition VIII, SPIE Vol 3371, 74-82.

Where did the research	Utah State University
When did the research	1993-1999
Project leader(s)	Berkey
Corresponding author for the publication	Chen
What did Li Chen do	Li Chen wrote the first rough draft. Dr. Berkey supervised the project and provided real data. The others helped for the published version.
Funding and supporter	Utah State University
Review and Citation Information	

Additional comments:

Authors' signature

Li Chen
Date

Frank T. Berkey
Date

Donald H. Cooley
Date 8/30/00

Yanxiang He
Date

Jianping Zhang
Date 7/27/2000

Lan Zhang
Date

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

Co-authored Paper Information Sheet*

For

Chen, L., Berkey, F., Cooley, D., He, Y., Zhang, J., Zhang, L., ``Intelligent curve tracking algorithms and implementations,'' SPIE Conference on Automatic Target Recognition VIII, SPIE Vol. 3371, 74-82.

Where did the research	Utah State University
When did the research	1993-1999
Project leader(s)	Berkey
Corresponding author for the publication	Chen
What did Li Chen do	Li Chen wrote the first rough draft. Dr. Berkey supervised the project and provided real data. The others helped for the published version.
Funding and support	Utah State University
Review and Citation Information	Tsai, L.-C. and F.T. Berkey, in press, vol. 35, No. 5, *Radio Science*, Sept/Oct, 2000.

Additional comments:
The methods described in this manuscript was carried out by Mr. Chen with minimal supervision on my part. His work was consistently professional and at a very high of quality. Again, the application to the method he developed to the analysis of ionograms was again a seminal work in the field.

Authors' signature FRANK T. BERKEY

Li Chen
Date 8/25/00

Frank T. Berkey
Date
8 AUGUST 2000

Donald H. Cooley
Date

Yanxiang He
Date

Jianping Zhang
Date

Lan Zhang
Date 8/25/00

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

Co-authored Paper Information Sheet*
For

Chen, L., Berkey, F., Cooley, D., He, Y., Zhang, J., Zhang, L., ``Intelligent curve tracking algorithms and implementations,'' SPIE Conference on Automatic Target Recognition VIII, SPIE Vol 3371, 74-82.

Where did the research	Utah State University
When did the research	1993-1999
Project leader(s)	Berkey
Corresponding author for the publication	Chen
What did Li Chen do	Li Chen wrote the first rough draft. Dr. Berkey supervised the project and provided real data. The others helped for the published version.
Funding and supporter	Utah State University
Review and Citation Information	

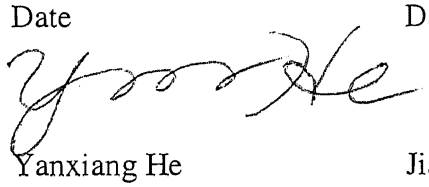
Additional comments:

Authors' signature

Li Chen
Date

Frank T. Berkey
Date

Donald H. Cooley
Date



Yanxiang He
Date 8/25/2000

Jianping Zhang
Date

Lan Zhang
Date

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

Co-authored Paper Information Sheet*
For

Chen, L., Cheng, H., and Zhang, J. (1994) `` Fuzzy subfiber and its application to seismic lithology classification," Information Science: Applications, 1(2), pp 77-95.

Where did the research	Utah State University
When did the research	1992-1993
Project leader(s)	Cheng, H., Zhang, J.
Corresponding author for the publication	Cheng, H.
What did Li Chen do	This paper expended lambda-connectedness to high dimensional range space and applied to geophysical lithology classification. Li Chen did the data processing, graphical display and wrote first rough manuscript under the supervision of Dr. Cheng and Dr. Zhang. Li Chen did the original mathematical analysis. Li Chen, Cheng and Zhang improved and corrected together.
Funding and supporter	Zhang's USU faculty funding
Review and Citation Information	Engineering Index, Computer Control Abstract, German Math Zbl, Listed by the book "Fuzzy Logic and Fuzzy Sets."

Additional comments:

Authors' signature

Li Chen
Date 7/25/00

Cheng, Hengda
Date

Zhang, Jianping
Date 7/27/2000

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

Co-authored Paper Information Sheet*
For

Chen, L., Cheng, H., and Zhang, J. (1994) `` Fuzzy subfiber and its application to seismic lithology classification," Information Science: Applications, 1(2), pp 77-95.

Where did the research	Utah State University
When did the research	1992-1993
Project leader(s)	Cheng, H., Zhang, J.
Corresponding author for the publication	Cheng, H.
What did Li Chen do	This paper expended lambda-connectedness to high dimensional range space and applied to geophysical lithology classification. Li Chen did the data processing, graphical display and wrote first rough manuscript under the supervision of Dr. Cheng and Dr. Zhang. Li Chen did the original mathematical analysis. Li Chen, Cheng and Zhang improved and corrected together.
Funding and supporter	Zhang's USU faculty funding
Review and Citation Information	Engineering Index, Computer Control Abstract, German Math Zbl, Listed by the book "Fuzzy Logic and Fuzzy Sets."

Additional comments:

Authors' signature

Li Chen
Date 7/25/2000

Cheng, Heng-Da
Date 8/10/00

Zhang, Jianping
Date

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

Co-authored Paper Information Sheet*
For

Chen, L., Cooley, H., and Zhang, L. (1998) ``An Intelligent data fitting technique for 3D velocity reconstruction," Application and Science of Computational Intelligence, Proc SPIE 3390, 1998, pp 103-112.

Where did the research	Wuhan University, Utah State University
When did the research	1989-1991, 1997-1998
Project leader(s)	Chen, L. Cooley, D.
Corresponding author for the publication	Chen, Li
What did Li Chen do	Li Chen wrote the first rough draft. Lan Zhang and others did data processing. Cooley did the most correction. Li Chen, Cooley and Zhang improved and corrected together.
Funding and supporter	Partially supported by Research Institute of Geophysical Prospecting for Petroleum And Utah State University
Review and Citation Information	Computer Control Abstract

Additional comments:

Authors' signature

Li Chen
Date 8/20/00

Donald H. Cooley
Date 8/30/00

Zhang, Lan
Date 8/20/00

*This Form is made for the Li Chen's PhD report to The University of Luton (UK)

Co-authored Paper Information Sheet*
For

Chen, L., Cooley, H., and Zhang, J. (1999) ``The equivalence between two definitions of digital surfaces," Information Sciences, 115(1), pp 201-220.

Where did the research	Utah State University
When did the research	1992-1994, 1996-1999
Project leader(s)	Zhang, Cooley
Corresponding author for the publication	Chen, Li
What did Li Chen do	Li Chen wrote the first rough draft and did the original mathematical analysis under the supervision of Dr. Cooley and Zhang. Li Chen, Cooley and Zhang improved and corrected together.
Funding and supporter	Partially supported by Zhang's USU faculty funding,
Review and Citation Information	Engineering Index, Computer Control Abstract, Math Review.

Additional comments:

Authors' signature

Li Chen
Date 7/25/2000

Donald H. Cooley
Date 8/30/00

Jianping Zhang
Date 7/27/2000

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

Co-authored Paper Information Sheet*
For

Chen, L., Zhang, J., and D. H. Cooley (1996) ``Regular polygons and their applications to digital curves, " Vision Geometry II, SPIE Proc. Vol. 2826, pp 133-139.

Where did the research	Utah State University
When did the research	1995-1996
Project leader(s)	Zhang
Corresponding author for the publication	Chen
What did Li Chen do	Li Chen wrote the first rough draft under the supervision of Dr. Zhang and Cooley
Funding and supporter	Computer science Dept at USU
Review and Citation Information	Engineering Index

Additional comments:

Authors' signature

Li Chen
Date 7/25/2000

Jianping Zhang
Date 7/27/2000

Donald H. Cooley
Date 8/30/2000

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

Co-authored Paper Information Sheet*
For

Chen, L., Berkey, F.T., and Johnson, S.A. (1994) ``The application of a fuzzy object search technique to geophysical data processing,'' Proc. of IS&T-SPIE Symposium on Electronic Imaging, SPIE Proc. Vol. 2180, 300-309.

Where the research was carried out.	The University of Utah, Utah State University
When did the research	1991, 1993-1994
Project leader(s)	Johnson, Berkey
Corresponding author for the publication	Li Chen
What did Li Chen do	This paper applied lambda-connectedness to three problems in geophysical data processing. Li Chen did the data processing, graphical display and wrote the initial manuscript. Johnson and his group did the reconstruction data and Berkey did the ionogram filtering. Li Chen did the detailed analysis under the supervision of Prof. Johnson and Prof. Berkey.
Funding and supporter	U.S. National Science Foundation
Review and Citation Information	Engineering Index, Computer Control Abstract, Cited by Tsai and Berkey's papers.

Additional comments:
Mr. Chen applied his knowledge of fuzzy analysis and his experience in geophysical data processing to an entirely new application in ionospheric physics. This was pioneering work in the field and has recently been cited as the genesis for work soon to be published in the International journal *Radio Science* (Sept/Oct 2000 issue).

Authors' Signature FRANK T. BERKEY

Li Chen	Berkey, F.T	Johnson, S.A
Date 7/20/2000	Date 8 AUGUST 2000	Date

Co-authored Paper Information Sheet*
For

Chen, L., Berkey, F.T., and Johnson, S.A. (1994) ``The application of a fuzzy object search technique to geophysical data processing," Proc. of IS&T-SPIE Symposium on Electronic Imaging, SPIE Proc. Vol. 2180, 300-309.

Where the research was carried out.	The University of Utah, Utah State University
When did the research	1991, 1993-1994
Project leader(s)	Johnson, Berkey
Corresponding author for the publication	Li Chen
What did Li Chen do	This paper applied lambda-connectedness to three problems in geophysical data processing. Li Chen did the data processing, graphical display and wrote the initial manuscript. Johnson and his group did the reconstruction data and Berkey did the ionogram filtering. Li Chen did the detailed analysis under the supervision of Prof. Johnson and Prof. Berkey.
Funding and supporter	U.S. National Science Foundation
Review and Citation Information	Engineering Index, Computer Control Abstract, Cited by Tsai and Berkey's papers.

Additional comments:
Mr. Chen applied his knowledge of fuzzy analysis and his experience in geophysical data processing to an entirely new application in ionospheric physics. This was pioneering work in the field and has recently been cited as the genesis for work soon to be published in the International journal *Radio Science* (Sept/Oct 2000 issue).

Authors' Signature Frank T. Berkey

Li Chen	Berkey, F.T	Johnson, S.A
Date	Date	Date
7/20/2000	8 AUGUST 2000	16 August 2000

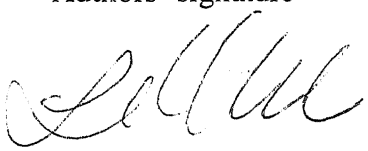
Co-authored Paper Information Sheet*
For

Chen, L. and Zhang, J. (1993b) ``Digital surface definition and the fast algorithms for decision and tracking,'' Vision Geometry II, SPIE Proc. Vol. 2060, pp 169-178.

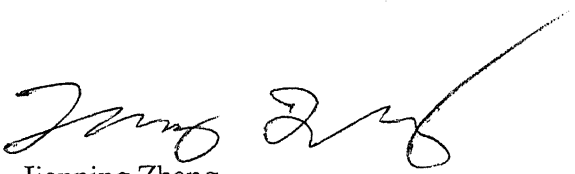
Where did the research	Utah State University
When did the research	1992-1993
Project leader(s)	J. Zhang
Corresponding author for the publication	Chen
What did Li Chen do	Li Chen wrote the first rough draft. Zhang finalized the published version. Chen and Zhang improved together.
Funding and supporter	Zhang's faculty funding.
Review and Citation Information	Math Review and Computer Control Abstract. Cited by Latecki at CVGIP(1997) and his book(1998).

Additional comments:

Authors' signature



Li Chen
Date 7/25/2000



Jianping Zhang
Date 7/25/2000

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

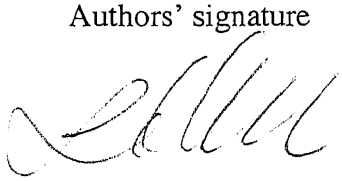
Co-authored Paper Information Sheet*
For

Chen, L. and Zhang, J. (1993c) ``Classification of simple digital surface points and a global theorem for simple closed surfaces," Vision Geometry II, SPIE Proc. Vol. 2060, pp 179-188.

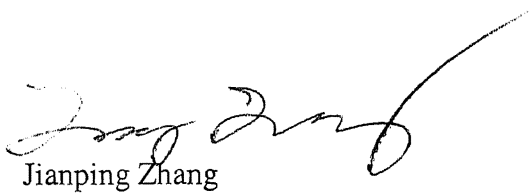
Where did the research	Utah State University
When did the research	1992-1993
Project leader(s)	Zhang
Corresponding author for the publication	Chen
What did Li Chen do	Li Chen wrote the first rough draft and finalized the published version. Chen did the classification work originally. Chen and Zhang improved together.
Funding and supporter	Zhang's faculty funding.
Review and Citation Information	Math Review and Computer Control Abstract. Cited by Latecki at CVGIP(1997) and his book(1998).

Additional comments:

Authors' signature



Li Chen
Date 7/25/2000



Jianping Zhang
Date 7/25/2000

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

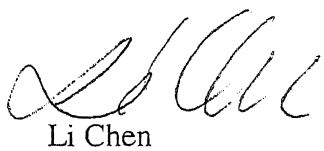
Co-authored Paper Information Sheet*
For

Chen, L. and Zhang, J. (1993a) ``Digital manifolds: An intuitive definition and some properties,'' Proc. Second ACM Symp. on Solid Modeling and Applications, pp 459-460.

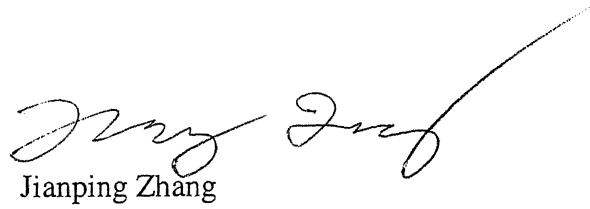
Where did the research	Utah State University
When did the research	1992-1993
Project leader(s)	Zhang
Corresponding author for the publication	Zhang
What did Li Chen do	Li Chen wrote the first rough draft under the supervision of Dr. Zhang. Zhang finalized the published version. Li Chen had the parallel-move concept and the digital surface definition. Chen and Zhang improved together.
Funding and supporter	Zhang's faculty funding.
Review and Citation Information	A reviewer said the paper is "very important" and "break through."

Additional comments:

Authors' signature



Li Chen
Date 7/25/2000



Jianping Zhang
Date 7/25/2000

*This Form is made for Li Chen's PhD report to The University of Luton (UK)

第24卷 第2期　　石　油　物　探　　Vol.24,No.2
1985年6月　GEOPHYSICAL PROSPECTING FOR PETROLEUM　June,1985

三维Fuzzy数字拓扑及应用(I)

陈　溧

（地质矿产部石油物探研究所）

摘　要

本文引入L-，*l*-连通度的概念，把Rosenfeld等人建立的三维数字拓扑模糊化，并在三维地震处理中找到了它的应用模型。

一、引　言

Rosenfeld等人建立了三维数字拓扑，并把它用于三维图象的曲面分类[1,2,3]。我们知道三维地震数据处理中，有大量的模糊现象。为了使数字拓扑方法有效地用于三维处理，我们把三维数字拓扑模糊化*，作为对数字拓扑的一点补充。

二、连　通　度

设Σ是三维网格点，$J:\Sigma\longrightarrow[0,1]$是一映射。我们用$J(\Sigma)$表示三维图象，即$J(\Sigma)$是$\Sigma$的一个模糊子集。

设Σ中的点x，y是邻接（6-邻或26-邻）的，x，y的连通度定义为：

$$\begin{cases} C(x,y)=1-\dfrac{|J(x)-J(y)|}{|J(x)+J(y)|} & J(x),\ J(y)\text{不全为}0 \\ C(x,y)=1 & \text{否则} \end{cases} \tag{1.1}$$

设π是一条路径，即$\pi=\pi_{(p_0,p_m)}=p_0p_1\cdots p_m$，其中$p_j$，$p_i$邻接当且仅当$j=i+1$，或$j=i+1 \mod(m)$。则：

$$L(\pi)=\prod_{i=0}^{m-1} C_{(p_i,p_{i+1})} \tag{1.2}$$

$$l(\pi)=\min\{C_{(p_i,p_{i+1})}\mid i=0,1,\cdots,m-1\} \tag{1.3}$$

分别称为π的L-流和*l*-流。

---

* 本文中的基本概念和定义见[1]，[4]

设S是Σ的子集，$\underset{\sim}{S}$是S的Fuzzy子集，即

$$\underset{\sim}{S}=\sum_{x_i\in S}J(x_i)/x_i \tag{1.4}$$

如果x，y∈S，那么x，y在S中的L-，l-连通度定义为

$$C_S(x,y)=\max\{L(\pi(x,y))\mid \pi\text{落在}S\text{中}\} \tag{1.5}$$

$$C'_S(x,y)=\max\{l(\pi(x,y))\mid \pi\text{落在}S\text{中}\} \tag{1.6}$$

命题1.1 (i)若π是S中的一条路径，则有$L(\pi)\geqslant l(\pi)$；

(ii) 对任意的$S\subset\Sigma$，有$C_S(x,y)\leqslant C'_S(x,y)$；

(iii)(a) $L(\pi)=1$当且仅当$l(\pi)=1$，

(b)$C_S(x,y)=1$当且仅当$C'_S(x,y)=1$。

证：(i)，(ii)成立是显然的。对于(iii)(a)，因为设$\pi=p_0p_1\cdots p_m$，$L(\pi)=1$，则$C_{(p_i,p_{i+1})}=1\ (i=1,2,\cdots,m-1)$。这就证明了$l(\pi)=1$。反之，$l(\pi)=1$，则$C_{(p_i,p_{i+1})}\geqslant 1$，故有$L(\pi)\geqslant 1$，再由(i)得$L(\pi)=1$。

对于(iii)(b)，设$C_S(X,y)=1$，则存在S中的某$\pi_0=\pi_0(x,g)$，使$L(\pi_0)=1$，因此有$l(\pi_0)=1$，即是$C'_S(x,y)=1$。反之亦然。

定义1.1 设$\underset{\sim}{R}_S$是$\underset{\sim}{S}$的L-连通关系矩阵，$\underset{\sim}{R}'_S$是$\underset{\sim}{S}$的l-连通关系矩阵；若$C_S(x,y)=1$则称x，y是L-完全连通的，若$C'_S(x,y)=1$则称x，y是l-完全连通的；若$C_S(x,y)\geqslant\lambda$（λ为[0,1]中的某常数）则称x，y是λ—L连通的，若$C'_S(x,y)\geqslant\lambda$则称x，y是λ—$l$连通的。

命题1.2 L-完全连通也是l-完全连通的。

命题1.3 完全连通是一等价关系，λ—l连通也是一等价关系；λ—L连通满足自反律、对称律。

上过命题的证明从略，利用完全连通(或λ—l连通)性可以对$\underset{\sim}{S}$(在$\underset{\sim}{R}_S$或$\underset{\sim}{R}'_S$上)进行等价分类；利用λ—L连通性可以进行聚类。地震处理中可以把这种类作为地层，即每一类作同一层。也可以把类作为同相轴。等价分类后，每一类称为S的部分（或λ—l部分）。

命题1.4 设X是S的λ—l部分，那么存在点$x\in X$，使$X-\{x\}$仍是λ—l连通的。

证：我们可以采用λ-截集方法，便X成为完全连通的，再根据[1]文中的证明；最后还原。

A，$B\subset S$称为邻接的，如果有一$a\in A$与B中某点邻接且连通度不为0。设A，B是完全连通的，且A，B邻接；当A∪B亦完全连通时我们称A与B同型。不难得到

命题1.5 完全连通集可分解为若干同型完全连通集的并。

定义1.2 设A，$B\subset S$；A，B间的L-，l-连通度分别定义为

$$C_S(A,B)=\max\{C_S(x,y)\mid x\in A,\ y\in B\} \tag{1.7}$$

$$C'_S(A,B)=\max\{C'_S(x,y)\mid x\in A,\ y\in B\} \tag{1.8}$$

命题1.6 设$S\subset\Sigma$，那么$C_S(A,B)\leqslant C'_S(A,B)$。

定理 设A，$B\subset S$是完全连通的，那么A中任意元到B中任意元的连通度相等。

证. 我们只证L-连通度的情形，若$a\in A$，$b\in B$. 设

$\pi_{min}(a,b)=ap_1p_2\cdots p_{n-1}b$，这里$L(\pi_{min})=C_S(a,b)$，又设a′，b′分别是A，B中的任意元素，由于A，B均完全连通，则存在从a′到a，b′到b的路径分别落在A、B中，记为$\pi_A(a',a)$和$\pi_B(b',b)$。显然$\pi_A(a',a)\ \pi_{min}(a,b)\ \pi_B(b,b')$与$\pi_{min}(a,b)$的L-流相等。定理得证。

事实上，$C_S(A,B)$表现了A，B之间的相似程度；如果A，B分别表示两地层，$1-C_S(A,B)$表示的正是A，B的差异。

三、洞和分离

关于洞，在参考文献[1]中已有确切的定义，而我们在此要稍加变形，使之适应模糊集的情形。

$S\subset\Sigma$，且不到达边界，Δ_λ是[0.1]中的一个普遍的闭区间（以后设此区间的长度不变）；且$x\geqslant\lambda$，若$x\in\Delta_\lambda$，

$$S_\lambda=\{x\mid J(x)\in\Delta_\lambda\ 且\ x\in S\} \tag{2.1}$$

S_λ的洞称为S的λ-洞，同样，与S_λ的补$\overline{S_\lambda}$邻接的非洞区城，称为S的λ-底。

给定S和$\tilde{S}$，检验S是否有λ-洞的算法是这样的，先求S_λ，再重复删去“单边界”点[1]，看其是否收缩到一点。

设A、B和C是Σ的三个独立子集，A与B，B与C是邻接的。如果，从A到C的任一路径必通过B，称B分离A，C。B对A，C的分离度定义为

$$S_B(A,C)=1-\max\{L(\pi)\mid \pi\text{在}A\cup B\cup C\text{中}\} \tag{2.2}$$

$$S'_B(A,C)=1-\max\{l(\pi)\mid \pi\text{在}A\cup B\cup C\text{中}\} \tag{2.3}$$

命题3.1 (i) $C(A,C)\geqslant 1-S_B(A,C)$

(ii) $C'(A,C)\geqslant 1-S_B'(A,C)$

定理 S的λ-洞与S-边的分离度大于λ。

我们不难想到，盖层是油气区和沉积的分离层，而油气区则是盖层的λ-洞。

四、曲 面

S称为λ-简单闭曲面，如果S_λ是一简单闭曲面。定义λ-曲面是因为，真正的油气边界是一个λ-简单闭曲面，而不是简单闭曲面。我们可以证明

定理　若S是λ-简单闭曲面，则S把$\overline{S}$分割成它的λ-洞和底两部分。

定理　不存在6-连接的λ-交叉帽。

朱子强同志提出许多宝贵意见，在此表示感谢。

参 考 文 献

〔1〕A.Rosenfeld, Three-Dimensional Digital Topology, Information and control, 50, 119～127(1981).

〔2〕D.Morgenthaler and A.Rosenfeld, Surfaces in Three-Dimensional Digital Images, Information and control, 51, 227～247(1981).

〔3〕G.Reed and A.Rosenfeld, Recognition of Surfaces in Three-Dimensional Digital Images, Information and Control, 53, 108～120(1982).

〔4〕J.Chassery and M.Chenin, Topologies on Discrete Space, Digital Image Processing (J.Simonad and R.Haralich eds.), 59～66(1981).

（收稿日期：1984年8月）

3-D FUZZY DIGITAL TOPOLOGY AND ITS APPLICATION

Chen Li

Abstract

In this paper, the author introduced the notation of L, 1-connectivity and made the 3-D digital topology founded by Rosenfeld et al fuzzified. The model used in 3-D processing has also been found.

THE NECESSARY AND SUFFICIENT CONDITION AND THE EFFICIENT ALGORITHMS FOR GRADUALLY VARIED FILL*

CHEN Li (陈 澍)

(*Department of Computer Science, Wuhan University, Wuhan 430072, PRC*)

Received June 24, 1989.

Keywords: gradually varied fill, digital manifold, algorithm.

Filling is an important part in pattern recognition and computer vision. In the discrete plane (grid of plane) Σ_2, fill means "determination of the region D enclosed by the simple closed curve J which is given beforehand"[1,2]. In other words, if the valuation of the point on the contour J is 1, we want to do a valuation for Σ_2 to make the valuation of the point p be 1 iff point p belongs to D.

But in reality, values of the contour J are variable, it is not single valued. In most cases, values of the contour J are varying gradually. That is to say, not only the region D should be determined but also the value of each point in D should be determined and such values should be proved to be gradually varied. In the present note, only the second part will be examined. As for the first part, it may be seen in Refs. [1] and [2].

In continuous spaces, this problem is trivial. If J is a Jordan curve in plane and $f: J \to R$ is the continuous function (or belongs to $C^{(n)}$), obviously there is the continuous ($C^{(n)}$) extension of f on D. Whereas, the result is not true in a discrete space. On the contrary, the concerned results in continuous space are corollaries in discrete space.

Note that, in a discrete plane, some of the simple closed curves are not Jordan curves, it is dependent on the definition of connectivity (4-adjacent is Jordan, 8-adjacent is not)[3,4]. We will see that it is very important for the design of the algorithm for gradually varied fill and its time complexity.

Given the formal definition of gradual variation. Let $A_1, A_2, \cdots, A_m$ be the m rational numbers (or real numbers), and $A_1 < A_2 < \cdots < A_m$, i.e. $\{A_1, A_2, \cdots, A_m\}$ gives the system of gray level. And let $f: \Sigma_2 \to \{A_1, \cdots, A_m\}$ be the gray scale.

We say that f is gradually varied on two adjacent points p and p' of Σ_2 (or other discrete space), if $f(p) = A_i$ then $f(p')$ is A_{i-1}, A_i or A_{i+1}. f is gradually varied on digital

* Project supported by the Science Fund of the Chinese Academy of Sciences for Young Scientists.

manifold if f is gradually varying on any two adjacent points of digital manifold (curve, surface, etc.).

Repeat two basic concepts as follows:

(1) Length of path. Let $P=p_1,p_2,\cdots,p_n$ be the path, the length of P is $n-1$. The length of the shortest path between p and p' is expressed by $d(p,p')$.

(2) Difference of gray level. Suppose $f(p)=A_i$ and $f(p')=A_j$, the difference of gray level on p and p' is $|i-j|$. It is denoted by $LF(p,p')$.

Let D be the simply connected set in the discrete space (such as Σ_2) and the contour J of D be simply closed. If $f_J: J\to\{A_1,\cdots,A_m\}$ is gradually varied on J, the problem of gradually varied fill is defined as: there is an extension of f_S on D, $f_D: D\to\{A_1,\cdots,A_m\}$, such that f_D is gradually varied on D. Here the extension means $f_D(p)=f_J(p)$ if $p\in J$.

Theorem. *The necessary and sufficient condition under which there exists the gradually varied fill on D is that for any two points p and p' in J, the length of the shortest path between p and p' in D is not less than the difference of gray level of p and p'.*

Proof. (1) Suppose f_D is the gradually varied function on D, then f_D is gradually varied on every curve (path) in D. The curve may be the shortest path between two points p and p' in the contour J of D. Hence, the length of the shortest path is not less than the difference of gray level of p and p', i.e. $d(p,p')\geq LF(p,p')$. (2) Conversely, suppose $d(p,p')$, if $p,p'\in J$ in D, the following proof is consturctive (an algorithm). For every $p\in J$, let $f_D(p)=f_J(p)$ and $f_D(p)=\theta$ and if $p\in D-J$, denote

$$D_0=\{p: f_D(p)\neq\theta,\ p\in D\}.$$

For any $p, p'\in D_0$, we have $d(p,p')\geq LF(p,p')$.

(a) If $D_0\neq D$, we can find a point $r\in D_0$, such that there is an adjacent point x of r, $x\in D-D_0$. We may as well suppose $f_D(r)=A_i$.

(b) Let $f_D(x)=f_D(r)=A_i$, and let

$$m(x)=\{p: f_D(p)<f_D(x),\ p\in D_0\}.$$

$$M(x)=\{q: f_D(q)>f_D(x),\ q\in D_0\}.$$

There exist three cases:

(i) If there is a $p\in m(x)$, such that $d(x,p)<LF(x,p)$. $d(r,p)\leq d(r,x)+d(x,p)$ and $d(r,x)=1$, hence $d(r,p)\leq 1+d(x,p)$ and $d(x,p)\geq d(r,p)-1$. In addition, $d(r,p)\geq LF(r,p)$ and $LF(r,p)=LF(x,p)$, thus $d(x,p)\geq d(r,p)-1\geq LF(x,p)-1$. According to the assumption $d(x,p)<LF(x,p)$, hence $d(x,p)=LF(x,p)-1$. For any $q\in M(x)$, $f_D(p)<f_D(x)<f_D(q)$, so $LF(p,q)=LF(p,x)+LF(x,q)$. Again $d(p,x)+d(x,q)\geq d(p,q)\geq LF(p,q)$, then

$$d(p,x)+d(x,q)\geqslant LF(p,x)+LF(x,q)\geqslant d(p,x)+1+LF(x,q),$$

hence $d(x,q)\geqslant LF(x,q)+1$. Thus, we have proved that if there exists $p\in m(x)$ such that $d(x,p)<LF(x,p)$, then we have

$$\forall p\in m(x)\ (d(p,x)\geqslant LF(p,x)-1)\ \ \&\ \forall q\in M(x)\ \ (d(x,q)\geqslant LF(x,q)+1).$$

We modify the valuation of x, let $f_D(x)=A_{i-1}$. Obviously, for the new value, we have

$$d(y,x)\geqslant LF(y,x),\ \text{if}\ \ y\in D_0\ \ \&\ \ f_D(y)\neq A_i.$$

$$d(y,x)\geqslant LF(y,x)=1,\ \text{if}\ \ y\in D_0\ \ \&\ \ f_D(y)=A_i;$$

(ii) If there is $q\in M(x)$, such that $d(x,q)<LF(x,q)$. Similarly, we have $d(x,q)\geqslant LF(x,q)-1$ for any $q\in M(x)$, and $d(p,x)\geqslant LF(p,x)+1$ for any $p\in M(x)$. Hence, modifying $f_D(x)=A_{i+1}$, we obtain $\forall y\in D_0\ (d(x,y)\geqslant LF(x,y))$.

(iii) If (i) and (ii) are not satisfied, then $f_D(x)=A_i$ will just be required.

(c) Let $D_0\leftarrow D_0\bigcup\{x\}$, thus for any p and p' in D_0, we have

$$d(p,p')\geqslant LF(p,p').$$

Return to (a), and loop this process until $D_0=D$. For any x,y in D, if x and y are adjacent, i.e. $d(x,y)=1$ because $d(x,y)\geqslant LF(x,y)$, then $LF(x,y)=0$ or 1. That is, f_D is gradually varied on x and y, then f_D be gradually varied on D.

In the above proof, we do not make particular restriction except using the symmetric relation and the triangle inequality of distance. Hence, this theorem may be generalized to finite digital manifold. The digital manifold means the "discretization or net sampling" of some manifold.

Corollary 1. *Let D be the simply connected digital manifold, and the boundary J of D be simply closed. Suppose $f_J: J\to\{A_1,\cdots,A_m\}$ is gradually varied on J. Then there is a gradually varied extension of f_J on D, such that $f_D: D\to\{A_1,\cdots A_m\}$ iff the length of the shortest path between any two points p and p' in J is not less than the difference of gray level of p and p' in D.*

Now we discuss the algorithms for gradually varied fill which can be distinguished into two kinds, the first kind is the decision algorithm by which we solve whether "there is the gradually varied fill". The other is the fill itself, or we call the constructive algorithm. In fact, we have already shown these two kinds of algorithms in the proof of the theorem. According to Dijkstra's algorithm and Floyd's algorithm for finding the shortest path, it is not difficult to prove Corollary 2.

Corollary 2. (1) *There is a decision algorithm which has the time complexity $O(|J||D|^2)$.*

(2) *There is the $O(|D|^3)$ time algorithm for gradually varied fill.*

Under usual instance, the number of adjacent points of a point is finite in discrete spaces; and the distance of two points may be calculated in a constant (or linear) time. Thus, we can introduce many concrete results (see "On Gradually Varied Fill Algorithms"). Particularly, if D has Jordan property, i.e. a simple closed curve or super-surface segments the rest of D into two disconnected parts (or several components in discrete case). The illustration is shown in Fig. (1). Here, $\forall x \in D_1 - \{J \bigcup J_1\}$ & $\forall y \in D_2 - \{J \bigcup J_1\}$, every path to join x and y must contain the point of $J_1 \bigcup J$. Thus, if f_1 is gradually varied on D_1 and f_2 is on D_2, for any $x \in J$, we have $f_1(x) = f_2(x)$. Then f is gradually varied on D that is composed by f_1 and f_2. This method is similar to the divide-and-conquer techniques, i.e. we first define the valuation of the shortest path between p and p', then consider D_1 and D_2. If D is defined by 8-adjacent way (non-Jordan see Fig.(2)), then f is not gradually varied on D, here f is composed of f_1 and f_2. Jordan property has essential sense for us to design fill algorithms.

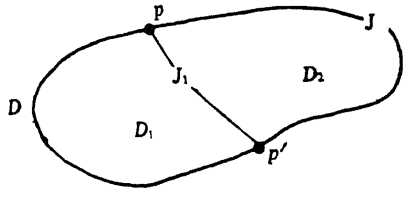


Fig. 1

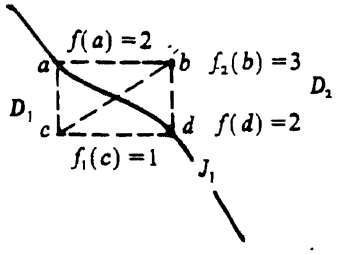


Fig. 2

Corollary 3. *Let D be a square in 4-adjacent Σ_2, then there is an $O(|D| \cdot \log_2 |D|)$ time algorithm for gradually varied fill.*

Actually, these results may be generalized in the sense of λ-connection (or continuity), and can be used for intelligent data fitting[5,6]. The gradually varied fill is also a technique of surface construction (interpolation) under which boundary valuations are known. Like Bezier polynonmial fitting and B-spline, the gradually varied function (surface) which is constructed by the algorithm in this note has a good property.

Corollary 4. *The gradually varied surface is in the convex closure of its guiding points.*

References

[1] Pavlidis, T., *Algorithms for Graphics and Image Processing*, Computer Science Press, Inc., 1982.

[2] Tang, G. Y., *CVGIP*, **42**(1988), 3 : 279—305.

[3] Rosenfeld, A., *Inform. and Control*, **50**(1981), 2 : 119—127.

[4] Kong, T. Y. & Roscoe, A. W., *CVGIP*, **29**(1985), 1 : 60—86.

[5] 陈 漾, 中国人工智能年会文集, 武汉大学出版社, 1989, 165—167.

[6] Besl, P. J. & Jain, R. C., *IEEE Tran.*, PAMI, **10**(1988), 2 : 167—192.

渐变填充的性质和算法

陈　　溧

（武汉大学计算机科学系，武汉 430072）

THE PROPERTIES AND THE ALGORITHMS FOR GRADULLY VARIED FILL

Chen Li

(Wuhan University, Wuhan 430072)

Abstract: Gradually varied fill (GVF) on Jordan digital manifold is discussed. The constricting GVF algorithm and the separating GVF algorithm are developed. It is proved that there exists an $O(|D||\partial D|)$ time algorithm for GVF on Jordan convex set D in discrete grid spaces. Finally, three $O(|D|\log_2|D|)$ time separating algorithms for the Jordan square D, the Jordan triangle D and the Jordan disk D are given, respectively.

Key Words Gradually varied fill, Jordan curve theorem, convex set, fast algorithm.

摘要： 本文着重研究了 Jordan 数字流形上的渐变填充，设计了紧缩渐变填充算法和分裂渐变填充算法；并证明：如果D是离散网格空间上的 Jordan 凸集，那么存在 $O(|D||\partial D|)$ 时间的紧缩算法去做渐变填充。最后，我们对 Jordan 正方形区域、三角域和圆盘，分别给出了它们各自的 $O(|D|\log_2|D|)$ 时间的分裂渐变填充算法。

关键词： 渐变填充，Jordan 曲线定理，凸集，快速算法。

一、引　　言

填充是模式识别和计算机视觉中的一类重要问题[1,2]。在文献[3]中，我们对灰度图象提出了它的“渐变填充”问题：设在一单连通的区域D的边界∂D上有一逐渐变化的灰度值，问是否存在一灰度赋值在D上亦逐渐变化。证明了D上有渐变填充的充要条件是边界上任两点在D中的最短路长度不小于这两点的灰度级差。

本文得到中国科学院青年奖励基金资助，本文 1989 年 4 月 24 日收到。作者陈溧系武汉大学讲师，目前从事计算机理论、图象处理和识别的研究工作。

我们不难看出，渐变填充的解正好是已知边界值的插值曲面，称之为渐变(填充)曲面．在大多数情况下，渐变曲面不是唯一的；我们将证明渐变曲面唯一的充要条件是D中任一点都含于∂D中某两点间在D上的一条无丰（non-abundant）最短路．并给出渐变曲面唯一性的有效判定算法．我们还将讨论渐变曲面的计数问题及计数算法．

尽管对渐变填充有一个$O(|D|^3)$时间的算法，但当$|D|$较大时，这个算法也是非常耗时的．为了解决这个一般性的问题对我们来说是不容易的．然而，在通常的情况下，我们确实可以找到渐变填充的快速算法．特别是，当D有 Jordan 性质时，即D中一条简单闭曲线或超曲面把D分为两个不连通的部分(离散情况可能有多个分支)，我们将分别利用紧缩方法和分治技术得出适用于 Jordan 数字流形的渐变填充算法——紧缩算法和分裂算法．我们有，

(1) 如果D是离散网格点空间中的 Jordan 凸集，那么存在$O(|D||\partial D|)$时间的紧缩算法去做渐变填充．

(2) 如果D是 Jordan 正方形(三角形或圆盘)，那么用分裂算法做渐变填充的时间为$O(|D|\log_2|D|)$．

把单纯剖分方法用于 Jordan 集，可以进一步证明：Jordan 曲面上的渐变填充的时间复杂性不超过$O(|D||\partial D|+|\partial D|^2\log_2|\partial D|)$．

二、渐变填充和它的通用算法

设$A_1, A_2, \cdots, A_m$是m个有理数(或实数)且$A_1 < A_2, \cdots, < A_m$；即$A_1, \cdots, A_m$给出一组灰度级．又设$f: \Sigma_2 \to \{A_1, \cdots, A_m\}$是一灰度图象．

定义 1. 称f在Σ_2中两邻点p, p'上是渐变的，如果$f(p)$和$f(p')$在$\{A_1, \cdots, A_m\}$中相邻或相等．称f在数字流形(曲线、曲面等)上渐变，如果f在该数字流形的任两个邻点上渐变．

设J是Σ_2中一简单闭曲线，填充是指确定了所包围的区域．换句话说，如果J中的点赋值为1，我们要对Σ_2中的各点赋一值，使P在J的内部当且仅当关于P的赋值为1．渐变填充是指设D是Σ_2的某单连通子集(或其它数字流形)，D的边界$\partial D = J$是简单封闭的，已知$f_J: J \to \{A_1, \cdots, A_m\}$．求$f_J$在$D$上的扩张$f_D: D \to \{A_1, \cdots, A_m\}$使$f_D$在$D$上渐变．更一般的可设$J$为$D$的非空子集

为了讨论方便，我们重复两个基本概念：

(1) 路径长度，设$P = p_1 \cdots p_n$是一路径，P的长度为$n-1$．p和p'间的最短路长度记为$d(p, p')$．

(2) 灰度级差，设$f(p) = A_i$及$f(p') = A_j$；p和p'上的灰度级差为$|i-j|$，记作$LF(p, p')$．

当$\max\limits_{p,p' \in J} LF(p, p') \leqslant 1$时，渐变填充是平凡的，本文将不考虑这样的情况．

定理 1. D上存在渐变填充的充要条件是，J上任两点在D中的最短路长度不小于这两点的灰度级差．

上述定理的正确性证明见文献[3]．现在我们给出基于这个定理的通用填充算法．

算法 A：通用渐变填充算法

输入：D, J 及 $f_J: J \to \{A_1, \cdots, A_m\}$，

输出：D 上的渐变函数 $f_D: D \to \{A_1, \cdots, A_m\}$，或指出 D 上无 f_J 的渐变扩张。

过程 A

步骤 1：用 Floyd 最短路径算法计算 D 中各对点之间的最短路及长度。

步骤 2：检验 J 上的每两点 p 和 p'，是否都满足。

$$d(p,p') \geqslant LF(p,p')$$

若有一对不满足，指出 D 上不存在渐变填充，并停机。

步骤 3：置 D_0 为空集。

步骤 4：$D_0 \leftarrow J$ 并对 D_0 中每个点 r 令 $f_D(r) \leftarrow f_J(r)$。

步骤 5：如果 $D_0 = D$ 则输出所有的 $f_D(r)$ 并停机，否则找一点 $r \in D_0$，使 r 有一邻点 x 在 $D - D_0$ 中。不妨设 $f_D(r) = A_i$。

步骤 6：$f_D(x) \leftarrow f_D(r)$，并作

$$m(x) = \{p \mid f_D(p) < f_D(x), p \in D_0\}$$
$$M(x) = \{q \mid f_D(q) > f_D(x), q \in D_0\}$$

步骤 7：如果有一 $p \in m(x)$ 使 $d(x,p) < LF(x,p)$，令 $f_D(x) \leftarrow A_{i-1}$。如果有一 $q \in M(x)$ 使 $d(x,q) < LF(x,q)$，令 $f_D(x) \leftarrow A_{i+1}$。

步骤 8：$D_0 \leftarrow D_0 \cup \{x\}$，转步骤 5.

定理 2. 算法 A 的计算时间为 $O(|D|^3)$。如果 D 中点的邻点数是 K 有界的，则算法 A 的计算时间即是 $O(K|D|^2 + |D|^2\log_2|D|)$.

证明．我们仅对算法 A 的主要步骤的计算时间进行简要的分析.

步骤 1 的计算时间为 $|D|$ 个顶点图的所有对最短路计算所需时间，用 Floyd 算法其时间为 $O(|D|^3)$[4]。如果 D 中点的邻点数是 K 有界的，那么所考虑的图中的边数不超过 $\frac{1}{2}K|D|$；我们用 Dijkstra 算法逐点计算，总时间为 $O(K|D|^2 + |D|^2\log_2|D|)$[4].

步骤 2 的时间耗费不超过 $O(|J|^2)$.

显然，步骤5的计算时间不超过 $O(|D_0| \cdot |D|)$，当邻点数 K 有界时，计算时间不超过 $O(|D_0| \cdot K)$。而步骤 6 ～ 7 的计算时间为 $O(|D_0|)$。那么步骤 5 ～ 7 的计算总耗时为 $O(|D| \cdot |D| \cdot |D|)$；在邻点数 K 有界时为 $O(|D|^2 \cdot K)$.

因此，算法 A 的时间为：

$$O(|D|^3) + O(|J|^2) + O(|D|^3) = O(|D|^3)$$

在邻点数 K 有界的情况下为：

$$O(|D|^2 \cdot K + |D|^2\log_2|D|) + O(|J|^2) + O(|D|^2K)$$
$$= O(K|D|^2 + |D^2| \log |D|).$$

三、渐变曲面的唯一性和计数

我们知道渐变填充的目的是求 D 上的函数 $f_D: D \to \{A_1, \cdots, A_m\}$，其几何意义可等

价于定义在D上的曲面，它显然是已知边界值f_J的插值曲面．称之为渐变填充曲面或渐变曲面．与Coons曲面和其它计算方法不同的是，前者是直接从离散输入得出离散结果，后者是通过连续方法所得．

根据定理1，我们知道如果对任意的$x,y\in J$，有$d(x,y)\geqslant LF(x,y)$．那么D上存在一渐变填充．当有一对点$x,y\in J$使$d(x,y)=LF(x,y)$时，则在所有连接x,y的最短路上各点的待定函数值是唯一确定的．换句话说，如果$\forall x,y\in J$，$d(x,y)>LF(x,y)$，必然存在两种和两种以上的渐变填充．

定义2. 如果$xy\in D$且$f(x)$，和$f(y)$已确定，称$AN(x,y)=d(x,y)-LF(x,y)$为x,y的半数（abundant number）．如果$AN(x,y)=0$称x,y间的最短路为无丰最短路．而

$$AN(D,J,f_J)=\min_{x,y\in J}\{AN(x,y)\}$$

称为(J,f_J)在D上的丰数．如果$AN(D,J,f_J)=0$称(J,f_J)在D上无丰，f_D称为无丰填充．

不难看出，无丰渐变扩张也不一定是唯一的，但唯一渐变填充必是无丰的．

定理3. 唯一渐变填充的充要条件是，边界上的无丰最短路覆盖$3D$，即D中任一点都在J中某两点的一条无丰最短路上．

证明．必要性，设f_D是D上关于(J,f_J)的唯一渐变曲面．对任一点$p_i\in D$．我们要证明p_i必在J中两点间的一条无丰最短路上．

设$U(p_i)$为D中p_i的邻域即p_i在D中的所有邻点的集合．设$f_D(p_i)=A_i$，如$p\in U(p_i)$，则$f_D(p)\in\{A_{i-1},A_i,A_{i+1}\}$．又由$f_D(p_i)$的唯一性，得出$U(p_i)$中必有两点$p$和$q$使$f_D(p)=A_{i-1}$及$f_D(q)=A_{i+1}$．不妨令$p_{i-1}=p$，$p_{i+1}=q$．同理$U(p_{i-1})$中必有一点$p_{i-2}$使$f_D(p_{i-2})=A_{i-2}$及$U(p_{i+1})$中必有点$p_{i+2}$使$f_D(p_{i+2})=A_{i+2},\cdots\cdots$．这一过程必然终止在边界$J$上．这样我们就得到一个序列$p_\xi,\cdots,p_i,\cdots,p_\eta$且$f_D(p_t)=A_t$，$t=\xi,\cdots,i,\cdots,\eta$；此时$p_\xi$，$p_\eta\in J$．可见路径$p_\xi\cdots p_i\cdots p_\eta$的长为$|\eta-\xi|$，而$LF(p_\xi,p_\eta)=|\eta-\xi|$．这说明$p_\xi\cdots p_i\cdots p_\eta$必为$p_\xi$和$p_\eta$之间的最短路，否则$d(p_\xi,p_\eta)<|\eta-\xi|$即是$d(p_\xi,p_\eta)<LF(p_\xi,p_\eta)$，这与存在渐变曲面的假设相矛盾．因此$p_i$在无丰最短路$p_\xi\cdots p_i\cdots p_\eta$上，其中$p_\xi,p_\eta\in J$．

充分性是显然的，证毕．

推论1. 存在一$O(|J||D|^2)$时间的算法去判定D上是否存在唯一渐变曲面．

根据下面的算法B，推论1是不难证明的．

算法B： 唯一渐变曲面判定算法

输入：同算法A．

输出：回答是否存在渐变填充，渐变曲面是否唯一．

过程B

步骤1：用Dijkstra算法[4]对每个$p\in J$，计算p到D中所有元素的距离$d(p,x)$，$p\in J\&x\in D$．

步骤2：检验J中每两元p,q是否有

$$d(p,q)\geqslant LF(p,q),\quad p,q\in J$$

如果都有上式，回答D上有渐变填充，转步骤3，否则无渐变填充、停机．

步骤 3：对 $D-J$ 中的每个元素 x，如果都有

$$\exists p,q\in J(d(p,x)+d(x,q)=LF(p,q))$$

则 D 上的渐变曲面是唯一的．否则渐变曲面不是唯一的，停机．

定义 3. 设有区域 D 和它的边界 J，如果存在一渐变的 f_J，使 f_J 有唯一的渐变扩张 f_D，称 D 是正则填充集．

并非所有的区域 D 都是 E 则填充集，如图 1 所示的区域就不是 E 则填充集(不考虑平凡情形)．我们容易知道 E 则填充集的连通子集亦是 E 则的，两个 E 则填充集的交、差都是 E 则的．但目前尚不知道二个 E 则填充集的并是否 E 则．

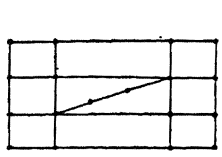


图 1

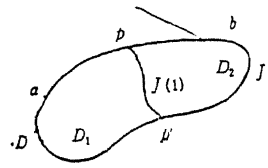


图 2

对于确定的 D,J 和 f_J，有多少渐变扩张 f_D，这就是渐变曲面计数问题，渐变曲面的个数我们用 $\#\{f_D\}$ 表示．一般来说半数 $AN(D,J,f_J)$ 越大则 $\#\{f_D\}$ 越大．目前我们尚不能证明这个计数问题是 NP-困难的，还是 $\#P$-完全的[5]．但由于 $D-J$ 中的每个点的值在填充时至多只有 3 种选择．于是

引理 1. $\#\{f_D\}\leqslant 3^{|D|-|J|}$，对确定的 (J,f_J)

最简单的计数方法是边构造曲面边作计数，这显然不是有效算法．如果 D 有 Jordan 性质(见文献[3]和本文第四节)计数的难度要小一点．例如 D 是平面网格 Σ_2 的子集，见图 2，设 $\{f_{J(1)}\}$ 为 p 与 p' 间的简单路径 $J_{(1)}$ 上渐变曲线的集合，则关于 (J,f_J) 的渐变曲面计数 $\#\{f_D,(J,f_J)\}$ 可表为：

$$\#\{f_D,(J,f_J)\}=\sum_{f\in\{f_{J(1)}\}}\#\{f_{D_1},(J_a,f_{J_a})\}\times\#\{f_{D_2},(J_b,f_{J_b})\}$$

这里 J_a 为 D_1 的边界，J_b 为 D_2 的边界．

关于渐变曲线的计数，如 $\#\{f_{J(1)}\}$ 我们可以得到确定的结果．设

$$\begin{aligned}\sigma(m,n,k)&=\sigma(d(p,p'),LF(p,p'),d(p,p')-LF(p,p'))\\&=\#\{f_{J(1)}\}\end{aligned}$$

可得

$$\sigma(m,n,k)=\begin{cases}0 & k<0\\ 1 & k=0\\ \sigma(m-1,n,k-1)+\sigma(m-1,n-1,k)+\sigma(m-1,n+1,k-2) & k>0\end{cases}$$

尽管渐变曲面的计数是困难的，但是我们可以得到一个有趣的性质：所有渐变曲面均含于某两个渐变曲面所包围的区域之中．由于本文篇幅所限，我们在“渐变曲面的唯一性和渐变曲面的包络”一文中有详细证明．

四、Jordan 集上的渐变填充、凸集和单纯剖分

通用渐变填充算法 A 是笨拙的，因为去确定一个未赋值点的值要考虑所有已赋值点．我们自然会想到，能否把大区域上的渐变填充划分为在小区域上进行．在〔3〕中，我们曾简要证明了可作划分(渐变)填充的充要条件是 D 有 Jordan 性质，即 D 上任一条简单闭曲线 C 把 $D-C$ 分割为若干个互不连通的部分． 下面我们给出适用于 Jordan 数字流形的两种快速渐变填充算法——紧缩算法和分裂算法．

算法 C：紧缩渐变填充算法

输入：单连通的 Jordan 集 D 及 J, f_J．

输出：同算法 A

过程 C

步骤 1：$D'\leftarrow D$，$D_0\leftarrow J$，$J\leftarrow\phi$；对每个 $p\in J$ 令 $f_D(p)=f_J(p)$．

步骤 2：把 D_0 在 D 中的所有邻点存入 J，如 $J=\phi$，则停机并输出 f_D；否则取 D_0 中一点 r 和它在 J 中一邻点 x，令 $f_D(x)=f_D(r)=A_i$，并执行：

(1) 如果对所有的 $p\in M(x)\cup m(x)$，有 $d(p,x)\geqslant LF(p,x)$，x 上的值 $f_D(x)$ 不变．

(2) 如果有一 $p\in m(x)$，使 $d(x,p)<LF(x,p)$ 且对所有的 $q\in M(x)$ 有 $d(x,q)\geqslant LF(x,q)+1$，令 $f_D(x)=A_{i-1}$．

(3) 如果有一 $q\in M(x)$，使 $d(x,q)<LF(x,q)$ 且对所有的 $p\in m(x)$ 有 $d(x,q)\geqslant LF(x,q)+1$，令 $f_D(x)=A_{i+1}$．

如果上述 3 条均不满足，则 D 上无渐变的 f_D，停机．否则 $D_0\leftarrow D_0\cup\{x\}$，继续这一步骤直到 J 中所有元的值都被确定．

步骤 3：$D'\leftarrow(D'-D_0)\cup J$，$D_0\leftarrow J$，$J\leftarrow\phi$ 转步骤 2．

算法 D：分裂渐变填充算法

输入、输出同算法 C

过程 D

步骤 1：$D_0\leftarrow J$；对每个 $p\in J$ 令 $f_D(p)=f_J(p)$．

步骤 2：在 J 上取两点 p 和 p'．求出 p 和 p' 在 D 中的一条最短路 $P(p,p')=p_1p_2\cdots p_{n-1}p'$，并设 $p_1=p$；从 $i=2$ 开始，令 $x=p_i$．且 $f_D(x)=f_D(p_{i-1})=A_i$，并执行同过程 C 中步骤 2 一样的(1)、(2)和(3)；如果这三条均不满足，则 D 上无渐变的 f_D 并停机．否则 $D_0\leftarrow D_0\cup\{x\}$，$i\leftarrow i+1$ 重复这一步骤直到 $i=n-1$．

步骤 3：对 J 和 $P(p,p')$ 所划分的 D 的所有分支 $D_1,\cdots,D_k$；分别(递归)调用本算法．

图 3(a),(b) 分别给出算法 C 和算法 D 的示意

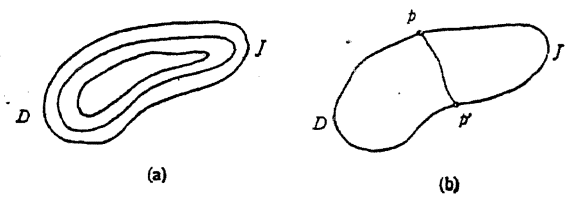


图 3

定理 4. 对离散网格点空间 Σ_n 中的 Jordan 凸集 D，分别存在 $O(|D||J|)$ 时间的紧缩渐变填充算法和分裂算法.

证明. 我们知道 Σ_n 中两点间距离的计算是常数时间的（或为 n 的线性函数）. 又因为 D 是凸集，所以距离所表示的最短路有一条一定落在 D 内. 还因为 Σ_n 中点是均匀分布的，所以边界 J 的元数不少于 J 在 D 中邻点集的元数.

在算法 C 中，确定一个点 x 上的值的时间不超过 $O(|J|)$. 所以算法 C 的计算时间为 $O((|D| - |J|) \cdot |J|)$.

在算法 D 中，确定最短路上每一点的时间亦不超过 $O(|J|)$. 而在 $D_1, \cdots, D_k$ 中如果 $J_1, \cdots, J_k$ 分别是它们的边界，则 $|J_i| \leqslant |J|$，$i = 1, \cdots, k$，且 D_i 亦是凸集. 所以算法 D 的计算时间不超过 $O(|D| \cdot |J|)$. 证毕.

对任意的 Jordan 流形 D，由于最短路计算可能很耗时间，所以算法 C、算法 D 对一般的 D 其时间复杂性仍为 $O(|D|^3)$. 如果 $D \subseteq \Sigma_n$，根据几何拓扑的知识[6]，可以说我们所能遇到的数字流形都可以作单纯（凸集）剖分. 我们不难证明

引理 2. 设 D 是 Σ_n 中的 2 维或 3 维 Jordan 数字流形，且其边界 J 是单连通的. 则存在 $O(|J|^2 \log_2 |J| + |D|)$ 时间的算法去找一组极大单纯剖分.

综合算法 C 和 D 即可得：

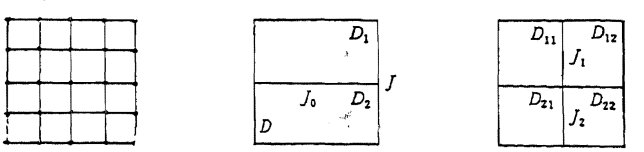
推论 2. 设 D 是 Σ_n 中的 2 维或 3 维 Jordan 数字流形，且 J 是单连通的. 则存在 $O(D|J| + |J|^2 \log_2 |J|)$ 时间的渐变填充算法.

五、方型域、圆型域和三角域上的渐变填充

我们均采用分裂渐变填充算法.

1. 方形域

D 为 Σ_2 上的正方形区域且为 4-邻接的. 因此 D 是 Jordan 凸集. 它在图象处理和识别中常常出现，见图 4(a) D 中的点可表为一 $n \times n$ 的矩阵 $M = \{a_{ij}\}_{n \times n}$，$a_{ij}$ 和 $a_{i'j'}$ 的距离为 $|i - i'| + |j - j'|$，及确定其最短路的时间为 $O(|i - i'| + |j - j'|)$.



(a)　(b)　(c)

图 4

设 D 的边长为 2^k，即 $n = 2^k + 1$；首先把 D 等分为二，见图 4(b)，即确定 J_0 中各点的赋值，需时间 $(4n - 4)(n - 2)$. 然后确定 D_1 中 J_1 上点的赋值及 D_2 中 J_2 上点的赋值，见图 4(c)，需时间 $2 \cdot (3n - 3)(n - 3)/2$. 设 $T_1(n) = T_1(2^k + 1)$ 为方型域 D 的填充时间，则

$$T_1(n) = 4T_1\left(\frac{n+1}{2}\right) + 4(n-1)(n-2) + 3(n-1)(n-3)$$

六、讨　论

我们认为有两个方面的问题值得考虑，其一是渐变填充的时间复杂性下界，如是否为 $O(|D|\log_2|D|)$？其二作为一种离散插值方法的渐变填充与其它连续型插值方法的关系．我们知道渐变填充可以在 λ-连通意义上加以推广，而后者是适用于连续空间的[7,8]。

参　考　文　献

[1] T. Pavlidis, Algorithms for Graphics and Image Processing, Computer Science Press, Inc., 1982.

[2] G. Y. Tang, Region filling with the use of the discrete Green theorem, *CVGIP*, 42: 3(1988), 279—305.

[3] 陈　澡，渐变填充的充要条件和有效算法，科学通报，**34**: 20(1989)1622—1624.

[4] A. V. Aho, J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Addison-Weslen, 1982.

[5] M. R. Garey and D. S. Johnson, Computers and Intractability, W. H. Freeman, San Francisco, 1979.

[6] E. E. Moise, Geometric Topology in Dimension 2 and 3, Springerverlag, 1977.

[7] 陈　澡，智能数据拟合、模式分类和 λ-连通分割，中国人工智能学会第三届年会文集，武汉大学出版社，1989，165—167.

[8] P. J. Besl and R. C. Jain, Segmentation through variable-order surface fitting, *IEEE Tran. PAMI*. 10: 2(1988), 167—192.

第六届全国计算机辅助设计及图形学学术会议简报

由中国计算机学会计算机辅助设计及图形学专业学会、江西省计算机学会、江西省航空学会联合主办的第六届全国计算机辅助设计及图形学学术会议于 1990 年 10 月 6 日—10 月 8 日在江西省南昌市南昌飞机制造公司召开．

在这次学术会议期间共举行了一次全体会议，20 次分组会议进行学术交流，南昌飞机制造公司总工程师张彤、中科院计算所计算机辅助设计开放实验室主任林宗楷研究员分别作了题为“重应用、讲实效，在新机研制中积极推进 CAD/CAM 技术”及“九十年代 CAD 技术展望”的学术报告．中科院计算所研究员刘慎权、中科院软件所副研究员吴恩华分别介绍了 1990 年 9 月欧洲图形学会议及 1990 年 8 月美国图形学会议的情况，均引起了与会代表的较大兴趣．

在本次会议上共宣读论文 145 篇，其中在分组会议上宣读了 141 篇．

会议期间还观看了太极计算机公司所作的新产品演示；听取了 IBM CADAM 公司所作的 R6000 工程工作站上的机械 CAD 软件的介绍；观看了南昌飞机制造公司关于 CAD/CAM 应用的录像片以及中科院 CAD 开放实验室及综考会的三维地貌辅助设计的录像片，使到会代表对我国 CAD 技术的发展及国内外动向有了进一步的了解．

与会代表认为清华大学计算机系特别是唐泽圣教授，对本次会议进行了卓有成效的准备工作，印刷出版的论文集质量较高，南昌飞机制造公司领导对此次会议给予高度重视和大力支持，该公司计算中心承担会务工作的同志们特别是主任刘炎同志付出了辛勤的劳动，给会议提供了良好的环境，为会议的顺利进行创造了条件．代表们对他们表示衷心的感谢！

此次会议还得到了南昌飞机制造公司科学技术协会、太极计算机公司、联想计算机集团公司、江西省航空学会及 IBM CADAM 公司的资助，与会代表对此表示感谢！

会议决定第七届全国计算机辅助设计及图形学学术会议于 1992 年四季度在江苏省无锡市召开，由江南计算所负责筹备，会议欢迎全国的计算机辅助设计及图形学工作者带着更高水平的研究论文，更广泛的应用成果参加下一届学术会议．

（真鸣）

λ-连通分割和最优区域分并分割算法

陈　　澍

（武汉大学计算机科学系，武汉 430072）

THE λ-CONNECTED SEGMENTATION ALGORITHM AND THE OPTIMAL ALGORITHM FOR SPLIT-AND-MERGE SEGMENTATION

Chen Li

(Wuhan University, Wuhan 430072)

Abstract: An $O(m|\Sigma_m|)$ time λ-connected segmentation algorithm is given, here m is the dimension of the discrete space Σ_m. The Length k-limited λ-segmentation is developed and it is proved that, if the connected quantity of the image in Σ_m is not greater than $1/2|\Sigma_m|$ then $O((m-1)\ln n) < k \leqslant O((m+1)\ln n)$. An $O(|\Sigma_m| \log_2|\Sigma_m|)$ time algorithm for split-and-merge (quadtree) image segmentation is presented. These two algorithms are compared in theory and applications.

Key Words: λ-connected, split-and-merge, segmentation, algorithm.

摘　要： 本文提出适用于多维灰度图象的 λ-连通分割算法，其计算时间为 $O(m|\Sigma_m|)$；这里 m 为空间 Σ_m 的维数。我们对 λ-连通分割作了误差分析，并利用长度 k-局部受限的概念，证明当图象在 Σ_m 中的连通量不大于 $\frac{1}{2}|\Sigma_m|$ 时，k 必须大于 $O(m-1)\ln n)$ 且几乎不需要超过 $O((m+1)\ln n)$.

我们改进了经典的区域分并（四叉树）分割方法，得到其时间复杂性为 $O(|\Sigma_m| \cdot \log_2|\Sigma_m|)$ 的算法，并从理论和应用两方面对这两种方法作了比较。

关键词： λ-连通，分并技术，分割，算法.

一、引　　言

分割是在一幅图像上辨别出观察者看上去均匀的各个区域，它是图像处理和识别中

本文得到中国科学院青年奖励研究基金资助，于 1989 年 4 月 25 日收到. 作者介绍见本刊第 3 期.

的一类重要方法[1,2]. 最近，Wilson 和 Spann (1988)有专著论述这一专题[3]. 常见的分割方法有：阈值分割、聚类分割、区域生长分割和边缘检测与跟踪. 其中区域生长分割，特别是四叉树区域生长分割一直为人们所重视[4,5,6].

区域生长分割的思想是 Grimsdale 提出的[7]，它是在区域中寻求一点然后由这一点根据均匀性的准则生长扩充. 四叉树分割是使用二维分裂与合并技术去扩充区域；在高维时，这种方法称为区域分并分割，它被认为是一种高级分割方法[8].

本文根据 1984 年我们提出的 λ-连通概念[9,10]，证明了 λ-连通不仅可以作为一种分类方法，而且可以是一种分割方法，称之为 λ-连通分割. 我们设计了一个最优算法去实现 λ-连通分割. 最近，我们还扩充了 λ-连通的内涵，讨论它与智能拟合的关系，并证明任何一种模式分类都等价于某度量空间上的 λ-连通分割[11].

本文改进了经典的分并分割算法，证明了改进后算法的计算时间为 $O(|\Sigma_m|\log_2|\Sigma_m|)$，这一结果达到了最优. 最后，我们比较了 λ-连通分割和区域分并分割，并综合这两种分割简述了正则 λ-连通分并分割.

二、λ-连通分割与受限 λ-连通分割

我们先描述阈值分割，不妨以二值图像为例. 设 Σ_m 为 m 维网格空间，$S:\Sigma_m \to \{0,1\}$ 是 Σ_m 上的二值图像. 如果 x，x' 是 Σ_m 中的两邻点，当 $S(x)=S(x')$ 时，称 $P_x=(x, S(x))$ 和 $P_{x'}=(x', S(x'))$ 为连通. 根据 Tarjan 的先深(宽)查找算法[1]，阈值分割算法如下.

算法 A:

Step 1 对 Σ_m 中任一点 x，在 x 上打 * 标记，即 $L(x)\leftarrow *$. 顺序检查 x 的各邻点，如果有 x'，使 $|S(x)-S(x')|=0$，且 $L(x')\neq *$、$**$，则 $x\leftarrow x'$. 重复 Step 1.

Step 2 如果关于 x 点没有上述的邻点 x' 存在，令 $L(x)\leftarrow **$，找到进入 x 的点 y，退回 y，即 $x\leftarrow y$，返回 Step1.

显然，打上标记"$**$"的点构成图像 S 的一部分. 如对算法 A 稍作修改就可以找到 S 的所有部分.

设 S 是 Σ_m 上的灰度图像，$S:\Sigma_m \to [0,1]$；如仍然使用算法 A，那么划分出的连通块有时会很小，而且不能抗干扰. 在文献[9,10]中，我们对两邻点 x，y 定义象点 p_x 和 p_y 的连通度为

$$C(p_x,p_y)=\begin{cases}1-\dfrac{|S(x)-S(y)|}{|S(x)+S(y)|} & S(x)+S(y)\neq 0\\ 1 & \text{否则}\end{cases} \tag{1}$$

设 π 是一条长度为 k 的路径，即 $\pi=\pi(x_0,x_k)=x_0x_1\cdots x_k$，其中 x_i,x_{i+1} 是邻接的. 在大多情况下要求 π 是简单路径. 称

$$CP(\pi)=\min\{C(p_{x_i},p_{x_{i+1}})\mid i=0,\cdots,k-1\} \tag{2}$$

为路径 π 上的连通量.

S 上任两点 p，q 的连通度定义为

$$C(p,q)=\max\{CP(\pi)\mid \pi=\pi(x,y)\text{ 为 }\Sigma_m\text{ 中的路径，且 }p=(x,S(x))\text{ 及 }q=(y,S(y))\} \tag{3}$$

定义 1. 称 p, q 为 λ-连通的，如果 $C(p,q)\geqslant\lambda$, $\lambda\in[0,1]$.

容易证明，由(1)，(2)和(3)确定的连通度在 λ-连通的意义下构成一等价关系，因而可以用于分类. 然而，使 λ-连通构成等价关系的连通度函数是很多的，它只需满足以下三个条件即可：(i) $C(p,q)\leqslant c$, c 为一常数；(ii) $C(p,q)=C(q,p)$；(iii) $C(p,q)\geqslant\min\{C(p,r),C(r,q)\}$.

本文依然讨论由(1)、(2)和(3)确定的连通度，它已成功地用于地球物理数据处理[10].

不难看出，p 和 q 是 λ-连通的当且仅当存在一路径 $\pi(x_0,x_k)=x_0x_1\cdots x_{k-1}x_k$，这里 $p=(x_0,S(x_0))$ 和 $q=(x_k,S(x_k))$，使 $\forall i C((x_i,S(x_i)),(x_{i+1},S(x_{i+1}))\geqslant\lambda$. 这样，我们仅需稍微修改算法 A，就可获得 λ-连通分割算法了.

算法 A′：

Step 1　对 Σ_m 中任一点 x，在 x 上打上＊标记，即 $L(x)\leftarrow *$，顺序检查 x 的各邻点. 如果有 x'，使 $C((x,S(x)),(x',S(x')))\geqslant\lambda$ 且 $L(x')\neq *$，＊＊；则 $x\leftarrow x'$. 重复 Step 1.

Step 2　如果对点 x 没有上述的点 x' 存在，那么令 $L(p)\leftarrow **$，并找到进入 x 的点 y，退回 y 即 $x\leftarrow y$. 返回 Step 1.

打上＊＊标记的就是一个 λ-连通部分. 由于 Σ_m 中的一点至多有 $2m$ 个直接邻点，或至多有 3^m-1 个任意邻点(在 Σ_2 中为 4-邻接和 8-邻接). 那么有

引理 1. 在一个 $O(m|\Sigma_m|)$ 时间的先深(宽)查找算法完成 λ-连通分割. 如果邻点定义是非直接的，该算法的计算时间为 $O(3^m|\Sigma_m|)$.

后文主要讨论直接邻接，任意邻接可类似考虑.

引理 2. 设 p 和 q 是通过长为 k 的路径 λ-连通的，那么 $S(x_p)/S(x_q)$ 最大可达 $\left(\frac{2-\lambda}{\lambda}\right)^k$. 这里 $p=(x_p,S(x_p))$ 及 $q=(x_q,S(x_q))$.

证明. 设路径 $\pi=\pi(x_0,x_k)=x_0x_1\cdots x_k$ 上的连通量不小于 λ，这里 $x_0=x_p$, $x_k=x_q$. 所以

$$1-\frac{|S(x_0)-S(x_1)|}{|S(x_0)+S(x_1)|}\geqslant\lambda$$

不妨设 $S(x_0)\geqslant S(x_1)$，那么

$$S(x_0)/S(x_1)\leqslant(2-\lambda)/\lambda$$

又因为当 $S(x_0)\geqslant S(x_1)\geqslant\cdots\geqslant S(x_k)$ 时达到极大. 所以

$$\frac{S(p)}{S(q)}=\frac{S(x_0)}{S(x_k)}=\frac{S(x_0)}{S(x_1)}\cdot\frac{S(x_1)}{S(x_2)}\cdots\cdots\frac{S(x_{k-1})}{S(x_k)}\leqslant\left(\frac{2-\lambda}{\lambda}\right)^k \tag{4}$$

上不等式中“等于”是可能出现的. 证毕.

上述讨论告诉我们，尽管 λ-连通抗干扰，并有一快速算法去实现，但盲目使用将导致较大的误差. 为了保证分割的合理性，必需限制误差的范围.

定义 2. 称 λ-连通是 k 局部受限的，如果对任意长度不大于 k 的路径 $\pi(x_p,x_q)$，当 $CP(\pi(x_p,x_q))\geqslant\lambda$ 时，有 $S(x_p)/S(x_q)\leqslant 1+\varepsilon_k$, $\varepsilon_k\geqslant 0$.

引理 3. 对于 k 局部受限的 λ-连通，设 ε_k 给定，则 λ 可取的最小值为

$$\lambda_{\min} = 2/(1 + \sqrt[k]{1 + \varepsilon_k}).$$

定理 1. k 局部受限的 λ-连通是一个等价关系，这里 $\lambda \in [2/(1+\sqrt[k]{1 + \varepsilon_k}),\ 1]$.

证明 只需证明传递性. 设 p 和 q，q 和 r 分别都是 k 局部受限 λ-连通的，即存在 $\pi(x_p, x_q)$ 和 $\pi(x_q, x_r)$ 使 $CP(\pi(x_p, x_q)) \geqslant \lambda$ 及 $CP(\pi(x_q, x_r)) \geqslant \lambda$. 如果 $\pi(x_p, x_q)$ 或 $\pi(x_q, x_r)$ 的长度不大于 k，那么 $S(x_p)/S(x_q) \leqslant 1 + \varepsilon_k$ 或 $S(x_q)/S(x_r) \leqslant 1 + \varepsilon_k$.

设 k_1 为 $\pi(x_p, x_q)$ 的长，k_2 为 $\pi(x_q, x_r)$ 的长. 如果 $k_1 + k_2 > k$，则 $\pi(x_p, x_r) = x_p \cdots x_q \cdots x_r$ 不在限制范围内，因而满足条件.

如果 $k_1 + k_2 \leqslant k$. 由引理 2、3 得

$$S(x_p)/S(x_p) \leqslant \left(\frac{2-\lambda}{\lambda}\right)^{k_1}$$

及

$$S(x_q)/S(x_r) \leqslant \left(\frac{2-\lambda}{\lambda}\right)^{k_2}$$

推出

$$\begin{aligned} S(x_p)/S(x_r) &\leqslant (S(x_p)/S(x_q)) \cdot (S(x_q)/S(x_r)) \\ &\leqslant \left(\frac{2-\lambda}{\lambda}\right)^{k_1+k_2} \leqslant \left(\frac{2-\lambda}{\lambda}\right)^{k} \leqslant 1 + \varepsilon_k. \end{aligned}$$

所以 p，r 是 k 局部受限 λ-连通的. 证毕.

局部受限的 λ-连通分割算法与算法 A' 基本相同，只要先根据 k 和 ε_k 计算出 λ 的取值范围即可；其计算时间仍为 $O(m|\Sigma_m|)$.

三、局部受限的期望值

在上节中，如果 k 较小，则分割过粗误差大；如果 k 较大，则分割过细，类别过多. 于是问，k 为何值时合理？下文根据随机图论的知识来估计 k 的期望值.

设 $G_p = (\Sigma_m, E_p)$ 是随机图，$(a, b) \in E_p$ 当且仅当 (a, b) 在 Σ_m 中邻接且选择 (a,b) 的概率为 p.

引理 4. m 维超立方体中有 $2^{m-i}\binom{m}{i}$ 个 i 维超立方体.

引理 5. m 维超立方体中度为 $m + i$ 的点数为 $(n-2)^i\binom{m}{i}2^{m-i}$，其中 n 为超立方体的边点数.

上述两引理的证明从略.

定理 2. 边点数为 n 的 m 维超立方体 Σ_m 中点度数的期望值为 $\left(1 + \frac{n-2}{n}\right)m$.

证明 根据引理 5，点度数的期望值为

$$E_d=\frac{\sum_{i=0}^{m}(m+i)(n-2)^i\binom{m}{i}2^{m-i}}{n^m}=m+\frac{\sum_{i=0}^{m}i(n-2)^i\binom{m}{i}2^{m-i}}{n^m}$$

由于 $|\Sigma_m|=n^m=(n-2+2)^m=\sum_{i=0}^{m}2^{m-i}\binom{m}{i}(n-2)^i$

考虑 $(x+2)^m=\sum_{i=0}^{m}2^{m-i}\binom{m}{i}x^i$，于是

$$\frac{d(x+2)^m}{dx}=\sum_{i=0}^{m}i\cdot 2^{m-i}\binom{m}{i}x^{i-1}$$

故

$$\frac{xd(x+2)^m}{dx}\Big|_{x=n-2}=(E_d-m)n^m$$

即

$$E_d=\frac{(n-2)m(n-2+2)^{m-1}}{n^m}+m=\left(1+\frac{n-2}{n}\right)m.$$ 证毕.

设 E_k 为 $G_p=(\Sigma_m,E_p)$ 中长度为 k 的路径的个数的期望值，不难证明.

$$n^m\cdot m^k\cdot p^k\leqslant E_k<n^m\cdot E_d^k\cdot p^k \tag{5}$$

我们还看到，如果 $p\cdot E_d>1$ 即 $p>\left[\left(1+\frac{n-2}{n}\right)m\right]^{-1}$，那么 E_k 是 k 的非减函数. 这将给我们的计算带来困难. 设图象 S 在 Σ_m 中的连通量为：

$$CH(S,\Sigma_m)=\frac{1}{2}\sum_{q,r\text{相邻}}C(q,r) \tag{6}$$

假设均匀分布，即可设 $CH(S,\Sigma_m)=\frac{1}{2}E_d\cdot|\Sigma_m|\cdot p$. 如果 $CH(S,\Sigma_m)\leqslant\frac{1}{2}|\Sigma_m|$，则 $p\leqslant\left[\left(1+\frac{n-2}{n}\right)m\right]^{-1}$.

定理 3. 当 $CH(S,\Sigma_m)\leqslant\frac{1}{2}|\Sigma_m|$ 时，有

(i) 如果 $k>(m+1)\ln n/-\ln\left[\left(1+\frac{n-2}{n}\right)mp\right]$，那么几乎所有的 Σ_m 上的图像都没有长度为 k 的路径.

(ii) 如果 $k\leqslant(m-1)\ln n/-\ln(mp)$，那么几乎所有的 Σ_m 上的图像都有长度为 k 的路径.

证明　事实上定理 3 的证明等价于证明：

(i) $P(X_k>0)\to 0$ 当 $k>(m+1)\ln n/-\ln\left[\left(1+\frac{n-2}{n}\right)mp\right]$ 时，

(ii) $P(X_k>0)\to 1$，当 $k\leqslant(m-1)\ln n/-\ln(mp)$ 时；X_k 是图上长度为 k 的路径的个数.

令 $n^m\left[\left(1+\frac{n-2}{n}\right)mp\right]^d=1$，即

$$d = m\ln n/-\ln\left[\left(1+\frac{n-2}{n}\right)mp\right] \tag{7}$$

若 $k > d$，不妨设 $k \geqslant d + r$，根据(5)有

$$E_k < n^m\left[\left(1+\frac{n-2}{n}\right)mp\right]^{d+r}$$

$$< e^{r\ln\left[\left(1+\frac{n-2}{n}\right)mp\right]}$$

如果 $r = -\ln n/\ln\left[\left(1+\frac{n-2}{n}\right)mp\right]$，则

$$E_k < \frac{1}{n} \to 0 \tag{8}$$

故有 $P(X_k > 0) \to 0$，所以(i)成立。事实上只要

$$r = -\ln f(n)/\ln\left[\left(1+\frac{n-2}{n}\right)mp\right],\quad f(n) \to \infty \text{ 即可。}$$

现在我们来证明(ii)。已知

$$P(X_k = 0) \leqslant \sigma_k^2/E_k^2,\ \sigma_k^2 \text{ 为 } X_k \text{ 的方差。}$$

显然，如果 $P(X_k = 0) \to 0$，则 $P(X_k > 0) \to 1$。因为 $P(X_k \geqslant 0) = 0$，而 $P(x_k > 0) = P(X_k \geqslant 0) - P(X_k = 0)$。设 E_{k^2} 为 X_k 的二阶原点矩，则

$$\frac{\sigma_k^2}{E_k^2} = \frac{E_{k^2}}{E_k^2} - 1 \tag{9}$$

我们知道，如果 $\mathscr{H}(G)$ 是某一类 $\Psi = \{F_1, F_2, \cdots\}$ 中 G 的导出子图的数目[13]，有

$$E(\mathscr{H}^2(G)) = \sum_{(F',F'')\in\Psi} P(G \text{ 包含 } F' \text{ 和 } F'') \tag{10}$$

本文中的 E_{k^2} 意指恰有长度为 $l(l = 0, 1, \cdots, k)$ 的路径对之和，即

$$E_{k^2} \leqslant E_k \cdot E_1 + E_k\sum_{l=1}^{k}(k-l)(E_dp)^{k-l}$$

$$\leqslant E_1^2 + E_k \cdot \frac{E_dp[1-(k-1)(E_dp)^{k-1}]}{1-E_dp}$$

$$P(X_k = 0) \leqslant \frac{\sigma_k^2}{E_k^2} \leqslant \frac{E_dp[1-(k-1)(E_dp)^{k-1}]}{E_k(1-E_dp)}$$

由(5)得

$$P(X_k = 0) \leqslant \frac{E_dp}{n^m m^k p^k(1-E_dp)}$$

设 $n^m m^{d'} p^{d'} = f(n)$，$f(n) \to \infty$，有

$$d' = [m\ln n - \ln f(n)]/-\ln(mp) \tag{11}$$

当 $k \leqslant d'$ 时，$P(X_k = 0) \to 0$。特别地，当 $f(n) = n$ 时，我们有 $d' = (m-1)\ln n/-\ln(mp)$。(ii) 得证。证毕。

如果图像的连通量 $CH(S, \Sigma_m) > \frac{1}{2}|\Sigma_m|$，上述证明的某些过程不收敛，也就是 k 与 $|\Sigma_m|$ 成正比。

四、改进的二维图像四叉树分割

区域生长分割常常用平均亮度作为均匀性准则[4]. 这一准则不能保证分割结果构成数学上的分类，这样连通性算法将不能使用. 解决这个问题有两种方法，其一是针对性的给出启发函数[5]. 其二就是采用四叉树区域分割方法. 如果图像不是二维的，四叉树分割称作区域分并分割[1,3,8]. 它使用分并（divide-and-conquer）策略，以小区域代替象素去生长，主要优点是降低了对噪声的灵敏度.

设 Σ_2 的边长为 n，$n=2^l$. 对于某确定的 ε，如果区域 $A\subset\Sigma_2$ 满足：

$$\forall x\in A\left(\left|S(x)-\frac{1}{|A|}\sum_{y\in A}S(y)\right|\leqslant\varepsilon\right) \tag{12}$$

那么称 A 是均衡的，也可以把 $(A,S(A))$ 作为一连通部分或一块.

我们先从一个简单的例子来看区域生长分割. 设图 1 所示的图像 $S:\Sigma_2\to\{0,0\}$

$$(\Sigma_2,S)\triangleq \tag{13}$$

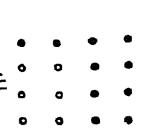


图 1

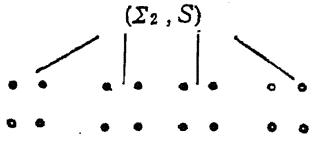


图 2

Σ_2 是非均匀的，我们分上式图 1 为四份，见图 2.
然后看其第一儿子结点见图 3.

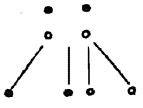


图 3

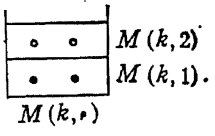


图 4

图 3 第一子均衡，进栈 $M(k,1)$，第二子进栈 $M(k+1,1)$，$\cdots$；直到四子均进栈. 然后合并，见图 4.
其余 $M(k+1,\bullet)$，$\cdots$ 清零待用. 类推，我们得到四个栈的内容见图 5.

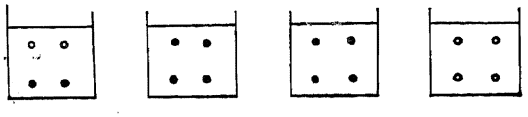


图 5

合并这四个栈（横放）见图 6.

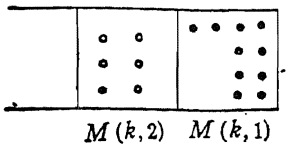


图 6

可见，一个栈是某次分解的结果．上述过程的时间复杂性为：

$$T(n)=4T\left(\frac{n}{2}\right)+c\cdot U\left(\frac{n}{2}\right) \tag{14}$$

式(14)中 $U\left(\frac{n}{2}\right)$为合并所需时间，$c$ 为常数．

合并即是在两个栈中检查有无两相邻且均衡的块．由于每个栈可能有 $\frac{n}{2}\times\frac{n}{2}$ 个元，那么邻元检查时间一般估计为 $O\left(\left(\frac{n}{2}\right)^4\right)$，由(14)得

$$T(n)=4T\left(\frac{n}{2}\right)+c\cdot n^4=O(n^4)=O(|\Sigma_2|^2) \tag{15}$$

上述分析表明减小整个算法的时间关键在减小合并时间．下文讨论这个问题．

首先对栈中内容作些加强，称被用于划分的区域为基区域，一个栈对应一个基区域．不妨用 $M(k),k=1,2,\cdots,$ 表示各个栈，用 $M(k,l),\ l=1,2,\cdots,B(k)$ 表示第 k 栈中的第 l 块，$M(k,l,i)$ 表示第 k 栈中，第 l 块的第 i 个元素，令

$$\sigma=\sum_{x\in M(k,l)}S(x)$$

$$n=|M(k,l)|$$

$$\min=\min_{x\in M(k,l)}S(x),\quad \max=\max_{x\in M(k,l)}S(x)$$

$$PR(k,l)=(\sigma,n,\min,\max)$$

显然，$\left|\min-\frac{\sigma}{n}\right|\leqslant\varepsilon$ & $\left|\max-\frac{\sigma}{n}\right|\leqslant\varepsilon$ 也是均衡的充要条件，并设

$$\begin{cases}f(k,l,i)=M(k,l,i+1)\\ g(k,l,i)=M(k,l,i-1)\\ f(k,l,n)=*\\ g(k,l,1)=*\end{cases} \tag{16}$$

而对于任一点 x，令 $k(x)=k$，$l(x)=l$ 和 $i(x)=i$ 即是 $M(k(x),l(x),i(x))=x$．

设有边长为 t 的四个待合并的基区域(如图 7)合并算法见算法 B．

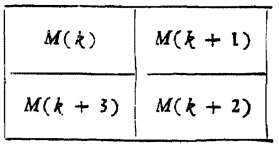


图 7

算法 B：分并分割的合并算法

```
begin
  for K: = k to k + 3 do begin
    for L: = 1 to B(K) do begin
      Star(M(K,L)): = **; i: = 1; p: = M(K,L,i)
1:    if (p 在合并的边界上) then begin
        while (p' 是 p 的外邻点) do begin
          if (Star(M(k(p'),l(p'))≒**) then
            begin
              if(|MIN(min, min') − (σ+σ')/(n+n')| ≤ε & |MAX(max,max') − (σ+σ')/(n+n')| ≤ε then
                begin
                  p 的后点 f(p) 进栈 Stack, 以便把新链插入老链
                  根据 g(p'),…,g^{n-1}(p') 找到 g^n(p') = *.
                  f(p): = g^{n-1}(p'), 即把原链解开接在新链之后
                  p: = g^{n-1}(p'); Star(M(k(p'), l(p')) = **
                  goto 1
                end;
              else
                if (f(p) = *) then
                  if (Stack = φ) goto 2
                else
                  f(p): = S+ack Top (栈顶送 f(p))
              end;
            end;
          end;
        p: = f(p)
        goto 1
2:    end;
    end;
  end
```

最后根据 f 和 g 把所有块存入一个栈.

我们看到,合并总是要通过边界的,而边界点只有 $O(t)$ 个,也就是说仅有 $O(t)$ 个部分合并在一起. 在算法 B 中区域内每点至多经过两次,而边界点也只经过常数次(邻点个数为常数). 因此

引理 6. 算法 B 的计算时间为 $O(t^2)$.

定理 4. 二维图像的区域分并分割的时间复杂性为 $O(|\Sigma_2|\log_2|\Sigma_2|)$.

证明 由引理 6 和(14)有

$$T(n) = 4T\left(\frac{n}{2}\right) + c \cdot \left(\frac{n}{2}\right)^2 = O(n^2\log_2 n)$$

$$= O(|\Sigma_2|\log_2|\Sigma_2|). \text{ 证毕.}$$

五、多维图像的区域分并生长分割

上节中的算法 B，可以推广到三维乃至多维图像. 由于维数增加，基区域和合并边界数随之增加.

定理 5. 多维灰度图像的区域分并分割的计算时间为 $O(|\Sigma_m|\log_2|\Sigma_m|)$.

略证 由于一个基区域被分开成 2^m 个基区域，这样合并边界就有 $m\cdot 2^{m-1}$ 个. 这时算法 B 的时间复杂性为 $O(m\cdot 2^{m-1}\cdot t^m)=O(m2^m t^m)$，所以

$$T(2t)=2^m T(t)+c\cdot m2^m t^m$$

$$T(n)=O(mn^m\log_2 n)=O(|\Sigma_m|\log_2|\Sigma_m|).$$

证毕.

由于分并方法导致 2^m-树有 m 层，这样所得到的 $O(|\Sigma_m|\log_2|\Sigma_m|)$ 达到了最优. 除非不用分并技术.

六、两种分割方法的比较

λ-连通分割与区域分并分割相比，具有节省时空、编程简单的优点；然而后者较前者稳定，不会出现大的误差. 在国家七·五攻关项目(75—54—02—08—12)"物探资料灰色识别方法研究"和地矿部七五项目"三维地层层位的拓扑识别"中，我们用这两种方法分别对塔里木北部某工区和川中某工区作实际处理，获得了满意的结果，正确划分出地层层位. 图 8 是用 λ-连通分割在某测线上作出的速度层位划分，图 9 是用区域分并分割的结果.

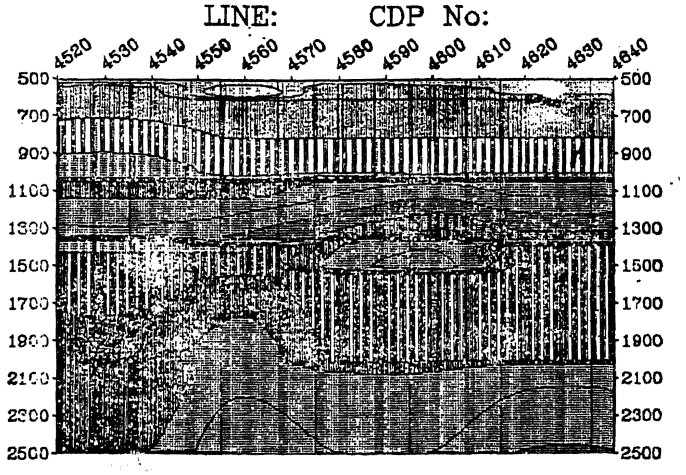


图 8

在地球物理资料处理中，我们发现区域分并分割不如 λ-连通分割适用. 原因是，地球物理数据大多呈渐变特征，λ-连通正好予以刻划，而区域分并用平均亮度准则就不合适了.

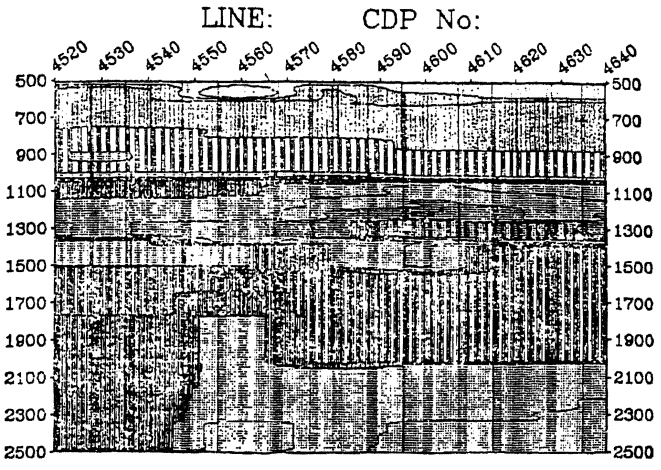


图 9

对于一集合，如果其中每一点都与其在集合中的邻点 λ-连通，这个集合叫做正则 λ-连通集，它的特例是渐变曲面或渐变流形[14,15]。正则 λ-连通不构成等价关系，用于分割只能采用分并策略。合并过程的平均亮度准则修改为邻点的 λ-连通准则。这就是四叉树正则 λ-连通分割或区域分并 λ-连通分割。

张　兰、周东兵、刘　岱、尹云峰和王长春等同志参加了这方面的工作并使之用于实际资料处理，地矿部石油物探研究所提供宝贵的资料．一并致谢．

参　考　文　献

[1] T. Pavlidis, Algorithms for Graphics and Image Processing, Computer Science Press, Inc., 1982.
[2] A. Rosenfeld, Three-dimensional digital topology, *Information and Control*, 50: 2(1981), 119—127.
[3] R. Wilson and M. Spann, Image Segmentation and Uncertainty, John Wiley & Sons Inc., 1988.
[4] S. W. Zucker, Region growing: childhood and adolescence, *Comput. Graphics Image Processing*, 5: 4 (1976), 382—399.
[5] O. Monga, A new segmentation method and its application to stereo vision, Proc. International Workshop on Industrial Application of Machine Vision and Machine Intelligence, 1987, 176—181.
[6] M. Spann and R. Wilson, A quadtree approach to image segmentation that combines statistical and spatial information, *Pattern Recognition*, 18: 3(1985), 257—259.
[7] A. Ronsenfeld and A. C. Kak, Digital Picture Processing, Academic Press, 1982.
[8] 边肇祺等，模式识别，清华大学出版社，1988.
[9] 陈　漠，三维数字拓扑与曲面识别，石油物探译丛，6(1984).
[10] 陈　漠等，三维 Fuzzy 数字拓扑及应用 (I,II,III)，石油物探，24: 2(1985)，26: 2(1987)，29: 3(1990).
[11] 陈　漠，数据智能拟合，模式分类和 λ-连通分割，中国人工智能年会文集，武汉大学出版社，1989.
[12] A. V. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.
[13] B. Bollobas, Random Graphs, Academic Press, 1985.
[14] 陈　漠，渐变填充的充要条件和有效算法，科学通报，34: 20(1989).
[15] 陈　漠，渐变填充的性质和算法，计算机学报，见本刊本期.

RANDOM GRADUALLY VARIED SURFACES FITTING*

CHEN LI (陈 澍)

(*Department of Computer Science, Wuhan University, Wuhan* 430072, *PRC*)

Received December 20, 1990; revised August 12, 1991.

Keywords: gradually varied surface, random algorithm, probability distribution.

Let D be a digital manifold, J the subset of non-empty of D; and let A_1, A_2, $\cdots$, A_m be m real numbers with $A_1<A_2<\cdots<A_m$. Supposing function $f_J: J \to A_1,\cdots, A_m$, we want to find a function $f_D: D\to\{A_1,\cdots, A_m\}$, such that f_D is gradual variation on D, having $f_D(x)=f_J(x)$, $x\in J$. Such f_D is called the gradually varied extension or interpolation of f_J.

Theorem 1. *For any two points x and y in J, if the length $d(x, y)$ of the shortest path between x and y in D is not less than the difference of gray level of the two points $LF(x, y)$ (or $d(f_J(x), f_J(y))$), then there exists a gradually varied extension f_D of f_J; and vice versa.*

Theorem 2. *If there is the gradually varied extension f_D of f_J, then there exist two gradually varied extensions f^1_D and f^2_D of f_J, such that for every gradually varied extension f_D, $f^1_D(x)\leqslant f_D(x)\leqslant f^2_D(x)$, $x\in X$.*

We gave an algorithm to compute f_D, but it made f_D in the convex closure of its guiding points. In case the holding convexity is not necessary, we should find a random gradually varied surface which can be regarded as interpolating surface between f^1_D and f^2_D.

Algorithm A (Random algorithm for f_D).

Input: D, J and f_J.

Output: gradually varied f_D or indicating no gradually varied extension for f_J.

begin

1. for (every pair p, p' in D) do
   to compute $d(p, p')$ by Floyd's algorithm;
2. for (every pair p, p' in D) do
   if ($d(p, p')<LF(p, p')$) then (no f_D and helt);
3. $D_0:=J$;
4. while ($r\in D_0$) do ($f_D(r):=f_J(r)$);
5. using depth-first-searching algorithm, to connect all points in D except D_0, denoted by H;

* Project supported by the Science Fund of the Chinese Academy of Sciences for Young Scientists.

6. if ($D_0=D$) then (output f_D and helt) else
begin
choose x that is the endpoint of H and choose $r\in D_0$, $d(r, x)=1$,
without loss of generality assume that $f_D(r)=A_i$,
$v(1):=A_{i-1}$, $v(2):=A_i$, $v(3):=A_{i+1}$;
for $t:=1$ to 3 do
7. begin
while ($p\in D_0$) do if ($d(x, p)<d(v(t), f_D(p))$) goto 8,
$v(t):=1$;
8. end
$R:=\{t|v(t)=1\}$
random choose $j\in R$
$j:=i+j-2$, $f_D(x):=A_j$
$H:=H-\{x\}$
$D_0:=D_0\bigcup\{x\}$
goto 6
end
end

Lemma 1. *The computing time of Algorithm* A *is* $O(|D|^3)$.

There is a defect in Algorithm A, i.e. its valuation procedure has an order which was determined beforehand; consequently the randomness of the surface is restricted. The following algorithm A′ such that $x\in D$ is also random.

Algorithm A′

Input, Output and Step 1 until Step 4 are similar to those in Algorithm A.

begin
.....
5. random choose x in $D-D_0$;
6. for $t:=1$ to m do begin
while ($p\in D_0$) do if ($d(x, p)<d(A_i, f_D(p))$) goto 8
$v(t):=1$;
8. end
$R:=\{t|v(t)=1\}$
random choose $j\in R$
$f_D(x):=A_j$
$D_0:=D_0\bigcup\{x\}$
goto 5
end

Theorem 3. *f_D is gradually varied which is obtained by Algorithm* A′.

Proof. We need only to prove that if there is a gradually varied extension for f_J, then output f_D of Algorithm A′ is gradual variation. Suppose there is a gradually varied extension. Therefore $\forall x, y \in J$, $d(x, y) \geqslant LF(x, y)$. In the following, we shall prove that an integer t exists such that $v(t) = 1$ at Step 6 of Algorithm A′.

On the contrary, let such t do not exist, i. e.

$$\forall\ t \in \{1, \cdots, m\},\quad \exists\ y \in D_0,\ d(x, y) < d(A_t, f_D(y)). \tag{1}$$

However, according to Theorem 1, if

$$\forall\ p, q \in D_0,\ d(p, q) \geqslant d(f(p),\ f(q)), \tag{2}$$

then there exists an extension $f: D \to \{A_1, \cdots, A_m\}$, such that

$$\forall\ p, q \in D,\ d(p, q) \geqslant d(f(p),\ f(q)). \tag{3}$$

which is in contradiction with (1). Thus, such t exists, and it is easy to see that f_D is gradual variation on D.

Only $|D|$ elements in $A_1, \cdots, A_m$ are efficient if $m > |D|$. Therefore, we have

Lemma 2. *The computing time of Algorithm* A′ *is* $O(|D|^3)$.

Let Σ_n be an n-dimensional grid space, and D the connected subset of Σ_n. If D is the convex set, then it is not necessary that the length of the shortest path is computed by Floyd's algorithm, and Algorithms A and A′ are simplified relevantly.

Theorem 4. *Let D be the convex set in Σ_n. Algorithm* A *needs* $O(|D|^2)$ *time, but Algorithm* A′ *requires still* $O(|D|^3)$ *time.*

If Σ_2 is 4-adjacent, i. e. $D \subseteq \Sigma_2$ has Jordan property, then much faster algorithm can be designed.

Corollary 1. *Let D be the Jordan square in Σ_2. Then there exists an* $O((|D| - |J|)\log_2(|D| - |J|) + |J|\sqrt{|D| - |J|}\ \log_2(|D| - |J|))$ *time improved Algorithm* A.

Given the distribution of f_D in $\{A_1, \cdots, A_m\}$, let $\xi(f_D, A_i) = \xi_i$, $\sum_{i=1}^{m} \xi_i \neq |D|$ and $P(A_i) = \xi_i / |D|$. For convenience, we say $\{\xi_i\}$ is the distribution. Set $N(S) = \{p: p \in D - S$ and p is adjacent to some points of $S\}$, and then $N(S)$ is called the adjacent point set of S.

Lemma 3. *For any S with $|S| = \{\xi_i\}$, if $N(S) > \xi_{i-1} + \xi_{i+1}$, then there are no gradually varied surfaces with distribution $\{\xi_i\}$.*

Not considering gradual variation, there are valuations of $|D|! / \prod_{i=1}^{m} \xi_i!$ kinds on D with distribution $\{\xi_i\}$. That is to say, it is impossible to construct gradually varied surfaces by one-by-one testing method.

Theorem 5. *There is no algorithm with polynomial-time to construct a gradually varied*

surface whose distribution was given beforehand unless $P=NP$.

Proof. We need only to transform an NP-complete problem to our problem in polynomial time[2]. PARTITION problem is selected: Let $R_1,\cdots, R_n$ be n sets, and $N=\{1, \cdots, n\}$. Is there a subset $N'\subset N$ such that $\sum_{i\in N'}|R_i|=\sum_{i\in N-N'}|R_i|$?

For any $i\in N$, connect any two points in R_i, where R_i and R_j are not adjacent, $i\neq j$. Suppose a point $r\notin\bigcup_{i=1}^{n}R_i$, such that r is connected with every point in $\bigcup_{i=1}^{n}R_i$. Thus, we obtain a special digital manifold. Let $\{A_i\}=\{A_1, A_2, A_3\}$ and $\{\xi_i\}=\left\{\sum_{i=1}^{n}|R_i|/2, 1, \sum_{i=1}^{n}|R_i|/2\right\}$. Obviously, the valuation of r is only A_2, or else f_D is not gradually varied or does not satisfy distribution $\{\xi_i\}$. The above procedure can be run in polynomial time.

Thus, there exists a gradually varied surface with distribution $\{\xi_i\}$ on $D=\{r\}\bigcup\bigcup_{i=1}^{n}R_i$ if and only if there is $N'\subseteq N$ such that $\sum_{i\in N'}|R_i|=\sum_{i\in N-N'}|R_i|$. That is to say, we turn the PARTITION problem to construction (or decision) problem of a gradually varied surface whose distribution was given by polynomial transformations.

According to Theorem 5, we only obtain the gradually varied surface approximating the distribution. However, in some simple case, it is possible to construct a gradually varied surface with distribution $\{\xi_i\}$ such as

Lemma 4. *For a rectangle grid D in plane, if $\xi_1<\cdots<\xi_m$ or $\xi_1>\xi_2>\cdots\xi_m$, there exists a gradually varied surface with distribution $\{\xi_i\}$.*

Let D_0 be the simply connected subset of D and $f(D_0)$ be single valued, such as A_i. If $N(D_0)$ and the boundary ∂D of D do not intersect, D_0 is called the extreme region.

Lemma 5. *For a Jordan digital manifold D in which the number of adjacent points of every point is nearly identical, if S is a connected subset with $|S|=n$ such that $|N(S)|$ is minimal, then for any S_1, $S_2\subseteq D$ and $|S_1|+|S_2|=n$, it holds that*

$$|N(S)|\leqslant|N(S_1)|+|N(S_2)|.$$

By Lemma 5, we approach an algorithm for the gradually varied surface with one extreme region whose distribution was given as follows.

Algorithm B (The approached algorithm for the gradually varied surface with one extreme region).

Input: Jordan set D, $\{A_i\}$ and $\{\xi_i\}$; we may suppose $\xi_1\leqslant\xi_m$.

Output: surface f satisfying distribution $\{\xi_i\}$.

```
begin
  S: =∅
  for i: =1 to m do begin
    for j: =1 to ξ(i) do begin
      choose x, S: =S⋃{x} such that |N(S)|=min
      f(x): =A_i
    end
  end
end.
```

Algorithm B is in $O(|D|^3)$ time, but f may be not gradually varied unless D is a closed manifold. The following technique can be used for any surface f, to compute the gradually varied surface g which approximates to f uniformly.

Let $F_i(p)=\{f(p)+t|t=0, \pm 1, \cdots, \pm i\}$, here might as well let $A_i=i$, $p\in D$; and let

$$F_i(p)/F_i(q)=\{x|x\in F_i(p), \text{ there is } y\in F_i(q), \text{ such that } d(p,q)\geqslant d(x,y)\}, \quad (4)$$

and

$$\begin{cases} F_i^{(k)}(p)=\bigcap\limits_{q\in J} F_i^{(k-1)}(p)/F_i^{(k-1)}(q), \\ F_i^{(0)}(p)=F_i(p). \end{cases} \quad (5)$$

We can prove that if

$$\exists\, k, \forall p\in D, F_i^{(k)}(p)=F_i^{(k+1)}(p)\neq\emptyset. \quad (6)$$

then for every $p\in D$, setting $g(p)=\inf\{F_i^{(k)}(p)\}$, we know that g is gradual variation and $\forall p\in D$, $d(f(p), g(p))\leqslant i$. Consequently, let $i=1, 2, \cdots$ until (6) is satisfied. Then we can obtain the optimal uniformly approximation g of f. This method has been used for stratum surface fitting[3].

The author is grateful to Prof. Y. T. Zhang for several results which were jointly attained by them.

References

[1] Chen Li. *Chinese Science Bulletin*, 35(1990), 10: 870.

[2] Graey, M. R. & Johnson, D. S., *Computers and Intractability*, W. H. Freeman and Company, 1979.

[3] Chen Li et al., *Annals of the Geophysics Society of China* (in Chinese), 1990, Seism Publishing House, 1990, pp. 42—43.

Gradually Varied Surface and Its Optimal Uniform Approximation

Li Chen
Department of Computer Science
Utah State University
Logan, UT 84322-4205
and
Department of Computer Science
Wuhan University
Wuhan 730072, China

ABSTRACT

A new digital surface called the gradually varied surface is introduced and studied in digital spaces, especially in digital manifolds. Generally, a digital surface is a mapping from a n dimensional digital manifold D into a m dimensional grid space Σ_m. In this paper, we have proved a constructive theorem: Let i_Σ_m be an indirectly adjacent grid space. Given a subset J of D and a mapping $f_J : J \to i_\Sigma_m$, if the distance of any two points p and q in J is not less than the distance of $f_J(p)$ and $f_J(q)$ in i_Σ_m, then there exists an extension mapping f of f_J, such that the distance of any two points p and q in D is not less than the distance of $f(p)$ and $f(q)$ in i_Σ_m, we call such f a gradually varied surface. In other word, any digital manifold (graph) can normally immerse an arbitrary i_Σ_m. We also show that any digital manifold (graph) can normally immerse an arbitrary tree T. Furthermore, we will discuss the gradually varied function, which is a gradually varied surface in the case of $m = 1$ in i_Σ_m. An envelop theorem, A uniqueness theorem, and an extension theorem which concerns with having the same norm are obtained. Finally, we will show an optimal uniform approximation theorem of gradually varied functionals and develop an efficient algorithm for the approximation.

1. BASIC DEFINITIONS

A digital manifold means the discretization of a mathematical manifold, such as the set Σ_m of all integer coordinate points of a m dimensional Euclid space or a simplicial subdivision Δ_n of a n dimensional manifold [1-4]. Generally, a digital manifold is a graph with geometric meaning [5].

Definition 1.1 Let D_1 and D_2 be two digital manifolds. If a mapping $f : D_1 \to D_2$ satisfies: x and y are adjacent in D_1, implies $f(x) = f(y)$ or $f(x)$ and $f(y)$ are adjacent in D_2, then f is an immersion from D_1 to D_2 or a gradually varied operator. If $D_2 = \Sigma_m$, then f is called a gradually varied surface.

Lemma 1.1 If f is a gradually varied mapping, then

$$\forall p, q \in D_1[d(p, q) \geq d(f(p), f(q))], \tag{1.1}$$

where $d(x, y)$ is the length of the shortest path between x and y.

Definition 1.2 Let J be a subset of D_1 and f be a mapping $f_J :\to D_2$, which satisfies:

$$\forall p, q \in J[d(p, q) \geq d(f_J(p), f_J(q))]. \tag{1.2}$$

If there exists an extension f of f_J such that $f : D_1 \to D_2$ is a gradually varied mapping, then we say (J, f) is immersion-extendable. If every (J, f) satisfying (1.2) is immersion-extendable, then D_1 can normally immerse to

0-8194-1477-8/94/$6.00

D_2.

Mathematically, it might be very difficult to decide whether or not a general graph/digital manifold can normally immerse to another one. This problem may be NP-hard [6]. However, in the use of data fitting, pattern recognition and computer vision, we can find several important and useful digital manifolds in which we can discuss normal immersion and generate gradually varied surfaces based on its guiding points.

It is well-known, a discrete/digital space is a graph. There are two kinds of the most important digital spaces which we show below.

(1) The m dimensional grid space Σ_m. It is the set of all points in n-dimensional Euclidean space with integer coordinates. There are many kinds of adjacency, and we just state two of them, which are mostly often used in discrete geometry [7]. First, two points $p = (p_1, ..., p_m)$ and $q = (q_1, ..., q_m)$ are directly adjacent if and only if $\sum_{i=1}^{m}(|p_i - q_i|) = 1$. Such a digital space is called a directly adjacent space denoted by d_Σ_m. Second, two points $p = (p_1, ..., p_m)$ and $q = (q_1, ..., q_m)$ are indirectly adjacent if and only if $Max_{i=1}^{m}(|p_i - q_i|) = 1$. Such a digital space is called a indirectly adjacent space denoted by i_Σ_m.

(2) n-simplicial subdivision space Δ_n. A Δ_3 is shown in Figure 1.1 .

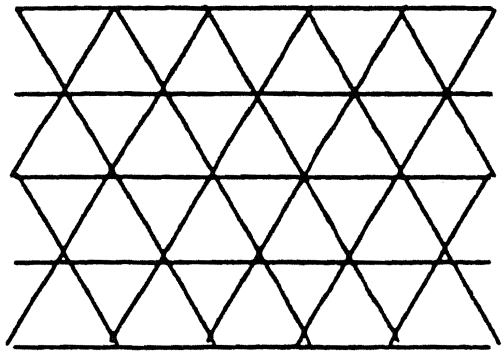


Figure 1.1 A n-simplicial Subdivision Space

2. GRADUALLY VARIED EXTENSION AND NORMAL IMMERSION

In this section, we will show two basic theorems in this paper, which can be used for interpolation problems. These two theorems are also important for discussing gradually varied functionals which will be described in the following sections. In previous papers [1-4], we showed a gradually varied extension theorem as follows:

Theorem 2.1 Let J be a subset of a digital manifold D, and a mapping $f_J :\rightarrow \Sigma_1$ satisfies (1.2), where Σ_1 can be viewed as $N = \{0, 1, 2, 3, ...\}$, a chain, or a totally ordered set. Hence, There exists a gradually varied extension f of f_J on D.

Theorem 2.1, in other word, states that any digital manifold can normally immerse to Σ_1 or a chain. We can show a more general theorem in the following.

Theorem 2.2 Any graph D (or digital manifold) can normally immerse an arbitrary tree T.

Proof Suppose that $J \neq \emptyset$ is a subgraph of D, and a mapping $f_J : J \rightarrow T$ satisfies (1.2), where T is a tree. We will prove that there exists a gradually varied mapping $f : D \rightarrow T$ with $f_J(p) = f(p)$, $p \in J$.

The proof of the theorem is a constructive proof. For every $p \in J$, let $f(p) = f_J(p)$, and let $f(p) = \theta$ if $p \in D - J$. Denote

$$D_0 = \{p | f(p) \neq \theta \& p \in D\}. \tag{2.1}$$

For any p and p' in D_0, we have $d(p, p') \geq d(f(p), f(p'))$.

If $D_0 \neq D$, we can find a point x in $D - D_0$ such that x has a adjacent $r \in D_0$. There are two cases:

1) If for every $p \in D_0$ with $p \neq r$, $d(r, p) > d(f(r), f(p))$. Let $f(x) = f(r)$. Since $d(r, x) = 1$ not less than $d(f(x), f(r)) = 0$ and

$$d(x, p) \geq d(r, p) - 1 \geq d(f(r), f(p)) - 1,$$

$d(x, p) \geq d(f(r), f(p)) \geq d(f(x), f(p))$. Let $D_0 \leftarrow D_0 \cup \{x\}$ and return to repeat.

2) If there is a point $p \in D_0$ such that $d(r, p) = d(f(r), f(p))$. Let

$$R_1 = \{p | d(r, p) = d(f(r), f(p)), p \in D_0\}. \tag{2.2}$$

We are continuing the analysis, and there are only two cases:

2.1) If for every p in R_1, $d(r, p) < d(r, x) + d(x, p)$. Therefore, $d(r, p) < 1 + d(x, p)$, so $d(x, p) \geq d(r, p)$. Let $f(x) = f(r)$. If $p \in R_1$, then $d(x, p) \geq d(r, p) \geq d(f(r), f(p)) \geq d(f(x), f(p))$; else, i.e. p in $D_0 - R_1$, since $d(r, p) > d(f(r), f(p))$ then

$$d(x, p) \geq d(r, p) - 1 \geq d(f(r), f(p)) \geq d(f(x), f(p)).$$

Let $D_0 \leftarrow D_0 \cup \{x\}$ and return to repeat.

2.2) If there is some p in R_1 such that

$$d(r, p) = 1 + d(x, p) = d(r, x) + d(x, p),$$

x is obviously in a shortest path between r and p. As the following, we will fix the point p. For every $q \in D_0$ with $q \neq r$ and $q \neq p$, we have $d(p, q) \leq d(p, x) + d(x, q)$. Let

$$m(x) = \{q | d(x, q) < d(q, r), q \in D_0\}. \tag{2.3}$$

There are only two cases needed to discuss:

2.2.1) If q in $m(x)$, then $d(p, q) < d(p, x) + d(x, q) < d(r, p) - 1 + d(r, q)$. Also, since p belongs to R_1,

$$d(p, q) \leq d(r, p) + d(r, q) - 1 \tag{2.4}$$

and

$$d(p, q) \leq d(f(r), f(p)) + d(r, q) - 1. \tag{2.5}$$

Three cases can be derived by the above equation:

2.2.1.1) If $d(r, p) \geq d(f(r), f(q)) + 2$, then $d(x, q) \geq d(r, q) - 1 \geq d(f(r), f(q)) + 1$. Therefore, $d(f(x), f(r)) \leq 1$ implies $d(x, q) \geq d(f(x), f(q))$.

2.2.1.2) If $d(r, q) = d(f(r), f(q))$, by using (2.4) and $p \in R_1$

$$d(p, q) < d(f(r), f(q)) + d(f(r), f(p)) - 1. \tag{2.6}$$

Because T is a tree, T has and only has a path between $f(p)$ and $f(p)$, which must be a shortest path. If this path contains the point $f(r) \in T$, then

$$d(f(p), f(q)) = d(f(r), f(q)) + d(f(r), f(p)) \tag{2.7}$$

and by using (2.6), we can obtain

$$d(p, q) \leq d(f(p), f(q)) - 1. \tag{2.8}$$

A contradiction occurs for the assumption of p and q in D_0. Hence, this path is not through $f(r)$.

2.2.1.3) If $d(r,q) = d(f(r), f(q)) + 1$, then

i) If $d(x, q) \geq d(r, q)$, then $d(x, q) \geq d(f(r), f(q)) + 1$. This case is similar as 2.2.1.1).

ii) If $d(x, q) = d(r, q) - 1$ and according to the fixed p with $d(r, p) = 1 + d(x, p)$ and $d(r, p) = d(f(r), f(p))$, we have

$$d(p, q) \leq d(p, x) + d(x, q) \leq d(r, q) + d(r, p) - 2,$$

so $$d(p, q) \leq d(f(r), f(q)) + 1 + d(f(r), f(p)) - 2.$$

It is similar as 2.2.1.2).

2.2.2) If $q \in D_0 - m(x)$, i.e., $d(x, q) \geq d(r, q) + 1$, then

$$d(x, q) \geq d(f(r), f(q)) + 1.$$

It is similar as 2.2.1.1).

To summarise, If there is a point p in R_1, such that

$$d(r, p) = 1 + d(x, p) = d(r, x) + d(x, p),$$

then for any q in $D_0 - \{r, p\}$, it must be included in M_1 or M_2 :

$$M_1 = \{q | d(x, q) \geq d(f(r), f(q)) + 1, q \in D_0 - \{r, p\}\},$$

we can choose $f(x) = f(r)$ or $f(x)$ is adjacent to $f(r)$ in the tree T.

$$M_2 = \{q | \text{ the shortest path between } f(p) \text{ and } f(q) \text{ is not through } f(r),\ q \in D_0 - \{r, p\}\ \}$$

Let $F_2 = \{f(q) | q \in M_2\}$. We can view that the point $f(r)$ is the root of the tree T. It is easier to see that every point in $F_2 \cup \{f(p)\}$ is included in a subtree of the tree. Thus, there exists a node t in T, such that $d(f(r), t) = 1$ and

$$\forall_{s \in F_2 \cup \{f(p)\}} d(f(r), s)) = d(t, s) + 1.$$

and then let $f(x) = t$. In fact, if the path between $f(r)$ and $f(p)$ is $f(r), s_1, ..., s_{l-1}, f(p)$, then $t = s_1$. Up to now we have obtained the proof of theorem 2.2.

Corollary 2.1 Any graph/digital manifold can normally immerse an arbitrary forest.

Theorem 2.3 Any graph/digital manifold D can normally immerse i_Σ_m.

Proof Let D be a digital manifold. Suppose that J is a subset of D and $J \neq \emptyset$, and a mapping $f_J : J \to i_\Sigma_m$ satisfies (1.2). We can denote $f_J(x) = (f_J^{(1)}(x), f_J^{(2)}(x), ..., f_J^{(m)}(x))$, and we have, for any two points p and q,

$$d(p, q) \geq d(f_J(p), f_J(q)).$$

Since

$$d(f_J(p), f_J(q)) = Max\{d(f_J^{(k)}(p), f_J^{(k)}(q)) | k = 1, ..., m\},$$

then

$$d(p,q) \geq d(f_J^{(k)}(p), f_J^{(k)}(q)),\ k = 1, ..., m.$$

Let $f_J[k] : J \to i_\Sigma_m$, where

$$f_J[k](x) = (0, ..., 0, f_J^{(k)}(x), 0, ..., 0) \tag{2.9}$$

By theorem 2.1, we have m gradually varied extensions as follows:

$$f[k] : D \to i_\Sigma_m,\ k = 1, ..., m. \tag{2.10}$$

Finally, we can combine the m gradually varied extensions to a unique mapping:

$$(x) = \sum_{k=1}^{m} f[k](x).$$

and $f(p) = f_J(p)$, $p \in$J. Also, if p and q be adjacent in D, then by (2.10)

$$d(f(p), f(q)) = Max\{d(f[k](p), f[k](q)) | k = 1, ..., m\} \leq 1.$$

Hence, f is a gradually varied extension of f_J.

Theorem 2.4 (1) A d_Σ_m can not normally immerse itself. (2) A Δ_n, $n > 2$, can not normally immerse itself. (3)A d_Σ_m or i_Σ_m can not normally immerse Δ_n, $n > 2$. (4) A Δ_n, $n > 2$, can not normally immerse d_Σ_m.

However, we have,

Lemma 2.2 A complete graph K_n can normally immerse any graph, and each graph can normally immerse K_n.

3. DIGITAL MANIFOLD AND GRADUALLY VARIED FUNCTIONALS

Suppose that M is a constructive-compact-metric-space [8], then there exists a routine generating a sequence of finite set $M_1, ..., M_n, ...$, such that M_n is a 2^{-n}_net of M. Then M_n is a digital manifold, and the adjacent set of $x \in M$ is $\{y | y \in M_n \& w(x,y) < 2^{-n}$, where w is the metric of M $\}$. Let f be a continuous functional on M (f is also an uniform continuous functional [8]). Therefore, for any l, there exists n such that: if $w(x,y) < 2^{-n}$, then $|f(x) - f(y)| < 2^{-l}$.

Now, we construct a functional f_n on M_n as follows. Let $A(s) = s \cdot 2^{-l}$, and set $f_n(x) = A(s)$, where $s = Min\{t | f(x) \leq A(t) \& x \in M_n\}$. Then f_n be 2^{-n}-uniform approximation of f on M_n.

Lemma 3.1 $f_n : M_n \to \{A(s)\}$ is gradually varied on M_n.

Proof If x and y are adjacent in M_n, then $w(x,y) < 2^{-n}$, and $|f(x) - f(y)| < 2^{-l}$. If $f_n(x) \leq f_n(y)$, then $f(x) \leq f_n(x) \leq f_n(y)$. Thus,
$0 \leq f_n(y) - f_n(x) \leq f_n(y) - f(x) \leq f_n(y) - f(y) + f(y) - f(x) < 2 \cdot 2^{-l}$, so $|f_n(x) - f_n(y)| < 2 \cdot 2^{-l}$. Similarly, in case of $f_n(x) > f_n(y)$, we can also obtain $|f_n(x) - f_n(y)| < 2 \cdot 2^{-l}$. This is, assume $f_n(x) = A(s)$ and $f_n(y) = A(t)$, then we must have $|t - s| < 2$. Hence, f_n is gradually varied.

(M_n, f_n) is said $(n,1)$-uniform approximation of (M, f). Obviously, i_Σ_1, d_Σ_1 and $\{A(s) | s \in N\}$ are equivalent. By theorem 2.1 and theorem 2.2, we can immediately obtain the gradually varied function interpolation theorem.

Theorem 3.1 Let J be a subset of D and $f_J : J \to \Sigma_1$. Then there is an extension f of f_J on D, $f : D \to \Sigma_1$ is gradually varied on D if and only if $\forall_{p,p' \in J} d(p,p) \geq d(f(p), f(p'))$.

The following three theorems deal with envelope, uniqueness, and extension of having the same norm, respectively. Because the proofs of three theorems are too long, we will give the proofs in another paper.

Theorem 3.2 If J is a non-empty subset of D and a mapping $f_J : J \to \Sigma_1$ has a gradual varying extension on then there are two gradually varied functions S_1 and S_2 on D, such that every gradually varied function S is tween S_1 and S_2. In other word, for any $x \in D$, $S_1(x) \geq S(x) \geq S_2(x)$.

Suppose that p and p' in J satisfy $d(p,p) = d(f(p), f(p'))$, then the valuations of each point in the shortest path tween p and p' are unique. This path is called non-abundant.

Theorem 3.3 Let J be a non-empty subset of D and a mapping $f_J : J \to \Sigma_1$ has a gradual varying extension a D, then there is only a unique gradually varied function on D if and only if D is covered by non-abundant iortest paths whose two end points are in J.

Let f_D be a gradually varied functional on D, the length of the longest non-abundant path is called the norm f f_D , which is denoted by $||f_D||$. $||f_D||$ has some strongly geometric meanings.

Theorem 3.4 Let f_J be a functional defined on a non-empty subset J of D. If for some $n > 0$ we have:
1) for all p and p' in J, $d(p,p') \geq d(f_J(p), f_J(p'))$;
2) if $d(f_J(p), f_J(p')) = k \cdot n + i$, $0 \leq i < n$ then
   a) $d(p,p') \geq k \cdot n + (k-1)$, $i = 0$,
   b) $d(p,p') \geq k \cdot n + k + i$, $i \neq 0$,

hen there exists a gradually varied extension f_D of f_J with $||f_D|| \leq n$.

Corollary 3.1 Let D be a digital manifold and J be the (semi-) convex submanifold of D. If $f_J : J \to \{A_s | s \in N\}$ satisfies (1.2), then there exists a gradually varied extension on D of f_J, f_D, having the same norm as $||f_J||$ on J.

4. OPTIMAL UNIFORM APPROXIMATION

In previous sections, we described the necessary and sufficient condition of the gradually varied extension. Given a digital manifold D, its a non-empty subset J, and a function $f_J : J \to \{A_s | s \in N\}$, where $A_{s-1} < A_s$. However, f_J may not satisfy the condition of gradually varied extension: for all p and p' in J, $d(p,p') \geq d(f_J(p), f_J(p'))$. In this section, we will give an algorithm for its optimal uniform approximation.

There are two kinds of common approximations, uniform approximation and least square approximation. In the case of gradually varied approximation, the uniform approximation is to find a gradually varied function f_D such that $Max\{d(f_D(p), f_J(p)) | p \in J\}$ is minimum; the least square approximation is to find a gradually varied function f_D such that $\sum_{p \in J}(f_D(p) - f_J(p))^2$ is minimum.

This paper only concerns with the uniform approximation. Without loss of generality, we can assume $A_s = s$ for convenience. Also, we denote

(1) $F_i(p) = \{f_J(p) \pm t | t = 0, 1, ..., i\}$,

(2) $F_i(p)/F_i(q) = \{x | x \in F_i(p)$ and there is $y \in F_i(q)$ such that $d(p,q) \geq d(x,y)\}$,

(3) $F_i^k(p) = \cap_{q \in J} F_i^{k-1}(p)/F_i^{k-1}(q)$,

where $F_i^0(p) = F_i(p)$.

Lemma 4.1 (1) For every p and k, $F_i^k(p) \subset F_i^{k-1}(p)$. (2) If there exists a k, for all $p \in J$, $F_i^k(p) = F_i^{k+1}(p)$. Thus, for all $N > k$, $F_i^k(p) = F_i^N(p)$.

Theorem 4.1 If $\exists k \forall p \in J(F_i^{k+1}(p) = F_i^k(p) \neq \emptyset)$, then for every $p \in J$, let $g_J(p) = inf\{F_i^k(p)\}$. Then, g_J

satisfies that for any $p, q \in J$, $d(p,q) \geq d(g(p), g(q))$, and $\forall_{p \in J}(d(g(p), f(p)) \leq i)$.

An algorithm for the uniform approximation can be designed as follows:

Step 1 Initially, let $i = 1$, and compute $F_i(p)$, $F_i(p)/F_i(q)$, and $F_i^k(p)$.

Step 2 Check if the condition of theorem 4.1 is satisfied. If it does not held, $i \leftarrow i + 1$ and go to step 1.

Step 3 Let $g_J(p) = inf\{F_i^k(p)\}$ for each $p \in J$.

Step 4 Get a gradually varied extension f_D of g_J.

A procedure for generating a gradually varied extension is given below:

Procedure A: Given a digital manifold D, its subset J and $f_J : J \rightarrow \{A_1, ..., A_m\}$, $A_1 < ... < A_m$, the procedure is to obtain a gradually varied function $f : D \rightarrow \{A_1, ..., A_m\}$ or state that there is no any gradually varied extension f of f_J.

```
BEGIN
      FOR (every pair p, p' in D) DO
         compute d(p, p') by Floyd's algorithm [6];
      FOR (every pair p, p' in J) DO
         IF (d(p, p') < d(f_J(p), f_J(p')) THEN (there is no f and halt;)
      D_0 := J;
      FOR (every r ∈ D_0) DO ( f(r) := f_J(r); )
  5   IF (D_0 = D) THEN
         (output f and halt )
      ELSE
         BEGIN
            choose x in D − D_0 such that x has an adjacent
            r ∈ D_0. (without loss of generality assume f(r) = A_i)
         END
      f(x) := A_i;
      FOR (every p ∈ D_0) DO
         BEGIN
            IF( d(x, p) < d(f(x), f(p)) ) THEN
               IF ( f(x) > f(p) ) THEN ( f(x) := A_{i-1} ) ELSE ( f(x) := A_{i+1} );
         END
      D_0 := D_0 ∪ {x};
      GOTO 5
END
```

In an application of the algorithm, we have used it to fit the surfaces in seismic prospecting. The Figure 4.1 shows a series of interfaces in a stratum by using the algorithm in an area located in the southwest of China.

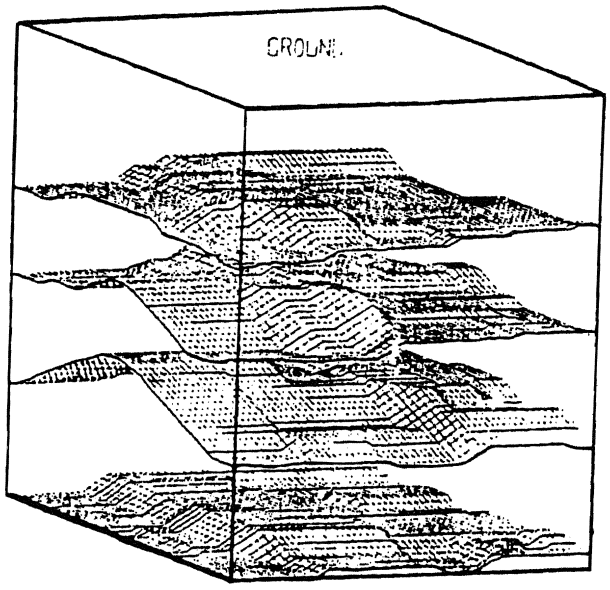


Figure 4.1 A Real Series of Interfaces by Using Gradually Varied Surface Approximation.

Acknowledgments: This research was supported by the Science Fund of the Chinese Academy of Sciences for Young Scientists.

REFERENCES

1. L. Chen, "The necessary and sufficient condition and the efficient algorithms for gradually varied fill," *Chinese science Bulletin*, Vol 35,10(1990). *Abstracts of SIAM Conference on Geometric design*, Temple, AZ, 1989.

2. L. Chen, The properties and the algorithms for gradually varied fill, chinese J. of computers, Vol 14, 3(1991).

3. L. Chen, "Random gradually varied surface fitting," *Chinese Science Bulletin*, Vol. 37, 1992. *Abstracts of Second SIAM Conference on Geometric design*, Temple, AZ, 1991.

4. L. Chen, "Gradually varied surfaces on digital manifold," *Abstracts of Second International Conference on Industrial and Applied Mathematics*, Washington, DC, 1991.

5. F. Harary, *Graph Theory*, Addison-Wesley, Readding, 1972.

6. A. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

7. A. Rosenfeld and A. Kak, *Digital Picture Processing*, Vol. I and II, Academic Press, Orlando, 1982.

8. E. Bishop and D. Bridges, *Constructive Analysis*, Springer Verlag, 1985.

Generalized Discrete Object Tracking Algorithms and Implementations

Li Chen
Scientific and Practical Computing
P.O. Box 6081
North Logan, UT 84341-6081
and
Department of Computer Science
Utah State University
Logan, UT 84322-4205
chenl@sys6.cs.usu.edu

ABSTRACT

This paper presents a general-purpose definition for discrete curves, surfaces, and manifolds. We also focus on their tracking algorithms and implementations. This definition only refers to a simple graph, $G = (V, E)$, which is a generalized discrete space. The idea presented in this paper is to recursively define discrete curves, surfaces, solid objects and so forth. Obviously, a vertex is a point-unit-cell, and an edge is a line-unit-cell. A surface-unit-cell is a simple closed path C (also called a cycle) if there is no pure subset of C can be a cycle. Based on this intuitive idea, n-D unit-cells can be defined in G for $n > 2$. For a graph G, i-D unit-cells, $i = 0, ..., n+1$, provide a topological structure to the discrete space G. A n-D discrete manifold M is defined as (1) M consists of n-D unit-cells and any two n-D unit-cells are $(n-1)$-D connected, (2)each $(n-1)$-D unit-cell in M is contained by one or two n-D unit-cells, and (3) there is no $(n+1)$-D unit-cell in M. This definition extends a definition of digital manifolds[1]. We have developed an linear time algorithm to decide if a subset is a discrete n-D manifold, a linear time algorithm to obtain the k-D boundary for n-D manifold. Such algorithms have been implemented in Σ_3, a space containing all integer points. This paper also discusses the non-orientable surface, quadtree surface-unit-cell representation, and an octree solid-unit-cell representation.

Keywords: digital geometry and topology, image processing, graph, curves, surfaces, manifold, tracking algorithm, implementation.

1. INTRODUCTION

Curve and surface tracking is growing in importance and application for image processing, computer vision, and computer graphics. Most of the related research work deal with curves and surfaces in Σ_m, a space containing all integer grid points. However, numerous applications need more generalized curves and surfaces, such as meshes in surface fitting. The triangulation of a continuous surface forms a "discrete surface," which is mostly like a graph. The digital curve can be defined as a simple path, but the definition of digital surfaces is more difficult. In earlier of the 80's, two kinds of digital surfaces were studied by Artzy, Frieder and Herman[2], and Morgenthaler and Rosenfeld[3] respectively. In Artzy-Frieder-Herman's paper[2], a point in 3D spaces was viewed as a voxel or a unit-cub. The boundary surface of a 3D object consists of some faces of unit-cubs. An efficient tracking algorithm to search boundary surface was also developed[2]. On the other hand, Morgenthaler-Rosenfeld presented a mathematical definition for closed digital surfaces. In their paper, a point is just an element of the array Σ_3. An improved definition

SPIE Vol. 3168 • 0277-786X/97/$10.00

of digital surfaces was given in Reference 1,4,5. In fact, a voxel can be represented by eight èlements in Σ_3. Some interesting ideas were considered by Latecki[6].

This paper gives a formal and generalized definition for discrete curves, surfaces, and manifolds in a simple graph $G = (V, E)$, which is treated as a discrete space first used by Herman [7]. The definition has no conflicts from the previous work. Based on the definition of discrete manifolds, some efficient tracking algorithms will be given in this paper. We have implemented the tracking algorithms in 2D spaces and 3D spaces using Borland C++ .

2. BASIC CONCEPTS

This section briefly reviews some basic concepts of digital geometry and topology. Although our goal is to give a curve and surface definition in a general discrete space, $G = (V, E)$, in order to present the basic idea we only concern the special discrete space Σ_m in this section. Let Σ_m be a m dimensional array of integer coordinate points. A point p in Σ_2 has four horizontal and vertical neighbors, namely $(x \pm 1, y)$ and $(x, y \pm 1)$; and p also has four diagonal neighbors, namely $(x \pm 1, y \pm 1)$ and $(x \pm 1, y \pm 1)$. We say that horizontal and vertical neighbors are directly adjacent to p (or 4-adjacent to p); and we say that both types of neighbors are adjacent to p (or 8-adjacent to p) [8].

In general, the points $p = (x_1, x_2, ..., x_m)$ and $q = (y_1, y_2, ..., y_m)$ are two directly adjacent points in Σ_m if $d_D(p, q) = \sum_{i=1}^{m} |x_i - y_i| = 1$. We say p and q are adjacent points if $d(p, q) = \max_{1 \le i \le m} |x_i - y_i| = 1$. We only consider direct adjacency in this section. Let p and q be a pair of (directly) adjacent points of Σ_m. Intuitively, $\{p, q\}$ is a line-unit-cell. A surface-unit-cell is a set of 4 points which form a unit square parallel to coordinate planes. A 3-D unit-cell (or solid-unit-cell) is a unit cub which has 8 points. More generally, a k-D-unit-cell (k-D-cub) has exactly 2^k points, $p_1, ..., p_{2^m}$, where each of $p_1, ..., p_{2^m}$ has m components, denoted by $p_i = (x_1^{(p_i)}, ..., x_m^{(p_i)})$. There are $m-k$ components which do not change value for all $p^{(1)}, ..., p^{(2^m)}$. For p_i and $p_j, i \neq j$, $1 \le \sum_{t=1}^{m} |x_t^{(p_i)} - x_t^{(p_j)}| \le k$, and $\max_{t=1}^{m} |x_t^{(p_i)} - x_t^{(p_j)}| = 1$. This definition is not very intuitive. We will give an recursive form in Section 3. Also, we sometime call a 0-D-unit-cell a point-unit-cell, 1-D-unit-cell a line-unit-cell, and 2-D-unit-cell a surface-unit-cell.

Two unit-cells u and v are point-adjacent if they share a point, line-adjacent if they share a line-unit-cell, and k-D-adjacent if they share a k-D-unit-cell. u and v are called k-D-connected in S if there exists a path $u_0 = u, u_1, ..., u_n = v$ in S, such that u_i and u_{i+1} are $k - D$-adjacent, $i = 0, ..., n-1$. Figure 2.1 shows some examples of adjacency and connectedness.

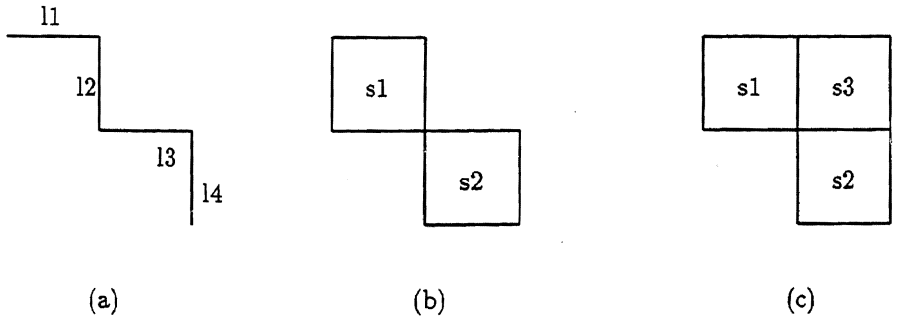


Figure 2.1 Adjacency and connectedness

Definition 2.1[1,4] Let A be subset of Σ_m. A' is called a parallel_move of A if (1) $|A| = |A'|$, (2) $A \cap A' = \phi$, and (3) there exists a biconjunction mapping $f : A \to A'$ such that $d(a, f(a)) = 1$ for all $a \in A$.

Definition 2.2[1,4] A connected subset M of Σ_m is a n dimensional digital manifold if any point $p \in M$ is included in some n-D unit-cell of M, and (1) any two n-D unit-cell are $(n-1)$-D connected in M, (2) every $(n-1)$-D unit-cell in M has only one or two parallel-moves, and (3) M does not contain any $(n+1)$-D unit-cell.

Since the concept of parallel-move can not be used to general discrete spaces, definition 2.2 can not be easily extended to general cases.

3. DISCRETE SURFACES AND MANIFOLDS

Let $G = (V, E)$ be a simple graph, where V is the vertex set and E is the edge set. For a simple graph, if $a \in V$ then (a, a) is not in E. If (p, q) is in E, p is called adjacent to q. Let $p_0, p_1, ..., p_{n-1}, p_n$ be $n+1$ vertices in V. If (p_{i-1}, p_i) is in E for all $i = 1, ..., n$, then $\{p_0, p_1, ..., p_{n-1}, p_n\}$ is called a path. If $p_0, p_1, ..., p_{n-1}, p_n$ are distinct vertices, the path is called a simple path. We say that a simple path $\{p_0, p_1, ..., p_{n-1}, p_n\}$ is closed (cycle) if (p_0, p_n) is an edge of G. Two vertices p and q are connected if there is a path $\{p_0, p_1, ..., p_{n-1}, p_n\}$ such that $p_0 = p$ and $p_n = q$. G is called connected if every pair vertices in G is connected. We always assume G is connected.

In order to define discrete curves and surfaces, we need to define the basic units of them, i.e. the line-unit-cell and the surface-unit-cell. The line-unit-cell is quit simple, i.e., a subgraph just contains two adjacent vertices. In section 2, we can easily define the basic members of Σ_m : point-unit, line-unit, and surface-unit. In a general discrete space G, we can use the same definitions for point-unit and line-unit, but it is not trivial to give the definitions of surface-unit. The reason is that a surface-unit could be a triangle, a square, or a polygon.

Let S be a set and S' be a subset of S, denoted by $S' \subseteq S$. If S is not a subset of S', then S' is called a pure-subset of S, $S' \subset S$. Let $G' = (V', E')$ be a graph and $V' \subset V$. If E' consists of all edge in G whose end vertices are in V', we say that $G' = (V', E')$ is a subgraph of G, denoted by $G' \preceq G$. If V' is a pure-subset of V, then we say $G' \prec G$.

A subgraph D of G is a simple closed path if D is closed and there is no any D's pure-subgraph D_0 which is a closed path. Such D is also called simple cycle. A surface-unit must be a simple cycle, but a simple cycle may not be a surface-unit. Suppose that we have a G shown in Figure 3.1. In (a) and (b), every surface-unit is clear. However, G in (c) and (d) has the same V and E, but one could have different interpretations. If simple cycle (a, b, c, f, i, h, g, d) is not a surface-unit, then the G looks a plane; if the cycle is a surface-unit, then the G looks having a 3-dimensional-unit.

A geometrical interpretation of G is necessary in this paper. This interpretation is to give a geometrical frame (a topological structure) to G. Let $\mathcal{C}$ be the set of all simple cycles in G. Defining a subset of $\mathcal{C}$ to be the set of minimum cycles or surface-unit is a way for generating a geometrical frame to G.

Definition 3.1 Let $\mathcal{C}$ be the set of all simple cycles in G. A subset of $\mathcal{C}$, U_2 is called a minimum cycle set if for any two different simple cycles in $\mathcal{C}$, u and v, $u \cap v$ is connected (in $u \cap v$). Each of the element of U_2 is called surface-unit respecting to the pair of $< G, U_2 >$.

We want that $u \cap v$ is a connected path because we want that the intersection of two surface-unit is on the "edges" of these two surface-unit. If $u \cap v$ is just a node, we say u and v are point-connected. If $u \cap v$ is just a line-unit, we say that u and v are 1-line-connected. Furthermore, u and v are called $(|u \cap v| - 1)$-line-connected if u and v share $(|u \cap v| - 1)$ line-unit. In this paper, we mostly consider 1-line-connectedness among surface-unit like G in Figure 3.1 (a) (b).

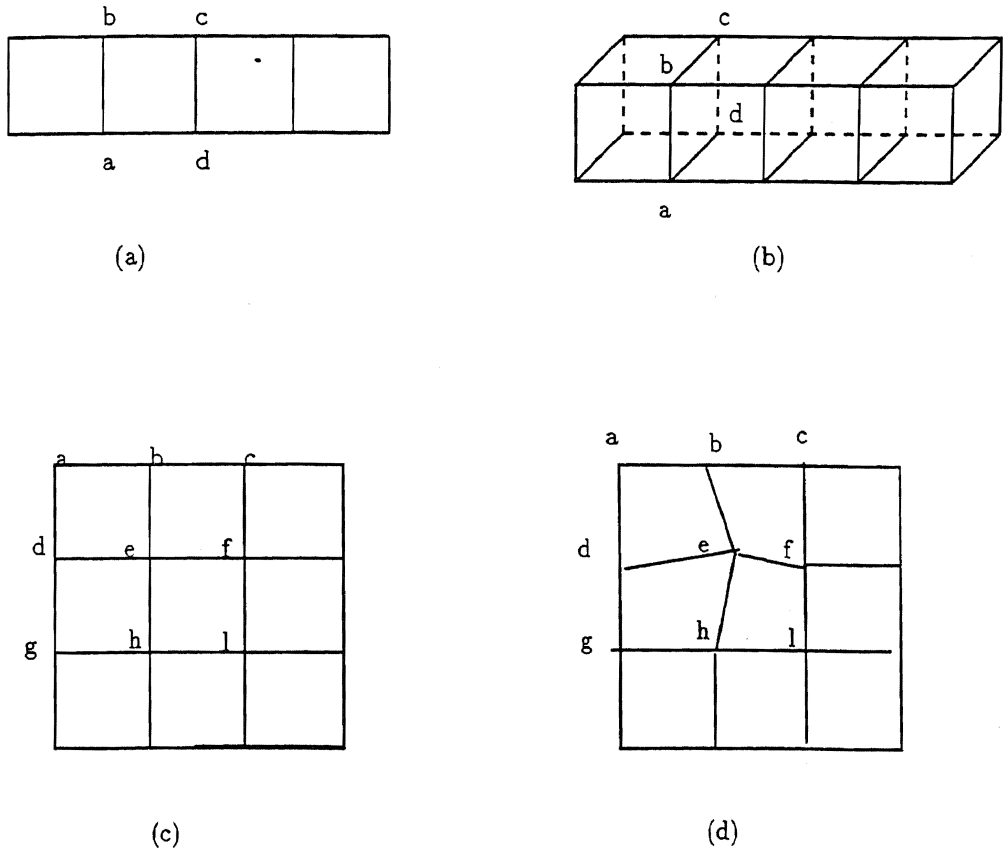


Figure 3.1 Examples of discrete spaces

Definition 3.2 (1)A subgraph D of G is called a (simple) semi-curve if D is a simple path.

(2)D is called a curve if D is simple path and D does not contain any minimum cycle (surface-unit).

A common definition of curve, "A simple discrete (or digital) curve is just a simple path in G"[8] is a semi-curve in this paper. The reason is there is a simple path such as $\{e, f, l, h\}$ Figure 3.1 (c), that can not separate the surface into two disconnected nonempty sets. This property is called Jordan theorem which is held in continuous space.

In order to discuss the properties for all discrete spaces without to prepare a surface-unit set U_2 first. We give a default definition of U_2 here. If a special definition of U_2 is not given in later discussion, we suppose to use this default definition of U_2.

Definition 3.3 Let G be a simple graph, the default definition of surface-unit of G is :

(1) Assume m is the minimum number of the path length for all closed paths. All closed paths with the length m are surface-unit.

(2) If a point p is not covered by any surface-unit in (1), a minimum number of a closed path containing p will be a surface-unit.

Definition 3.4 Subgraph D of G is a semi-surface if and only if each line-unit of D is included in one or two surface-unit in D. D is called a closed semi-surface if and only if each line-unit is included in exact two surface-unit

of D.

A 3-dimensional-unit is a closed semi-surface, but a closed semi-surface may not be a 3-dimensional-unit. As the same as we did for surface-unit, we define

Definition 3.5 Let $\mathcal{S}$ be the set of all closed semi-surface in G. A subset of S, U_3 is called a minimum closed semi-surface set if for any two different elements in U_3, u and v, $u \cap v$ is connected (in $u \cap v$) and all surface-unit in $u \cap v$ are line-connected . Each of the element of U_3 is called $3D$-unit respecting to the pair of $< G, U_3 >$.

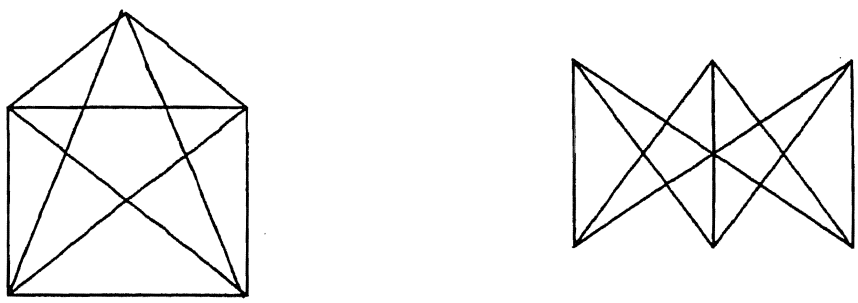
Definition 3.6 $S \preceq G$ is a discrete surface if and only if S is a semi-surface and S does not contain any subset which is a 3-dimensional-unit. The boundary of surface S, denoted by ∂S, is a subset of E and each element of ∂S is exactly contained by one surface-unit in S.

Corollary 3.1 A discrete surface is closed if and only if $\partial S = \phi$.

Proof: According to the definition of boundary of discrete surfaces, it is easy to see.$\diamondsuit$

Lemma 3.1 Both the complete graph K_5 and the bicomplete graph $K_{3,3}$ are not discrete surfaces.

Proof: The complete graph K_5 and the bicomplete graph $K_{3,3}$ are shown in Figure 3.2 below:



(a) (b)

Figure 3.2 The complete graph K_5 and the bicomplete graph $K_{3,3}$

Using default surface-unit definition, every 3-point-cycle is a surface-unit in K_5 and every 4-point-cycle is a surface-unit in $K_{3,3}$. One can see, every edge in (a) is contained by more than three surface-unit. In (b), every edge is also contained by at least three surface-unit.$\diamondsuit$

Theorem 3.1 A discrete surface is a planar graph, but a planar graph may not be a discrete surface.

Proof: We knew planar graph decision theorem is that a G is a planar graph if and only if G does not contain K_5 and $K_{3,3}$. According to Lemma 3.1, a discrete surface is a planar graph. However, look at the follow example, one can see this planar graph is not a surface.$\diamondsuit$

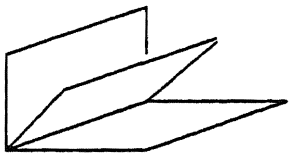


Figure 3.3 A planar graph but not a discrete surface

Thus, Jordan's theorem of planar graph can be applied to discrete surfaces.

Theorem 3.2 Discrete surfaces S have Jordan's properties: A closed curve D on S divides S into two disconnected components. In other words, $S - D$ consists of at least two disconnected components.

The proof of the theorem is very long, we will give the proof in another paper.

Theorem 3.2 The simplex division of a continuous surface forms a discrete surface.

Proof: We define the U_2 as

$$U_2 = \cup\{\partial\Delta\}$$

where Δ is a simplex. By the definition of the simplex dividing, any line segment is contained by at most two simplex. On the other hand, the intersection of two simplex is empty, a point or a line segment. That exactly matches the definition of discrete surfaces in this section.$\Diamond$

Generally, we can give a recursive definition for n-D discrete manifolds as follows: Assume we have a $G = (V, E)$, every element in V is a 0-D unit-cell, and every element in E is an 1-D unit-cell.

(1) A n-D unit-cell u is a subset of V, $n \geq 2$, which consists of several $(n-1)$-D unit-cell, and each $(n-2)$-D unit-cell is exactly contained by two $(n-1)$-D unit-cell in u. In addition, for any two n-D unit-cells, u and v, $u \cap v$ consists of several connected (in $u \cap v$) $(n-1)$-D unit-cell. A n-D unit-cell is also called minimum closed $(n-1)$-D semi-manifold. That is to say, a n-D unit-cell set, U_n, is based on the existing $U_0 = V, U_1 = E, ...U_{n-1}$. We also call $U_0, U_1, ...U_n$ a n-D discrete-topological-structure (DTS). We can see if $U_i = \phi$ then $U_k = \phi$ for all $k > i$. The dimension of G is defined as the largest index of $U_i \neq phi$ for all possible selection of DTS. Two n-D unit-cells u, v are k-D adjacent if $u \cap v$ consists of several k-D unit-cells. u and v is k-D connected if there is a path $u_1, ..., u_m$ where u_t, u_{t+1} are k-D adjacent, $t = 1, ..., m-1$.

(2) Given a $(n+1)$-D discrete-topological-structure of G, a connected subset M of G is a n-D discrete manifold if for any $a \in M$ then there exists a $u_i \subset U_i$ such that $a \in u_i$ and $u_i \subset M$, $i = 0, ..., n+1$, and (a) any two n-D unit-cell are $(n-1)$-D connected in M, (b) every $(n-1)$-D unit-cell in M is contained by only one or two n-D unit-cell. (c) M does not contain any $(n+1)$-D unit-cell.

(3) Let M be a n-D discrete manifold. The boundary of M, ∂M, is the set of all $(n-1)$-D unit-cell in M, each of which is contained by only one n-D unit-cell in M.

4. TRACKING ALGORITHMS

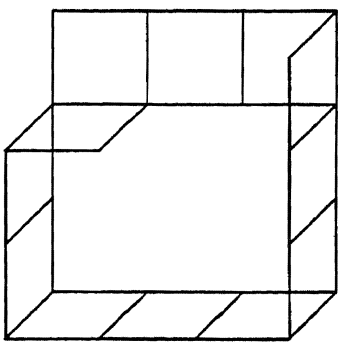
Curve and surface tracking are basic techniques to image segmentation and image registration. Typically, tracking is to determine objects and extract their boundaries in a discrete spaces. Most tracking problems are only deal with one object, such as "chain code" in a 2-D image. In this case, a segmentation must be used first to partition the image to each single object. That is not practically efficient because more extensive time and space

(in memory) are needed. The tracking algorithms developed in this paper do not need a segmentation.

There are many different tracking problems. The simplest tracking may be curve decision problem: Given a subgraph C of G, decide whether or not C is a curve. We can simply test if any 0-D unit-cell is contained by one or two 1-D unit-cells and other two conditions. This problem may not very useful in practical image processing. However, we still can define surface decision problem and manifold decision problem using the same way. Their algorithms are simple too. There is a decision problem which is important to mention here. How to decide a discrete non-orientable surface (or containing a Mobuis Band)? We will develop an algorithms later. Boundary tracking probably is the most important tracking problem in image processing and computer vision. In 2D, we want to track the boundary curves of 2D objects. In 3D, we want to track the boundary curves of 2D objects as well as boundary surfaces of 3D objects. We will discuss a representation-independent method to the problem. Because some images are represented by quad-trees and octrees, we also consider the fast algorithm just for these representations.

4.1 Nonorientable Surface Decision

In this section, we let $G = \Sigma_3$ for simplicity. Let S be a surface. If S contain a Mobuis band, then S is non-orientable. As a example, we show a digital Mobuis band below.



0 0 0 1	1 1 1 1
1 1 0 1	1 1 1 1
1 0 0 1	1 0 0 1
1 1 1 1	1 1 1 1
First Section	Second section

Figure 4.1 A Mobuis band

How to decide a nonorientable surface? In pure mathematics, one tests whether the surface contain a subsurface which is topological equivalent a Mobuis band. Let p_1, p_2, p_3, p_4, $d_D(p_i, p_{(i+j) mod(4)}) = 1$, be a surface-cell; then its two normal lines can be represented as (p_1, p_2, p_3, p_4) and (p_4, p_3, p_2, p_1) (see Figure 4.2 (a)). Meanwhile, all of the loop-shifts of them are equivalent. As an example, $(p_1, p_2, p_3, p_4) = (p_3, p_4, p_1, p_2)$.

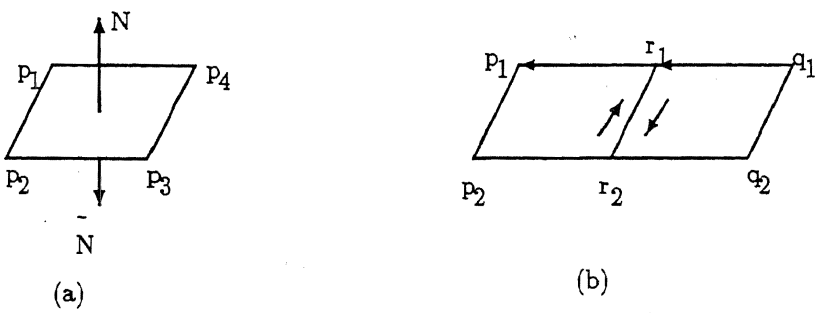


Figure 4.2 Normal lines

Let A and A' be two surface-cell, and they are line-adjacent (see Figure 4.2 (b)). There are 4 normal lines as (p_1, p_2, r_2, r_1) , (r_1, r_2, p_2, p_1), (r_1, r_2, q_2, q_1) and (q_1, q_2, r_2, r_1); (p_1, p_2, r_2, r_1) and (r_1, r_2, q_2, q_1), and (r_1, r_2, p_2, p_1)

and (q_1, q_2, r_2, r_1) are called adjacent normal-line. Generally, if N and N' are two normal lines of two adjacent surface-cell A and A', respectively; N and N' are called adjacent when the orders of r_1, r_2 appearing in N and N' are different (e.g. we use anti-clockwise), where $A \cap A' = \{r_1, r_2\}$. We can determine the connectedness of normal lines based on their adjacency.

Definition 4.1 A surface S is called nonorientable if all normal lines of S are connected.

Lemma 4.1 Let S be a surface, N and $\bar{N}$ are two normal lines (one is anti-clockwise and another one is clockwise) of a surface-cell of S. S is nonorientable if N and $\bar{N}$ are connected.

We can design an optimal algorithm, $O(|S|)$ time, to decide whether a surface S is orientable or nonorientable.

Algorithm 4.1 A orientable surface tracking algorithm
Step 1 Get N and $\bar{N}$ for all surface-unit-cells in S.
Step 2 Using breadth-first-search technique[9] search normal lines for adjacent surface-unit-cells.
Step 3 Once the number of connected normal lines of surface-cells is greater than the number of surface-unit-cell of S, then S is nonorientable.

4.2 Object and Boundary Tracking

Let S be a subset of G, a point (vertex) p is a i-D point in S if there is a i-D unit-cell in S containing p and there is no $(i+1)$-D unit-cell in S containing p .

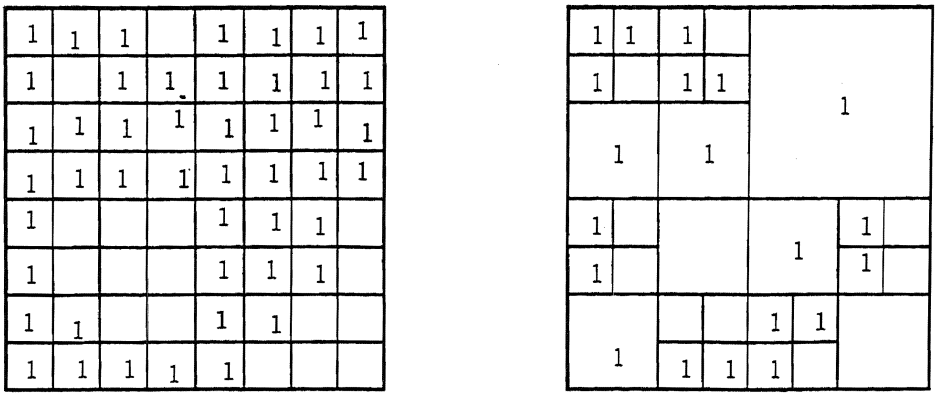
There are two kinds of tracking problems: (1) Object tracking, select an integer i to get all i-D points in S, denoted by $S^{(i)}$; segment $S^{(i)}$ into connected components using adjacency if necessary. (2) Boundary tracking, select an integer i to get a set of i-D points in S, denoted by $\partial S^{(i)}$; if $p \in \partial S^{(i)}$ and p also belongs to a $(i-1)$-D unit-cell, the $(i-1)$-D unit-cell is only contained by just one i-D unit-cell in S; segment $\partial S^{(i)}$ into connected components using adjacency if necessary.

According to the definition of discrete manifolds, it is easy to see all above tracking have (linear) optimal algorithms. If given adjacency-lists for point-unit-cells, line-unit-cells,..., then we can use breadth-first-search method to design more faster algorithms. However to obtain such adjacency-lists may not be simple. We will consider it in future research. Boundary of an image may not be a curve (or curves), and Boundary surface of a solid object may not be a surface using our definition. However, in some applications, "perfect" curves and surfaces may be not important.

4.3. Boundary Tracking of 2D and 3D Images Represented by Quadtrees and Octrees

Some real images are stored by quadtree and octree. Quadtree and octree technique is a very powerful method to represent a two or three-dimensional image. They are also used to image compression and solid modeling. In this section, we briefly give boundary tracking algorithms for 2D and 3D.

As we known, quadtree technique is a binary image data structure. Suppose we have an image shown in Figure 4.3 (a). Its quadtree is shown in Figure 4.3 (b). For simplicity, we call the space in which a point represents a small region (one unit edge square) as a AFH (Artzy-Frieder-Herman) space, The space in which a point represents a point as a MR (Morgenthaler-Rosenfeld) space. All vertices of unit square in a AFH space forms a MR space. The MR space is called the dual space of the AFH space.



(a) (b)

Figure 4.3 (a) A binary image, (b) The quadtree representation of (a)

We can see such representation is a AFH representation. In Figure 4.3 (a), a surface cell has four vertices and four edges in MR spaces (which is the dual space of the AFH space). But in Figure 4.3 (b), a surface cell could has more vertices and four edges. However, they satisfy Definition 3.1. The boundary tracking is to search all edges (line cells), each of which is contained by only one surface cell. This is very easy to implement. We programmed a C++ code and applied to real images.

An interest thing is that we use AFH spaces to represent images and use the algorithm in MR spaces to get the solutions. Such solutions are exactly the same as Herman's definition [7]. The advantages to use quadtree surface cells are: (1) satisfying the definition of surface cell naturally, (2) very small amounts of surface cells in general, (3) easy to implement.

Using the similar technique, we can track the boundary of a solid. First get octree representation of the solid. Then translate to MR spaces. Finally find all surface cells, each of which is contained by only one solid cell.

5. IMPLEMENTATION AND DISPLAY

Efficient tracking algorithms are very useful in real image processing. For example, "chain code" in 2D has a lot of usages such as shape recognition and compression. A fast algorithm for tracking a three dimensional object was given in Reference 2 . The algorithms given in section 4 can also use Tarjan's breadth first searching method. Because real images have noise, tracking boundary for all objects is not very easy. In addition, we will not only track "perfect" discrete manifolds.

For 2D images, in order to get information properly, we can develop a process to get all following information: (1) extract all curves, that is to get all points in curves. (2) extract all surfaces, that is to get all points of surfaces, and (3) extract all boundaries for all surfaces surfaces. For 3D images, we need to add (4) extract all solid objects, and (5) extract all boundaries for all solid objects.

Note, a point is in a curve of an image, then this point is not in any surface of the image, and so on.

It is so simple to get the job done because all algorithms in section 4 can be easily implemented in C++. We have built a demo system in Borland C++ 5.0 to perform these algorithms. We have only implemented these algorithms in Σ_2 and Σ_3. Figure 5.1 shows the results of boundary search of 2D objects in 3D spaces, and Figure 5.2 shows the results of boundary search of a 3D object which has two holes inside of the object in 3D space.

6. CONCLUSION

The definition of discrete manifolds we gave in this paper only refers to a simple connected graph $G = (V, E)$. We also focus on practical tracking problems and surface orientable problem. We have implemented these algorithms in 2D and 3D grid spaces. We will implement some algorithms in general G and apply to some practical problems such as image registration in the future. Because this paper only dealt with binary images and most of real images are gray images or color images, we will need a converting process when we try to apply these algorithms to complicated applications. We will consider some direct use of these algorithms by combining with gradual variation techniques[10] and some fuzzy segmentation techniques[11].

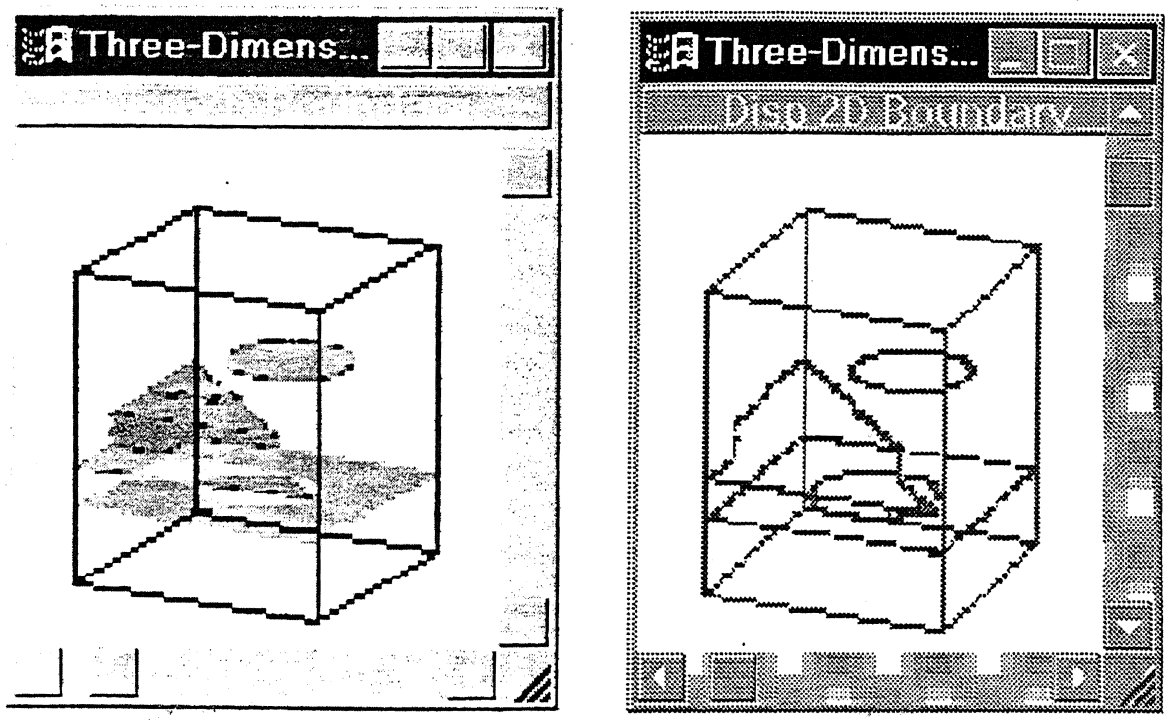


Figure 5.1 Boundary search of 2D objects in 3D spaces

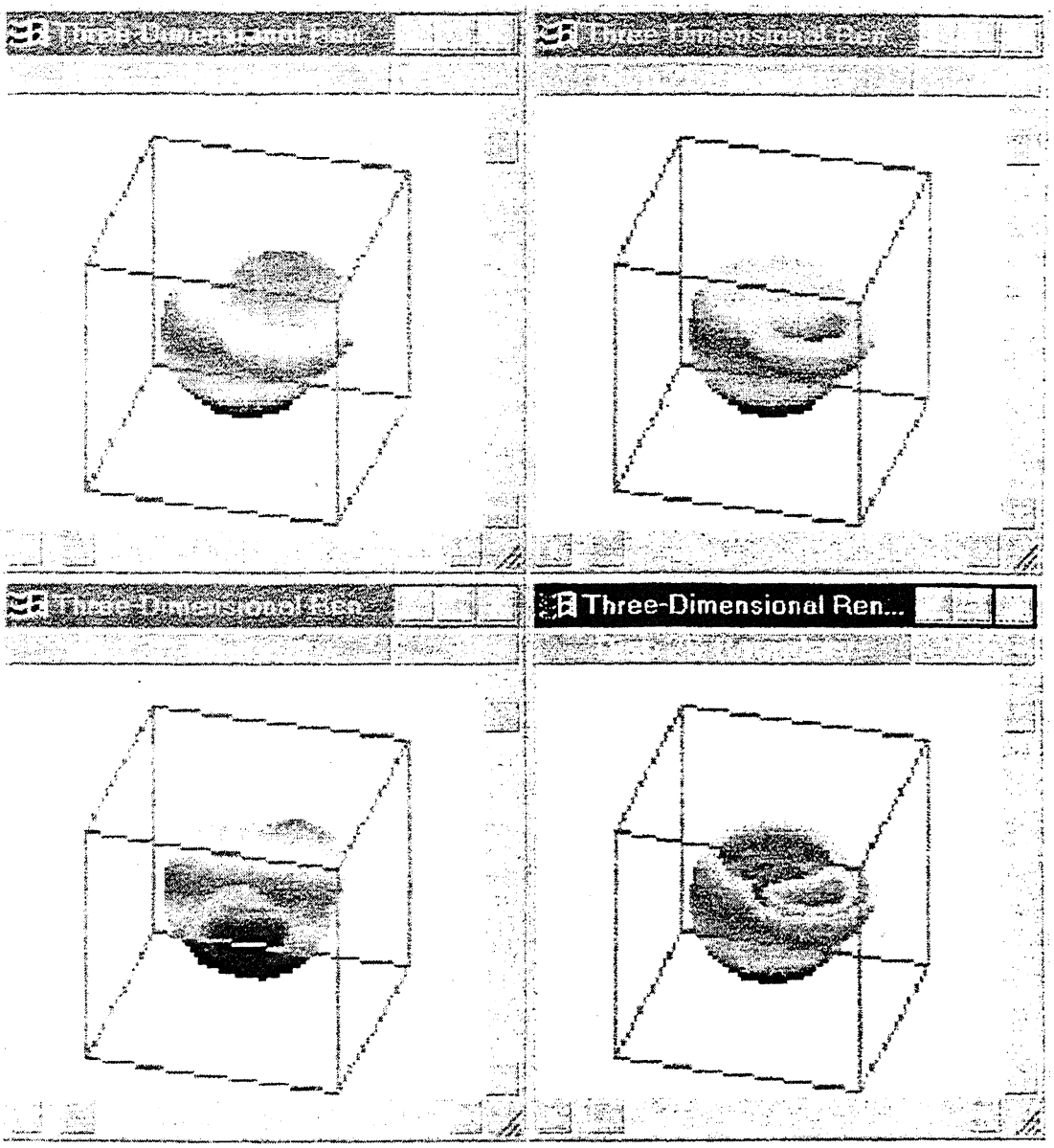


Figure 5.2 Boundary search of a 3D object in 3D spaces

7. REFERENCES

1. L. Chen and J. Zhang, "Digital manifolds: A Intuitive Definition and Some Properties", *The Proc. of the Second ACM/SIGGRAPH Symposium on Solid Modeling and Applications,* pp. 459-460, Montreal, 1993.

2. E. Artzy, G. Frieder and G. T. Herman, "The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm", *Comput. Vision Graphics Image Process.* Vol 15 , pp. 1-24, 1981.

3. D. G. Morgenthaler and A. Rosenfeld, "Surfaces in three-dimensional images", *Imform. and Control* Vol 51, pp. 227-247, 1981.

4. L. Chen and J. Zhang, "Digital surface definition and the fast algorithms for decision and tracking", *The Proc. of SPIE on Vision Geometry II,* Vol 2060, pp. 169-178, 1993.

5. L. Chen and J. Zhang, "Classification of Simple digital Surface Points and A Global Theorem for Simple Closed Surfaces", *The Proc. of SPIE on Vision Geometry II,* Vol 2060, pp. 179-188, 1993.

6. L. Latecki, "3D well-composed pictures", *The Proc. of SPIE on Vision Geometry IV,* Vol 2573, pp. 196-203, 1995.

7. G. T. Herman, "Discrete multidimensional Jordan surfaces", *CVGIP: Graphical Models and Image Process.* Vol 54, pp. 507-515, 1992.

8. T.Y.Kong and A.Rosenfeld, "Digital topology: Introduction and survey", *Comput. Vision Graphics Image Process.* Vol 48, pp. 357-393, 1989.

9. A. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

10. L. Chen, "Gradually varied surface and its optimal uniform approximation", *IS&T/SPIE Symposium on Electronic Imaging,* SPIE Proc. Vol 2182, pp. 300-307, 1994.

11. L. Chen, H. Cheng and J. Zhang, "Fuzzy subfiber and its application to seismic lithology classification", *Information Science: Applications*, Vol 1, No 2, pp. 77-95, 1994.

A Note on (α,β)-Type Digital Surfaces and General Digital Surfaces

Li Chen
Scientific and Practical Computing
P. O. Box 6081
North Logan, UT 84341
and
Department of Computer Science
Wuhan University
Wuhan 430072, China

Abstract

Morgenthaler and Rosenfeld gave the first formal definition of digital surfaces [1]. Kong and Roscoe embedded digital spaces to continuous spaces to show the intuitive meaning of Morgenthaler-Rosenfeld's surfaces [2]. Reed and Rosenfeld discussed the recognition algorithms for digital surfaces [3]. Chen and Zhang considered an intuitive and simple definition of digital manifolds in case of direct adjacency in digital spaces [4-5]. This note continues the task in [2] and [4-5]. This note will simplifier some concepts used in [2] and extended some concepts used in [4-5]. First, we continue to study all cases of (α,β)-surfaces and eliminate some trivial and overlapping cases. Among nine types of digital surfaces, there are only two useful: (6,26)-surfaces and (18,6)-surfaces. We will develop or modify three algorithms for the three definitions in [1], [2], and [5], respectively. Because some visually true digital surface (or surface points) is not included in any of (α,β)-surfaces (or surface points), we will give a more general definition for digital surface in both direct and indirect adjacency. In order to develop a fast algorithm for real surface tracking, a quasi-digital-surface is defined even through sometimes the definition is weak in terms of mathematics.

1. Introduction

In three-dimensional image processing, surfaces are represented by a set of points in three-dimensional (3D) arrays. Morgenthaler and Rosenfeld gave a formal definition of digital surfaces [1]. According to the nature of connectivity among points in digital spaces, Morgenthaler-Rosenfeld's definition has totally nine types of variations, which are usually denoted by (α,β)-surfaces, where α, β =6, 18, 26. However, this formal definition of digital surfaces is difficult to understand in the sense of geometric meaning. Kong and Roscoe embedded the digital space to a continuous space to show the intuitive meaning of Morgenthaler-Rosenfeld's surfaces [2].

Kong and Roscoe's continuous analogs of digital surfaces provided an intuitive view for each type of Morgenthaler-Rosenfeld's surfaces, and some concepts and results in [2] are intuitive. However, the method used in [2] is very difficult to understand, and the use of continuous analogs may not be necessary. Because a Morgenthaler-Rosenfeld's surface is a set of simple surface points, when one tries to recognize or decide if a set is a digital surface in real image processing, the one must go through all conditions that determine a simple surface point (see Reed and Rosenfeld [3]). This procedure is not fast. Although some properties pointed out by Kong and Roscoe [2] are very important and useful, a fast algorithm still can not be developed based on [2]. In addition, Morgenthaler-Rosenfeld's definition can only deal with closed surfaces, so it has some restrictions in practical use. We will give more discussions in section 4.

Part of the SPIE Conference on Vision Geometry VII • San Diego, California • July 1998
SPIE Vol. 3454 • 0277-786X/98/$10.00

For the purpose of simplifying Morgenthaler-Rosenfeld's definition, Chen and Zhang introduced an intuitive and tractable definition of digital surfaces [4-5]. In fact, some ideas in [5] first appeared in [2] in the continuous sense. However, the digital surface definition in [5] is totally defined in a digital space that is the same as Morgenthaler-Rosenfeld's definition. In addition, a surface-unit-cell is defined by the parallel-moving of a line-unit-cell in [4-5], and this principle can be used to develop a very fast recognition algorithm by employing Tarjan's breadth-first-searching technique [5]. The other advantages of the parallel-move-based digital surface definition are (1) the definition is for both closed and unclosed digital surfaces, and (2) the surface concepts in [4] can be generalized to define high dimensional manifolds [4]. However, the digital surface definition only dealt with direct adjacency, which are the case of (6,26)-surfaces in Morgenthaler-Rosenfeld's definition.

This note continues the work in [2] and [4-5]. This note will simplify some concepts used in [2] and extended the concepts in [4-5]. First, we continue to study all cases of (α,β)-surfaces and eliminate some trivial and overlapping cases. In section 3, we will find out the real useful types of (α,β)-surfaces based on [2]. In section 4, we will develop or modify three algorithms for the three definitions in [1], [2], and [5], respectively. Because each type of these (α,β)-surfaces has advantages and disadvantages in the real image processing and some visually true digital surface (or surface points) is not included in any of (α,β)-surfaces (or surface points), we will give a more general definition for digital surface in both direct and indirect adjacency. In order to develop a fast algorithm for real surface tracking, a quasi-digital-surface is defined even though sometimes the definition is too weak in terms of mathematics.

2. (α,β)-Surfaces and Continuous Analogs

In this section, we introduce some basic concepts of digital geometry and topology. These concepts come from [1-2] . Let $\sum_3$ be a 3D grid array with consecutive integer coordinates. Sometimes people use Z^3 to represent this array in the infinite case. An element of $\sum_3$ is also called a point. Two distinct points $p=(x,y,z)$ and $q=(u,v,w)$ in $\sum_3$ are 6-adjacent if $|x-u|+|y-v|+|z-w|=1$. We also say q is a 6-neighbor of p. Two distinct points $p=(x,y,z)$ and $q=(u,v,w)$ in $\sum_3$ are 18-adjacent if $|x-u|+|y-v|+|z-w| \le 2$ and $\max\{|x-u|,|y-v|,|z-w|\}=1$. Two distinct points $p=(x,y,z)$ and $q=(u,v,w)$ in $\sum_3$ are 26-adjacent if $\max\{|x-u|,|y-v|,|z-w|\}=1$. A point that is not at the border of $\sum_3$ has six 6-adjacent points, eighteen 18-adjacent points, and twenty-six 26-adjacent points. We also call 6-adjacency direct adjacency.

Let S be a subset of $\sum_3$. Two distinct points p and q in S are α-connected (α=6,18, or 26) if there is a sequence of points (called a path) $p=p_0,p_1,\ldots,p_n=q$ such that p_i and p_{i-1} are α-adjacent where $i=0,\ldots,n-1$. It is easy to know α-connectedness is an equivalence-relation. So, using α-connectedness, any subset of $\sum_3$ can be partitioned into several α-connected subsets (called α-components). We say S is α-adjacent to a point p if a point in S is α-adjacent to p.

Let p be a point (we always assume p is not at the border of $\sum_3$), N_p denotes the set of points containing p and all 26-adjacent points of p. N_p is the neighborhood of p in terms of digital space. If p is in S, define $S(p) = S \cap N_p$.

Definition 2.1 (Morgenthaler-Rosenfeld [1]). A point p in S is a (α,β)-simple surface point (or (α,β)-surface point) if and only if (1) $S(p)$ is a α-component, (2) N_p-$S(p)$ has exactly two β-components and each of which is β-adjacent to p, (3) each of α-neighbors is β-adjacent to both β-components of N_p-$S(p)$. If every point of S is a (α,β)-surface point, then S is called a (α,β)-surface.

It is easy to know a (α,β)-surface is a closed surface. This definition was the first formal definition of digital surfaces. The definition provides a direction of theoretical studies on digital surfaces. "However, Morgenthaler, Reed and Rosenfeld defined their 'simple surface points' axiomatically, and it is difficult to

understand this concept just by reading the axioms" [2]. Kong and Roscoe embedded the Σ_3 into R^3 to "present a visual interpretation of (digital) surface points." Because some concepts defined in [2] are very complicated, we just extract the most useful and meaningful concepts for simplicity. However, the simplification may not be strict in mathematics. One could find a complete definition in [2]. The rest of this section will roughly review main concepts and results of their paper [2].

In [2], a polyhedral surface is defined as a set of triangles $\{T_i\}$ in R^3 satisfying (1) $T_i \cap T_j$ is an empty set, or a side or corner point of both T_i and T_j for $i \neq j$. (2) Each side of a triangle is a side of at most one other in the triangle set. A plate π is a polyhedral surface such that the polyhedral surface is in a unit-cub, the boundary of π, $\partial\pi$, is on the boundary faces of the unit-cub, and each vertex of $\partial\pi$ is on the corner of the unit-cub.

Let p be a point in Σ_3. A plate cycle at p is a sequence of distinct plate $\{\pi_i | i=0,...,n\}$ such that:

1) There is a sequence $\{e_i | i=0,...,n\}$ in which e_i e_{i+1} are distinct edges of π_i, $i=0,...,n-1$, e_n and e_0 are distinct edges of π_n, and p is an endpoint of each e_i.

2) If $i \neq j$, then $\pi_i \cap \pi_j$ is the union of a number of straight line segments each of which is an edge of both plates and a set of points each of which is a vertex of both plates.

3) Any edge of a π_i is an edge of at most one other π_j.

The concept of plate cycles plays a very important rule in [2]. It is also very intuitive to describe what does a point on a digital surface look like. A plate cycle of p is a two-dimensional neighborhood at p. However, this concept has no algorithmic advantages because to decide if a plate set is a plate cycle is time consuming. We will discuss this issue in Section 4.

In [2], Kong and Roscoe obtained a series of results on (α,β)-surfaces:

Proposition 2.1 (Case (α,26)) If $p \in S \subseteq \Sigma_3$, then p is an $(\alpha,26)$-surface point of S iff the following all hold:

(i) No unit-cub in N_p contains eight points in S.

(ii) $F_{26}(S(p))\langle p\rangle$ is the plate set of a single plate cycle at p.

(iii) If q is an α-neighbor of p that is contained in S, then q is a vertex of some plate in $F_{26}(S(p))\langle p\rangle$.

Here, $F_{26}(T)$ is defined as the set of 1×1 squares whose corners all lie in T. If G is a set of plates, $G\langle p\rangle$ is all plates containing p.

Proposition 2.2 (Case (α,18)) If p is an $(\alpha,18)$-surface point of a set S, and all α-neighbors of p that lie in S are also $(\alpha,18)$-surface points of S, then the following all hold:

(i) Each unit-cub in N_p contains at most six points in S.

(ii) $F_{18}(S(p))\langle p\rangle$ is the plate set of a single plate cycle at p.

(iii) If q is an α-neighbor of p that is contained in S, then q is a vertex of some plate in $F_{18}(S(p))\langle p\rangle$.

Conversely, if $p \in S \subseteq \Sigma_3$ and (i),(ii),(iii) all hold, then p is an $(\alpha,18)$-surface point of S.

Here, $F_{18}(T)$ is defined as the set of 1×1 squares or compound plates whose corners all lie in T. A compound plate CP is defined as (1) has six vertices in a unit-cube, (2) two vertices not contained in (1) have distance $\sqrt{3}$, (3) let g be the centroid of the unit-cube, $CP=\cup\{$triangle $(uvg) \mid u,v$ are 6-adjacent points in that 6 vertices$\}$. See Figure 2.1.

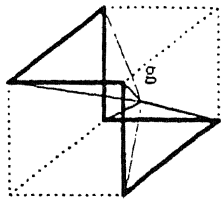


Figure 2.1 A compound plate in a unit-cube (3D-unit-cell)

One can see, a compound plate is just one possible interpretation for the six vertices set in a unit-cube. However, one can give other reasonable interpretations. Some of the interpretations do not satisfy Proposition 2.2.

Proposition 2.3 (Case (18,6)) Suppose that p is an (18,6)-surface point of a set S, and all 18-neighbors of p that lie in S are also (18,6)-surface points of S. Then the following all hold:

(i) Each unit-cub in N_p contains at most four points in S and every unit-cube containing four points in S is identical to unit-cube (a), (b), (c), or (d) in Figure 2.2.

(ii) No two unit-cubes sharing a face is as (f) in Figure 2.2.

(iii) $F_6(S(p))<p>$ is the plate set of a single plate cycle at p.

(iv) If q is an 18-neighbor of p that is contained in S, then q is a vertex of some plate in $F_{18}(S(p))<p>$.

Conversely, if $p \in S \subseteq \sum_3$ and (i),(ii),(iii), and (iv) all hold, then p is an (18,6)-surface point of S.

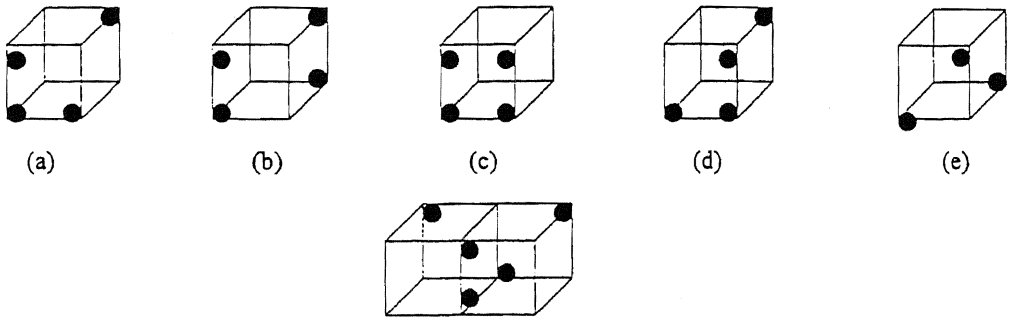


Figure 2.2 Points in S are marked ●

Proposition 2.4 (Case (26,6)) Suppose p is an (26,6)-surface point of a set S, and all 26-neighbors of p that lie in S are also (26,6)-surface points of S. Then the following all hold:

(i) Each unit-cub in N_p contains at most four points in S and every unit-cube containing four points in S is identical to unit-cube (b) or (d) in Figure 2.2.

(ii) $F_6(S(p))<p>$ is the plate set of a single plate cycle at p.

(iii) If q is a 26-neighbor of p that is contained in S, then q is a vertex of some plate in $F_{18}(S(p))<p>$.

Conversely, if $p \in S \subseteq \sum_3$ and (i),(ii),(iii), and (iv) all hold, then p is a (26,6)-surface point of S.

Here, $F_6(T)$ is defined as the collections of the following plates:

(1) Plate π is the union of triangles ΔABC and ΔBCD, where A, B, C, and D are the four points in a unit-cube of Figure 2.2 (a) which are in S, and $BC=\sqrt{3}$.

(2) Plate π is a $1\times\sqrt{2}$ rectangle whose corners are the four points in a unit-cube of Figure 2.2 (b) which are in S.

(3) Plate π is a $(\sqrt{2}, \sqrt{2}, \sqrt{2})$-triangle whose corners are the three points in a unit-cube of Figure 2.2 (e) which are in S.

(4) Plate π is a 1×1 rectangle whose corners are the four points in a unit-cube of Figure 2.2 (c) which are in S.

A (6,6)-surface does not have the same results above because there is no (6,6)-surface with finite points [2]. In addition, each (18,26) or (26,26)-surface must be a coordinate plane [2]. Each (26,18)-surface is a coordinate

plane or contains only compound plates. In the next section, we will discuss other trivial cases to reduce (α,β)-surfaces to real meaningful cases.

3. Nontrivial Cases of (α,β)-Surfaces

For the nine different categories of (α,β)-surfaces, α, β =6,18,26, Kong and Roscoe [2] already discovered some of the cases are not real exist or in some trivial forms. These cases are not really useful to image processing and computer graphics. These cases include (6,6)-surfaces, (18,26)-surfaces, and (26,26)-surfaces [2]. In addition, (26,18)-surfaces are either coordination plane or only consist of compound plates [2]. Therefore, (α,26)-surfaces can be reduced to only (6,26)-surfaces.

3.1 (6,18)-surfaces

This section will find more trivial (α,β)-surfaces. First, we consider (6,18)-surfaces and (18,18)-surfaces. Obviously, a (6,18)-surface is a (6,26)-surface, and it cannot contain any corner point as follows:

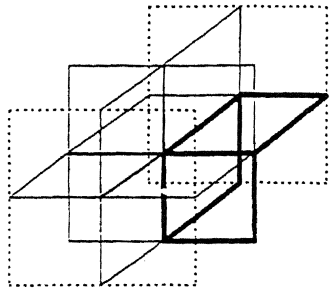


Figure 3.1 A corner surface point

Thus, a (6,18)-surface is a coordination plane or some simple "stage surfaces" shown in Figure 3.2.

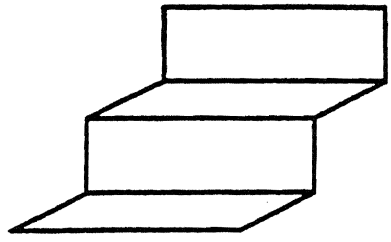


Figure 3.2 The general structure of (6,18)-surfaces

3.2 (18,18)-surfaces

For (18,18)-surfaces, we can have:

Proposition 3.1 If a compound plate PC shares an edge with a 1×1 plate PW, then there must be a point in $PC \cup PW$ that cannot be a surface point.

Proof: Assume a compound plate in S is as in Figure 3.3. Each edge has only two choices to be shared with a 1×1 plate. Therefore, suppose (p,a) is the edge. If $\{p,q,b,a\}$ is in S, then $\{c,d,p,q\}$ must be in S according to proposition 2.2. As we know, b is also a surface point, so $\{c, f, b, q\}$ is in S. Thus, p cannot be an (18,18)-surface point. We can get the same result for other cases.

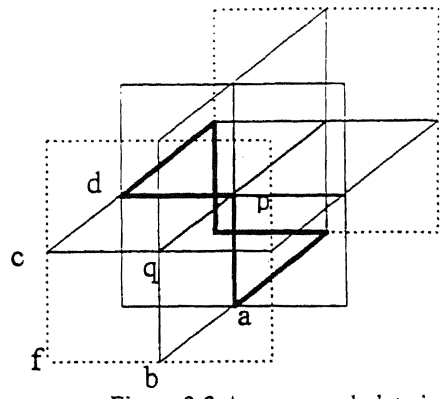


Figure 3.3 A compound plate in N_p

According to the above proposition, we have

Proposition 3.2 (18,18)-surfaces either only consist of 1×1 plates or only consist of compound plates.

One could easily prove that:

Proposition 3.3 If an (18,18)-surface only consists of 1×1 plates, then it is a coordinate plane.

Proposition 3.4 There is only one closed (18,18)-surface by compound plates (shown in Figure 3.4).

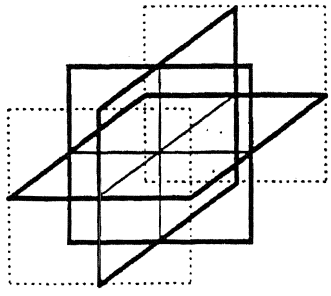


Figure 3.4 A closed (18,18)-surface

Proof: There are two methods to prove this proposition. For the first method, one can start with a compound plate and then try to link another compound plate, and so on. We can see there is only one solution like Figure 3.4. The second method is to use the Euler planar graph theorem "|edge set E|+2=|face set F|+|vertex set V|."

Let's consider the second method because it is more elegant.

Suppose that S is a closed (18,18)-surface by compound plates. S must be a planar graph. Thus,

$$|E|+2=|F|+|V|.$$

On the other hand, first, a face has six edges and an edge is included in exactly two faces, so

$$6|F|=2|E|.$$

Second, there are only two types of vertices. A vertex of the first type is contained by two faces, and a vertex of the second type is contained by four faces. Assume the set of the first type vertices is V_1 and the set of the second type vertices is V_2. Then,

$$|V_1| + |V_2| = |V| .$$

We can see, if two compound plates share an edge, then they share two edges. In addition, these two edges share a point which only be contained by two faces. So, a compound plate must share with three other compound plates. Therefore, in a face, there are three points in V_1 and three points in V_2. So,

$$3|F| = 2|V_1| \quad \text{and} \quad 3|F| = 4|V_2|.$$

We have $|V_1| = 2|V_2|$. We can get $|F| = 8$ and $|E| = 24$ and $|V| = 18$. That is the case of Figure 3.4.

Similarly, we can also prove that "there is only a type of closed (26,18)-surface by compound plates." Therefore, (α,18)-surfaces are not really useful to image processing.

3.3 (18,6)- and (26,6)-surfaces

A (26,6)-surface is an (18,6)-surface [2]. Since (α,18)-surfaces are useless surfaces, (α,26)-surfaces can be reduced to (6,26)-surfaces, and (α,6)-surfaces can be reduced to (18,6).

Therefore, there are only two really useful (α,β)-surfaces: (6,26)- and (18,6)-surfaces.

4. Algorithmic Considerations and Surface-Unit-Cell Traversal Algorithms

As we discussed above, Kong-Roscoe's analog results are very helpful for understanding, Morgenthaler-Rosenfeld's surfaces. An algorithm to find a plate cycle was discussed in [2]. However, if one performs this algorithm on each point of set S, that will be very slow because for each point one needs to find a plate cycle. We will show in this section that this algorithm is not faster than testing all conditions of Morgenthaler-Rosenfeld's surfaces as discussed by Reed and Rosenfeld. On the other hand, using continuous spaces to analog can help people to understand and reduce some ambiguities. However, discrete spaces and objects have their own properties. The continuous analog is just an single interpretation of a digital object, and it may have many interpretations.

Chen and Zhang's results [4-5] show that without using continuous analog, one still can get a true visually intuitive digital surface. Also, based on their definition, one can develop a fast global algorithm for surface decision and recognition. This algorithm does not work on each point but on a surface-unit. Chen and Zhang's results are totally built on digital spaces. That is much simpler than the analog process in [2]. In addition, Chen and Zhang's results can be used to determine an unclosed surface [4-5]. However, they only dealt with (6,26)-surfaces in [4-5]. In this section, we will show all three kinds of algorithms and analyze them.

4.1 Improved Morgenthaler-Reed-Rosenfeld's Algorithm

Morgenthaler-Reed-Rosenfeld's algorithm is to test all three conditions for each point in S in definition 2.1. The main task of the algorithm is to determine all "1" points is a α-component and all "0" points in N_p form two disconnected β-components. The fastest algorithm to search connected components can be developed by Tarjan's depth-first-search or breadth-first-search technique [6].

Even though one could use Tarjan's technique, this algorithm is still not very fast. The reason is that for each point one needs to find three components in N_p. Many steps are repeated because of adjacency in S. For example, assume p has four α-adjacent points q_1, q_2, q_3, q_4 in S, then the algorithm must reconsider p at least four times when the algorithm goes through q_1, q_2, q_3, q_4.

An improved Morgenthaler-Reed-Rosenfeld's algorithm: In this algorithm, we use C_1, C_2, to store two "0" value components (complement components in N_p) and A to store "1" value component. In order to reduce the repeat actions, after we finish a test (for three conditions in definition 2.1) for point p, we move to an α-adjacent point in S of p. Meanwhile, we do not empty C_1, C_2 and A, but update them.

Step 1. Assign a point of S to p, and let C_1, C_2, and A be empty sets. *STACK* is a stack. Get C_1, C_2, and A for p.

Step 2. Examine all three conditions in definition 2.1. If p is not a surface point, halt.

Step 3. Mark p, and push all unmarked α-adjacent points into the stack *STACK*.

Step 4. Pop up a point p' from *STACK* (this point is an α-adjacent point most of the time). If p' is adjacent to p, $N_p \cap N_{p'}$ has 18, 12, or 8 points depending on their adjacency. Get C_1, C_2, and A for p'. $p \leftarrow p'$.

Step 5. Goto Step 2 until *STACK* is empty.

4.2 Modified Kong-Roscoe's Algorithm

Kong and Roscoe showed an algorithm to find a plate-cycle for a point p in [2]. p is a surface-point if $\boldsymbol{P}\langle p\rangle$, all plates containing p, is a plate-cycle. Note, $\boldsymbol{P}\langle p\rangle$ is defined as 1x1 square for (6,26)-surface, and P<p> is defined as one of the cases in Figure 2.2 (a)-(e) for (18,6)-surfaces .The original Kong-Roscoe's algorithm is to find a largest plate cycle for a point p. We modified the algorithm and give the algorithm as follows [2]:

Modified Kong-Roscoe's algorithm (modified for simplification in this note):

Step 1. Let π' and π'' be in $\boldsymbol{P}\langle p\rangle$ and π' and π'' shares an edge containing p. Let π_0 be π' *and* π_1 be π''. Let e_0 be the common edge containing p.

Step 2. For i = 1,2,3,... if there exists an e_i and it is an edge of π_i that contains p but is different from e_{i-1}, and there exists a plate π_{i+1} different from π_i such that e_i is an edge of π_{i+1}. Until there is $m<i$ such that $e_i=e_m$.

Step 3. If all plates used in **Step 2** are equal to the set of $\boldsymbol{P}\langle p\rangle$, then p is a surface point. (One just needs to test if $|\boldsymbol{P}\langle p\rangle|$ is equal to m.)

This algorithm needs to get $\boldsymbol{P}\langle p\rangle$ for each p and store the set first. In Step 2, one needs to select an appropriate π_i . For (6,26)-surfaces, we can determine all 1x1 squares containing p to be $\boldsymbol{P}\langle p\rangle$ in N_p. For (18,6)-surfaces, it is not easy to determine $\boldsymbol{P}\langle p\rangle$ because there are many different kinds of plates. We can also see that this algorithm (in both cases) is not a fast algorithm.

Both Morgenthaler-Reed-Rosenfeld's algorithm and Kong-Roscoe's algorithm are not very fast because their definitions are based on local properties, i.e., they use surface-point to define a surface. In such a case, a local neighborhood N_p and $S \cap N_p$ must be considered extensively. In addition, such a definition cannot deal with boundary, so each surface under this definition must be a closed surface.

Chen and Zhang showed a definition of digital surfaces that surmounts these two problems [4-5]. We introduce their definition in the next subsection.

4.3 Surface-Unit-Cell Traversal Algorithm

Before we give the traversal algorithm, we introduce the definition of digital surfaces given by Chen and Zhang first [4-5].

Definition 4.1 Two 6-adjacent points is called a 6-line-unit-cell (or line-cell in this section). A line-cell $\{q,q'\}$ is called a parallel-move of line-cell $\{p,p'\}$ if p and p' are 6-adjacent to q and q' respectively; but neither p and q' nor p' and q are 6-adjacent.

We can see in Σ_3, each line-cell has four parallel-moves. A line-cell and each of its four parallel-moves form a surface-cell. Two surface-cells are line-cell-adjacent if they share a line-cell. Two surface-cells A and B are line-cell-connected if there is a surface-cell path. $A=P_0, P_1, \ldots, P_n=B$, where P_i and P_{i-1} are line-cell-adjacent.

Definition 4.2 A connected set S in Σ_3 is a digital surface if

(1) Each line-cell of S has at least one and at most two parallel-moves.

(2) Any two surface-cells of S are line-cell-connected in S.

(3) S does not contain any 3D-unit-cell (or unit-cube).

An algorithm based on Tarjan's breadth-first-searching technique can be developed to decide or track S in a

global way, which does not examine each N_p for S [5]. This algorithm traverses every surface-cell using a parallel-move of line-cell. We describe the algorithm below:

Surface-Unit-Cell Traversal Algorithm:
Step 1. Start with two 6-adjacent points $LC=\{p,q\}$ (a line-cell) in S.
Step 2. Put all (at most two if S is a surface, otherwise halt) parallel-moves of $\{p,q\}$ in a stack *STACK*.
Step 3. Mark line-cell $LC=\{p,q\}$. Pop up a line-cell to LC from *STACK*, repeat Step 2.
Step 4. S is a surface if all line-cells in S have been marked.

This algorithm is very simple and is optimized. However, there are two problems: (1) This algorithm cannot be used for (18,6)-surfaces; and (2) the surfaces defined by the definition 4.2 varies a little Morgenthaler-Rosenfeld's surfaces. We will consider the first problem in the next section. We could face the second problem below.

Definition 4.3 [4,7] Let p be a point of a surface S, p is regular if all surface-cells including p in S are line-connected. If p is both inner and regular, then p is called a regular inner surface point.

Theorem 4.1 [7] If S is a Morgenthaler-Rosenfeld's surface, then each point of S is a regular inner surface point. Conversely, if S is a closed surface (by Definition 4.2) and each point of S is a regular inner surface point, then S is a Morgenthaler-Rosenfeld's surface.

According to the proof of Theorem 4.1 (Theorem 4.1 in [7]), we can easily get:
Proposition 4.1 Let S be a surface by definition 4.2. If S does not contain any point p such that N_p is equivalent to Figure 4.1, then every point in S is regular.

Therefore, if we could quickly detect the case of Figure 4.1, we could complete the entire algorithm. A simple and fast way is described as follows and should be add to the end of the Surface-Unit-Cell Traversal Algorithm:

Step 5. Check all unit-cubes, each of which contains a point in S. If a cube contains seven points in S, assign p as the point whose diagonal point in the cube is not in S. If p has any 6-adjacent points not in the cube, then S is not a Morgenthaler-Rosenfeld's surface.

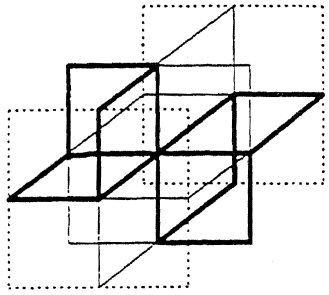


Figure 4.1 A non-regular inner surface point

We can see the Surface-Unit-Cell Traversal Algorithm is much simpler and faster than the other two. However, the disadvantage is that the Surface-Unit-Cell Traversal Algorithm is only for (6,26)-surfaces, but (6,26)-surfaces are the most important digital surfaces. In Section 5, we will discuss a more general definition of digital surfaces.

5. A New Definition of Digital Surfaces

Since (α,18)-surfaces are useless surfaces, (α,26)-surfaces can be reduced to (6,26)-surfaces, and (α,6)-surfaces can be reduced to (18,6)-surfaces. Therefore, there are only two really useful (α,β)-surfaces: (6,26)- and (18,6)-surfaces. On the other hand, there are many visually true surfaces (or surface points) that cannot be contained by any type of (α,β)-surfaces. For example, Figure 5.1 shows such a case *S(p)* in N_p. We can see *S(p)* is 18-connected (it is also 6-connected and 26-connected). β can not be 18 or 26; otherwise, there are no two β-components in N_p-*S(p)*. The only choice of β=6. If β=6, *q* can not β(=6)-adjacent to both β-components C_1 and C_2 in N_p-*S(p)*. So it is not any (α,β)-surface point.

Therefore, we can see that the (α,β)-surface is not a complete description of digital surfaces. A (α,β)-surface has blind sports and weakness to represent some digital surfaces, especially in computer graphics.

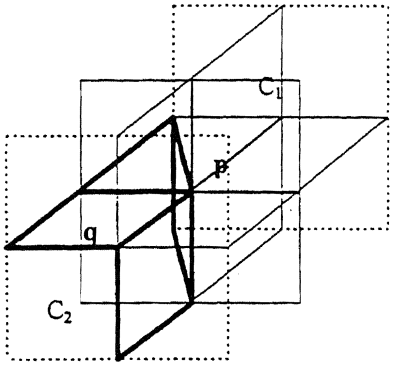
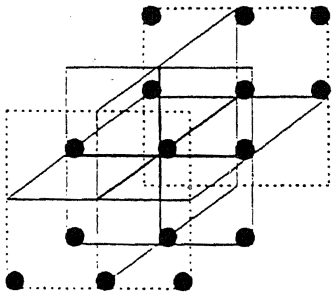
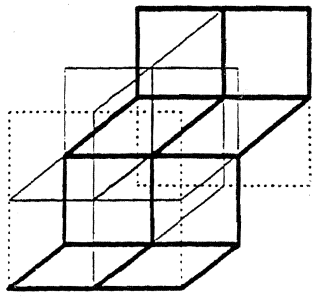


Figure 5.1 A visually true surfaces point, but not an (α,β)-surface point for any α and β.

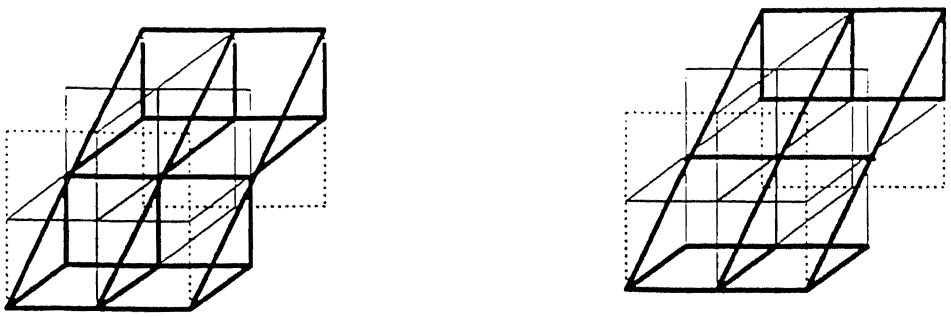
On the other hand, indirect adjacency causes ambiguous interpretations for a set S. Therefore, 18- or 26-adjacency introduces ambiguities for the interpretation of surfaces. For example, suppose that we allow 18-adjacency for a set S containing points as shown in the following Figure 5.2 (a). We can see that for 6-adjacency we have a unique interpretation. However, for 18-adjacency. We could have an interpretation like (b), which is a surface. If we interpret (a) like (c), it is not a surface. If we interpret (a) to be (d), it is a different surface.



(a)



(b)



(c) (d)

Figure 5.2 A digital points set in S and some of interpretations

All of these interpretations are reasonable under 18-adjacency. That is why a general solution allowing indirect adjacency is very difficult. As we know, there are only six types of simple surface points for direct adjacency [*], but it is really hard to get all simple surface points allowing indirect adjacency. The ambiguity is real existence, and one cannot reduce that to be unique. On the contrary, we can see "that is why indirect adjacency is much powerful."

This section will give a new definition of general digital surfaces for all 6-, 18-, and 26-connectivity. Based on the previous work of [1], [2], and [4], this definition contains (α,β)-surfaces to be special cases.

Definition 5.1 An α-surface-unit cell is a polygon on a plane in a unit-cube. Each pair of adjacent points on the polygon is α-adjacent. Such a pair is called an edge or line-unit cell of the α-surface-unit.

In addition, we could define an α-surface-unit is a minimized closed α-adjacent path, i.e., there is not pure subset of the path such that this subset can be a closed α-adjacent path.

Definition 5.2 A set S is a digital surface (α-surface) if there is a collection of α-surface-unit-cells in S, denoted by $U(S)$, satisfying:

(1) $S = \cup\{p \mid p$ is a point of an α-surface-unit-cells in $U(S)\}$,
(2) $U(S)$ is line-unit-connected in S.
(3) Each line-unit cell or edge of a surface-unit is contained by one or two α-surface-unit-cells in $U(S)$.
(4) S does not contain any unit-cube. (It is the only three-dimensional unit-cell we defined in this paper.)

Definition 5.3 A point p in an α-digital surface $(S, U(S))$ is a regular-surface point if the set of α-surface-unit-cells containing p is line-connected.

We can see a $U(S)$ is an interpretation of S. Using this definition, we still cannot get a fast algorithm easily. It is even worse in the design of algorithms. However, it provides a broad new area to study with. The following problems are meaningful.

(1) How many $U(S)$'s for a certain S ? What is the upper limit?
(2) How to find an optimal solution in some sense (e.g. minimum of area)?
(3) Is that finding all $U(S)$'s an NP-hard problem [6]?

We need to do more study to obtain a fast tracking algorithm. This definition is still good for representations of surfaces such as for computer graphics. However, in image processing, we sometimes do not desire a strict digital

surface, but rather a fast way to extract a surface. We can have a simple definition satisfying human intuition for surfaces. We use a different term: quasi-surfaces.

Definition 5.4 A point p in S is an α-quasi-surface point if
(1) $N(p) \cap S$ is an α-connected component.
(2) $N(p)$-S has two 6-connected components. (That is to say these two components are not 6-connected.)
(3) $N(p) \cap S$ does not contain any unit-cube (3D-unit-cell containing 8 points).

Definition 5.5 S is an α-quasi-surface if every point of S is an α-quasi-surface point.

It is not difficult to see,

Proposition 5.1 An (α,β)-surface is an α-surface by definition 5.2. Any point of an α-surface is an α-quasi-surface point by definition 5.4.

References

[1] D.G. Morgenthaler and A. Rosenfeld, Surfaces in three-dimensional images, *Information and Control*, Vol 51, 227-247, 1981.

[2] T.Y. Kong and A.W. Roscoe, Continuous analogs of axiomatized digital surfaces, *Computer Vision, Graphics, and Image Processing*, Vol 29, 60-86, 1985.

[3] M. Reed and A. Rosenfeld, Recognition of surfaces in three-dimensional digital images, *Information and Control*, Vol 54, 507-515, 1982.

[4] L. Chen and J. Zhang, Digital manifold: an intuitive definition and some properties, *Proceedings of the Second ACM/SIGGRAPH Symposium on Solid Modeling and Applications*, 1993.

[5] L. Chen and J. Zhang, A digital surface definition and fast algorithms for surface decision and tracking, in R.A. Melter and A.Y. Wu , *Vision Geometry* II , SPIE Proceedings 2060, 169-178.

[6] A. Aho, J. Hopcroft and J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison Wesley, 1974.

[7] L. Chen, D.H. Cooley, and J. Zhang, The equivalence between two definitions of digital surfaces, *Manuscript*, 1997.

[8] L. Chen and J. Zhang, Classification of simple digital surface points and a global theorem for simple closed surfaces, in R.A. Melter and A.Y. Wu , *Vision Geometry* II , SPIE Proceedings 2060.

Point Spaces and Raster Spaces in Digital Geometry and Topology

Li Chen
Scientific and Practical Computing
P.O. Box 6081
North Logan, UT 84341-6081
and
Department of Computer Science
Wuhan University
Wuhan 430072, China

ABSTRACT

In digital geometry and topology, there are two popular kinds of digital spaces: point spaces and raster spaces. In point-spaces, a digital object (e.g. a curve or surface) is represented by a set of elements (points). In raster spaces as defined in this note, a digital object is a subset of a "relation" on the space. In an Euclidean space, given a set S of points which are called sites, we can get the Voronoi diagram of S and its Delaunay triangulation (or simplicial decomposition in general). The Voronoi diagram is just a raster space as well as Delaunay simplicial decomposition is a point space. Thus, a point space is a dual space of a raster space. This note reviews some research results in point spaces and raster spaces and presents the author's opinions on the following problems: How to define digital curves, surfaces, and manifolds in point spaces or raster spaces? What are the difference and relationship between them? What are the advantages and/or disadvantages to use point spaces or raster spaces in practical computation? The purpose of the note is to show a global consideration and to unify some basic concepts in digital geometry and topology.

Keywords: Digital geometry and topology, point space, raster space, Voronoi diagram, Delaunay triangulation, curve, surface, manifold, algorithm, image processing.

1. Introduction

Discrete geometry deals with the geometric and topological properties of discrete objects or finite objects. Discrete geometry also includes packing, covering, tiling, convex polytops analysis, and computational geometry etc.[1] Digital geometry and topology is a sub-category of discrete geometry. Digital geometry and topology attempts to solve the theoretical problems of image processing, computer vision, and computer graphics in digital spaces.

Several methods of digital geometry and topology were proposed for image processing and computer vision because a two-dimensional (2D) image can be represented by a function on a two-dimensional array as well as a three-dimensional (3D) image can be represented by a function on a three-dimensional array in computers. In addition, such a function can be translated to a binary value $\{0,1\}$ function. Thus, to study geometric and topological properties for a connected "0" or "1" point set in a 2D array or 3D array has become an interesting research topic.[2,3]

Unfortunately, the concept of "digital geometry and topology" has not been well-recognized broadly. Three major reasons are:

(1)Lack of great significant discovery,

Part of the SPIE Conference on Vision Geometry VII • San Diego, California • July 1998
SPIE Vol. 3454 • 0277-786X/98/$10.00

(2)Not dispensable for most researchers in image processing and computer vision, and

(3)Lack of mathematical beauty.

For the first problem, if one can solve a very famous mathematical problem such as "four color problem" in digital geometry and topology, then digital geometry and topology will make a great impact. For the second problem, if we can find an efficient digital geometry method to track and register moving objects in computer vision, it will help to build the reputation of digital geometry and topology.

For the third problem, if we could simplify the terminology of digital geometry and topology, make its methods easier to understand, to use, and to implement, and make it more closer to traditional research fields such as discrete and computational geometry; then digital geometry and topology can attract more scientists and engineers to study and use it.

This note presents author's opinions on simplifying and unifying the basic concepts of two popular digital spaces, point spaces and raster spaces. (Note, if some of author's opinions was already appeared in some publications by others which is not cited in this note, please let the author know.)

2. Two Basic Digital Spaces in Euclidean Spaces

A real image is a continuous function, but it must be stored in a digital array in computers. So, we call the stored image a digital image. A digital space usually means a digital array. Mathematically, let E_m be the m-dimensional Euclidean space. Z^m is the set of all points $(a_1, ..., a_m)$ in E_m, where all $a_1, ..., a_m$ are integers. Z^m is called a digital space. Because the computer can only store finite amount of data, we usually consider a finite subset of Z^m, $\Sigma_m = \{(a_1, ..., a_m) | 0 \leq a_i \leq N; i = 1, ..., m\}$ for a certain N, as a digital space.

Digital geometry and topology deals with adjacency, connectivity, curves, surfaces, manifolds and etc. in Z^m. Unlike continuous geometry and topology, digital objects in digital spaces have concrete meanings. For example, a digital point may have length, digital curve may have area. Such properties are dependent on how to define a point, curve and so on. Thus, different interpretations in digital geometry and topology can yield different results.

Assume f is a two-dimensional image which is stored in a two-dimensional array in a computer. If (x, y) is an element in the array and $f(x, y)$ is the value on (x, y). Typically, $f(x, y)$ represents the average value of a unit square surrounding the point (x, y), and this unit square is called a pixel $Pixel(x, y)$. (A three-dimensional image can be represented by a function on a three-dimensional array, the value on a point is the average value of a unit cube called a voxel.)

Suppose that a connected region S in Fig. 2.1 (a) and we want to determine the boundary of S, then a problem will appear. What is the boundary of S? The most natural way is to consider that the boundary is B in Fig. 2.1 (b), but how to represent B using pixels? In this case, a piece of line is the intersection of two adjacent pixels, then we can use a set of pixel-pairs to represent the boundary. That is to say a line-cell (1-cell) is an element of a binary relation on Σ_2. By the way, a point (0-cell) is represented by the intersection of four pixels. However, in continuous spaces, the boundary is a curve. In order to keep consistency, we represent a curve by using a set of pixel-pairs. Such a digital space is called a raster-space in this note.

On the other hand, we can defined the boundary of S as C shown in Fig. 2.1 (c). That is to say, a curve can be represented by a set of pixels (or points) in the 2D array. Without loss generality, we can directly use a point (x, y) to represent a pixel and omit the $Pixel(x, y)$. This digital space is called a point space. A line-cell consists of two adjacent points in this case. **In this note, the difference between a point space and a raster space is how to define i-dimensional cells (or i-cells) for $i = 0, ..., m$ in an m-dimensional space.** We will define i-cells for both "point spaces" and "raster spaces" in this section. In addition, we will introduce the "connectedness" in

this section too.

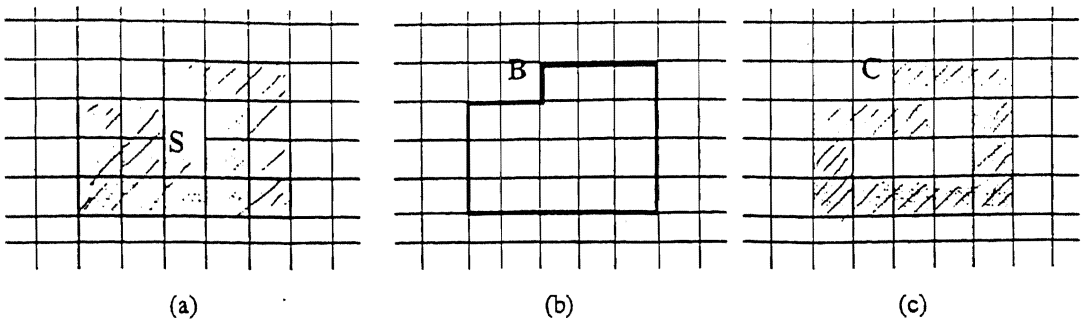


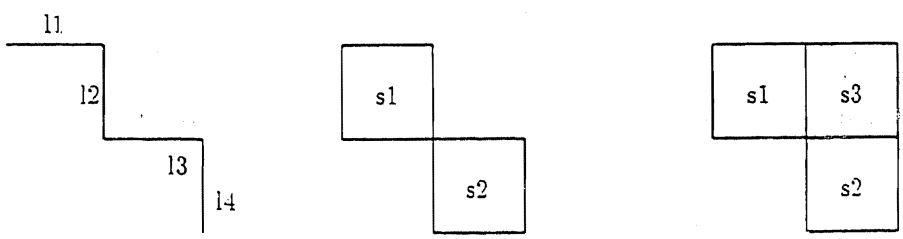
Fig. 2.1 (a) A connected set S;(b) B to be the boundary of S; (c) C to be the boundary of S.

2.1 Point Spaces in Direct Adjacency[4,5]

A point p in Σ_2 has four horizontal and vertical neighbors, namely $(x \pm 1, y)$ and $(x, y \pm 1)$; p also has four diagonal neighbors, namely $(x \pm 1, y \pm 1)$ and $(x \pm 1, y \pm 1)$. We say that horizontal and vertical neighbors are directly adjacent to p (or 4-adjacent to p), and we say that both types of neighbors are adjacent to p (or 8-adjacent to p) [2].

In general, the points $p = (x_1, x_2, ..., x_m)$ and $q = (y_1, y_2, ..., y_m)$ are two directly adjacent points in Σ_m if $d_D(p, q) = \sum_{i=1}^{m} |x_i - y_i| = 1$. We say p and q are general(directly or indirectly)-adjacent points if $d(p, q) = \max_{1 \le i \le m} |x_i - y_i| = 1$. We only consider direct adjacency in this sub-section. Let p and q be a pair of (directly) adjacent points of Σ_m. Intuitively, $\{p, q\}$ is a line-cell. A surface-cell is a set of 4 points that form a unit square parallel to coordinate planes. A 3-cell (or solid-cell) is a unit cub that has 8 points.

More generally, a k-cell (k-cub) has exactly 2^k points, $p_1, ..., p_{2^k}$ in direct adjacency, where each of $p_1, ..., p_{2^k}$ has m components, denoted by $p_i = (x_1^{(p_i)}, ..., x_m^{(p_i)})$. There are $m - k$ components that do not change value for all $p^{(1)}, ..., p^{(2^k)}$. For p_i and p_j, $i \neq j$, $1 \le \sum_{t=1}^{m} |x_t^{(p_i)} - x_t^{(p_j)}| \le k$, and $\max_{t=1}^{m} |x_t^{(p_i)} - x_t^{(p_j)}| = 1$. Also, we sometime call a 0-cell a point-cell, 1-cell a line-cell, and 2-cell a surface-cell. Two unit-cells u and v are point-adjacent if they share a point, line-adjacent if they share a line-cell, and k-adjacent if they share a k-cell. u and v are called k-connected in S if there exists a path $u_0 = u, u_1, ..., u_n = v$ in S, such that u_i and u_{i+1} are k-adjacent, $i = 0, ..., n - 1$. Figure 2.1 shows some examples of adjacency and connectedness.



point adjacency and connectedness line adjacency and connectedness

Fig. 2.2 Adjacency and connectedness in point spaces

2.2 Raster Spaces in Direct Adjacency[5,7]

A raster space, R_m, is partition (or decomposition) of E_m. Such partition splits E_m into m-dimensional unit-cells (m-cells) such that a such unit-cell contains a point of Z^m in the center of the cell.

For two distinct m-cells r and r', if $r \cap r'$ is an $(m-1)$-cell then the $(m-1)$-cell is called a surface-element[5,7]. So, a surface-cell can be represented by an element of a binary relation on R_m. Note, a surface-element is in $(m-1)$-dimension. In order to keep the consistency in this note, we will call it as $(m-1)$-cell. If $m = 3$, then a "surface-element" is a surface-cell.

In general, for 2^k m-cells $r_1, ..., r_{2^k}$, if $\cap_{i=1}^{2^k} r_i$ is an $(m-k)$-cell, then, an $(m-k)$-cell can be represented by an element of a 2^k-relation on R_m. A 2^k-relation in direct adjacency can be defined as follows:

$$\mathcal{R}^{(m-k)} = \{r = (r_1, ..., r_{2^k}) | r_1 \cap .... \cap r_{2^k} \text{ is an } (m-k)\text{-cell } \}.$$

Note, $(r_1, ..., r_{2^k}) = (p_1, ..., p_{2^k})$, where $\{p_1, ..., p_{2^k}\}$ is any permutation of $\{r_1, ..., r_{2^k}\}$. Such a relation is called a "symmetric relation."

We can see, although R_m can also be represented by Z^m. However, R_m has different implication as Σ_m. If we say the definition of k-cell in point spaces is "bottom to top," then the definition of k-cell in raster spaces is "top to bottom." They are reversed. In fact, let $P = \{p | p \text{ is 0-cell in } R_m\}$, then we can generate a similar k-cell structure in point-spaces.

For two t-cells, $t > k$, u and v are directly k-adjacent if $u \cap v$ contains a k-cell. u and v are called directly k-connected in S if there exists a path $u_0 = u, u_1, ..., u_n = v$ in S, such that u_i and u_{i+1} are k-adjacent, $i = 0, ..., n-1$.

It is not difficult to know R_m is the Voronoi decomposition (diagram) of "sites" Z^m. If we link every pair of points which are $(m-1)$D-adjacent in Z^m, then we get a Delaunay "triangulation" of the Voronoi diagram. Therefore, the point space Z^m is a dual space of R_m. We will discuss more in Section 5.

Fig. 2.3 shows 2D raster and point spaces and their 0-cells, 1-cells and 2-cells with direct adjacency in Fig. 2.3. We will consider indirect adjacency in Section 3.

3. Digital Manifolds in Point Spaces and Raster Spaces

In point spaces, a k-dimensional digital object (i.e. digital curve, surfaces, and manifolds ect.) is represented by a set of elements in Z^m. In raster spaces, a k-dimensional digital object is represented by a set of elements in $2^{(m-k)}$-relation on R_m.

3.1 Digital Manifolds in Point Spaces[4,5]

Definition 3.1 Let A be a subset of Σ_m. A' is called a parallel-move of A if

(1) $|A| = |A'|$.

(2) $A \cap A' = \phi$, and

(3) there exists a bijection mapping $f : A \to A'$ such that $d(a, f(a)) = 1$ (d is the distance between a and $f(a)$) for all $a \in A$. In addition, a and b are adjacent in A if and only if $f(a)$ and $f(b)$ are adjacent in A'.

Definition 3.2[4] A connected subset M of Σ_m is a k-dimensional digital manifold if any point $p \in M$ is included

in some k-cell of M, and

(1) any two k-cell in M are $(k-1)$-connected in M,

(2) every $(k-1)$-cell in M has only one or two parallel-moves in M, and

(3) M does not contain any $(k+1)$-cell.

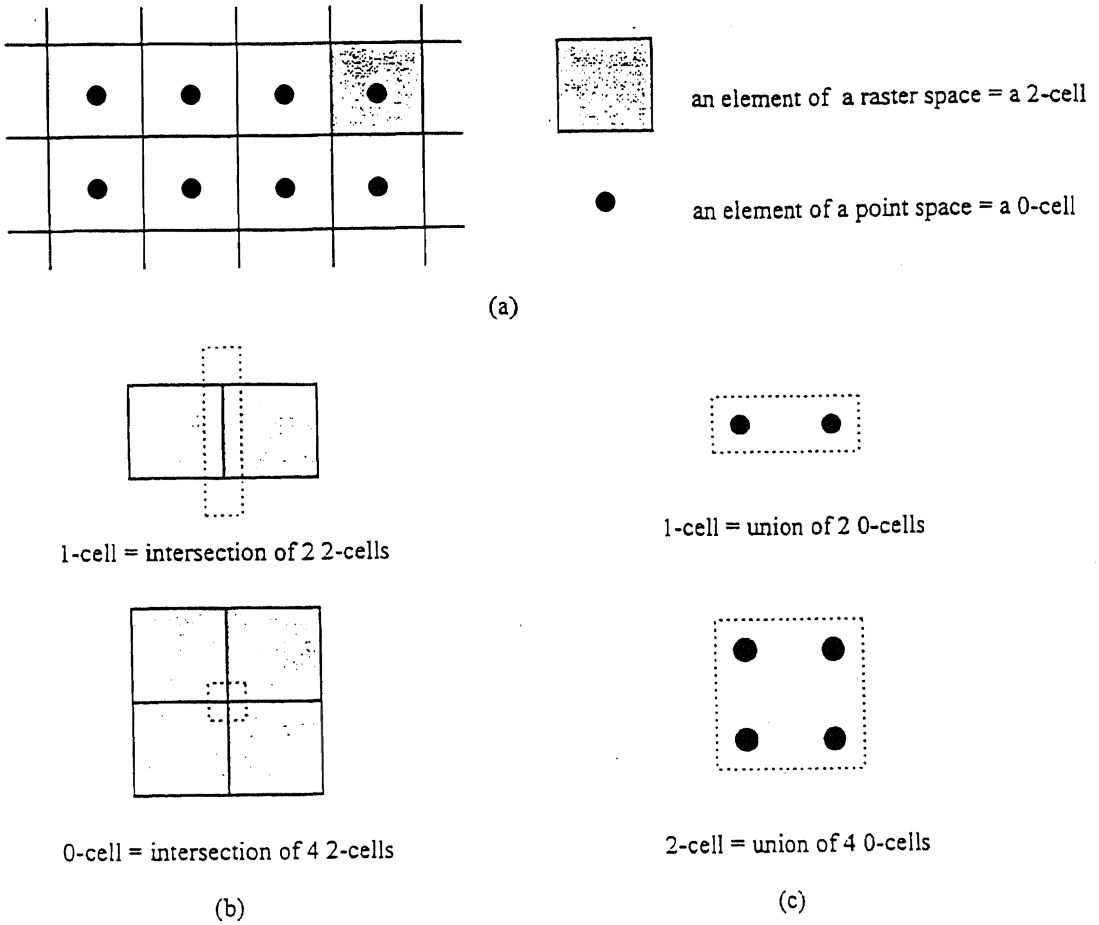


Fig. 2.3 0-cells, 1-cells, and 2-cells in 2D.

3.2 Digital Manifolds Raster Spaces[6,7]

According to the defintions in Section 2.2,

Definition 3.3 [6,7] Given R_m and $\mathcal{R}^{(m-1)}$, a nonempty subset of $\mathcal{R}^{(m-1)}$ is called a (super)-surface.

Herman presented a definition for any set V and a binary relation π on V [6,7]. We will discuss it later. Definition 3.3 is very general, but it is very weak. For a surface in 3D, it is too easy to break the rule of "a neighborhood of any point on the surface is homeomorphic to a 2D disk" in mathematics.

Definition 3.3 can be modified and extended as follows.

Definition 3.4 A k-D simple digital manifold is a nonempty subset of $\mathcal{R}^{(k)}$, denoted by S. And S satisfies:

(1) S is $(k-1)$-connected, i.e. any two k-cells in S are $(k-1)$-connected in S, and

(2) any $(k-1)$-cell in S (which means it contained by a k-cell in S) is contained by at least one and at most two k-cells.

This definition is similar to Definition 3.2.[8] Some similar ideas of Definition 3.4 was also appeared in Herman and Webster[9] and Kong and Roscoe[10] in 3D.

3.3 Allowing indirect adjacency

In fact, definition 3.2 and definition 3.4 only allow direct adjacency. For example, in 3D, a surface-cell in point spaces is defined as four points in a plane which is parallel to a coordinate plane. On the other hand, a surface-cell in raster spaces is also parallel to a coordinate plane. These definitions have an important property: A closed $(m-1)$-dimensional sub-manifold of an m-dimensional manifold divide it into two (or more in digital spaces) connected parts. Such property is called Jordan property.

However, in real image processing, we must consider indirect adjacency, such as 8-adjacency in 2D, and 18- or 26-adjacency in 3D. Allowing indirect adjacency will give you much power to represent complex shapes in image processing and graphics. However, it will also introduce ambiguity and non-Jordan cases.

For 3D point spaces, direct adjacency is also called 6-adjacency because of only six points directly adjacent to a point. Using the same way one can define 18, 26-adjacency and 18, 26-connectedness.[10,11] A definition of general digital surfaces in 3D point spaces was given by Morgenthaler and Rosenfeld[11]. The digital surfaces are called (α,β)-surfaces, $\alpha,\beta = 6, 18, 26$. Let S be a subset of Σ_3. If p is a point (element) of S, then N_p is the $3 \times 3 \times 3$ array containing p at the center. We also denote $S(p) = S \cap N_p$.

Definition 3.5[10,11] A point p in S is an (α,β)-(simple) surface point if

(1) $S(p)$ is an α-component,

(2) $N_p - S(p)$ has exactly two β-components and p is β-adjacent both β-components ,

(3) each of α-neighbors is β-adjacent both β-components of $N_p - S(p)$.

If every point of S is an (α,β)-surface point then S is called an (α,β)-surface.

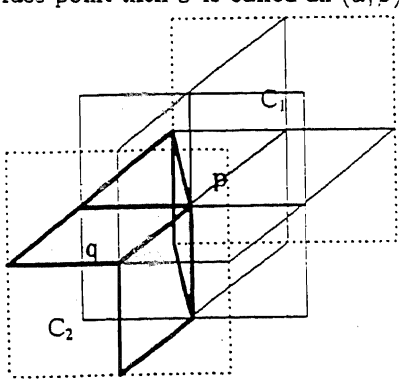


Fig. 3.1 A visually true surface-point but not an (α,β)-surface point.

This definition consider total nine cases, which contain not only direct adjacency ((6,26)-surfaces), but also other eight indirect adjacency. Many people thought that this definition is well-established and complete. In addition, a type of continuous analogs for each cases has been found by Kong and Roscoe.[10]

However, Chen found an anti-example to show that a simple and "visually true" surface point is not included in (α, β)-surfaces.[12] See Fig. 3.1. On the other hand, in the total nine variations, only (6,26)- and (18,6)- surfaces have non-trivial examples.[10,12]

Therefore, to give a complete definition of general digital surfaces (allowing indirect adjacency) is still an open problem.[12]

For raster spaces, it may be even worse. How to define a "face" (or 2-cell) between two voxels which share only a line-cell or a point-cell? It is hard to find an intuitive meaning for such a face.

4. The Relationship Between Point Spaces and Raster Spaces

This section presents three interesting facts in between a point space and a raster space. First, a raster space R_m with defined k-cells is a dual space of a point space. A raster-space is a Voronoi diagram, and its dual point-space is the Delaunay simplicial decomposition of the Voronoi diagram.[13] Herman used Voronoi diagrams to describe the raster space R_m.[6,7]

Second, for 3D point spaces, Chen and Zhang obtained: Simple surface points in $(6, 26)$-surfaces have exact six types.[14]. A regular inner surface point based on Definition 3.2 was introduced to get the results.[4,5] A $(6, 26)$-surface point defined by Definition 3.5 is equivalent to a regular inner surface point. Here, "regular" means all surface-cells containing p in S are line-connected in $S \cap N_p$.

For 3D raster spaces, if V is a subset of R_3, i.e. V is a set of voxels. Latecki found if V is well-composed in R_3, for every points p on the boundary surface S of V, $S \cap N_p$ is just one of the six types of simple surface points.[15] "Well-composed" can be defined as follows: Let v be a vertex, and there are eight voxels in R_3 containing v. If V contains two of the eight voxels, then these two do not share only the vertex v.

This fact provides us a very positive feeling to use point-cells (0-cell) in a raster space to represent the raster space by using point spaces' methods.

Last, let us consider to use a point space to represent a raster space. Using 0-cell in raster spaces, we can define a topological structure (such as adjacency and connectivity) in the same way we did in point spaces in Section 2.1. However, the following example will show that the two representations are not equivalent (Fig. 4.1).

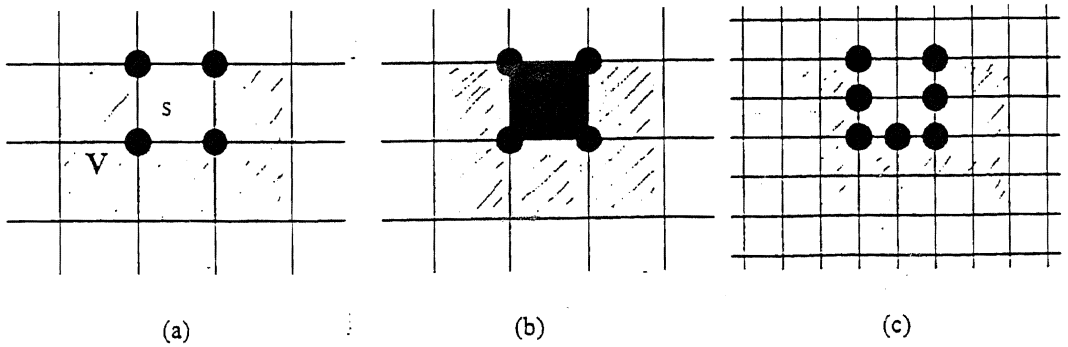


Fig. 4.1 Can a set in a raster space be represented by its 0-cells? (a) The pixel Set V, $s \notin V$; (b) s must be in V if using 0-cells to define a 2-cell in point spaces; (c) a refined V.

In the raster space in Fig. 4.1 (a), pixel s is not in set V, but if using point-cells to represent V, then the surface-cell s must in the translated object of V because all four corner points are in the object (see Fig. 4.1 (b)). However, if we refine the original raster space by 2, i.e. divide a pixel into 4 pixels in 2D and a voxel into 8 voxels in 3D, etc., we can get the exact same representation of original V by using 0-cells in refined raster spaces. (See Fig. 4.1 (c)) On the contrary, such mathematical equivalence may not be efficient in term of the costs of computing.

5. Discrete Surfaces and Manifolds in Voronoi Diagrams and Delaunay Decompositions

Given n points in the m dimensional Euclidean space. We can get a Voronori Diagram and Delarney triangulation (Simplicial Decomposition). To define a digital surface based on the structures of Voronoi Diagram and Delaunay triangulation are easier than the cases in indirect adjacency. The reason is that the structures of Voronoi Diagram and Delaunay triangulation have Jordan property. (See Section 3.3.)

Let P be a set of points in Euclidean space E_m . P is called sites. "The Voronoi diagram of P splits E_m into regions with one region for each site, so that the points in the region for site $p \in P$ are closer to p than to any other site in P." [13]

The Delaunay triangulation of P is the unique triangulation of P so that there are no elements of P inside the circumsphere of any triangle. Here "triangulation" has an extended meaning: it decomposes the convex hull of P into simplices using S as vertices. [13] In addition, the Delaunay triangulation need not be a proper triangulation.[13] In this note, we sometime use the term of "Delaunay decomposition" instead of Delaunay triangulation. (See Fig. 5.1 and Fig. 5.2 [13])

Voronoi diagrams is a central subject in computational geometry. Some efficient algorithms have been developed to compute the Voronoi diagram and Delaunay decomposition.[1,13]

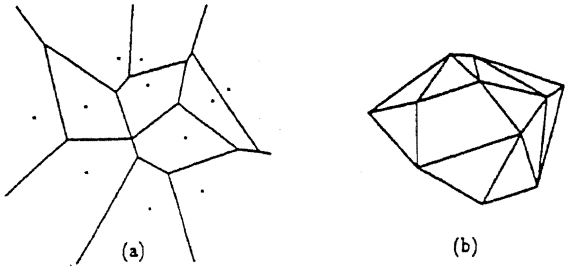


Fig 5.1 (a) A Voronoi diagram in dimension 2; (b) the Delaunay triangulation of the same sites

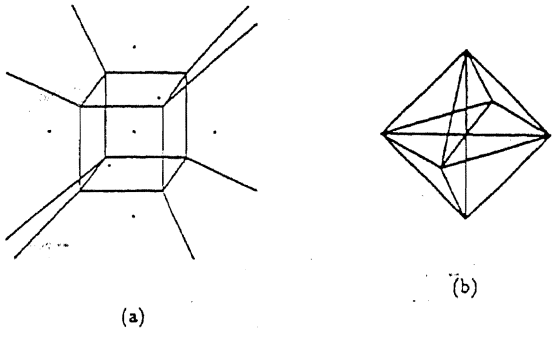


Fig 5.2 A Voronoi diagram in dimension 3; (b) the Delaunay triangulation of the same sites

In this section, we will discuss how to define a discrete surface (or manifold) in Voronoi diagrams or Delaunay decompositions. In order to eliminate some unusual cases, we do not consider any cells and points on the border of

Voronoi diagrams or Delaunay decompositions.

5.1 Point Spaces Method: Bottom-Up Method

Point spaces method is much like the method we used in Section 3.1. A general approach was considered by Chen [8], and it can be used in Voronoi diagrams and Delaunay decompositions. In 3D, a Voronoi diagram is a special "Graph" $G = (V, E)$, where V contains all vertices of the Voronoi diagram as well as E contains all Voronoi edges (Voronoi 1-cells). A Delaunay decomposition is also a special "Graph" $G = (V, E)$, where V is the set of sits, and E is the set of Delaunay edges (Delaunay 1-cells). We also have Voronoi 2-cells, Voronoi 3-cells, Delaunay 2-cells, and Delaunay 3-cells.

For a graph G, let $\tau(G)$ be a group of defined sets of 0-cells (V), 1-cells (E), 2-cells,..., k-cells,.... $\tau(G)$ is called a topological structure of G.

Definition 5.1 Given a G and $\tau(G)$. Subgraph D of G is a k-dimensional digital manifold (or k-manifold) if and only if

(1) any two k-cells in D are $(k-1)$-cell-connected in D,

(2) each $(k-1)$-cell in D is included in one or two k-cells in D,

(3) D does not contains any $(k+1)$-cell.

The boundary of a k-manifold D, denoted by ∂D, is a subset of $(k-1)$-cells contained in D so that each element of ∂D is exactly contained by one k-cell in D.

Note, for a general graph, to give a topological structure $\tau(G)$ is not an easy task.[8]

5.2 Raster Spaces Method: Top-Down Method

Raster spaces method is much like the method we used in Section 3.2. A general approach was considered in a communication between Herman and Chen.

Let V be a set, R be a symmetric binary relation, and τ be a partition of R. $H = (V, \tau)$ is a raster digital space. When $\tau = R$, i.e., partition to single element, then this definition is equivalent to the original definition by Herman.[7]

Definition 5.2[7] A n-D simple digital manifold is a subset S of V, and for any two elements $p, q \in S$, there is a path $(p, p_1), ..., (p_{n-1}, q) \in \tau_S$.

S is called Simply $(n-1)$-D connected if for any two elements $p, q \in S$, there is a path $(p, p_1), ..., (p_{n-1}, q) \in \tau_S$.[7] In fact, Simply $(n-1)$-D connectedness is equivalent to direct connectedness in Section 2.

Let $V, R, ..., R^{(n)}, \tau, ...\tau^{(n)}$, where V is a set and R is a symmetric binary relation on V, $R^{(i)}$ is a binary relation on $R^{(i-1)}$, $\tau^{(j)}$ is a partition of $R^{(i)}$, then we say $G = (V, R, ..., R^{(n)}, \tau, ..., \tau^{(n)})$ is a n-dimensional topological structure on V. Given a topological structure $G = (V, R, ..., R^{(n)}, \tau, ..., \tau^{(n)})$. Let S be a subset of V and

$$G_S = (S, R_S, ..., R_S^{(n)}, \tau_S, ..., \tau_S^{(n)}),$$

where $R_S^{(i)} = \{r = (a, b) | (r \in R^{(i)}) \& (a, b \in R_S^{(i-1)})\}$, and $\tau_S^{(i)} = \{r = (a, b) | (r \in \tau^{(i)}) \& (a, b \in \tau_S^{(i-1)})\}$.

If S is a subset of $\tau^{(i)}$, then for $j > i$, and let $S = \tau_S^{(i)}$, $R_S^{(j)} = \{r = (a,b) | (r \in R^{(j)}) \& (a,b \in R_S^{(j-1)})\}$, and $\tau_S^{(j)} = \{r = (a,b) | (r \in \tau^{(j)}) \& (a,b \in \tau_S^{(j-1)})\}$.

The boundary of S is defined as $\partial S = \{r = (a,b) | (r \in \tau^{(i+1)}) \&$ just one of $a, b \in S\}$.

Definition 5.3 A i-dimensional manifold is a subset S of $\tau^{(n-i)}$, and

(1) S is $(i-1)$-cell connected, i.e., for any two element $p, q \in R_S^{(n-i)}$ there is a path $(p,p_1),...,(p_{n-1},q) \in R_S^{(n-(i-1))}$.

(2) $\tau_S^{(n-(i-1))} = R_S^{(n-(i-1))}$, i.e., $\tau_S^{(n-(i-1))}$ is the single element partition. (Intuitively, any $(i-1)$-cell (p,p') is contained by at most two i-cells p, p' in S.)

For Voronoi diagrams and Delaunay decompositions, we can naturally get their topological structures $G = (V, R, ..., R^{(n)}, \tau, ..., \tau^{(n)})$.

5.3 Advantages and Disadvantages

Raster spaces as defined in this note are more meaningful in the nature, but they are difficult to represent in computers. On the other hand, point spaces are easy to represent, but sometime it is hard to match the pure mathematical meaning. They are dual spaces each other, so they have the "dual properties."

6. Algorithms

Boundary tracking of a solid object is probably the most important application of digital geometry and topology. A solid object is "connected" but it may have holes inside. For raster spaces, an optimal algorithm described by Herman to find a connected-component in the boundary (surface) of a solid object.[7] This algorithm is called "Fat-Flies," and it is easy to see that the algorithm is optimal because "Fat-Flies" used the same strategy as the "breadth-first-search" technique discovered by Targan.[16]

Some fast algorithms were discussed in point spaces [5,12], such as surface recognition and boundary tracking using breadth-first-search.

In fact, for raster spaces or point spaces, a 2-cell (surface-cell) is on the boundary of a solid object iff the 2-cell is contained by just one 3-cell in the object. This property is used by Chen to extract all boundary elements in a solid object.[8]

If an image is represented by a quadtree in 2D or an octree in 3D. A leaf can be defined as a 2-cell in 2D (a 3-cell in 3D) by Chen.[8] To extract all boundary elements is to find all 1-cells in 2D (2-cell in 3D) which is included in just one leaf in the object. This algorithm uses the properties in both raster spaces and point spaces.

A fast algorithm is also designed by Chen to decide if a surface is orientable in point spaces.[8] This algorithm with little modification can use to decide if a surface is orientable in raster spaces.

There are many real image processing problem in which we may use digital geometry and topology to develop efficient solving algorithms. Such as object shape analysis, target tracking, and image compression.

Remarks: Dr. G. Herman have discussed with the author on the problems of digital spaces (raster spaces in this note), and he provided the author an earlier version of his new book.[7] To give a general digital surface definition was also suggested by him. Dr. L. Latecki sent the author his research paper which pointed a path for the author's study.[15] This paper might be the first paper to deal with the relationship between a raster space and a point space.

Dr. H. Zhu talked with the author about computational geometry so that the author can study the concepts of Voronoi diagrams and Delaunay decompositions recently, and use them in this note.

References

1. J. E. Goodman and J. O'Rourke eds, "Handbook of Discrete and Computational Geometry, " CRC Press, New York, 1997.

2. A. Rosenfeld, "Digital topology," Amer. Math. Monthly, Vol 86, pp 621-630, 1979.

3. E. Artzy, G. Frieder and G. T. Herman, "The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm," Comput. Vision Graphics Image Process. Vol 15 , pp. 1-24, 1981.

4 L. Chen and J. Zhang, "Digital manifolds: A Intuitive Definition and Some Properties", The Proc. of the Second ACM/SIGGRAPH Symposium on Solid Modeling and Applications, pp. 459-460, Montreal, 1993.

5. L. Chen and J. Zhang, "Digital surface definition and the fast algorithms for decision and tracking", Melter and Wu ed, Vision Geometry II, SPIE 2060, pp. 169-178, 1993.

6. G.T. Herman, "Oriented surfaces in digital spaces," CVGIP: Graphical Models and Image Processing, Vol 55, No 5, pp 381-396, 1993.

7. G.T. Herman, "Geometry of Digital Spaces," Birkhauser, Boston, 1998.

8. L. Chen, "Generalized discrete object tracking algorithms and implementations," Melter, Wu, and Latecki ed, Vision Geometry VI, SPIE 3168, pp 184-195, 1997.

9. G.T.Herman and D. Webster, "A topological proof of a surface tracking algorithm," Comput. Vision Graphics Image Process. 23, 1983, 162-177.

10. T.Y. Kong, and A.W. Roscoe, "Continuous analogs of axiomatized digital surfaces," Computer Vision, Graphics, and Image Procssing, Vol 29, 60-86, 1985.

11. D. G. Morgenthaler and A. Rosenfeld, "Surfaces in three-dimensional images", Inform. and Control Vol 51, pp. 227-247, 1981.

12. L. Chen, A note on (α, β)-type digital surfaces and general digital surfaces, in Vision geometry VII, too appear.

13. S. Fortune, Vororoi diagrams and Delaunay triangulations, in D.Z. Du and F. K. Hwang ed, Computing in Euclidean Geometry, World Scientific Publishing Co, 1992.

14. L. Chen and J. Zhang. "Classification of Simple digital Surface Points and A Global Theorem for Simple Closed Surfaces", Melter and Wu ed, Vision Geometry II, SPIE 2060, pp. 179-188. 1993.

15. L. Latecki, "3D well-composed pictures", Melter and Wu ed, Vision Geometry IV, SPIE 2573, pp. 196-203,1995.

16. A. Aho, J. E. Hopcroft, and J. D. Ullman, "The Design and Analysis of Computer Algorithms, " Addison-Wesley, Reading, MA. 1974.

The Application of A Fuzzy Object Search Technique to Geophysical Data Processing

Li Chen Frank T. Berkey
Center for Atmospheric and Space Sciences
Utah State University
Logan, UT 84322-4405

Steve A. Johnson
Department of Bioengineering
The University of Utah
Salt Lake City, UT 84112

ABSTRACT

This paper describes the application of a fuzzy search technique to geophysical data processing. Here, we discuss three kinds of image object search problems: velocity layer search in Computerized Tomography in cross wells, surface search and reconstruction in three dimensional seismic images, and critical frequency curve search in digital ionograms. These three kinds of images have a common property: there are no crisp boundaries in these objects. For the first two applications, we want to search the small two dimensional regions called layers in a given image and apply a fuzzy segmentation technique called λ-connected segmentation. For the third application, we want to examine all of the reasonable curves in an ionogram; therefore, we have developed a dynamic λ-connected search technique which is combined with the use of genetic algorithms.

In this paper, we show: If the fuzzy connectedness function (E) between two pixels in an image has the properties (i) $E(u,u) = 1$, (ii) $E(u,v) = E(v,u)$, and (iii) $E(u,v) \geq min\{E(u,w), E(w,v)\}$, then the region growing technique using the fuzzy connectedness function E makes an equivalence classification for the image. If two pixels, u and v are λ-connected, which are related to a certain real number r, then $E(u,v) \geq r$. We can conclude that a fuzzy system approach has important advantages for geophysical data processing.

1. INTRODUCTION

The application of image processing and pattern recognition techniques to automate the process of object searches in geophysical data processing is currently very popular. In seismic image processing, several techniques have been used to find the events/layers in seismic profiles [1,2]. Using computerized tomography, techniques have been developed to search for the boundary curves and surfaces [3,4], and in ionosheric research, an automatic processing system has been developed to identify the virtual-height frequency curves in an ionogram [5].

In most cases, geophysical data and images have considerable uncertainty, and fuzzy boundaries between the different objects. In this paper we describe our application of a fuzzy segmentation and search technique to three image object search problems: velocity layer search in the computerized tomography of cross wells, surface search and reconstruction in three dimensional seismic images, and the identification of the critical frequency trace in digital ionograms.

 0-8194-1475-1/94/$6.0

In the first application, we search small, two-dimensional ($2D$) regions called layers in a velocity profile and apply a fuzzy segmentation technique called λ-connected segmentation. For the second application, we also apply λ-connected segmentation, then combine and reconstruct the boundary curves of these layers to form the surfaces of a three dimensional image. For the third application, we will examine all of the physically meaningful traces in an ionogram using a dynamic λ-connected search technique, which is combined with the use of genetic algorithms.

2. A FUZZY SEGMENTATION AND SEARCH TECHNIQUE

Image segmentation is applied to separate an image into several parts, each of which is a connected set and uniform in brightness or intensity. Mathematically, the formal definition of segmentation is as follows: let Σ_2 be a two dimensional array. A gray scale is a mapping $f : \Sigma_2 \rightarrow R$. Given a standard of "uniformity," if non-empty subsets $F_1, F_2, ..., F_m$ of (Σ_2, f) satisfy:

(1) $F_i \cap F_j = \phi$, $i \neq j$,

(2) $\cup F_i = (\Sigma_2, f)$, and

(3) Each F_i is connected and uniform,

then we can say the (Σ_2, f) is segmented by $F_1, F_2, ..., F_m$ corresponding to the uniformity.

There are many methods for segmentation, such as: measurement space clustering, region growing segmentation, split-and-merge segmentation, and edge detection. For a gray scale, we have developed a region growing technique called λ-connected segmentation that is used to search for an object in the image [3]. Assuming that p and q in Σ_2 are adjacent and f is a 2D fuzzy subset, the degree of adjacency $C_{adj}(p, q)$ between p and q is computed by the following formula:

$$C_{adj}(p,q) = \begin{cases} 1 - \frac{|f(p)-f(q)|}{|f(p)|+|f(q)|} & |f(p)| + |f(q)| \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

If $P(p_0, p_n) = p_0, p_1, ..., p_n$ is a path between p_0 and p_n, we define the degree of connectedness $C_{path}(p_0, p_n)$ between p_0 and p_n with respect to the path $P(p_0, p_n)$ as the minimum degree of adjacency between p_i and p_{i+1} for $i = 0, ..., n-1$,

$$C_{path}(P(p_0, p_n)) = min\{C_{adj}(p_i, q_i + 1) | i = 0, ..., n-1\}.$$

$C(p, q)$, the degree of connectedness between the two points p and q, is defined as the maximum degree of connectedness with respect to all paths between p and q,

$$C(p, q) = max\{C_{path}(P(p, q)) | P(p, q) \text{ is a path between } p \text{ and } q \}.$$

If $\lambda \in [0, 1]$, then p, q is called λ-connected if $C(p, q) \geq \lambda$ and $C(p, q)$ is called a fuzzy connectedness function. Generally, we can state the following theorem:

Theorem 2.1 If the fuzzy connectedness function (E) between two pixels in an image has the properties (i) $E(u, u) = 1$, (ii) $E(u, v) = E(v, u)$, and (iii) $E(u, v) \geq min\{E(u, w), E(w, v)\}$, then the region growing technique using the fuzzy connectedness function E and a selecting $\lambda \in [0, 1]$ satisfies an equivalence classification for the image.

The segmentation algorithm is given below:

Algorithm 2.1 A λ-connected Segmentation Algorithm.

Step 1: Let p_0 be a point in Σ_2. Set
$L(p_0) \leftarrow *$ and $STACK \leftarrow STACK \cup p_0$
i.e., labeling p_0 and p_0 is sent a stack $STACK$.

Step 2: If STACK is empty, go to step 4; otherwise,
$p_0 \leftarrow STACK$ (top of $STACK$). Then,
$L(p_0) \leftarrow 0$.

Step 3: For each p linking p_0 , if
$L(p) \neq 0, L(p) \neq *$, and $C(p, p_0) \geq \lambda$, then
$STACK \leftarrow STACK \cup p$ and $L(p) \leftarrow *$. Then, goto Step2.

Step 4: Stop. $S = \{p : L(p) = 0\}$ is singly λ-connected part.

Algorithm 2.1 is designed by using Tarjan's Breadth-first-search technique [6], so it is time optimal. Its time complexity is $O(|n \cdot \Sigma_2|)$. In theorem 2.1, we state that λ-connectedness is a mathematical equivalence relation. Therefore, it can be used to segment gray scale images directly.

The method! of λ-connectedness can also be used to develop a split-and-merge segmentation algorithm. Split-and-merge segmentation is an advanced segmentation technique, which is often termed quadtree segmentation in two dimensional image processing. Given the definition of "uniform" as follows: we say a region is uniform if the distance between the gray value of any point and the average value of the region is not greater than a given real number ϵ, or any two adjacent points in the region are *lambda*-connected (the region is called a normal λ-connected set) [7,8]. Split-and-merge segmentation is based on a quadrant subdivision of the image. We may split a subdivision if it is not uniform and merge adjacent regions if the resulting region is uniform. The following process is the procedure of split-and-merge segmentation.

If the whole image is uniform, then it is not necessary to subdivide it. Otherwise, we split it into four subdivisions and repeat this process until all subdivisions are uniform. The merge process is used to merge the uniform regions among four adjacent subdivisions, each of which have already been the union of several uniform regions. This determines whether or not the union of any two adjacent uniform regions, which belong to different subdivisions, are uniform and merges them if this condition is satisfied.

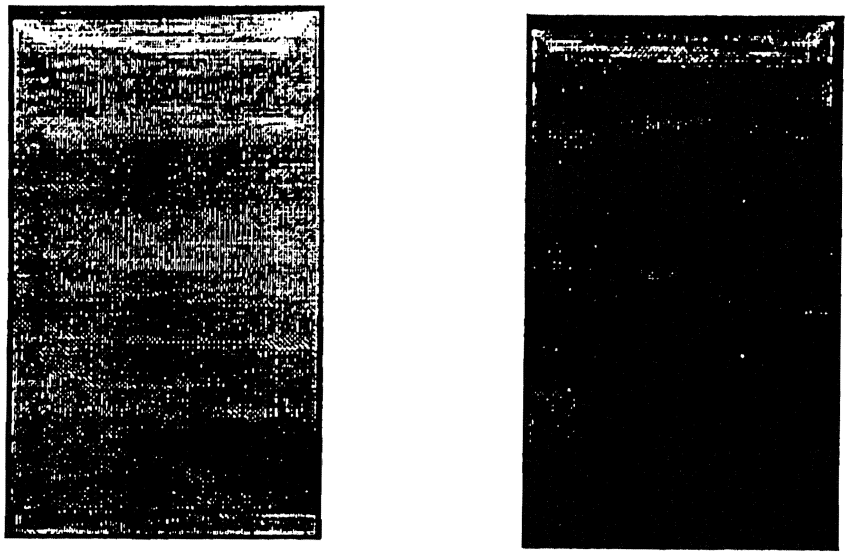
3. LAYER SEARCH FOR CROSS-WELL VELOCITY PROFILES

In this section, we apply the fuzzy segmentation technique discussed in section 2 to search the different layers in a cross-well velocity profile. The profile is represented by a 60 x 100 element gray scale image, which was obtained by applying an inversion method to a dataset provided by Exxon Corp. by Borup and Johnson [9] and shown in Fig. 3.1 (a).

In order to find the structure in a given stratum, cross-well tomography is to reconstruct the images in terms of geophysical parameters. Inverse scattering inversion, one of the most common techniques, is used to reconstruct the velocity of each location between two parallel wells. The velocity of the waveform in the stratum is one of the most important parameters used to identify the structure of layers.

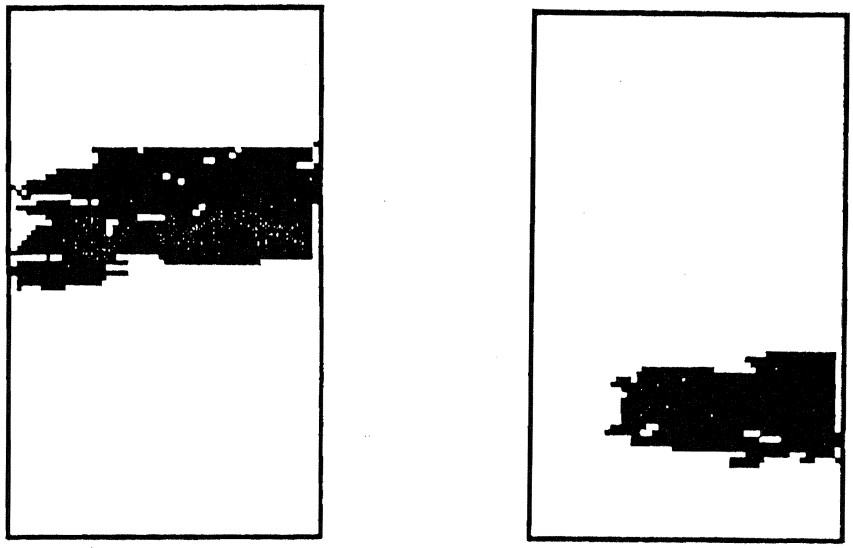
In this application, we need to search the layers in the velocity image. The reason for choosing λ-connected segmentation is that the variation of the stratum is gradual and not crisp. Theoretically, λ-connected segmentation can be used to find each segment in the image when we select a $\lambda \in [0, 1]$. However, for this particular use, the horizontal variation is smaller than the vertical variation in a layer. A modified λ-connected segmentation is used, which splits the unique λ into two parts: λ_h for horizontal direction and λ_v for vertical direction, with $\lambda_h > \lambda_v$. Figs. 3.2 (a) and (b) show two layers in the image. After determining the layers in the image, we can enhance the

image to form another image, as seen in Fig. 3.1 (b).



(a) (b)

Fig. 3.1 The Original Velocity Profile (a) and Its Fuzzy Processed Profile (b).



(a) (b)

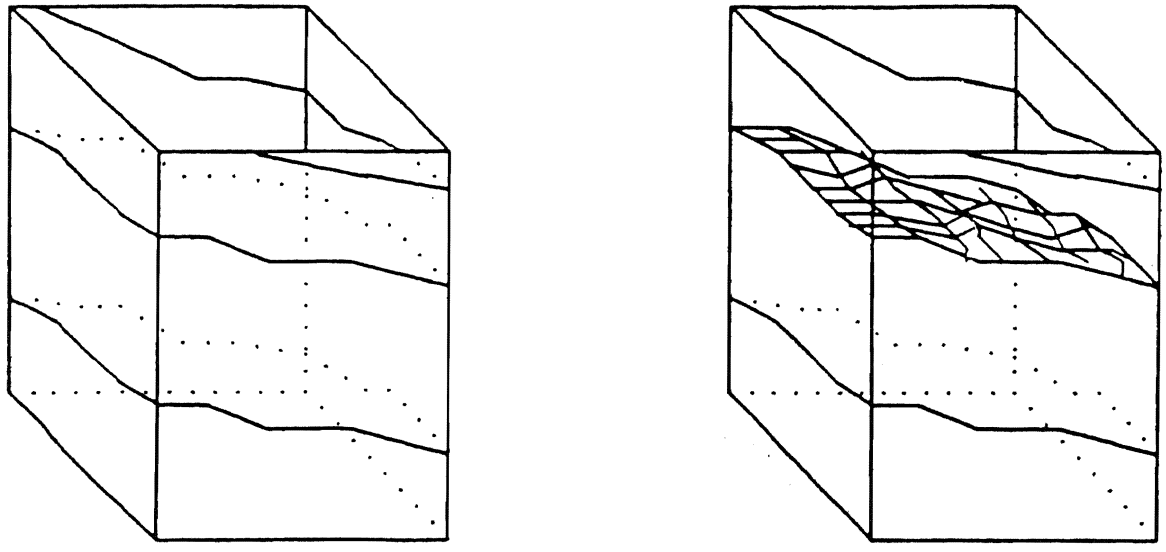
Fig. 3.2 Two λ-connected Layers Derived from Fig. 3.1 (a).

4. LAYER SEARCH BASED SURFACE RECONSTRUCTION FOR SEISMIC VELOCITY DATA

In Section 3, we described the application of the fuzzy segmentation technique to search the layers in a cross well image. In this section, we will develop a method which combines the fuzzy segmentation and fuzzy fitting methods to search and re-form the surfaces of layers in the strata.

Data fitting, interpolation and/or approximation, is used to obtain a relatively complete dataset from a incom-

plete dataset. In geophysical exploration, three dimensional seismic prospecting is very expensive. Geophysists often use the two dimensional profile (section) to reconstruct the three dimensional data (volume). In this case, given the data of four boundary sections of a $3D$ volume, see Fig. 4.1(a), there are two tasks: layer interface reconstruction (Fig. 4.1(b)) and entire volume data reconstruction for the $3D$ volume. In this paper we are only concerned with the layer interface reconstruction, where the interface is a surface.



(a) (b)

Fig 4.1 Four Boundary sections (a) and An Interface fitting (b) in A Three Dimensional Volume.

Assume that we have data for four boundary sections of a $3D$ volume. First, λ-connected segmentation is used to segment layers for each of the four boundary sections. Next, we can derive the boundary curves for every layer, collecting four curves which correspond to a surface. Finally, we fit the four curves to form the surface.

There are many methods for surface fitting, such as B-spline, Coons surface fitting, Bezier polynomial and Chebyshev polynomial, etc. One can choose any one of them to fit a surface. However, the fitting procedure must cause the resulting surface be included in the convex closure of its guiding points, in order to avoid having the upper or lower interfaces pass through each other.

It is very interesting that λ-connectedness can also be related to a fitting method having convexity. Because we can view λ-connectedness as a gradual-variation, we can show the necessary and sufficient condition and the algorithm as follows [10]:

Theorem 4.1 Suppose D is a connected subset in Σ_2, J is a non-empty subset of D, and $f_J : J \to \{A_1, A_2, ..., A_m\}$, where $A_1 < A_2 < ... < A_m$. The necessary and sufficient condition for the existence of a gradually varied interpolation surface of f_J is: for any two points $p, q \in J$, $d(p,q) \geq d'(f_J(p), f_J(q))$; where $d(p,q)$ means the length of the shortest path in D and $d'(f_J(p), f_J(q)) = |i-j|$, $f_J(p) = A_i$, and $f_J(q) = A_j$.

Theorem 4.2 Suppose D is a connected subset in Σ_2, J is a non-empty subset of D, and $f_J : J \to R$. The necessary and sufficient condition for existence of a λ-connected interpolation surface of f_J is: for any two points $p, q \in J$ and any shortest path $p_0, ..., p_n$ between p and q, there is a group of real numbers $f(p_0), ..., f(p_n)$, $f_J(p) = f(p_0)$ and $f_J(q) = f(p_n)$, such that p and q are λ-connected.

The following algorithm applies to gradually varying surface interpolation, including the interpolating surface

in the convex closure of the guiding points.

Algorithm 4.1: Guiding points are at any position.

Step 1: Save the guiding points in a queue $T = \{T(1), ..., T(n)\}$, let the pointer $k \leftarrow 1$.

Step 2: $p \leftarrow T(k)$;

Step 3: For every adjacent point q of p, if $q \in T$, then $q \leftarrow$ *the next adjacent point of* p *in* T; else $f(q) \leftarrow f(p)$. If there is a point $r \in T$, it satisfies $d'(f_J(q), f_J(r)) > d(q, r)$, then assign the value to $f(q)$ such that $d'(f_J(q), f_J(r)) = d(q, r)$ and $d'(f_J(p), f_J(q)) = 1$. Add q to the end of queue T.

Step 4: $k \leftarrow k + 1$; stop if k overflows (i.e. $k > |D| - |J|$), else go to step 2.

Algorithm 4.1 is used to fit a series of interfaces in real interval velocity sections in an area located in the southwest of China. The result is shown in Fig. 4.2.

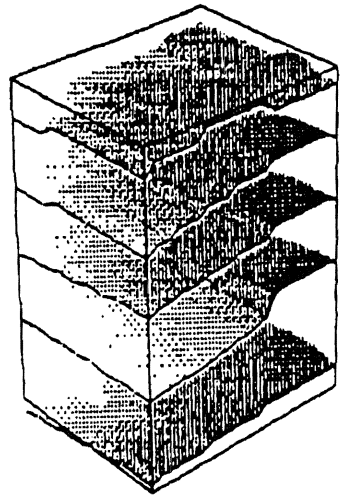


Fig. 4.2 A Real Series of Interfaces.

5. FUZZY CURVE SEARCH AND GENETIC ALGORITHMS APPLIED TO AUTOMATIC IONOGRAM PROCESSING

In this section, we discuss the application of the fuzzy segmentation technique to search and examine the frequency-height trace in ionograms acquired by a vertical incidence HF sounder (ionosonde). Since each trace is represented by a set of irregular points in the ionogram image, we have developed a dynamic genetic algorithm to find the optimal trace. This problem is also referred to as automatic ionogram scaling. The scaling of ionospheric parameters is a laborious task because of the complex structures which often appear in ionograms.

The original ionogram is composed of discrete points and often looks fuzzy, so the first stage in our process is fuzzification and filtration. Secondly, we use the fuzzy search method introduced in section 2 to segment the ionogram into several disconnected components corresponding approximately to the different ionospheric regions from which echoes are returned. Because each component consists of many curves, the third stage of the process is to search all possible curves in each segment. As the last stage, we will connect the curves, if possible, among

these components. In this application, we have not used the physical properties of the ionograms (echo amplitude, polarization, angle-of-arrival etc.) to segment and/or merge the various curves.

(1) Fuzzification and Amplitude-based Filter.

In order to use the fuzzy segmentation technique, we first make a fuzzification for the ionogram. It is a straightforward method, which converts the number of points in a small region into a number. The original ionogram is shown in Fig 5.1, and the result of fuzzification is shown in Fig. 5.2.

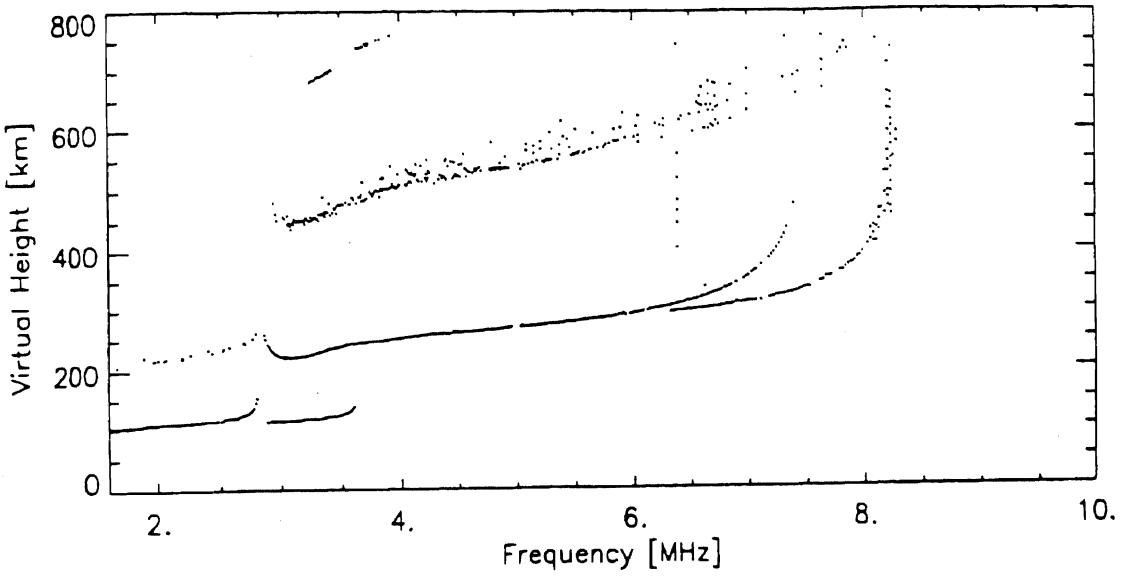


Fig. 5.1 An Original Frequency-Height Ionogram.

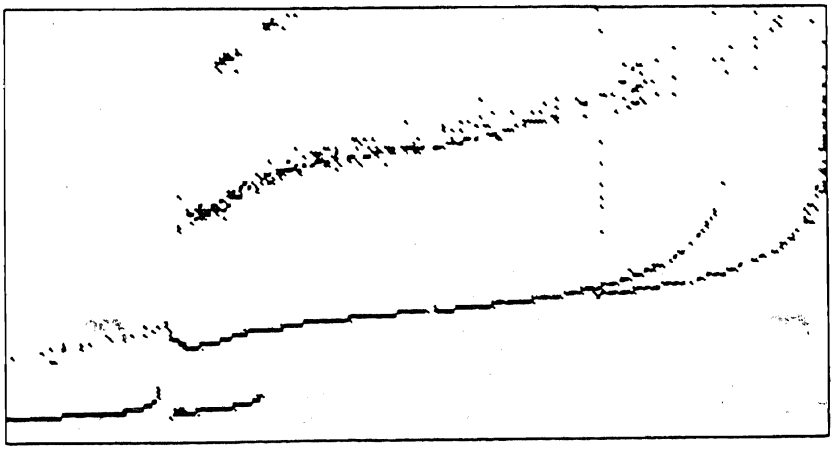


Fig. 5.2 The Fuzzification of the Frequency-Height Ionogram.

The amplitude-based filter is used to discriminate among the multiple reflections, based upon the amplitude-frequency image. This algorithm is also simple. It finds the maximum amplitude for each frequency value in the amplitude-frequency image, then in the height-frequency image we eliminate all points which are not the maximum amplitude points, as shown in Fig. 5.3.

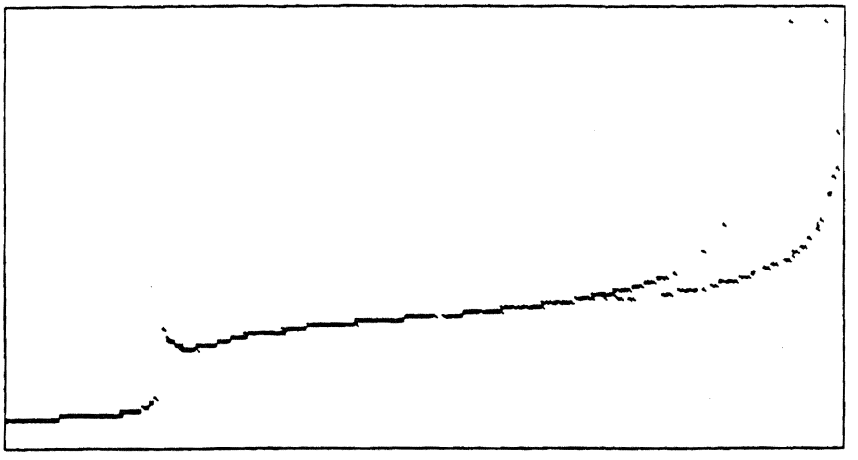


Fig. 5.3 The Filtered Frequency-Height Ionogram.

(2) Fuzzy Segmentation.

The purpose of applying fuzzy segmentation is to partition the height-frequency image into several inter-connected components. Any two parts are disconnected in the sense of λ-connectedness. We can directly apply algorithm 2.1 for this image, resulting in 22 components, which are shown in Fig 5.4., where different curve patterns show the different components.

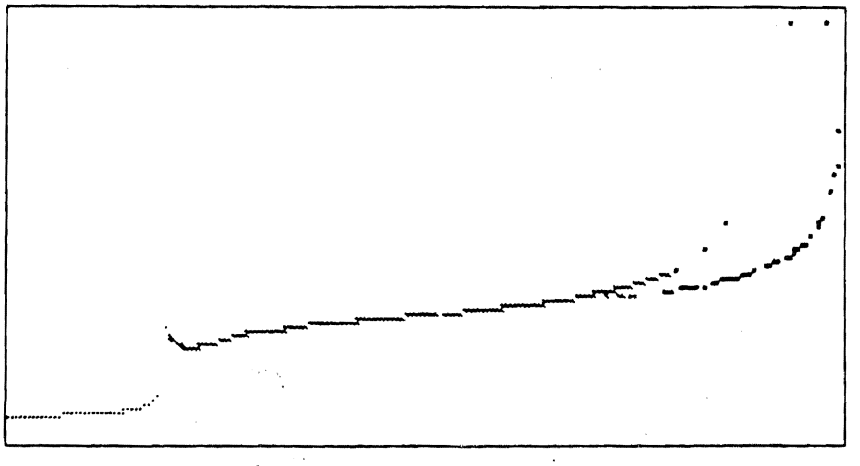


Fig. 5.4 The Filtered Frequency-Height Ionogram.

(3) Curve Search in a Component.

In order to find all physically realizable curves, we must search all of the possible curves in a component. We know that all components consist of curves although we do not know which is the most physically relevant curve. Before applying the curve-search algorithm, we build a point-successor list which represents the point (pixel) and its successors in a component. The curve-search algorithm uses the depth-first-search technique given by Tarjan [6], modified it for this particular problem. Also, the algorithm can eliminate some curves which are not relevant to our solution. For convenience, we call the curves in the components subcurves. It is not possible to discuss the curve-search algorithm in this paper; however, we will present the entire algorithm and its time analysis in another paper.

(4) Genetic Algorithms Application to Search Optimal Traces.

The most difficult task in searching for a curve in an ionogram is that of merging two or more subcurves in different components into a single curve. Once we have found each of the subcurves in the entire image, we can enumerate all possible curves and evaluate each theoretically. However, it is not feasible to do so if we take into consideration the computational overhead.

Genetic algorithms have many advantages for the optimal search in real-time computing [11]. When applying a genetic algorithm, the curve is represented as a string. Each element in the string is a number representing a subcurve in a component. The curve generated by some subcurves must have no overlap. The string is called a chromosome, and each number in the string is called a gene. The fittness function is used to evaluate whether a string is a viable curve. According to the properties of ionograms, the smoothness of a curve is a desirable property, required for further processing.

We now briefly describe the subcurve merging process using a genetic algorithm. First, we build a subcurve-successor list, which represents which subcurve can be followed by which subcurve. Secondly, we randomly choose an initial chromosome set (called the initial generation), each curve of which is a reasonable curve. Also, we compute their fittness. After that, we randomly choose the location to make a crossover computation if it is possible to form a new generation. We repeat these steps until the generation does not undergo any change. The longest trace found by the genetic algorithm is shown in Fig 5.5.

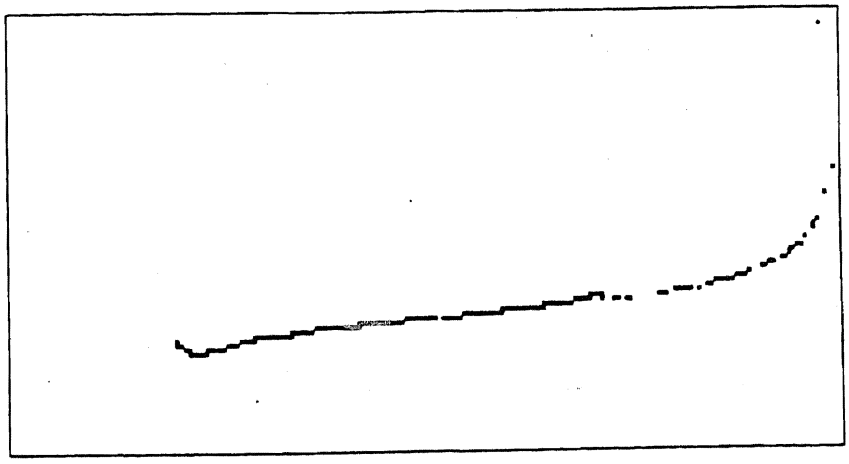


Fig. 5.5 An Identified Trace in the Frequency-Height Ionogram.

6. CONCLUSION

This paper has described a fuzzy segmentation technique and applied it to three real geophysical data/image processing problems. The fuzzy system approach has important advantages for geophysical data processing. However, in different applications, the fuzzy segmentation technique may need to be modified in order to obtain realistic results. The methodology of fuzzy logic and systems should be objective. In addition, the advanced artificial intelligence and image processing methods are related and associable. For example, neural networks and genetic algorithms could be combined with a fuzzy system approach to obtain optimal results. We have used genetic algorithms to search for the optimal trace in an ionogram in section 5. In future work, we intend to develop a fuzzy aided software system for geophysical data/image processing.

Acknowledgments: This research has been supported by NSF grant DPP91-19382 at Utah State University. We are deeply indebted to Dr. David T. Borup for providing the cross well velocity profile. We would like to thank The Research Institute of Geophysical Prospecting for Petroleum (China) for the seismic data, and Exxon Corporation for the cross well data.

REFERENCES

1. Y. C. Chen and S. Y. Lu, "Wave form correlation using tree matching," *7th International Conference on Recognition,* Vol.1, 350-353, 1984.

2. P. L. Love and M. Simaan, "Segmentation of a seismic section using image processing and artificial intelligence technique," *Pattern Recognition,* Vol 18, 409-419, 1985.

3. A. Rosenfeld and A. Kak, *Digital Picture Processing,* Vol. I and II, Academic Press, Orlando, 1982.

4. G. Read and A. Rosenfeld, "Recognition of surfaces in three-dimensional digital images," *Information and Control,* Vol. 53, 108-120, 1982.

5. S. Igi, H. Minakoshi and M. Yoshida, "Automatic ionogram processing system, 2. automatic ionogram scaling," *J. Commun. Res. Lab.,* Vol.39, 367-379, 1992.

6. A. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

7. L. Chen, "The λ-connected segmentation and the optimal algorithm for split-and-merge segmentation," *Chinese J. Computers* Vol. 14, 1991. *Computer and Control Abstracts,* Vol. 26, 63949, 1991.

8. L. Chen, H. Cheng and J. Zhang, "Fuzzy subfiber and its application to seismic lithology classification," *Information Science*, to appear.

9. D. T. Borup and S. Johnson, "Techniscan progress report to Exxon," *Techniscan Inc.,* July 1991.

10. L. Chen, "Random gradually varied surface fitting," *Chinese Science Bulletin,* Vol. 37, 1992. *Abstracts of Second SIAM Conference on Geometric design,* Temple, AZ, 1991.

11. L. Davis, *Handbook of Genetic Algorithms,* Van Nostrand Reinhold, 1991.

Intelligent Curve Tracking Algorithms and Implementations

Li Chen[a,d] Frank T. Berkey[b] Don Cooley[a]
Yexian He[c] Jianping Zhang[a] Lan Zhang[a]
[a]Department of Computer Science, Utah State University, Logan, UT 84322
[b]Space Dynamics Laboratory, Utah State University, Logan, UT 84322
[c]Department of Computer Science, Wuhan University, Wuhan 430072, PRC
[d]Scientific and Practical Computing, P.O. Box 6081, North Logan, UT 84341

ABSTRACT

Real time curve tracking is an important topic in radar image processing and can be applied to automated ionogram scaling and remote target tracking. A 2D grey-scale image only contains line segments and curves, and we want to extract them. For straight line tracking, Hough transform can be applied easily. However, to extract an arbitrary curve, Hough transform requires more processing and data storage costs. According to the nature of the problem, this paper try to find each piece of curves and then link some pieces together to form a curve. A systematic method has been studied in this paper. This method includes the fuzzification of images, fuzzy segmentation, sub-curve search, and genetic algorithm linking. The fuzzification and fuzzy segmentation may not be applied if the original image is clear. The genetic algorithms are designed to link sub-curves in this system. In addition, a fuzzy neural network is proposed and implemented for tracking curves in sequential images.

Keywords: image processing, curve tracking, fuzzy segmentation, neural networks, genetic algorithms

1. INTRODUCTION

Vertical incidence ionospheric radar systems receive data and interpret it into two dimensional images. Such an image may contain several traces (curves). Similarly, over-the-horizon radar (OTHR) have been successfully used for tracking remote targets on the sky, such as aircrafts. To track the trace of an object is mathematically equivalent to extract a curve in an image. The problem becomes more complex when more objects are on the target area. Since these curves may intersect each other, the number of possible choices of curves can be an exponential function respecting to the size of the image. Therefore, to track all possible curves may take several years. On the other hand, sometime the images are clear, and sometime they are fuzzy. Meanwhile, the background of the image may have a lot of noises. In such a situation, intelligent system techniques, such as fuzzy logic, genetic algorithms, and neural networks, have more advantages to solve the problem.

2. SEGMENTATION

Curve tracking problem usually deal with binary images. However, sometimes the image could be a gray-scale image. It is also possible to transfer a binary image to be a gray-scale image to reduce noise or for other purposes. Although one can set a clip level or threshold to get a binary image, such process may loss important information. For example, there are two curves in the image. One has intensity "1" and another has intensity "0.5". Using clip value "0.5", the two curve will be mixed. Using clip value "1", one curve will be dismissed.

Part of the SPIE Conference on Automatic Target Recognition VIII • Orlando, Florida • April 1998
SPIE Vol. 3371 • 0277-786X/98/$10.00

A fuzzy segmentation technique[1,2,3], λ-connected segmentation, can be used in such cases to obtain each λ-connected component. The purpose of applying fuzzy segmentation is to partition an image into several interconnected components. Any two parts are disconnected in the sense of λ-connectedness.

Assuming that p and q in Σ_2 are adjacent and f is a 2D fuzzy subset, the degree of adjacency $C_{adj}(p,q)$ between p and q is computed by the following formula:

$$C_{adj}(p,q) = \begin{cases} 1 - \frac{|f(p)-f(q)|}{|f(p)|+|f(q)|} & |f(p)| + |f(q)| \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

If $P(p_0,p_n) = p_0, p_1, ..., p_n$ is a path between p_0 and p_n, we define the degree of connectedness $C_{path}(p_0,p_n)$ between p_0 and p_n with respect to the path $P(p_0,p_n)$ as the minimum degree of adjacency between p_i and p_{i+1} for $i = 0, ..., n-1$,

$$C_{path}(P(p_0,p_n)) = min\{C_{adj}(p_i,q_i+1) | i = 0, ..., n-1\}.$$

$C(p,q)$, the degree of connectedness between the two points p and q, is defined as the maximum degree of connectedness with respect to all paths between p and q,

$$C(p,q) = max\{C_{path}(P(p,q)) | P(p,q) \text{ is a path between } p \text{ and } q\ \}.$$

If $\lambda \in [0,1]$, then p,q is called λ-connected if $C(p,q) \geq \lambda$ and $C(p,q)$ is called a fuzzy connectedness function.

The segmentation algorithm is given below:

Algorithm 2.1 A λ-connected Segmentation Algorithm.

Step 1: Let p_0 be a point in Σ_2. Set
$L(p_0) \leftarrow *$ and $STACK \leftarrow STACK \cup p_0$
i.e., labeling p_0 and p_0 is sent a stack $STACK$.

Step 2: If STACK is empty, go to step 4; otherwise,
$p_0 \leftarrow STACK$ (top of $STACK$). Then,
$L(p_0) \leftarrow 0$.

Step 3: For each p linking p_0 , if
$L(p) \neq 0, L(p) \neq *$, and $C(p,p_0) \geq \lambda$, then
$STACK \leftarrow STACK \cup p$ and $L(p) \leftarrow *$. Then, goto Step2.

Step 4: Stop. $S = \{p : L(p) = 0\}$ is singly λ-connected part.

Algorithm 2.1 is designed by using Tarjan's Breadth-first-search technique [4], so it is time optimal. Its time complexity is $O(|n \cdot \Sigma_2|)$.

We can apply algorithm 2.1 to an ionogram, resulting in 22 components, which are shown in Fig 2.1.

3. CURVE SEARCH IN A CONNECTED COMPONENT

After we get each λ-connected component, we want to get all pieces of curves in the component. In order to find all physically realizable curves, we must search all of the possible curves in a component. We know that all components consist of curves although we do not know which is the most physically relevant curve. Before applying the curve-search algorithm, we build a point-successor list which represents the point (pixel) and its successors in

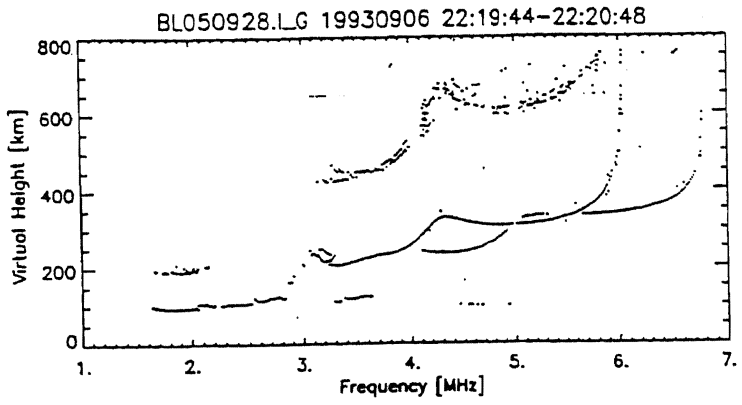


Fig 2.1 An Frequency-Height Ionogram.

a component. The curve-search algorithm uses the depth-first-search technique given by Tarjan [4], modified it for this particular problem. Also, the algorithm can eliminate some curves which are not relevant to our solution. For convenience, we call the curves in the components sub-curves.

In fact, there are two ways can be used in searching all pieces of curves. First, each curve piece does not contain any points of others excepting end-points. Second, a curve piece may contain some parts of other pieces, but it is restricted in a certain direction of the image.

4. GENETIC ALGORITHMS AND OPTIMAL CURVE TRACKING

An difficult task in searching for a curve in an image is that of merging two or more sub-curves in different components into a single curve. Once we have found each of the sub-curves in the entire image, we can enumerate all possible curves and evaluate each theoretically. However, it is not feasible to do so if we take into consideration the computational overhead.

Genetic algorithms are search algorithms based on the mechanics of natural selection[5]. Genetic algorithms have many advantages for the optimal search in real-time computing. When applying a genetic algorithm, the curve is represented as a string. Each element in the string is a number representing a sub-curve in a component. The curve generated by some sub-curves must have no overlap. The string is called a chromosome, and each number in the string is called a gene. The fitness function is used to evaluate whether a string is a viable curve. According to the properties of ionograms, the smoothness of a curve is a desirable property, required for further processing.

We now briefly describe the sub-curve merging process using a genetic algorithm. First, we build a subcurve-successor list, which represents which subcurve can be followed by which subcurve. Secondly, we randomly choose an initial chromosome set (called the initial generation), each curve of which is a reasonable curve. Also, we compute their fitness. After that, we randomly choose the location to make a crossover computation if it is possible to form a new generation. We repeat these steps until the generation does not undergo any change. The longest trace found by the genetic algorithm is shown in Fig 4.1.

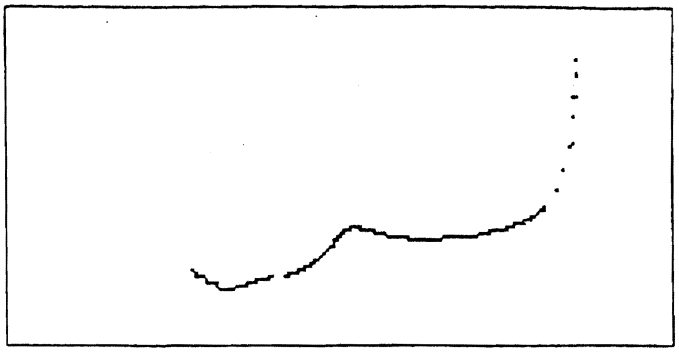


Fig 4.1 An Identified Trace in the Frequency-Height Ionogram.

5. SEQUENTIAL IMAGE PROCESSING USING NEURAL NETWORKS

This section consider curve tracking in sequential images. In this case, we assume that for a certain curve in the first image, there is a corresponding curve in the second image, ... In addition, the later corresponding curve has only small changes comparing with its immediately previous curves. Therefore, one only need to get all curves in the first image and do not have to go through all process we described before for rest of the images. We will use a possibility function-based neural network to search curves in the sequence. We will consider ionogram scaling as an example even though our method can deal with other applications.

Ionograms are acquired at intervals ranging from a few minutes up to an hour. We can use a search method, such as the fuzzy search method, to track all traces in each ionogram. However, the optimal solution for one ionogram in a sequence might not be an optimal solution for the entire sequence, because the quality of a single ionogram might not be high enough for searching . Second, this is a time-consuming process for all three traces in each isolated ionogram.

We propose to use a possibility function-based neural network to process the sequential ionograms excepting the first one. The first ionogram will be used as training data for the neural network.

For each point $p = (p_1, p_2)$ in an ionogram, we want to determine whether p is a point of E-layer, F_1-layer, or F_2-layer. That is determine the possibility that p belongs to T_1, T_2, or T_3. $P_T(p)$ represents the possibility that p is in trace T_1, which indicates a E-layer; $P_{F_1}(p)$ represents the possibility that p is in trace T_2, which indicates a F_1-layer; $P_{F_2}(p)$ represents the possibility that p is in trace T_3, which indicates an F_2-layer.

The possibility function-based neural network is shown in Figure *. This network has an input layer plus a converting layer. In order to process the specific problem, the computation formulas need to be changed.

We present general formulas for this type network. Let $X = (x_1, ..., x_m)$ be the input vector for one layer, and r_{ij}, which generates $\gamma_{ij}(u)$ and τ_{ij}, be the weights for this layer, where $i = 1, ..., m$ and $j = 1, ..., n$. The output of this layer is a group of possibility functions: $C = (c_1, ..., c_n)$ is defined by

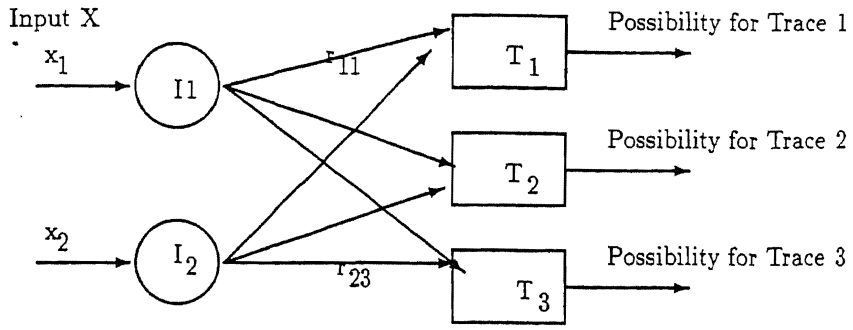


Fig 5.1 A Possibility-Based Neural Network to Search the Three Traces.

$$c_j = \mathcal{F}(\sum_{i=1}^{m} \tau_{ij} \cdot q_{ij})$$

where

$$q_{ij} = \min_{u \in [0,1]} \{(x_i(u) - \gamma_{ij}(u))^2\}.$$

and

$$\mathcal{F}(s) = 1 - s.$$

In this case, there are two types of training. First we want to find all r's by training on the first ionogram in the ionogram sequence. We can find the r's using the method described in the previous section. We will discuss it in the initialization stage later.

Here, we give a theoretical discussion of the first type of training.

Let u_0 be the point such that $(x_i(u_0) - \gamma_{ij}(u_0))^2 = \min\{(x_i(u) - \gamma_{ij}(u))^2\}$. The performance of the converting layer is defined as

$$E = \frac{1}{|S_T|} \sum_{p=0}^{|S_T|-1} E_p$$

S_T is the training set, and E_p is the output error for pth training sample. E_p is defined by

$$E_p = \frac{1}{2} \sum_{j=1}^{n} (T_j - c_j)^2$$

where $T = (T_1, \ldots, T_n)$ is the target output of input X.

The weights γ and τ are modified to minimize the error in the backpropagation algorithm according to the rule

$$(1)\gamma(u) \leftarrow \gamma(u) + \mu_\gamma \frac{\partial E}{\partial \gamma(u)}$$

$$(2)\tau \leftarrow \tau + \mu_\tau \frac{\partial E}{\partial \tau}$$

where μ_γ and μ_τ are two constant learning rates.

Algorithm 5.1

$$(3)\ \gamma_{ij}(u_0) \leftarrow \gamma_{ij}(u_0) - \mu_\gamma \cdot \tau_{ij}(T_j - c_j)(x_i(u_0) - \gamma_{ij}(u_0)),$$

$$(4)\ \tau \leftarrow \tau + \mu_\tau (T_j - y_j)(x_i(u_0) - \tau_{ij})^2.$$

The fuzzification of the input is simple. Let $p = (p_1, p_2)$ be a point in an ionogram. min_1 is the minimum value for p_1 in the entire ionogram sequence, max_1 is the maximum value for p_1 in the entire ionogram sequence, min_2 is the minimum value for p_2 in the entire ionogram sequence, and max_2 is the maximum value for p_2 in the entire ionogram sequence. The fuzzification of p is $x = (x_1, x_2)$, where

$$x_1(u) = \begin{cases} p_2 & \text{if } u = p_1, \\ 0 & \text{otherwise} \end{cases}$$

and

$$x_2(u) = \begin{cases} p_1 & \text{if } u = p_2, \\ 0 & \text{otherwise} \end{cases}$$

In this section, we discuss the second type training, which is important for sequential ionogram processing. We know, for sequential ionograms, two adjacent ionograms only have small differences. In other words, each of three traces shows little change in consecutive ionograms. We can use three recognized traces as the result to train the network in order to fit the consecutive ionogram.

Let $\mathcal{PN}_k$ be the possibility function-based neural network after the process of the kth ionogram in the ionogram sequence. Suppose that T_1, T_2, T_3 are these three traces found by $P_k(x) < 1$, and C_1, C_2, C_3 are three fitted curves for T_1, T_2, T_3, respectively.

In this application, we did not use Markov's law. In other words, we want to train a new network PN_{k+1} for $k+1$th ionogram by using C_1, C_2, C_3 as a set of training data. Let $(a_1, a_2) \in C_j$,

$$g_{1j}(u) = \begin{cases} a_2 & \text{if } u = a_1 \text{ and there is no other } a_2' \\ & \text{such that } (a_1, a_2') \text{ is also in trace } j \\ \phi & \text{there are two points } (a_1, a_2) \text{ and} \\ & (a_1, a_2') \text{ in trace } j \\ 0 & \text{otherwise} \end{cases}$$

and

$$g_{2j}(u) = \begin{cases} a_1 & \text{if } u = a_2 \text{ and there is no other } a_1' \\ & \text{such that } (a_1', a_2) \text{ is also in trace } j \\ \phi & \text{there are two points } (a_1, a_2) \text{ and} \\ & (a_1', a_2) \text{ in trace } j \\ 0 & \text{otherwise} \end{cases}$$

The following formulas are used to train the new γ-weight:

$$\gamma_{ij}(u) \leftarrow \gamma_{ij}(u) + \mu \frac{g_{ij}(u) - \gamma_{ij}(u)}{2}$$

where μ is a learning rate ($mu = 1$ for example).

We have applied the possibility function-based neural network to process a sequence of ionograms that has ten ionograms recorded at the Bear Lake Observation of Utah State University. The first and fifth ionograms are shown in Fig 5.2. Three F1-layer search in the first, fifth and tenth ionograms in a sequential ionogram are shown in Figure 5.3.

REFERENCES

1. L. Chen, "The λ-connected segmentation and the optimal algorithm for split-and-merge segmentation," *Chinese J. Computers* Vol. 14, No 3, 1991.

2. L. Chen, F.T. Berkey and S.A. Johnson, The application of a fuzzy object search technique to geophysical data processing, *IS&T/SPIE Symposium on Electronic Imaging*, (SPIE Proc. Vol 2180), 1994.

3. L. Chen, H. Cheng and J. Zhang, "Fuzzy subfiber and its application to seismic lithology classification," *Information Science: Application*, Vol 1, No 2, 1994.

4. A. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

5. L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.



Fig 5.2 Two Ionograms in a Sequential Ionograms.

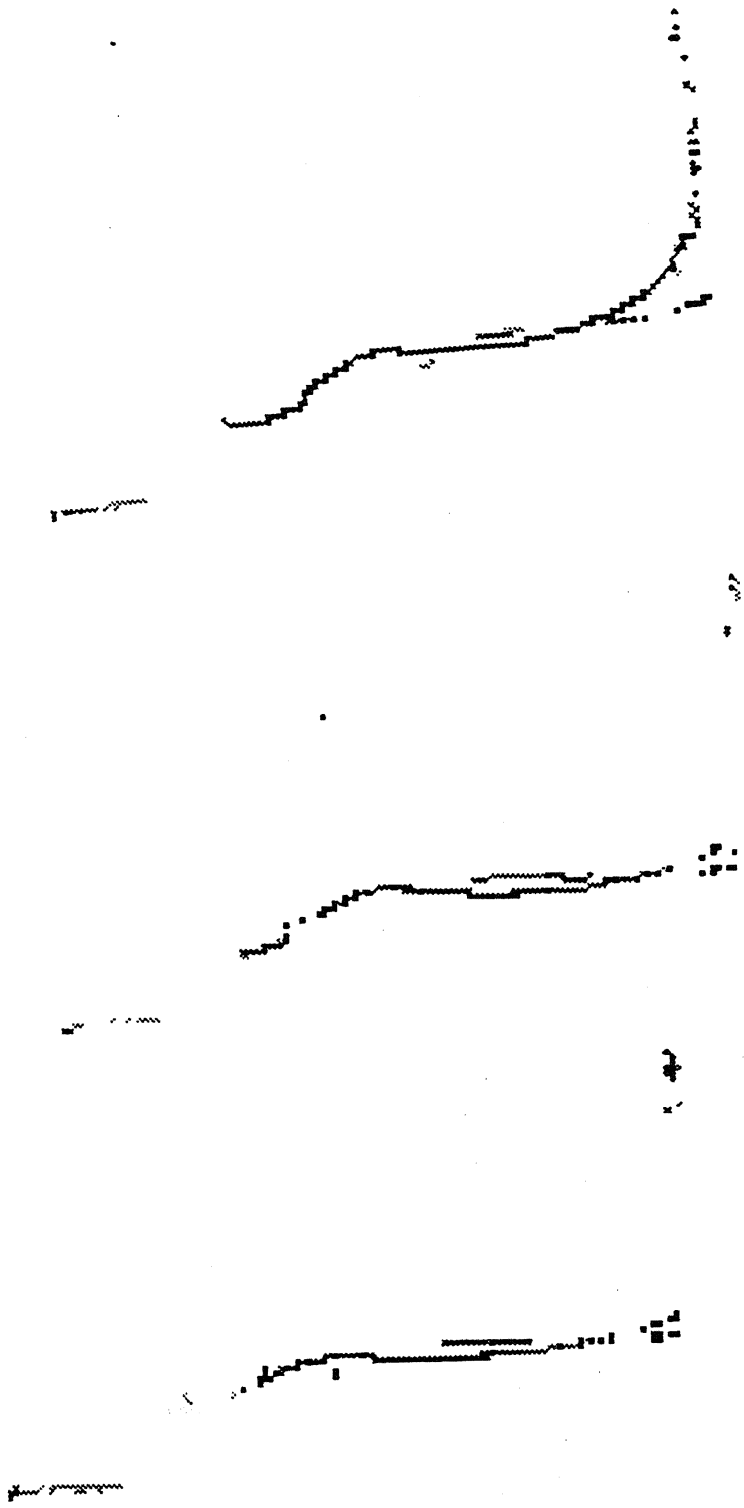


Fig 5.3 F1-layers in the Three Ionograms in the Sequential Ionograms.

Fuzzy Subfiber and Its Application to Seismic Lithology Classification

LI CHEN
HENG-DA CHENG

and

JIANPING ZHANG

Department of Computer Science, Utah State University, Logan, Utah 84321

ABSTRACT

Rosenfeld proposed the concept of the 2D fuzzy subset and successfully applied it to the problem of image segmentation. However, the 2D fuzzy subset approach could be used only for gray scale image segmentation because it fails to handle higher-dimensional range images such as color images. To deal with higher-dimensional range images, we introduce a new concept—fuzzy subfiber—which can be viewed as an extension of the 2D fuzzy subset. In this paper, we give the definition of fuzzy subfiber and discuss one of its most important properties: connectivity on fuzzy subfibers. This property enables us to develop fast image segmentation algorithms for higher-dimensional range images. Finally, we discuss lithology determination (classification) as a real application of fuzzy subfiber.

1. INTRODUCTION

Image understanding and recognition usually begin by segmenting the image into regions and computing various properties of and relationships among these regions [1]. In many cases, these regions could not be defined as crisp regions; that is, there is no crisp boundary between regions. To segment such kinds of images, Rosenfeld [1–4] proposed the concept of the two-dimensional fuzzy subset (2D fuzzy subset) and associated methods. In the 2D fuzzy subset methods, a gray scale picture in a two-dimensional discrete space is regarded as a fuzzy subset, and the connectedness is defined on the fuzzy subset [1–4]. As a result, the problem of segmentation of images is converted to the problem of searching the connected classes in the fuzzy subset of the image [1–5]. To extend Rosenfeld's method, another connectedness was defined on the 2D fuzzy subset [6]. This connectedness makes a mathematically equivalent classification.

© Elsevier Science Inc. 1994
655 Avenue of the Americas, New York, NY 10010 1069-0115/94/$7.00

Although the 2D fuzzy subset methods worked very well, they handle only gray scale images and fail to process high-dimensional range images. In a gray scale image, each point of the image is associated with a value which is the intensity of the point. By high-dimensional range images, we mean images in which each point is associated with a vector of values. For example, a color image is a three-dimensional range image, because each point of a color image is associated with three values. In order to deal with high-dimensional range images, we introduce a new concept: fuzzy subfiber. Fuzzy subfiber is a natural generalization of the 2D fuzzy subset and is able to represent high-dimensional range images. A 2D fuzzy subset is a mapping that maps each pixel of an image to a value between 0 and 1, whereas a fuzzy subfiber is a mapping that maps each pixel of an image to a vector of values between 0 and 1. One of the most important geometric properties of fuzzy subfiber is connectedness that enables us to develop fast algorithms for segmenting high-dimensional range images.

We applied the fuzzy subfiber approach to attack the problem of determining (classifying) the lithology (rock type) of a given area from seismic data, which is an important problem in oil–gas exploration [7–10]. Accurate classification of lithology can help petroleum geologists correctly locate lithological oil–gas reservoirs and save millions of dollars. Each given area may include a mixture of different lithologies, so the result of a classification is a vector of values instead of a single value. Each of the values is the possibility of one lithology in this area. Thus, a lithological image is a high-dimensional range image and cannot be represented as a 2D fuzzy subset. As a result, lithological determination is a typical problem where the fuzzy subfiber approach must be applied.

In Section 2, we briefly review the concept of 2D fuzzy subset defined by Rosenfeld and the fuzzy connectedness on a 2D fuzzy subset. Section 3 defines the concept of fuzzy subfiber and the corresponding fuzzy connectedness. In Section 4, we discuss some properties of λ-connectedness and give efficient algorithms for determining λ-connectedness [11]. Section 5 describes the application of the fuzzy subfiber approach to determining lithology. We conclude this paper in Section 6.

2. 2D FUZZY SUBSET AND ASSOCIATED FUZZY CONNECTEDNESS

In this section we review the 2D fuzzy subset and fuzzy connectedness proposed in [1–4] and their application to image segmentation. We also review λ-connectedness proposed in [6, 12].

A fuzzy subset of a set S is a mapping $\mu: S \rightarrow [0,1]$. For any element $p \in S$, $\mu(p)$ is called the degree of membership of p in μ. If S is an array

of integer coordinate point set Σ_2, (Σ_2, μ) is called a two-dimensional fuzzy subset, or simply 2D fuzzy subset [1, 2], in this paper. A gray scale image is equivalent to a 2D fuzzy subset because a gray scale image is a mapping f: $\Sigma_2 \to R$, where R is the real number set which can be normalized easily to $[0, 1]$.

Image segmentation is done by separating an image into several parts, each of which is a connected set and is uniform in brightness or intensity. Mathematically, the formal definition of segmentation is the following: Given a standard of "uniformity," nonempty subsets of (Σ_2, f), $F_1, F_2, \ldots, F_m$, satisfy (1) $F_i \cap F_j = \phi$, $i \neq j$, (2) $\cup F_i = (\Sigma_2, f)$, and (3) each F_i is connected and uniform.

"Segmentation of a picture into subsets represents a strong commitment; in many cases, it would be preferable to weaken this commitment by 'extracting' fuzzy subsets" [1]. For this reason, the concept of fuzzy connectedness on the 2D fuzzy subset was introduced. Two points in a subset S of Σ_2 are connected if there exists a path between these two points in S; otherwise they are not connected. In normal connectedness, two points are either connected or not connected. In fuzzy connectedness, two points may be connected with a degree which is called the degree of connectedness. The degree of connectedness between two points is calculated using the mapping function μ:

$$C_\mu(p, q) = \max\{\min\{\mu(r) : r \text{ in path } P\} : P \text{ is any path from } p \text{ to } q\}$$

where $C_\mu(p, q)$ is the degree of connectedness between the points p and q. Two points p and q are called Rosenfeld fuzzy-connected in μ if

$$C_\mu(p, q) \geq \min\{\mu(p), \mu(q)\}.$$

Each Rosenfeld fuzzy-connected part is a uniform region in a segmentation. Therefore, segmenting an image is to find all Rosenfeld fuzzy-connected parts. One problem with Rosenfeld's fuzzy connectedness is that it does not satisfy transitivity [1, 2], so it is not an equivalence relation. Therefore, it does not define a mathematical classification that partitions (Σ, μ) into "connected components." To overcome this problem, another fuzzy connectedness, which satisfies reflexivity, symmetry, and transitivity, was defined [6]. In Rosenfeld's fuzzy connectedness, the degree of connectedness of two points is calculated directly from the values associated with all points in the path between the two points, while in the fuzzy connectedness defined in [6], the degree of connectedness is computed indirectly from the values associated with the points of the path between the two points. The second fuzzy connectedness is stated below.

Assume that p and q in Σ_2 are adjacent and f is a 2D fuzzy subset. Then the degree of adjacency between p and q $C_{\text{adj}}(p,q)$ is computed by the formula

$$C_{\text{adj}}(p,q)=\begin{cases}1-\dfrac{|f(p)-f(q)|}{|f(p)|+|f(q)|}, & |f(p)|+|f(q)|\neq 0,\\ 1, & \text{otherwise.}\end{cases}$$

If $P(p_0,p_n)=p_0,p_1,\ldots,p_n$ is a path between p_0 and p_n, we define the degree of connectedness between p_0 and p_n with respect to the path $P(p_0,p_n)$, $C_{\text{path}}(p_0,p_n)$, as the minimum degree of adjacency between p_i and p_{i+1} for $i=0,\ldots,n-1$:

$$C_{\text{path}}(P(p_0,p_n))=\min\{C_{\text{adj}}(p_i,q_i+1)|i=0,\ldots,n-1\}.$$

The degree of connectedness between the two points p and q, $C(p,q)$, is defined as the maximum value of degree of connectedness with respect to all paths between p and q:

$$C(p,q)=\max\{C_{\text{path}}(P(p,q))|P(p,q) \text{ is a path between } p \text{ and } q\}.$$

We now review λ-connectedness [6]. Let $\lambda\in[0,1]$. Then p, q is λ-connected if $C(p,q)\geq\lambda$.

In [6], the theorem that states λ-connectedness is a mathematical equivalence relation. Because λ-connectedness on a 2D fuzzy subset is an equivalent relation, it can be used to segment gray scale images directly.

A 2D fuzzy subset can be used only to represent gray scale images. However, in many real world applications, images are not gray scale images, namely, they are high-dimensional range images. The most notable non-gray-scale images are color images. Another example of non-gray-scale images is lithological images which are discussed in detail in Section 5. A gray scale image is a mapping from a set of points to a range of numeric values, whereas a high-dimensional range image is a mapping from a set of points to a vector of ranges of values. Formally, we can define an N-dimensional range image as below.

Let Σ_M be an M-dimensional lattice space. An N-dimensional range image is a mapping f: $\Sigma_M\rightarrow R^N$.

Fuzzy subfiber, introduced in the next section, is designed to represent high-dimensional range images and λ-connectedness on a fuzzy subfiber is defined to segment high-dimensional range images.

3. FUZZY SUBFIBERS AND ITS FUZZY CONNECTEDNESS

In this section, we define fuzzy subfiber and show some of its algebraic properties. Also, we define fuzzy connectedness on fuzzy subfibers.

Before formally defining fuzzy subfiber, we first informally explain the reason for choosing the word "subfiber." In mathematics, fiber is a topological and geometric object which also has the meaning of multidimension. A 2D fuzzy subset is a mapping that maps each point of an array to a value in the range [0, 1] which is a one-dimensional space. A fuzzy subfiber is a mapping that maps each point of an array to a vector, each element of which is a value between 0 and 1. That is, a fuzzy subfiber maps an array to a value in a multidimensional space, which is why we call it a fuzzy subfiber instead of a fuzzy subset.

Let Σ_N be an N-dimensional discrete space. We now give the formal definition of a fuzzy subfiber.

DEFINITION 3.1. An N-dimensional and n-range fuzzy subfiber is a mapping from Σ_N to $[0,1]^n$:

$$f: \Sigma_N \to [0,1]^n, \qquad [0,1]^n = [0,1] \times \cdots \times [0,1].$$

We call it an (N,n)-fuzzy subfiber. When $N=2$ and $n=1$, we have

$$f: \Sigma_2 \to [0,1],$$

which is Rosenfeld's 2D fuzzy subset.

It can be seen easily that a high-dimensional range image is equivalent to a fuzzy subfiber. For example, a color image is equivalent to

$$f: \Sigma_2 \to [0,1]^3.$$

An (N,n)-fuzzy subfiber f can be decomposed to a vector of n 2D fuzzy subsets $f_1, f_2, \ldots, f_n$, where f_i $(i=1,\ldots,n)$ maps a point to a value in $[0,1]$. Namely, $f=(f_1, f_2, \ldots, f_n)$.

Similar to classical fuzzy set theory, three basic algebraic operators—negation, addition, and composition—can be defined on a fuzzy subfiber.

DEFINITION 3.2. Let $f=(f_1,\ldots,f_n)$ and $g=(g_1,\ldots,g_n)$ be two (N,n)-fuzzy subfibers and let x be a point in the corresponding N-dimensional

space. Algebraic operators negation, addition, and composition can be defined as follows:

1. $\neg f(x) = 1 - f(x)$.
2. $(f+g)(x) = (f_1(x) + g_1(x), \ldots, f_n(x) + g_n(x))$.
3. $(fg)(x) = (f_1(x)g_1(x), \ldots, f_n(x)g_n(x))$.

The following properties can be proved:

PROPERTIES 2.1.

1. Let f and g be two (N, n)-fuzzy subfibers. $(f, +)$ and $(f, \cdot)$ are two semigroups.
2. $\neg(f+g) = (\neg f)(\neg g)$.
3. $\neg(fg) = \neg f + \neg g$.
4. $f(g+h) = fg + fh$.

These properties could be very useful in image processing. The major topic of this paper is the concept that fuzzy subfiber and its geometric and topological properties can be used to perform image segmentation, so the above mentioned properties and their applications to image processing are not addressed in this paper.

As fuzzy connectedness on a 2D fuzzy subset, fuzzy connectedness on a fuzzy subfiber is the most important concept for image segmentation. We now start to formally define this concept. Before giving the definition of fuzzy connectedness, we first define the degree of point adjacency, which is the basis of the degree of connectedness.

DEFINITION 3.3. Let f: $\Sigma_N \to [0,1]^n$ be a fuzzy subfiber. The degree of adjacency between two points x and y with respect to f in Σ_N is

$$C_{\text{adj}}(x, y) = \begin{cases} E(f(x), f(y)), & x \text{ and } y \text{ are adjacent,} \\ 0, & \text{otherwise,} \end{cases}$$

where E is a function on two vectors,

$$E: [0,1]^n \times [0,1]^n \to [0,1],$$

and E satisfies

$$E(u, u) = 1,$$

$$E(u, v) = E(v, u).$$

have a path from x to z denoted by $R(x,z)$: $x_0=x, x_1, \ldots, x_s=y=y_0, y_1, \ldots, y_t=z$. Because $C(x,z) \geq C_{\text{path}}(R(x,z))$ and $C_{\text{path}}(R(x,z)) = \min\{C_{\text{path}}(P(x,y)), C_{\text{path}}(Q(y,z))\} \geq \lambda$, we can conclude that $C(x,z) \geq \lambda$. ■

COROLLARY 3.1. $C(x,z) \geq \min\{C(x,y), C(y,z)\}$.

According to Theorem 3.1, λ-connectedness can be used to segment the fuzzy subfiber into several regions (actually, the Σ_N is segmented to several parts based on the fuzzy subfiber). Therefore, it can be used for multidimensional range image segmentation directly. We could say that λ-connected segmentation about the gray scale or fuzzy subset λ-connected segmentation is a kind of region-growing segmentation [13].

4. PROPERTIES AND EFFICIENT ALGORITHMS FOR λ-CONNECTED SEGMENTATION

We gave the definitions of fuzzy subfiber and λ-connectedness on the fuzzy subfiber. In this section, we discuss some basic properties of λ-connectedness and develop several efficient algorithms for finding all λ-connected parts of a fuzzy subfiber. When we say λ-connectedness, we mean λ-connectedness on a fuzzy subfiber. Because a high-dimensional range image is equivalent to a fuzzy subfiber and a region of a segmentation of an image is equivalent to a λ-connected part of a fuzzy subfiber, the problem of segmentation of a high-dimensional range image is the same as the problem of finding all λ-connected parts of a fuzzy subfiber.

LEMMA 4.1. *If $\lambda_1 > \lambda_2$ and x and y are two points in the same λ_1-connected class, then x and y must be in the same λ_2-connected class.*

LEMMA 4.2. *If $\lambda_1 > \lambda_2$, then λ_1-connected segmentation is a refinement of λ_2-connected segmentation, where by a λ-connected segmentation, we mean the segmentation generated using λ-connectedness.*

The λ-connectedness defined in Section 3 is a very weak condition when it is applied to image segmentation. Under this λ-connectedness definition, two adjacent points p and p' in one λ-connected class may have very low degree of adjacency as long as there exists one path Π between p and p' so that $C(\Pi)$ is larger than or equal to λ. In the following definition we define a λ-normal-connected subset which is a very strong condition for image segmentation. Inside a λ-normal-connected class, the degree of adjacency of any two adjacent points must be larger than or equal to λ.

DEFINITION 4.1. Let S be a subset of Σ_N and let f be a subfiber on Σ_N. If for any path Π in S, $C_{\text{path}}(\Pi) \geq \lambda$, then we say S is λ-normal-connected with respect to f.

LEMMA 4.3. *$S \subset \Sigma_N$ is a λ-normal-connected subset with respect to the fuzzy subfiber f if and only if for any two adjacent points p and p' in S, $C_{\text{adj}}(p, p') \geq \lambda$.*

THEOREM 4.1. *λ-normal-connectedness is not an equivalence relation.*

Theorem 4.1 holds, because λ-normal-connectedness does not satisfy the transitive property.

Although the λ-normal-connectedness is not an equivalence relation, the split-and-merge technique in pattern recognition [13] can be used to obtain the normal λ-connected subsets.

We already introduced the concepts of fuzzy subfiber, λ-connectedness, and λ-normal-connectedness, and showed that they can be used for high-dimensional range images segmentation in multidimensional spaces. Now we start to discuss the algorithms for λ-connected segmentation and λ-normal-connected segmentation.

Let f: $\Sigma_N \to [0,1]^n$ be a fuzzy subfiber, and let the function of adjacency, $E(u,v)$ and $C_{\text{adj}}(x,y)$ be as given. For arbitrary $\lambda \in [0,1]$, the algorithm that can find all λ-connected parts in Σ_N, is shown as follows.

ALGORITHM A. *Breadth-first-search technique for λ-connectedness.*

Step 1: Let p_0 be a point in Σ. Set $L(p_0) \leftarrow *$ and $STACK \leftarrow STACK \cup p_0$ i.e., labeling p_0 and p_0 is sent a stack *STACK*.

Step 2: If *STACK* is empty, go to Step 4; otherwise, $p_0 \leftarrow STACK$ (top of *STACK*). Then, $L(p_0) \leftarrow 0$.

Step 3: For each p linking p_0, if $L(p) \neq 0$, $L(p) \neq *$, and $C(p, p_0) \geq \lambda$, then $STACK \leftarrow STACK \cup p$ and $L(p) \leftarrow *$. Then, go to Step 2.

Step 4: Stop. $S = \{p: L(p) = 0\}$ is one λ-connected part.

LEMMA 4.3. *Algorithm A is time optimal and the time complexity of Algorithm A is $O(|n \cdot \Sigma_N|)$.*

Proof. We view the Σ_N as a graph whose nodes are the points of Σ_N and whose edges are the pairs (x, y), where x and y are adjacent points in Σ_N. Therefore, the theorem is true based on Tarjan's result [11]. ■

We now give an algorithm for finding λ-normal-connected classes. Usually, for multidimensional binary images, the quadtree and/or octree data representation approaches should be used to store the images first; then do the expansion [14]. In this paper, we directly process our problem dependent upon the split-and-merge technique because we cannot easily find a 2^N-tree approach to represent the fuzzy subfiber.

ALGORITHM B. *Split-and-merge technique for λ-normal-connectedness. Here we are concerned with (2, n)-fuzzy subfibers.*

Step 1: Let Σ_2 be a $(2^l \times 2^l)$ array. $L \leftarrow \Sigma_2$, where L is a set.

Step 2: If region L is already homogeneous, i.e., L is a λ-normal-connected set, $NOMERGE \leftarrow NOMERGE \cup L$ and go to Step 3; otherwise, divide L into four subsquares, L_{11}, L_{12}, L_{21}, and L_{22}; then $NOSPLIT \leftarrow NOSPLIT \cup \{L_{11}, L_{12}, L_{21}, L_{22}\}$.

Step 3: If there exist four regions which are four sons of some region L in *NOMERGE*, namely, L_{11}, L_{12}, L_{21}, and L_{22}, they will be conquered by the merging approach. The merging approach is described as follows. Test any two λ-normal-connected subsets which are neighbors and belong to different regions in L_{11}, L_{12}, L_{21}, and L_{22}. Merge these subsets if their union is a λ-normal-connected subset of L. Then $NOMERGE \leftarrow NOMERGE - \{L_{11}, L_{12}, L_{21}, L_{22}\}$. Otherwise, if there are no such four regions, then stop when *NOSPLIT* is empty; else choose a region from *NOSPLIT*, send it to L, and go to Step 2.

THEOREM 4.2. *There is an $O(n|\Sigma_2| + |\Sigma_2|\log|\Sigma_2|)$ time split-and-merge algorithm for normal λ-connected segmentation on the (2, n)-fuzzy subfiber.*

Proof. First, compute λ-adjacency of all adjacent pairs. There are $O(|\Sigma_2|)$ adjacent pairs, and each function $C(p, q)$ takes $O(n)$ for computing. Thus, $O(n|\Sigma_2|)$ time is needed to compute λ-adjacency of all adjacent pairs.

In [12], a data structure for split-and-merge segmentation with respect to average brightness and an $O(|\Sigma_2|\log|\Sigma_2|)$ time algorithm were discussed. Steps 1 and 2 of Algorithm B are just like those in [12], but the condition of merging changed from average brightness to λ-normal-connectedness.

Suppose that we know the adjacency list of all λ-normal-sets Adjlist(L_{ij}) in each L_{ij}, $i, j = 1, 2$, as well as that we have four such regions in Step 3. Let L_{ij} be a $t \times t$ square. Then we have two boundary lines $R(L_{ij})$ and $B(L_{ij})$ in L_{ij}, each of which has t points and will be considered for merging.

Let A be a λ-normal set in L_{11}. Find all points of A that are in $R(L_{11})$, namely, P_1. For each point in P_1, test if it is λ-adjacent to its adjacent points which are in opposite line denoted $L(L_{12})$ in L_{12}. We denote

$$\text{AOpps}(A) = \{x: x \in L(L_{12}) \text{ and there is a point } p \in P_1$$

$$\text{such that } x \text{ and } p \text{ are adjacent}\},$$

$$\mathrm{AOpps}_\lambda(A) = \{x: x \in L(L_{12}) \text{ and there is a point } p \in P_1 \text{ such that } C_{\mathrm{adj}}(x,p) \geq \lambda\}.$$

Pick up a point x in $\mathrm{AOpps}_\lambda(A)$. If $x \in B$, a λ-normal set in L_{12}, then find out all points of B that are in $L(L_{12})$, namely, Q_1. $A \cup B$ is a λ-normal set if and only if $\mathrm{AOpps}(A) \cap Q_1 = \mathrm{AOpps}_\lambda(A) \cap Q_1$. We consider the following two cases:

1. If $A \cup B$ is a λ-normal set, then union A and B, let

$$\mathrm{AOpps}(A \cup B) = \mathrm{AOpps}(A) \cup \mathrm{AOpps}(B) - \mathrm{AOpps}(A) \cap Q_1 - \mathrm{AOpps}(B) \cap P_1,$$

$$\mathrm{AOpps}_\lambda(A \cup B) = \mathrm{AOpps}_\lambda(A) \cup \mathrm{AOpps}_\lambda(B) - \mathrm{AOpps}(A) \cap Q_1 - \mathrm{AOpps}(B) \cap P_1.$$

Choose x in $\mathrm{AOpps}_\lambda(A \cup B)$ and repeat until $\mathrm{AOpps}_\lambda(A)$ becomes empty. However, in this case, suppose B' in L_{12} satisfied above (just like B) for $A \cup B$, but B and B' are adjacent in $\mathrm{Adjlist}(L_{12})$. Then $A \cup B \cup B'$ is not a λ-normal set.

2. If $A \cup B$ is not a λ-normal set, delete x from $\mathrm{AOpps}(A)$ and $\mathrm{AOpps}_\lambda(A)$. Choose x in $\mathrm{AOpps}_\lambda(A)$ and repeat until $\mathrm{AOpps}_\lambda(A)$ becomes empty.

After we complete merging on a pair opposite lines, we do another pair, i.e., L_{21} and L_{22}, and then we merge $L_{11} \cup L_{12}$ and $L_{21} \cup L_{22}$.

In addition, we need to construct the new adjacent list of λ-normal sets for L.

Now, we start to analyze the time requirement for the algorithm. We consider that if the point p in L_{11} is not on the boundary of L_{11}, then p is met only twice when we compute the P_1 and/or Q_1 and it was added to a new λ-normal set. If the point p in L_{11} is on the boundary of L_{11}, then p is met $O(t)$ times. The reason is that deciding whether $\mathrm{AOpps}(A) \cap Q_1 = \mathrm{AOpps}_\lambda(A) \cap Q_1$ needs to check each element in these two set linear times, and there are at most $O(t)$ such points. Thus the time needed for the preceding consideration is $O(t^2)$.

On the other hand, the number of total adjacent pairs at most is $O(t^2)$ in L, so the time complexity of constructing the adjacent list of λ-normal sets for L is $O(t^2)$. Actually, we have an easier procedure to construct the adjacent list: (1) Collect four adjacent lists of λ-normal sets for L_{ij} and (2) if sets A and B were merged, then add joined pairs into the new adjacent list.

Let $T(m)$ be the time cost of the algorithm, where $m = 2^l$ is length of each edge in the region, and $m^2 = |\Sigma_N|$. $T(m)$ is the sum of time costs of three kinds of subroutings:

1. Compute λ-adjacency for all adjacent pairs, which costs $O(nm^2)$.
2. Time costs for segmenting four son regions is $4T(m/2)$.
3. Merging four son regions' time costs is $O((m/2)^2)$.

So, we have

$$T(m) = 2^2 T(m/2) + c_1(m/2)^2 + c_2(nm^2) = O(m^2 \log_2 m + nm^2).$$

Thus, $T(m) = O(n|\Sigma_2| + |\Sigma_2| \log |\Sigma_2|)$. ■

COROLLARY 2. *There is an* $O(nN|\Sigma_N| + |\Sigma_N| \log |\Sigma_N|)$ *time complexity split-and-merge algorithm for normal* λ*-connected segmentation on the* (N, n)*-fuzzy subfiber. For the indirect adjacency, the time complexity of the algorithm is* $O(n2^N|\Sigma_N| + |\Sigma_N| \log |\Sigma_n|)$.

5. APPLICATION OF FUZZY SUBFIBERS TO LITHOLOGY DETERMINATION

In this section, we discuss an application of fuzzy subfibers to lithology determination that is a very important problem in oil–gas exploration. The ultimate goal of oil–gas exploration is to bear oil–gas reservoirs or locate stratigraphic traps [7]. No doubt, the most precise method is to drill many wells in an exploration area. In this way, petroleum geologists know the exact locations of oil–gas reservoirs and their capacities. Unfortunately, this method costs too much (tens of millions of dollars for drilling one well), so this method is not affordable. A feasible approach adopted by petroleum geologists can be described as follows. Based on the geological background and data obtained from some geological surveys, petroleum geologists choose an area that is highly probable to be an oil–gas reservoir. Then a few wells are drilled and logged in the area and seismic data are collected for the area using seismic methods. Well-log data are used by experienced human interpreters (maybe aided by computers) to determine the geological properties of the area. Finally, these human interpreters are able to decide if the area has an oil–gas reservoir and estimate the capacity of the reservoir. Because massive amounts of well-log data need to be analyzed, this task is tedious, time consuming, and error prone. Therefore, to effectively and efficiently determine lithology, petroleum geologists need automated methods to help perform the task [8–10].

There are two types of oil–gas reservoirs: structural reservoirs and lithological reservoirs. Structural reservoirs are determined based on the shapes of the layers, while lithological reservoirs are determined based on the lithology of the area. In lithological oil–gas reservoirs, sandstones and carbonates are the most common reservoir rocks, that is, rocks capable of containing gas, oil, or water. Therefore, if petroleum geologists can determine the lithology of stratum of an area, they would be able to judge if the area is a lithological oil–gas reservoir or not.

Lithology determination means to decide the rock types in stratum. The lithology of stratum is extremely complex and a small area may contain many different types of rocks. For example, in some area Σ, the stratum has five different types of lithologies: limestone, dolomite, sandstone, gypsum, and coal. A small region X in this area may contain 35% limestone, 40% dolomite, and 25% gypsum, so the lithology of this small area can be represented as the vector

$$\text{Litho}(X) = (0.35/\text{limestone}, 0.4/\text{dolomite}, 0.25/\text{sandstone}, 0/\text{gypsum}, 0/\text{coal}).$$

In our approach, each small region forms a pixel. We call the vector $\text{Litho}(X)$ as a lithological vector of X. A lithological section is a two-dimensional array of such small regions (pixels), each of which is associated with a lithological vector. Thus, each lithological section is a five-dimensional range image or a fuzzy subfiber:

$$\text{Litho}(X) = (\text{Litho}_1(X)/u_1, \text{Litho}_2(X)/u_2, \ldots, \text{Litho}_n(X)/u_n)$$

or

$$\text{Litho}(X) = (\text{Litho}_{u_1}(X), \text{Litho}_{u_2}(X), \ldots, \text{Litho}_{u_n}(X)).$$

Lithological determination consists of two steps. The first step is to determine the lithology for each small area, namely, to generate a lithological vector for each pixel so as to create a fuzzy subfiber. The second step is called trap search and it finds regions with the same lithology in the given lithological section. It can be seen that the second step segments the lithological section. The first step is a classification problem and the second step is a problem of segmentation of high-dimensional range image. Therefore, our fuzzy subfiber approach is applied in the second step. To give a whole picture of the approach, we first briefly discuss an approach applied in the first step.

We applied two approaches—a fuzzy approach and a machine learning approach—to attack the problem of determining lithology of small regions. In this paper, we briefly discuss the fuzzy approach. The machine learning approach is reported in [16].

In this application, two kinds of data are available. One is so called well-log data and the other is seismic data. Each data item is about a small region and is described by a few seismic parameters including amplitude, interval velocity, frequency, porosity, decline and absorption, and Poisson ratio. The lithologies of well-log data are known. In contrast, the lithologies of seismic data are unknown and need to be determined.

In our approach, we use a fuzzy classification model to determine the lithology of small regions. A training algorithm is used to train the model using well-log data. Once the model is trained, it is then applied to determine the lithology of small regions described by seismic data.

We now briefly describe the fuzzy classification model. The training algorithm is not given in this paper. This is a general classification model and can be applied to other classification problems.

Let us assume that each data item (the feature of a small region) is described by a set of seismic parameters, $v_1, v_2, \ldots, v_m$. Although each small region is very small, it still contains many points, each of which may take a different value for a given parameter. Thus, in a region X, we have a possibility distribution function for each parameter. A possibility distribution function of a parameter shows the properties of the parameter in the region X. A possibility distribution function is a mapping Possi $\Sigma \times \text{Domain}(v) \to [0,1]$. Notice that X is a point in Σ, but it is a region in a seismic section. Each region X is described by m possibility distribution functions. If $X \in \Sigma$, then $g_v(\text{Possi}_v(X), a)$ is the possibility that the value of the parameter v is a. In general,

$$\text{Possi}(X) = \left(\text{Possi}_{v_1}(X), \text{Possi}_{v_2}(X), \ldots \text{Possi}_{v_m}(X)\right)$$

shows m possibility functions regarded as the parameters $v_1, \ldots, v_m$.

The fuzzy classification model used in this application is

$$\text{Litho}(X) = \text{Possi}(X) * \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m) * \begin{pmatrix} R_{u_1 v_1} & R_{u_2 v_1} & \cdots & R_{u_n v_1} \\ R_{u_1 v_2} & R_{u_2 v_2} & \cdots & R_{u_n v_2} \\ \cdots & \cdots & \cdots & \cdots \\ R_{u_1 v_m} & R_{u_2 v_m} & \cdots & R_{u_n v_m} \end{pmatrix}$$

and we define

$$L_{u_i}(X) = \sum_{j=1}^{m} \lambda_j \sum_{t=1}^{k} \text{Possi}_{v_j}(X, t_0 + (t-1)\delta_t) \cdot R_{u_i v_j}(t_0 + (t-1)\delta_t),$$

$$\text{Litho}_{u_i}(X) = L_{u_i}(X) \Big/ \left(\sum_{i=1}^{n} L_{u_i}(X) \right).$$

Here, $\lambda_1, \lambda_2, \ldots, \lambda_m$ are the weights,

$$\text{diag}(\lambda_1, \ldots, \lambda_m) = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \lambda_m \end{pmatrix}$$

and $R_{u_i v_j}$ is the relationship between parameter v_j and lithology u_i. $R_{u_i v_j}$ is also a possibility function as

$$R_{u_i v_j}: \text{Domain}(v_j) \to [0,1],$$

which means the compatbility of v_i's value a and the possibility that the lithology belongs to u_i, where $a \in \text{Domain}(v_j)$. The matrix $[R_{u_i v_j}]_{m \times n}$ is denoted by R.

The input data Possi(X) can be obtained from the known data, such as amplitude profile, velocity profile, porosity profile, etc. The feature Possi(X) can be extracted from m seismic parameter profiles, and the lithology of each small region X can be computed through the trained fuzzy classification model.

We now discuss how to apply the fuzzy subfiber approach to search lithology traps. The change of lithology in strata does not follow any rules. The change may be crisp or may be fuzzy. The λ-connected segmentation on fuzzy subfibers suits this kind of problems.

Using Algorithm A or Algorithm B in Section 4, we could start with any point in a lithological fuzzy subfiber. We can get a λ-connected (or λ-normal-connected) subset in the lithological fuzzy subfiber, and we believe the subset is a lithological trap. After that, we delete this subset in the lithological fuzzy subfiber and repeat the procedure to track the λ-connected subsets until nothing is in the lithological fuzzy subfiber.

We directly apply this method to a group of real seismic data. The exploration area was a 1-km^2 area located in the southwest of China. Data

used included well-log data as well as three-dimensional seismic prospecting data obtained from some seismic methods. The well-log data were taken from two wells. Each well-log datum is a five element vector which describes a small region in one well. The first element of the vector is the lithology of the region. The remaining four elements are the possibility distribution functions of the seismic parameters: porosity, amplitude, velocity, and frequency. Porosity is the percentage of the total pore volume of the rock. Velocity used here is the interval velocity of a seismic wave traveling through the rock. Amplitude and frequency are two features of the seismic wave. Similar to the well-log data, each seismic datum describes a small region. The difference is that a seismic datum is a four element vector which does not inlcude the lithology of the region.

The lithologies of the area include carbonate (limestone and dolomite), gypsum, shale, silt, sandstone, and coal. According to the well-log data and expert experiments, the reservoir rock type of the area is pordolomite, which is one kind of dolomite. Pordolomite is a necessary condition for an oil–gas reservoir, but it is not a sufficient condition. We used five different categories of lithologies: carbonate, pordolomite, gypsum, shale/silt/sandstone, and coal. Because shale, silt, and sandstone are not crucial for discovering an oil–gas reservoir in this area, we combine them into one category. On the other hand, pordolomite is the only reservoir rock type in this area, so we separate it from carbonate. In this stratum, gypsum and carbonate prevent the oil–gas from shifting. We aimed at discovering the region where lithology is pordolomite surrounded by gypsum and carbonate. The well-log data were used to train the fuzzy classification model.

In this 3D object, we choose one section of data as test data. After performing recognition, we obtained the lithology distribution in this section; refer to Figure 1.

When the lithology of each small region is known, geologists wish to search and pick up similar lithological traps [15]. That is why we developed λ-connected segmentation on the fuzzy subfiber. Because the lithology of the small regions varies little, we chose the λ-connected segmentation on the fuzzy subfiber of lithology.

First, we need to select the function of fuzzy connectivity. There is no method telling us what the best choice is, so we should select it according to the situation of the investigated area. However, we often choose the function

$$E_1(u,v) = 1 - \frac{\|u-v\|}{\|u\|+\|v\|}.$$

Second, we use Algorithm A defined in Section 4 to search lithology traps. Figure 2 shows an example of lithology traps.

(a) Traditional representation

(b) Fuzzy subfiber representation

Fig. 1. Lithology distribution representations.

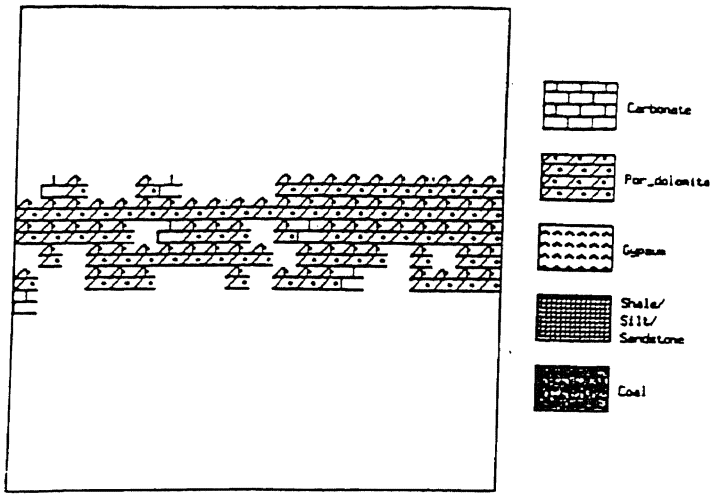


Fig. 2. Lithology trap obtained by applying λ-connected segmentation approach Figure 1(b).

6. CONCLUSION

The definition and properties of subfibers have been studied. The suitability of describing fuzzy multifeature images in multidimensional space using subfibers is discussed. We have applied the subfiber method to determine/classify the seismic lithology. The experimental results indicate that the proposed approach and algorithms are useful for image processing and pattern classification, in particular, for oil–gas exploration.

REFERENCES

1. A. Rosenfeld, Fuzzy digital topology, *Inform. Control* 40:76–87 (1979).
2. A. Rosenfeld, On connectivity properties of grayscale pictures, *Pattern Recognition* 16:47–50 (1983).
3. A. Rosenfeld, Fuzzy geometry of image subsets, *Pattern Recognition Lett.* 2:311–317 (1984).
4. A. Rosenfeld and R. Klette, Degree of adjacency or surroundness, Technical Report CAS TR-53, CS-TR-1380, Center for Automation Research, University of Maryland, 1984.
5. S. K. Pal and A. Ghosh, Fuzzy geometry in image analysis, *Fuzzy Sets and Systems* 48:23–40 (1992).
6. L. Chen, Three-dimensional fuzzy digital topology and its applications (I), *Geophys. Prospecting Petrol.* 24(2):86–89 (1985).

7. F. Aminzadeh, *Pattern Recognition and Image Processing*, Seismic Exploration Series, 1987.
8. P. L. Love and M. Simaan, Segmentation of a seismic section using image processing and artificial intelligence technique, *Pattern Recognition* 18:409–419 (1985).
9. J. M. Busch, W. G. Fortney, and L. N. Berry, Determination of lithology from well logs by statistical analysis, *SPE Formation Evaluation* 2 (1987).
10. S. R. Rogers and J. H. Fang, Determination of lithology from well logs using a neural network, *Amer. Assoc. Petrol. Geol. Bull.* 76:731–739 (1992).
11. A. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.
12. L. Chen, The λ-connected segmentation and the optimal algorithm for split-and-merge segmentation, *Chinese J. Comput.* 14(5) (1991); *Comput. Control Abst.* 26(306):639–649.
13. T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD, 1982.
14. H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1989.
15. J. Jacobs, A theory for seismic interpretation, *62nd Annual SEG Meeting, Expanded Abstracts*, 1991, pp. 231–234.
16. J. Zhang, L. Chen, and H. D. Cheng, Determination of lithology using instance-based learning, *Proceedings of the Sixth International Symposium on Artificial Intelligence*, 1993.

Received 1 April 1993; revised 10 October 1992, 10 April 1993; accepted 3 September 1993

ELSEVIER

Information Sciences 115 (1999) 201–220

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

The equivalence between two definitions of digital surfaces

Li Chen *, Donald H. Cooley, Jianping Zhang

Department of Computer Science, Utah State University, Logan, UT 84322-4205, USA

Received 1 January 1997; received in revised form 1 June 1998; accepted 15 September 1998

Abstract

Digital surfaces deal with properties of surfaces in digital spaces. In the early 80's, researchers began to establish definitions for digital surfaces. Unlike surfaces in continuous spaces, digital surfaces have different characteristics. A general and intuitive definition for digital surfaces is still an open problem. This paper presents a proof of the equivalence of two digital surface definitions. One of the definitions was developed based on simple surface points given by Morgenthaler and Rosenfeld [1], and the second uses a parallel-move concept as given by direct adjacency [2,3]. A by-product of this proof of equivalence is that any simple surface point in a simple surface belongs to one of six types of simple surface points. © 1999 Elsevier Science Inc. All rights reserved.

1. Introduction

A digital surface usually means the digitized points of a continuous surface. Different sampling ratios can generate different digital surfaces for a continuous surface. On the other hand, for a set of sampled surface points (e.g. from CT), one usually generates a fitted continuous surface to represent the original points. Different fitting algorithms can produce different continuous surfaces. However, such translations are not always necessary. How to deal with a surface in a digital domain has become an important topic. Researchers think that studies of digital curves, surfaces, and solid objects may provide a mathematical foundation for image processing, recognition, and representation [4].

* Corresponding author. E-mail: chenl@sys6.cs.usdu.edu

0020-0255/99/$19.00 © 1999 Elsevier Science Inc. All rights reserved.
PII: S0020-0255(98)10072-5

A digital space is a domain which can be stored in computers. Typically, a one-dimensional digital space means the integer set $Z = \{...., -3, -2, -1, 0, 1, 2, 3, ...\}$, and a m-dimensional digital space is $Z^m = Z \times Z... \times Z$. Because computers can only store limited data, we sometimes use $\Sigma_1 = \{1, 2, ... N\}$ to represent a one-dimensional space. Thus, an m-dimensional digital space is an m-dimensional array Σ_m. We also call Σ_m an m-dimensional grid space. Intuitively, a digital surface is a point set of Σ_m which looks like a surface but does not look like a curve or solid object.

Digital surfaces were first studied by Artzy, Frieder, and Herman in 1981 [5]. They defined a digital surface as the face of a solid object which consisted of unit cubes (called voxels). This is a very intuitive concept, but it is very weak. One can easily generate examples so that they do not satisfy the mathematical definition of surfaces: a neighborhood of each point on a surface is homeomorphic to a 2D disk. In addition, such a surface is always closed. Morgenthaler and Rosenfeld [1] on the other hand, only considered discrete points to form a "digital" surface in Σ_3. Morgenthaler and Rosenfeld gave the first mathematical definition for digital surfaces which is not dependent on continuous surfaces. Unfortunately, Morgenthaler–Rosenfeld's surfaces are not very intuitive and hence are not easy to understand. In order to verify Morgenthaler and Rosenfeld's surfaces are visually reasonable, Kong and Roscoe [6] discussed the continuous analog of Morgenthaler and Rosenfeld's surfaces. In fact, Kong and Roscoe gave an interpolation in three dimensional Euclidean spaces for each type of "simple surface points" in Morgenthaler and Rosenfeld's surfaces. However, all of these can only deal with the "local property" of a point so that they only considered closed surfaces.

In order to assimilate both intuitiveness and strictness in defining a digital surface, we introduced the concept of parallel-moves and gave an intuitive definition for digital surfaces in 3D grid spaces [2,3]. Although some intuitive ideas appeared in Kong–Roscoe [6], the surface definition of [2,3] used a recursive methodology which can be extended to more general cases such as general digital manifolds.

This parallel-move based definition is able to deal with boundary points as well as "inner" points. Using this definition, we have developed several fast algorithms for surface decision, boundary searching, and surface tracking [3]. Furthermore, these algorithms can be modified to the recognizing digital manifolds [2]. Thus, with parallel-move based definition, one can consider both closed and unclosed digital surfaces and design fast algorithms for surface decision, boundary searching, and surface tracking.

However, the parallel-move based surface definition in [2,3] only consider direct adjacency. Allowing indirect adjacency, Morgenthaler and Rosenfeld's surfaces have nine types of digital surfaces. Kong and Roscoe also considered all nine cases [6].

There are two reasons we discuss only direct adjacency in this paper. First, among the total nine types of surfaces, there are only two of them contain non-trivial examples [6,7]. Second, only direct adjacency has the property of unique interpretation. An example with multiple interpretations of indirect adjacency will be shown in Section 2. We also give an example to show that a visually true surface point is not in any of nine types. That is to say, Morgenthaler–Rosenfeld's definition is not complete. Therefore, the case of direct adjacency is the most basic and important one. For indirect adjacency, it is still an open problem to give a simple and intuitive definition of digital surfaces [7].

In this paper, we will prove that Morgenthaler–Rosenfeld's simple surface point in direct adjacency is the regular (inner) surface point defined in [2,3]. Therefore, we show the equivalence between Morgenthaler–Rosenfeld's surfaces and the regular closed surfaces defined in [2,3]. This proof brings interesting result: Morgenthaler–Rosenfeld's simple surface points have exactly six types [8]. The by-product of this paper may be more important to digital topology.

We will introduce the basic concepts and Morgenthaler–Rosenfeld's simple surfaces in Section 2 and the parallel-move based digital surfaces in Section 3. Then, we will prove the equivalence between these two definitions of digital surfaces in Section 4.

2. Simple digital surfaces in Σ_3

Digital surfaces were first studied by Artzy et al. [5]. However, the first mathematical definition of closed digital surfaces in discrete spaces was given by Morgenthaler and Rosenfeld [1]. Based on this definition, many important theoretical results were obtained, e.g., the discrete Jordan theorem [1,4]. An alternative approach was introduced by Kovalevsky [9]. In his work, digital images were represented by topological spaces. Recently, Rosenfeld, Kong and Wu have examined and proven some basic properties of digital surfaces [10].

To begin with, we define the concept of point adjacency. Let us first consider a two-dimensional digital space Σ_2. A point $p = (x, y)$ in Σ_2 has two horizontal $(x, y \pm 1)$ and two vertical neighbors $(x \pm 1, y)$. These four neighbors are called directly adjacent points of p. p also has four diagonal neighbors: $(x \pm 1, y \pm 1)$. These eight (horizontal, vertical and diagonal) neighbors are called indirectly adjacent points of p. Here, direct adjacency is a part of indirect adjacency. In 2D, directly adjacent points are called 4-adjacent points, and indirectly adjacent points are called 8-adjacent points.

Let Σ_3 be a 3-dimensional digital space. Two points $p = (x_1, x_2, x_3)$ and $q = (y_1, y_2, y_3)$ in Σ_3 are directly adjacent points if

$$\sum_{i=1}^{3} |x_i - y_i| = 1.$$

Directly adjacent points are also called 6-adjacent points because the number of directly adjacent points to p are 6 in Σ_3. Note, we always assume p is not at the border of Σ_3. p and q are 18-adjacent points if

$$\max_{1\leqslant i\leqslant 3}|x_i - y_i| = 1,$$

and

$$\sum_{i=1}^{3}|x_i - y_i| \leqslant 2.$$

p and q are 26-adjacent points if

$$\max_{1\leqslant i\leqslant 3}|x_i - y_i| = 1,$$

and

$$\sum_{i=1}^{3}|x_i - y_i| \leqslant 3.$$

In a three-dimensional space Σ_3, a point has six 6-adjacent points, eighteen 18-adjacent points, twenty six 26-adjacent points. 6-, 18-, 26-adjacent is denoted as α-adjacent, where $\alpha = 6, 18, 26$. The term "indirect adjacency" means all three types of adjacency are allowed.

A simple α-path is a point set $v_0, v_1, \ldots, v_n$, where v_i and v_{i+1} are α-adjacent for all $i = 0, \ldots, n-1$ and $v_i \neq v_j$ for all i, j. Two points p, q are α-connected if there is a simple α-path $v_0, v_1, \ldots, v_n$ such that $v_0 = p$ and $v_n = q$. Suppose that S is a subset of Σ_3, we say S is α-connected or an α-component if any two points in S are α-connected, i.e. there is an α-path in S linked with these two points.

Let $\bar{S} = \Sigma_3 - S = \{a \mid a \in \Sigma_3 \text{ and } a \notin S\}$, and $N_{27}(p)$ be the $3 \times 3 \times 3$ neighborhood of a point p, i.e. $N_{27}(p) = \{a \mid a \text{ is 26-adjacent to } p\}$.

Definition 2.1 (*Morgenthaler–Rosenfeld's surfaces*)[1]. Let $S \subset \Sigma_3$ and $p \in S$, p is called an (α, β)-simple (digital) surface point, $\alpha, \beta = 6, 18, 26$, if

1. $S \cap N_{27}(p)$ has exactly one α-component α-adjacent to p, denote this component A_p.
2. $\bar{S} \cap N_{27}(p)$ has exactly two β-components, c_1 and c_2, β-adjacent to p.
3. If $q \in S$ and q is α-adjacent to p, then q is β-adjacent to both c_1 and c_2.

S is called a (α, β)-simple (digital) surface if every point in S is a (α, β)-simple surface point.

Hence, there are totally nine types of simple (digital) surfaces. According to this definition, a Morgenthaler–Rosenfeld's surface is a simple closed surface because no boundary is defined in this definition. The definition of Morgenthaler–Rosenfeld surfaces focuses on each individual point and its neighbor-

hood. This definition provides a mathematical description for digital surfaces. However, it is difficult to understand the real meaning of the three conditions in the definition. In order to present a visual interpretation of Morgenthaler–Rosenfeld's simple surface points, Kong and Roscoe "transform (hard) digital topology problems into (fairly easy) problems of continuous topology, which are able to solve"[6].

Kong and Roscoe considered continuous surface-unit-cells in continuous space R^3 which are bounded by points in Σ_3 [6]. Note, for raster images Herman and Weber presented the concept of surface-unit-cells in 1982 [11]. A concept of "plates" such as a surface-unit-cell and "plate cycle" such as all plates around a point in $N_{27}(p)$ provide an intuitive interpretation of a simple surface-point. However, the "plates" and "plate cycle" were defined in a very difficult way. Kong and Roscoe considered all nine types of (α, β)-surface points and gave a continuous interpolation for each type [6]. Kong and Roscoe's work is also important theoretically. With a simple supplement to Kong and Roscoe's work, the nine types of surface points can be reduced to only two types: (6,26)- and (18,6)-simple surface points [7]. The remaining digital surfaces only contain trivial examples [6,7]. However, Kong and Roscoe's work is also difficult to understand since they introduced many new topological concepts.

(6, 26)- and (18, 6)-simple surfaces are the only useful cases. The example in Fig. 1 will show that the (18,6)-simple surface has no unique interpretations.

Kong and Roscoe [6] forced such cases into a unique interpretation. That is to say Morgenthaler–Rosenfeld's surfaces are reasonable, but do not give a complete description. Since the ambiguity here is real existence, so for different cases, one should have different interpretations. In Morgenthaler–Rosenfeld's surfaces, only (6,26)-surface has the property of consistency.

In addition, the definition of Morgenthaler-Rosenfeld's surfaces (Definition 2.1) is also not visually complete. Let us look at the following example (Fig. 2), point p is surely a surface-point in human interpretation, but it cannot be a (α, β)-simple surface point.

Although, to allow indirect adjacency in real image processing is necessary, to give a solid mathematical analysis is not simple. One reason for this difficulty is the indirect adjacency break Jordan property "a closed curve separates the rest of the plane into two components." Discussion of indirect adjacency is given in [7].

In this paper, we only consider direct adjacency. In what follows, we omit the word direct, that is, adjacency means direct adjacency. We also simplify Definition 2.1 to the following form:

Definition 2.1' (*Morgenthaler–Rosenfeld's surfaces*)[1]. Let $S \subset \Sigma_3$ and $p \in S$, p is called a simple surface point, if

1. $S \cap N_{27}(p)$ has exactly one 6-component adjacent to p, denote this component A_p.

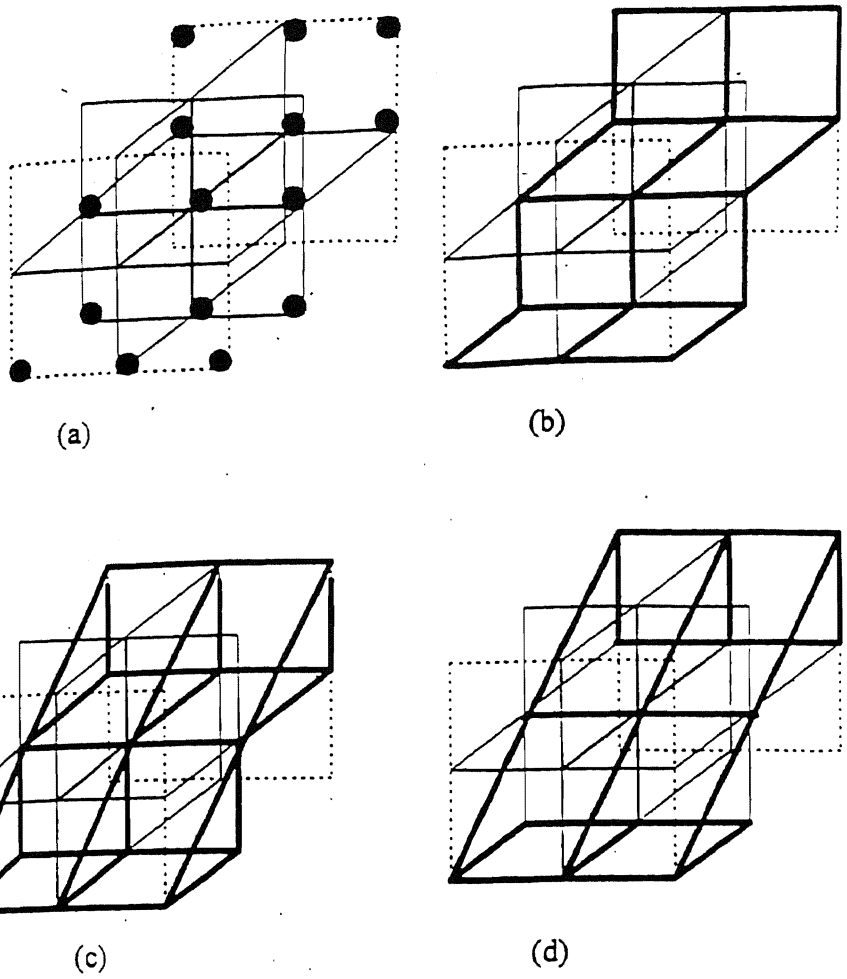


Fig. 1. (a) A digital point set, (b) first interpretation, (c) second interpretation which is a solid object and (d) third interpretation.

2. $\bar{S} \cap N_{27}(p)$ has exactly two 26-components, c_1 and c_2, 26-adjacent to p.
3. If $q \in S$ and q is 6-adjacent to p, then q is 26-adjacent to both c_1 and c_2.

3. Parallel-move based surfaces in Σ_3

Without making use of continuous space, Chen and Zhang [2,3] gave a recursive definition for k-dimensional unit cells using parallel-moves in direct adjacency. Based on line-unit-cells, surface-unit-cells and 3D-unit-cells, an

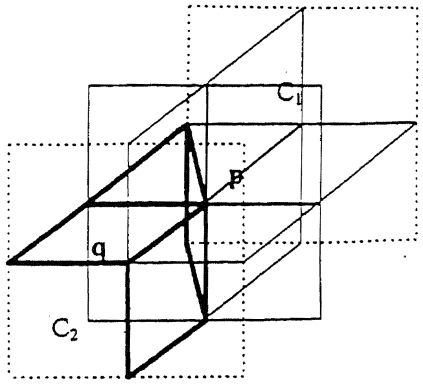


Fig. 2. A visually true surface point but not a (α,β)-surface point for any α and β.

intuitive definition of surfaces was proposed. An optimal surface tracking algorithm was designed by parallelly moving a line-unit-cell. This technique philosophically matches the idea of "curves made by dots and surfaces made by curves."

A point $p=(x,y,z)$ in Σ_3 is called a point-unit-cell (or point-cell). A pair of points $p=(x_1,x_2,x_3)$ and $q=(y_1,y_2,y_3)$ in Σ_3 is called a line-unit-cell (or line-cell), if p and q are adjacent points, i.e.,

$$d(p,q)=\sum_{i=1}^{3}|x_i-y_i|=1.$$

A line-cell is denoted as a set of p and q, $\{p\}\cup\{q\}$ or $\{p,q\}$.

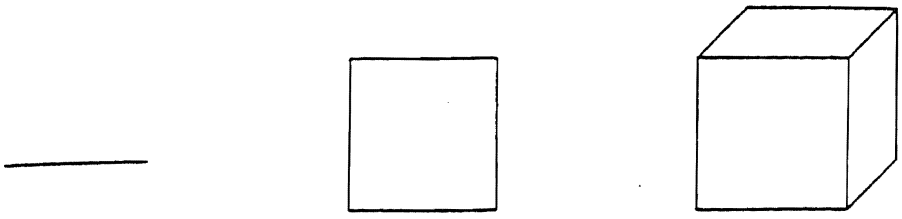
A surface-unit-cell (or surface-cell) is a set of 4 points which form a unit square parallel to coordinate planes. So, a surface-cell is a union of two line-cells $\{p,p'\}$ and $\{q,q'\}$. The set of $\{p,p',q,q'\}$ are four distinct points, satisfying:

$$d(p,p')=1 \quad \text{and} \quad d(q,q')=1,$$

$$d(p,q)=1 \quad \text{and} \quad d(p',q')=1.$$

A 3D-unit-cell (or 3D-cell) is a unit cube which includes 8 points. A 3D-cell is the union of two surface-cells $\{p_1,p_2,p_3,p_4\}$ and $\{q_1,q_2,q_3,q_4\}$ such that $\{p_1,p_2,p_3,p_4\}\cap\{q_1,q_2,q_3,q_4\}=\emptyset$ and for all $i=1,2,3,4$, $d(p_i,q_i)=1$.

In this note, each line-cell, surface-cell, or 3D-cell is a set of points in Σ_3. We sometimes call a point p a point-cell denoted $\{p\}$. All the operations of sets can be used for line-cells, surface-cells, and 3D-cells. Fig. 3(a)–(c) shows a line-cell, a surface-cell and a 3D-cell, respectively. Some general considerations of unit-cells are given in [2,12].



(a) 1_D cell (b) 2_D cell (c) 3_D cell

Fig. 3. Three kinds of unit-cells.

Definition 3.1. A line-cell $\{q,q'\}$ is called a parallel-move of line-cell $\{p,p'\}$ if $\{p,p'\} \cap \{q,q'\} = \emptyset$, and $d(p,q) = 1$ and $d(p',q') = 1$.

In the same way, we can define parallel-moves for point-cells, surface-cells, etc.

Lemma 3.1. *In Σ_3, each line-cell has four parallel-moves, and a line-cell and each of its four parallel-moves form a surface-cell.*

In fact, a point-cell and one of its parallel-moves forms a line-cell, and a surface-cell and one of its parallel-moves forms a 3D-cell. We only use the parallel-moves of a line-cell in this paper.

Two line-cells (or surface-cells) are point-adjacent if they share a point. For example, line-cells $l1$ and $l2$ are point-adjacent in Fig. 4(a), and surface-cells $s1$ and $s2$ in Fig. 4(b) are point-adjacent. Two surface-cells are line-adjacent if they share a line-cell. For example, surface-cells $s1$ and $s3$ in Fig. 4(c) are line-adjacent. In order to keep the concept in a simple form, we do not introduce a general definition of unit-cell adjacency here, see [2] for a general definition.

Two line-cells are point-connected if they are two end elements of a line-cell path with point-adjacency. For example, line-cells $l1$ and $l4$ in Fig. 4(a) are point-connected. Two surface-unit-cells are line-connected if they are two end elements of a surface-cell path with line-adjacency. For example, $s1$ and $s2$ in Fig. 4(c) are line-connected.

We now introduce our digital surface definition. We call such a surface a parallel-move based (digital) surface to distinguish it from a Morgenthaler–Rosenfeld's simple surface of Definition 2.1.

Definition 3.2[2,3]. A (point-)connected set $S \subset \Sigma_3$ is a parallel-move based digital surface if

1. Each line-cell of S has at least 1 and at most 2 parallel-moves in S.
2. Any two surface-cells of S are line-connected in S.
3. S does not contain any 3D-cell.

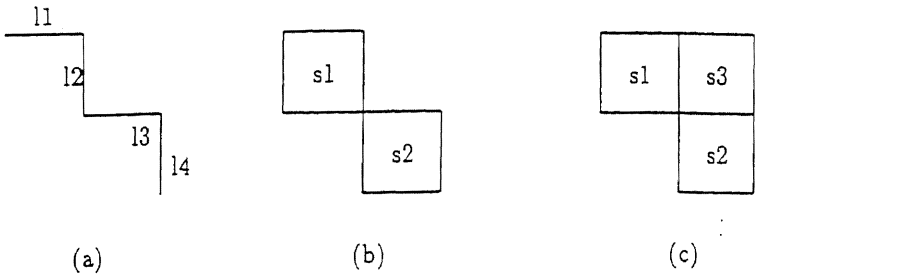


Fig. 4. Adjacency and connectedness.

All of the three conditions in Definition 3.2 are necessary conditions. If the first condition is not satisfied, there is one line-cell in S which has 3 parallel-moves, then S cannot be a surface, (see Fig. 5(a)). If the second condition is violated, that is, there exist two surface-cells in S that are not line-connected, then there must exist two surface-cells in S which are point-connected only (see Fig. 5(b)). If the last condition is not satisfied, S contains some 3D-cell; hence, S cannot be a surface.

Definition 3.3. p is an inner point in surface S if every line-cell containing p has exactly 2 parallel-moves in S. p is a boundary point in S if there exists a line-cell containing p that has only one parallel-move in S. The set of all boundary points is denoted by ∂S, (see Fig. 6).

The differences between Morgenthaler–Rosenfeld's simple surfaces definition and this definition of parallel-move based surfaces are: the former definition is dependent on the point and its neighborhood, but the latter definition is based on a line-cell and its parallel-moves so it can easily deal with boundary points. The Kong and Roscoe's analog is still restricted in the neighborhood of

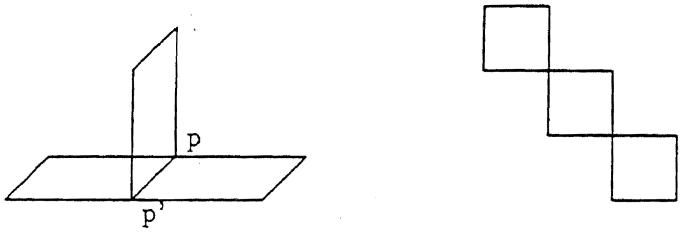


Fig. 5. Two instances which are not considered in surfaces.

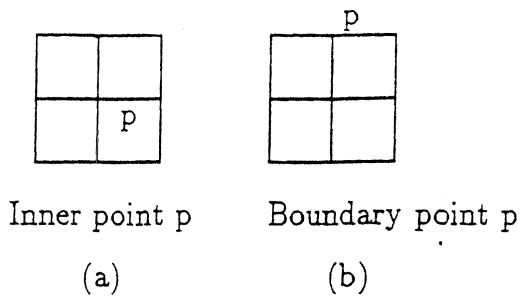


Fig. 6. Inner and boundary point.

a point [6]. They gave an intuitive interpretation but did not simplify the definition. This analog may make surface recognition procedures even more complicated [7]. In addition, because of the restriction of the neighborhood $N_{27}(p)$, both Morgenthaler–Rosenfeld surfaces and Kong and Roscoe's analog cannot deal with the boundary points of a surface.

4. Simple surface points and regular inner surface points

The definition of parallel-move based surfaces is simple and intuitive. The question is "what is the relationship between such a digital surface and a Morgenthaler–Rosenfeld surface?" This section will show that a closed regular surface is precisely a Morgenthaler–Rosenfeld simple surface. In order to establish the relationship between Morgenthaler–Rosenfeld's simple surfaces and our parallel-move based surfaces, we need to introduce the concept of regular surface points.

If p is a point of a parallel-move based surface S, p is regular if all of S's surface-cells including p are line-connected in S. If p is both inner and regular, then p is called a regular inner surface point. Although most surface points are regular, we show two examples for non-regular surface points in Fig. 7.

To deal with general cases, we can expand the meaning of a regular surface point to any (point-)connected set as follows.

Definition 4.1. Let S be a connected subset of Σ_3. Assume $p \in S$ and $S(p) = S \cap (N_{27}(p) \cup \{p\})$. p is called a regular surface point of S if:

1. Each line-cell in S containing p has at least 1 and at most 2 parallel-moves in $S(p)$.
2. Any two surface-cells containing p in S are line-connected in $S(p)$.
3. $S(p)$ does not contain any 3D-cell.

We say p is a regular inner surface point if p is a regular surface point and each line-cell containing p has exactly two parallel-moves in $S(p)$.

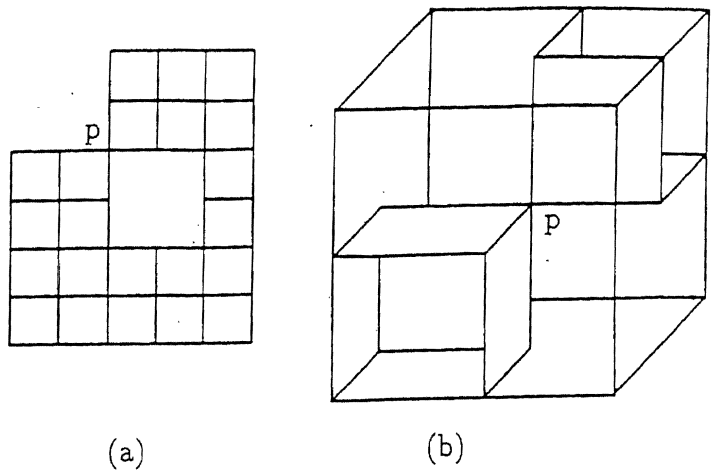


Fig. 7. (a) Non-regular point p; (b) Inner but non-regular surface point p.

Theorem 4.1. *A simple surface point of a set $S \subset \Sigma_3$ is a regular inner surface point, and vice versa.*

Proof. To simplify the proof, we separate it into two parts. In the first part we prove that if p is a simple surface point, then p is also a regular inner surface point. In the second part we prove that if p is a regular inner surface point, then p is a simple surface point.

Proof of part 1. Suppose that p is a simple surface point as reviewed in Section 2. We want to show the following:

1. $S(p) = S \cap (N_{27}(p) \cup \{p\})$ does not have any 3D-cell,
2. each line-cell containing p in S has exactly two parallel moves $S(p)$, i.e. p is inner, and
3. any two surface-cells containing p in S are line-connected in $S(p)$.

First, suppose p is a simple surface point in S. Obviously, p cannot be a corner point of any 3D-cell in S, hence we establish statement 1.

For 2, to begin with, p has three or more (directly) adjacent points in $S \cap N_{27}(p)$. Otherwise, $\bar{S} \cap N_{27}(p)$ is connected, so p is not a simple surface point in accordance with condition (2) of its definition in Section 2.

We define here a grid plane is a set of all points with a fixed z $P_z = \{(x,y,z)|(x,y,z) \in \Sigma_3\}$, all points with a fixed y $P_y = \{(x,y,z)|(x,y,z) \in \Sigma_3\}$, or all points with a fixed x $P_x = \{(x,y,z)|(x,y,z) \in \Sigma_3\}$.

Next, let $p' \in S$ be an arbitrary adjacent point of p. Therefore, there is a grid plane (such as plane1, plane2 or plane3 in Fig. 8) which contains p in $N_{27}(p)$ and does not contain p'. Thus, all directly and indirectly adjacent points of p' are in one side of the grid plane (including the plane). Dependent upon the third condition of the definition of simple surface points, each $c_1(p)$ and $c_2(p)$,

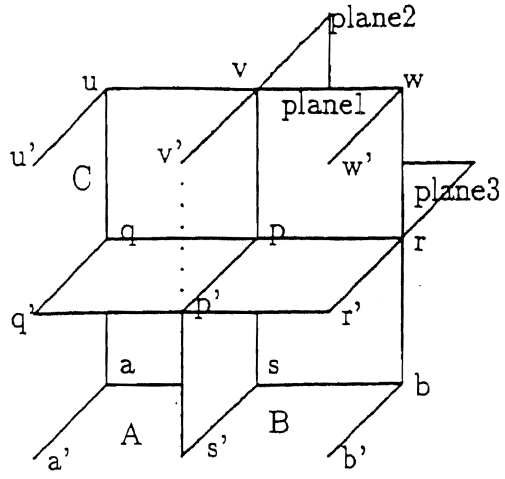


Fig. 8. S in $N_{27}(p)$.

which are defined in Section 2, have one point in the side of the plane because they must be indirectly adjacent to p'. Also, all of the parallel-moves of line-cell $\{p,p'\}$ are in the side of the plane.

If $\{p,p'\}$ has no two parallel-moves in S, then all of the points which are in $\bar{S} \cap N_{27}(p)$ and are in the side of the plane are connected; thus, $c_1(p)$ and $c_2(p)$ are connected. So, $\{p,p'\}$ has two or more parallel-moves.

Now, we prove line-cell $\{p,p'\}$ has no more than two parallel-moves. In contrast, suppose $\{p,p'\}$ has three parallel-moves; without loss of generality, we let the three parallel-moves $\{q,q'\}$, $\{r,r'\}$, and $\{s,s'\}$ of $\{p,p'\}$ be described in Fig. 8.

Suppose we are sure that $\{p,p',q,q',r,r',s,s'\}$ are in S. We let A be the set whose elements are in $\{a,a'\}$ but not in S, and B be the set whose elements are in $\{b,b'\}$ but not in S. Because $S \cap N_{27}(p)$ does not have any 3D-cells, A and B are not empty.

Since p is a simple surface point; A and B, A and C, or B and C must be indirectly connected according to the second condition of the definition of simple surface points in Section 2. Using the same reasoning for A and C, and B and C, we only need to prove: p is no longer a simple surface point when A and B, or A and C are indirectly connected.

(i) If A and B are indirectly connected, then $a \in A$ and $b \in B$; otherwise, A does not connect with B in $N_{27}(p) \cup p$. We know that a and b are indirectly connected. Meanwhile, every s's indirectly adjacent point in $N_{27}(p) \cup p$ is below plane3, and two of them are contained in $c_1(p)$ and $c_2(p)$, respectively. Thus, all of s's indirectly adjacent points must be indirectly adjacent to a or b. Then $c_1(p)$ and $c_2(p)$, denned in Section 2 are indirectly connected, so p is not a simple surface point.

(ii) If A and C are indirectly connected, a must belong to A. We may suppose that $A \cup C \subset c_1(p)$ and $B \subset c_2(p)$. We now must discuss the following two cases.

(ii.a) If $b \in S$ then $c_2(p) = b'$. Thus, q cannot be indirectly adjacent to b, i.e. $c_2(p)$. According to the third condition of the simple surface points' definition, p is not a simple surface point.

(ii.b) If b is not in S, i.e. $b \in c_2(p)$; then we can see if q connects to b and r connects to a, they must pass the plane 2. On the other hand, if $\{u, v, w\} \subset S$, then there is a point of $c_1(p)$ in $\{u', v', w'\}$ based on no 3D-cell in S. However, the point and A cannot be indirectly connected in $N_{27}(p)$, that is, A and C are not indirectly connected. Thus, there must be a point δ in u, v, and w such that it is in $c_1(p) \cup c_2(p)$. If δ is in $c_1(p)$ then each point in plane2 indirectly connects with a or δ. q cannot indirectly connect with the point b. If δ is in $c_2(p)$ then r cannot indirectly connect with the point a. According to the third condition of the simple surface point definition, p is not a simple surface point.

Therefore, we have proven statement 2. We now prove statement 3 to complete part one of the proof. Statement 3 says that any two surface-cells of $S(p)$ are line-connected if p is a simple surface point.

Actually, there are only two cases for two surface-cells A and B including p in $N_{27}(p) \cup p$ which are not line-adjacent (See Fig. 9). In the following we show that these two surface-cells A and B are line-connected in $N_{27}(p) \cup p$ when p is a simple surface point and both A and B are in S.

For case (a), if one of u, v, or w is in S, then surface-cells A and B are line-connected. Otherwise, because each (p, a) and (p, b) have two parallel-moves, C and D must be in S. By the same reasoning, points e and f are in S, so we can see that the p is not a simple surface point because $\bar{S} \cap N_{27}(p)$ has three indirectly connected parts.

For case (b), if u and v are in S, then A and B are line-connected. Otherwise, C and D, or $\{a, e, x, p\}$ and $\{a, p, y, w\}$ must be in S; so we have a case like case (a). Thus, p is not a simple surface point if A and B are not line-connected.

We have thus proven statement 3. To summarize, p is a regular inner surface point if p is a simple surface point.

Proof of part 2. Suppose p is a regular inner surface point. We want to show that p is a simple surface point, i.e. the following three statements are true:

1. $S \cap N_{27}(p)$ has exactly one component adjacent to p, denote this component A_p.
2. $\bar{S} \cap N_{27}(p)$ has exactly two 26-connected components, c_1 and c_2, 26-adjacent to p.
3. If $q \in S$ and q is adjacent to p, then q is 26-adjacent to both C_1 and c_2.

Our strategy for proving part two is different from part one's proof. We just enumerate all possibilities in which a regular inner surface point p can appear, and we then prove that all possible regular inner surface points are simple surface points.

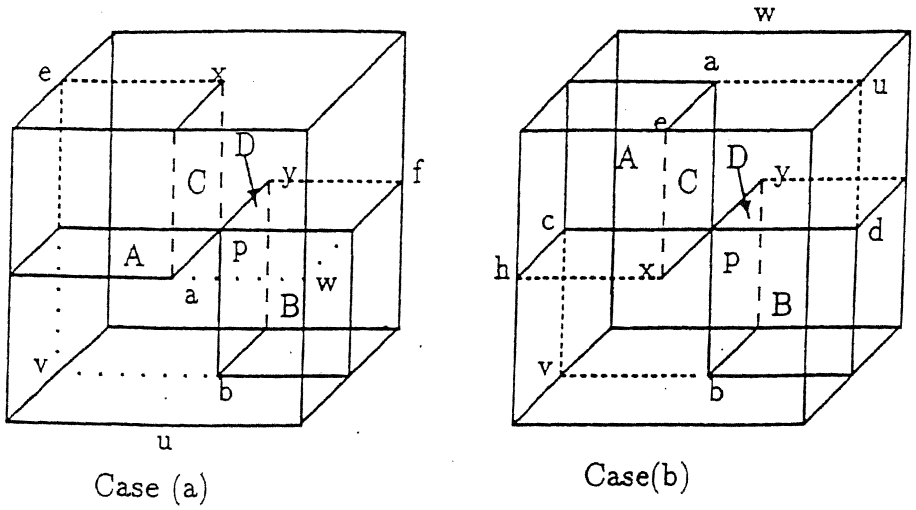


Fig. 9. Two cases in which surface-cells are not line-adjacent.

We know if p is a regular surface point of S, then p is not a corner point of any 3D-cell which is in S. Because of the (point-) connectivity of S, p has a adjacent point denoted by p'. Also, $\{p,p'\}$ has two parallel-moves, denoted by $\{a,a'\}$ and $\{b,b'\}$; therefore, both a and b are also adjacent to p.

Suppose the surface-cell that is formed by $\{p,a\}$ and $\{p,b\}$ is in S, then there are three surface-cells that are line-connected to each other in $S(p)$. Therefore, they can be illustrated as shown in Fig. 10.

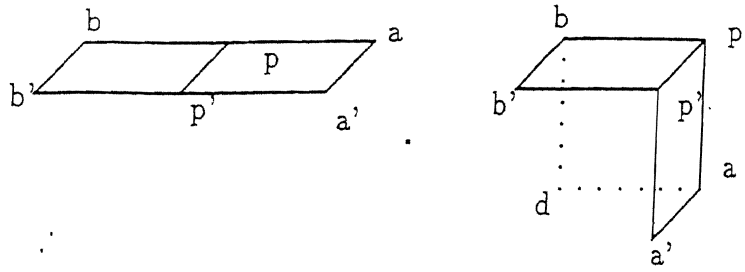


Fig. 11. Two cases.

Here x must not be in S as p is not a corner of any 3D-cell. If there is a point u, v, or w that is adjacent to p; then A, B or C must be in S because any line-cell has exactly two parallel-moves. In this case, there exist two surface-cells which are not connected in $N_{27}(p) \cup \{p\}$, so p is not a regular surface point. Thus, there is no such point u, v, or w which is adjacent to p, so we can see p satisfies the definition of a simple surface point.

On the other hand, if the surface-cell formed by $\{p, a\}$ and $\{p, b\}$ is not in S. We can derive only two different cases as shown in Fig. 11.

We now illustrate all possible developments based on the above cases, meaning that p is kept as a regular inner surface point. We only consider the points that are adjacent to p and the surface-cells that contain p. It is straightforward to see that there is only one possible case for a regular inner surface point with exactly three surface-cells.

(i) If there are exactly four surface-cells in $S(p)$, we have only the following two possible cases keeping p as a regular inner surface point (Fig. 12). We can see that p in either (a) or (b) is a simple surface point.

(ii) If there are five or more surface-cells in $S(p) = S \cap (N_{27}(p) \cup p)$, then only two cases which contain three surface-cells can be developed to generate different results. The two cases are given in Fig. 13.

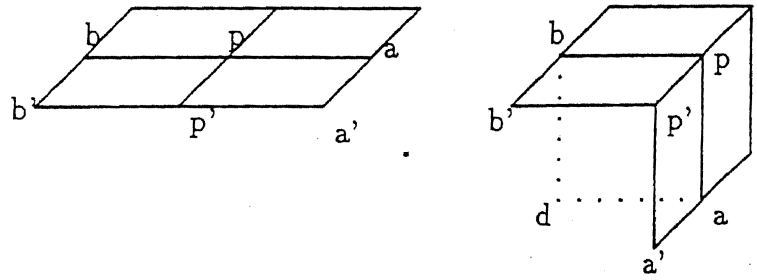


Fig. 12. $S(p)$ has exactly four surface-cells.

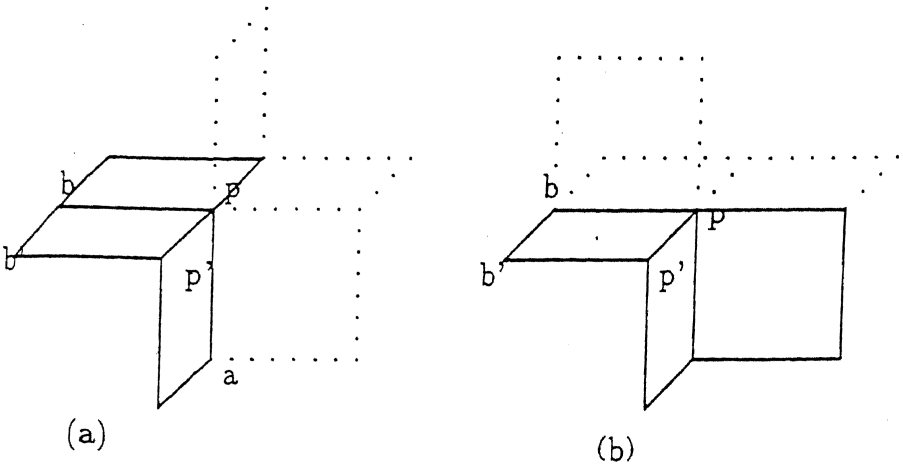


Fig. 13. Two cases that can be developed.

Each (a) and (b) in Fig. 13 has three possible means to add one more surface-cell, some of which overlap. We can reduce such cases to four distinct cases as follows (Fig. 14).

Next, we continuously add new surface-cells to the above cases and maintain p as a regular inner surface point; so that we arrive at the following seven cases (Fig. 15). We can see that (e) already arrives at the final state, where it has only five surface-cells in $S(p)$. On the other hand, (c) also arrives at the final state because it cannot be a regular inner surface point.

Next, We can see that there is only one choice to add a possible surface-cell for each case in Fig. 15 except (c) and (e). After adding a surface-cell, (a), (b), (d), (f), and (g) can be reduced to two cases. Each of which have six surface-cells including p, where p is a regular inner surface point. We show the two cases as follows (Fig. 16).

Finally, we now show why there are no seven or more surface-cells including p in $S \cap (N_{27}(p) \cup \{p\})$ when p is a regular surface point in S. We know p has six adjacent points in $N_{27}(p) \cup \{p\}$; in other words, there are six line-cells including p. If a surface-cell A including p is in S, then A contains two of the six line-cells. When S has seven surface-cells, there must exist a line-cell which is included by three surface-cells. Therefore, S is not a surface.

From the preceding 5-step process, we obtain three types of regular inner surface points contained by five or six surface-cells. Considering Figs. 11 and 12, we have one regular inner surface point with three surface-cells and two regular inner surface points with exactly four surface-cells. There are only six possibilities for p to be a regular inner surface point [6]. We can see that all of the three kinds of the regular inner surface points satisfy the definition of the simple surface points.

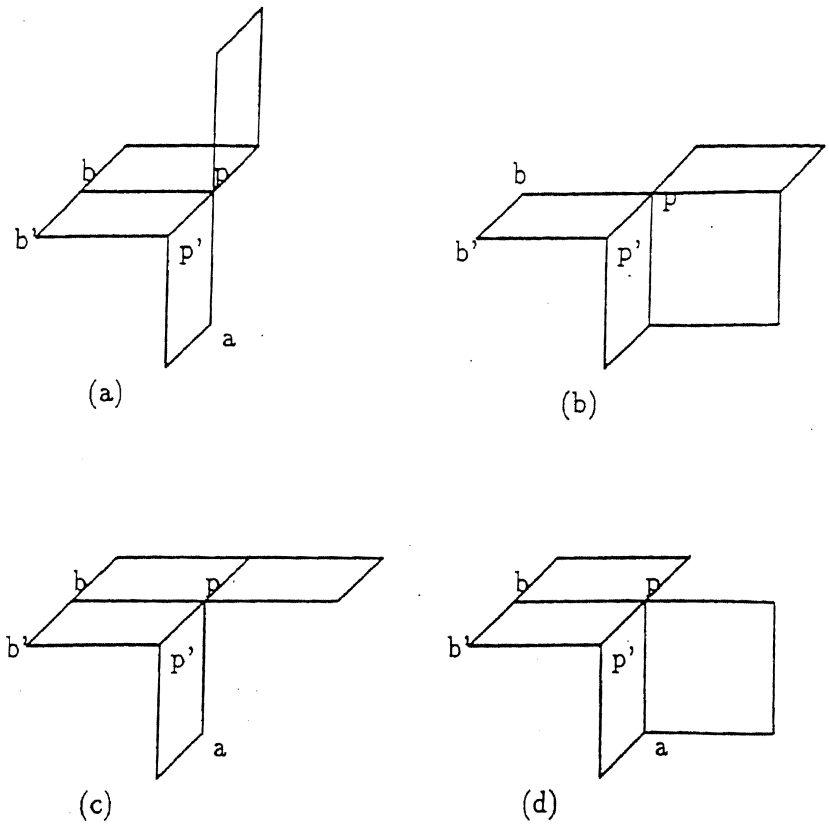


Fig. 14. Four cases.

We now have completed the proof of part 2. Therefore, every regular inner surface point is a simple surface point. Since none of the discussions used the first condition of Morgenthaler–Rosenfeld's definition,

Corollary 4.1. *The second and third conditions of the definition of Morgenthaler–Rosenfeld's surface imply the first condition.*

5. More about regular surface points

In Section 4, we define a regular surface point based on parallel-move based surfaces. We have obtained exact six types of regular inner surface points [8]. In practice, it is may useful to extract all simple surface points in a set S by matching the six types of simple surface points. Theoretically, to decide if S is a

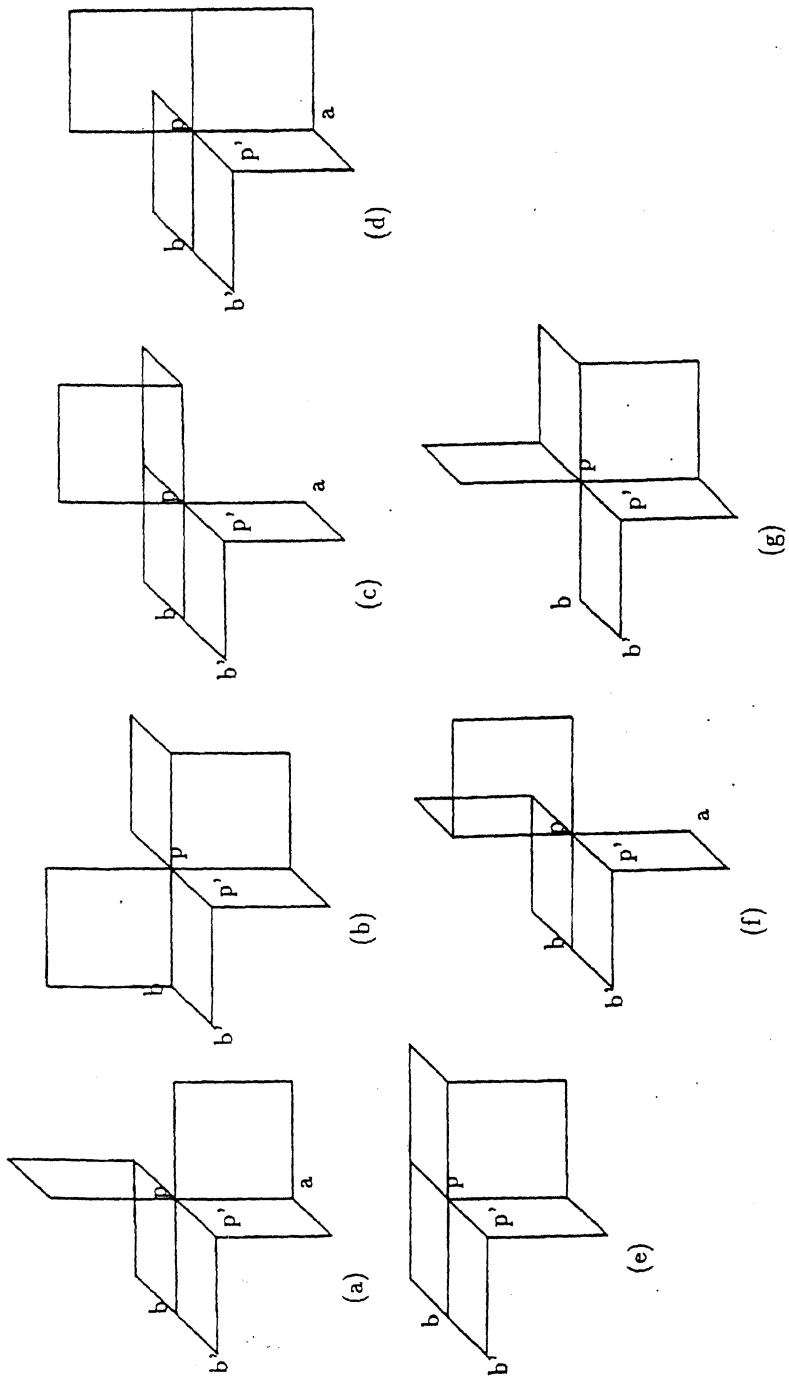


Fig. 15. Seven cases.

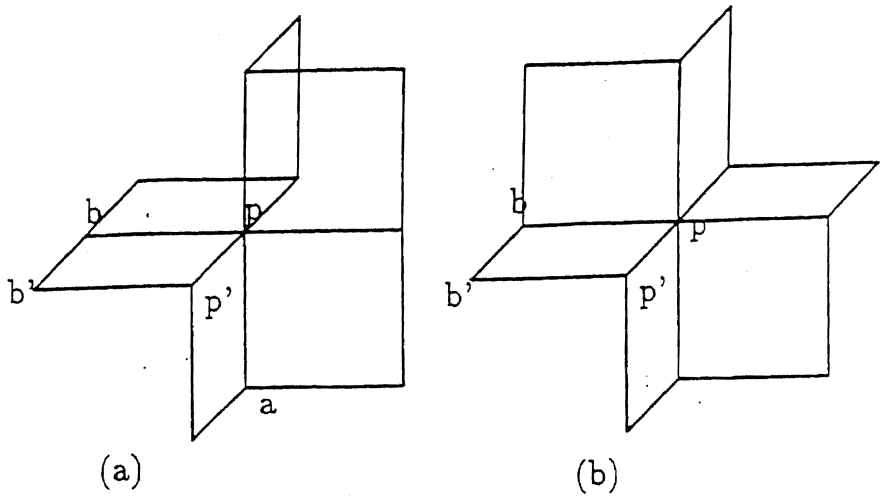


Fig. 16. Two cases.

simple surface need not to test all three conditions of definition 2.1 or to match six types of regular inner surface points.

A Morgenthaler–Rosenfeld's simple surface S is closed, and each point is a simple surface point. A fast algorithm can be developed by testing if S is a parallel-move based closed surface first, and then testing if S contains the case of Fig. 7(b).

In addition, the definition of simple surface points cannot deal with boundary points. The drawback of the definition of regular surface points is how to expand it to deal with indirect adjacency. We discuss this issue in [7]. Comparing with Kong and Roscoe's results [6], Definition 4.1 has some similarities to Proposition 6 in [6]. However, Definition 4.1 is much simpler, intuitive and general.

In summary, this paper proved that Morgenthaler–Rosenfeld's simple surfaces are just cases of regular closed surfaces defined by the authors in [2,3] in direct adjacency. We also can see there are only six types of simple surface points in three dimensional grid spaces [8].

References

[1] D.G. Morgenthaler, A. Rosenfeld, Surfaces in three-dimensional images, Inform. and Control 51 (1981) 227–247.

[2] L. Chen, J. Zhang, Digital manifolds: A intuitive definition and some properties, Proceedings of the Second ACM/SIGGRAPH Symposium on Solid Modeling and Applications, Montreal, 1993.

2. SEGMENTATION AND FITTING

There are many kinds of segmentation, such as measurement space clustering, region growing segmentation, split-and-merg segmentation, edge detection, etc. [1]. In order to segment velocity sections in seismic data effectively, one needs to develc a region growing fuzzy segmentation technique. We term this technique λ-connected segmentation [2-4]. In this section, w present a mathematical definition of segmentation and then introduce the principals of region growing segmentation an split-and-merge segmentation.

A digital image F is a mapping from a grid-space Σ to real set R (or R^n in general). Two points p, q in Σ ar adjacent if the distance between these two points is 1. Two points p, q in $S \subseteq \Sigma$ are connected if there is a path whic consists of adjacent points that links p and q. S is connected if each pair of points in S are connected. We say S is uniforn if F on S (a sub-image) has properties of uniformity. A popular uniformity measure is defined in terms of the maximur difference between any pixel value and the mean value of S.

Since we will be working to develop uniform segments, we define segmentation as a partition of an image int connected subsets, each of which is uniform, but such that no union of adjacent subsets is uniform. The formal definition o segmentation is then:

In a digital image F, if there exists a non-empty segmentation $F_1,F_2,...,F_m$ satisfying

(1) $F_i \cap F_j = \Phi$, if i=1,..,m, j=1,..m, i≠j
(2) $\cup_{i=1,...,m} F_i = F$,
(3) Each F_i is connected,
(4) Each F_i is "uniform",
(5) If F_i and F_j are adjacent, then $F_i \cup F_j$ is not uniform.

Then, $\{ F_1,F_2,...,F_m \}$ is called a segmentation of F.

Region growing segmentation: A region in an image may be viewed as a connected group of pixels, all with similar brightness. The method begins with a single pixel, and then by examining neighbors tries to find a maximum sized connected region of similar pixels. In this manner regions grow from single pixels [1]. One also can use a region or group of pixels as a seed instead of a single pixel. In this case, after selecting the partition (group of pixels), a uniformity test is applied to the region to see if it qualifies as a partition. If the test fails the region is subdivided into smaller regions. This process is repeated until all regions are uniform. The major advantage of using small regions rather than single pixels is reduced sensitivity to noise.

Split-and-Merge Segmentation: A split-and-merge algorithm is an extension of region growing algorithms. It is an advanced segmentation technique, sometimes called quadtree segmentation, used in two dimensional image processing. Split-and-merge segmentation is based on a quadtree partition of an image. One starts with tree nodes (representing square regions of the image) at some intermediate level of the quad tree. If a square is found to be nonuniform, it is replaced by its four subsquares (split). Conversely, if four squares form a uniform square, they are replaced by a single square(merge). This process continues recursively until no further splits or merges are possible [1].

3. λ-CONNECTED SEGMENTATION IN METRIC SPACES

Let (X, D) and (Y, D') be two metric spaces and $f: X \to Y$ be a function. If we define f(X)={y | y =f (x), x ∈ **X**}, then $(f(X),D'_f)$ is the subspace of (Y,D'). Here, point $(x,f(x))$ is denoted by p(x), and $P=\{p(x) \mid x \in X\}$. If

$$d_1 (p(x),p(y))=D(x,y) + D'(f(x),f(y)) \tag{3.1}$$

and

$$d_2 (p(x),p(y))= (D^2(x,y) + D'^2(f(x),f(y)))^{1/2} \tag{3.2}$$

then (P, d_1) and (P, d_2) form two new metric spaces, defined as $(X,d) \otimes (f(X),D_f)$.

[3] L. Chen, J. Zhang, A Digital Surfaces Definition and Fast Algorithms for Surface Decision and Tracking, in: Melter, Wu (Eds.), Vision Geometry II, Proceedings of SPIE 2060, 1993.
[4] T.Y. Kong, A. Rosenfeld, Digital topology: Introduction and survey, Comput. Vision Graphics Image Process 48 (1989) 357–393.
[5] E. Artzy, G. Frieder, G.T. Herman, The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm, Comput. Vision Graphics Image Process 15 (1981) 1–24.
[6] T.Y. Kong, A.W. Roscoe, Continuous analog of axiomatized digital surfaces, Computer Vision, Graphics, and Image Process. 29 (1985) 60–86.
[7] L. Chen, A Note on (α,β)-type digital surfaces in general digital surfaces, Manuscript, 1997.
[8] L. Chen, J. Zhang, Classification of Simple digital Surface Points and A Global Theorem for Simple Closed Surfaces, in: Melter, Wu (Eds.), Vision Geometry II, Proceedings of SPIE 2060, 1993.
[9] V.A. Kovalevsky, Finite topology as applied to image analysis, Comput. Vision Graphics Image Process. 46 (1989) 141–161.
[10] A. Rosenfeld, T.Y. Kong, A.Y. Wu, Digital surface, CVGIP: Graphical Models and Image Process. 53 (1991) 305–312.
[11] G.T. Herman, D. Webster, A topological proof of a surface tracking algorithm, Comput. Vision Graphics Image Process. 23 (1983) 162–177.
[12] L. Chen, J. Zhang, D.H. Cooley, Regular Polygons and Their Applications to Digital Curves, in: Melter, Wu, Latecki (Eds.), Vision Geometry V, Proceedings of SPIE 2826, 1993.

An Intelligent Data Fitting Technique For 3D Velocity Reconstruction

Li Chen Donald H. Cooley Lan Zhang

Department of Computer Science
Utah State University
Logan, UT 84322-4205
chenl@sys6.cs.usu.edu

ABSTRACT

Intelligent data fitting is often used in seismic data processing to reconstruct large three-dimensional data volumes based on relatively small amounts of 2D seismic and/or well-log data. This technique is useful for such applications because of the many different layers and lithologies in stratum. However, if such data fitting is performed over two different layers in the stratum, it often results in incorrect profiles. A more accurate way to perform such fitting is to first perform segmentation of the data into known classes, and then fit the data inside of each segmented class.

In seismic data processing, interval velocity is one of the most important factors in rock type identification, i.e. lithology classification. To find a region with low velocity generally indicates a lithology of high porosity. Such a region is much more likely to contain oil and/or gas. Velocity data is more difficult to obtain than seismic waveform, which is obtained by 3D seismic prospecting. To obtain a vertical trace of interval velocity data requires extensive processing. Generally, such a velocity volume is obtained by means of a data fitting technique. This paper presents a general intelligent data fitting approach to reconstruct interval velocity volumes. This technique uses a process we term *SEGFIT-segmentation followed by fitting*. In *SEGFIT,* we use a fuzzy connected segmentation technique for the segmentation, which we term λ-connected followed by a specialized fitting method which maintains the λ-connectedness of the fitted data in each segmented class. In this paper we show that a real velocity volume can be obtained using this technique.

Key Words: intelligent data fitting, segmentation, volume reconstruction, seismic interval velocity, λ-connectedness.

1. INTRODUCTION

In image processing and computer vision, data fitting is an important and basic tool. For example, simple enlargement of an image requires data fitting as the image grows in size and volume. For many problems the data fitting and interpolation required is much more complex than that for, say, a simple enlargement. For example, in seismic data processing, interpolation between known and unknown data points in a regular grid is a difficult task because of changing geological conditions such as varying lithologies, faults, etc. In such cases, one must use irregular two-dimensional data to reconstruct a three-dimensional data volume. Geophysicists often first choose those points that belong to the same lithology layer to perform the interpolation., fitting only among those points that are in the same lithology or class.

The goal of intelligent data fitting is to keep all fitted data in the same class as that of their guiding points. While ultimately classification is the goal of any segmentation algorithm, in this paper we will be concerned only with segmentation. In this paper, we first extend the concepts of λ-connectedness and λ-connected segmentation to metric spaces. We then discuss interpolation and approximation in metric spaces. Finally, we apply this technique to real velocity volume reconstruction of a seismic image.

Definition 3.1. In $(X,d)\otimes(f(X),D_f)$, if $p(x)=(x,f(x))$ and $p(y)=(y,f(y))$ satisfy $d(x,y)\le\varepsilon$, then $p(x)$ and $p(y)$ are called ε-adjacent.

Definition 3.2. In $P=(X,d)\otimes(f(X),D_f)$, if function $CA_\varepsilon: P\times P\to[0,1]$ satisfies

(1) $CA_\varepsilon(p(x),p(x))=1$,
(2) $CA_\varepsilon(p(x),p(y))=0$, if $d(x,y)>\varepsilon$,

then, $CA_\varepsilon(p(x),p(y))$ is called the adjacent connectivity of $p(x)$ and $p(y)$.

A finite sequence $\Omega(p(x_1),p(x_n))=\{p(x_1),p(x_2),\ldots,p(x_n)\}$ is called an ε-path, if $d(x_i,y_j)\le\varepsilon$, $j=i+1$ and $x_i\neq x_j$, $i\neq j$ excepting $x_1=x_n$. The connected value of an ε-path $\Omega(p(x_1),p(x_n))$ is denoted by

$$CP(\Omega(p(x),p(x_n)) = \min\{CA(p(x_i),p(x_j)) \mid i=1,\ldots,n-1;\ j=i+1\} \tag{3.3}$$

or

$$CP'(\Omega(p(x),p(x_n)) = \prod CA(p(x_i),p(x_j)),\quad i=1,\ldots,n-1;\ j=i+1,\ i=1,\ldots,n-1 \tag{3.4}$$

Finally,

$$C(p(x),p(y)) = \max\{CP(\Omega(p(x),p(y)) \mid \Omega(p(x),p(y)) \text{ is a } \varepsilon\text{-path}\} \tag{3.5}$$

is called the connectivity of p(x) and p(y) under ε-path.

Definition 3.3 $p(x)$ and $p(y)$ in $(X,d)\otimes(f(X),D_f)$ are called direct λ-connected if $CA(p(x),p(y))\ge\lambda$, $\lambda\in[0,1]$; and λ-connected if $C(p(x),p(y))\ge\lambda$.

Theorem 3.1 λ-connectedness is an equivalence relation, i.e., it can be used for classification.

A λ-connected segmentation partitions a metric space into several λ-connected components. Generally, for any function $E: S\times S\to R$ in the set S possessing the following properties:

(i) $E(u,u)\le c$, c is a constant,
(ii) $E(u,v)=E(v,u)$,
(iii) $E(u,v)>=\min\{E(u,w),E(w,v)\}$,

the λ-connectedness of (S,E) is an equivalence relation. λ-connectedness is a general dynamic connectedness relation.

There are many functions satisfying conditions (i), (ii) and (iii) ,such as

$$FA_\varepsilon(p(x),p(y)) = \begin{cases} 1-\dfrac{D(f(x),f(y))}{M}, & d(x,y)\le\varepsilon \text{ and } D(f(x),f(y))\le M \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

and

$$GA_\varepsilon(p(x),p(y)) = \begin{cases} 1-\dfrac{D(f(x),f(y))}{D(f(x0),f(x))+D(f(x0),f(y))}, & d(x,y)\le\varepsilon \text{ and } f(x0)=0 \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

In this paper, the connectivity functions defined as F and G are arrived at by applying (3.4) to FA and FG, respectively. To arrive at a λ-connected segmentation we use a breadth-first-search technique based on Tarjan's algorithm [5].

Algorithm A : Searching a λ-connected part.

Step 1. Let $p0=(x0,f(x0))$. Set $L(p0) \leftarrow *$ (* is a label), and $STACK \leftarrow STACK \cup \{p0\}$
Step 2. If $STACK$ is empty, goto step4; otherwise, $p0 \leftarrow$ top of $STACK$ and $L(p0) \leftarrow 0$;
Step 3. For every element $p=(x,f(x))$ satisfying $d(x0,x) \leq \varepsilon$ and $L(p) \neq 0$ and $L(p) \neq *$; if $C(p,p0) \geq \lambda$, then $STACK \leftarrow STACK \cup \{p\}$ and $L(p) \leftarrow *$, goto **Step 2.**
Step 4. Stop, and $S=\{p \mid L(p)=0\}$ is one λ-connected part.

This algorithm requires linear time, and hence cannot be improved upon. Although this λ-connected segmentation algorithm is a relatively fast equivalence relation classification, the connectivity of a λ-connected set can be very weak [3]. For such a case, we develop the concept of a normal λ-connected set.

Definition 3.4 If x and x' in S are ε-adjacent (ε-neighborhood) implying that $p(x)$ and $p(x')$ are (adjacently) λ-connected, then $(S, f(S))$ is called a normal λ-connected set.

Because an adjacent λ-connected set does not form an equivalence relation, if we use it to segment a set, there may be many solutions. The split-and-merge (quadtree in 2D) segmentation technique can be used in such a case. Let E_m be a m-dimensional grid space, for an image f: $E_m \rightarrow [0,1]$. Also, let $\varepsilon=1$ or $\varepsilon=(m)^{1/2}$. We can describe split-and-merge segmentation as follows:

Algorithm B : Split-and-merge technique for normal λ-connected segmentation (2D case).

Step 1. Build a quadtree structure where each node represents a fixed square region of the image. Start with the root node: $NODE \leftarrow root$. (Technically one can start with the nodes at some intermediate level of the quad tree.).
Step 2. If $NODE$ is a normal λ-connected set, goto **Step 3** for merging; otherwise, break $NODE$ into four children and repeat **Step 2** for every child of $NODE$.
Step 3. Check all four children for a node, and merge all possible normal λ-connected sets based the boundary of each node.

4. INTELLIGENT DATA FITTING AND λ-CONNECTIVITY

In mathematics, the term fitting can describe both interpolation and approximation. Based on a set of guiding points (sample points), a fitting procedure attempts to connect all data points in a region. For example, in a 2D image, for a line one wishes to connect all points that are a part of the line. Sometimes, a linear regression method is used to fit the line. Linear regression has limitations. For example, if there are two lines, linear regression will only find one. In such cases a more intelligent data fitting algorithm must be employed.

Intelligent data fitting in this paper means that (1) fitting is based on segmentation, and (2) fitted data segments the image into uniform regions based on the guiding points.

Let (X,d) be a metric space and (Y,D) be a normed linear space. Also, $X_s \subseteq X$ and f: $X_s \rightarrow Y$. Suppose X_s is segmented by the λ-connected segmentation of (X_s,d) $\otimes(f(X_s), D)$, denoted by $S_\lambda(X_s, f)=\{S_1,...,S_m\}$. Assume there is a F: $X \rightarrow Y$, $S_\lambda(X, F) = \{R_1,...,R_n\}$. If

$$\text{(i)} \quad \forall R_i \, \exists S_j \{ S_j \subseteq R_i \}, \text{ and} \tag{4.1}$$
$$\text{(ii)} \quad R_i \cap S_j = S_j \text{ or } R_i \cap S_j = \phi, \tag{4.2}$$

then we can say F and f are λ-compatible, and $n \leq m$. If $n=m$, F is called a λ-fitting of f.

Let F be a λ-fitting of f, that satisfies

$$f(x)=F(x), \quad \text{if } x \in X_s, \tag{4.3}$$

then F is called a complete λ-interpolation of f.

We now develop a λ-compatible theorem for linear interpolation. The connectivity function was defined in (3.6) and (3.7).

Theorem 4.1 Let $S_\lambda(X_s, f)=\{S_1,...,S_m\}$, and $S_\lambda(X, F) = \{R_1,...,R_m\}$. If

(i) $S_i \subseteq R_i$, $i=1,...,m$ and
(ii) $f(x)=F(x)$, $x \in X_s$ and
(iii) For every $u \in R_i$, there exists $(x, f(x))$ and $(x', f(x'))$ which are adjacent

λ-connected, where $x, x' \in S_i$, $d(u,x) \le \varepsilon$, $d(u, x') \le \varepsilon$, such that

$$F(u) = r f(x)+(1-r) f(x'), \qquad 0 \le r \le 1$$

Then, F and f are λ-fit.

Proof: We just need to prove that, if $u, v \in R_i$, then $(u,F(u))$ and $(v,F(v))$ are λ-connected.

(a) Let the connectivity function be GA and given by (3.7). (Y,D) is a normed linear space, $D(a,b)=||a-b||$. Suppose that $u \in R_i$, we have $x_1,x_2 \in S_i$, $||x_1-x_2|| \le \varepsilon$ and such that

$$C(x_1, x_2) = 1 - \frac{||f(x_1) - f(x_2)||}{||f(x_1)||+||f(x_2)||} \ge \lambda$$

and

$$F(u)=r f(x_1)+(1-r) f(x_2), \qquad 0 \le r \le 1.$$

Let

$$A = 1 - \frac{||F(u) - f(x_1)||}{||F(u)||+||f(x_1)||}$$
$$= 1 - (1 - r)||f(x_2) - f(x_1)||/(||F(u)||+||f(x_1)||)$$

$$B = 1 - \frac{||F(u) - f(x_2)||}{||F(u)||+||f(x_2)||}$$
$$= 1 - r||f(x_2) - f(x_1)||/(||F(u)||+||f(x_1)||)$$

If $A < \lambda$ and $B < \lambda$, then

$$A < C(x_1,x_2) \;\&\; B < C(x_1,x_2),$$

so,

$$(1-r)(||f(x_1)|| + ||f(x_2)||) > ||f(x_1)|| + ||F(u)||, \text{ and}$$
$$r(||f(x_1)|| + ||f(x_2)||) > ||f(x_1)|| + ||F(u)||,$$

implies

$$||f(x_1)|| + ||f(x_2)|| > ||f(x_1)|| + ||f(x_2)|| + 2||F(u)||.$$

But this is impossible. Thus $A \ge \lambda$ or $B \ge \lambda$. Then $(u,F(u))$ and $(x_1,f(x_1))$ are adjacent λ-connected, or $(u,F(u))$ and $(x_2,f(x_2))$ are adjacent λ-connected. Similarly, for any $v \in R_i$ there must exist $y \in S_i$ such that $(v,F(v))$ and $(y,f(y))$ are λ-connected. Because $(x_1,f(x_1))$, $(x_2,f(x_2))$ and $(y,f(y))$ are λ-connected to each other, $(u,F(u))$ and $(v,F(v))$ are λ-connected

based on Theorem 3.1 .

(b) One can similarly prove for a connectivity function FA, as given in (3.5).

Corollary 4.1 Let $S_\lambda(X_s, f)=\{S_1,...,S_m\}$, and (Y,D) be the real space. Assume $X=\{R_1,...,R_m\}$ with $S_i \subseteq R_i$, $i=1,...,m$. If F_i is a function fitted by a Bezier polynomial or parametric B-spline on $P_i=\{(x,f(x)) \mid x \in Si\}$, then $F = \{F_1, F_2,..., F_m\}$ is λ-compatible with f.

5. NORMAL λ-CONNECTED INTERPOLATION AND APPROXIMATION

λ-connectedness between two points only deals with a path linking these two points, so a λ-connected set usually cannot simulate a continuous function in two-dimensions. In this section we will consider a special λ-connected set in which every path is λ-connected. We call such a set a normal λ-connected set. A normal λ-connected set has some interesting mathematical properties. If the λ-connected fitting is a relaxed fitting, then a normal λ-connected fitting is a strict fitting. For example, any continuous fitting in a discrete space is a normal λ-connected fitting in the sense of a single domain component .

Definition 5.1 Let $S_0 \subseteq S$ and f_0: $S_0 \rightarrow R$, where S is a metric space and R is the real space. For two points p, q $\in S_0$, if there exists a ε-path in S, $p_0=p, p_1,..., p_n=q$, and real numbers, $g(p_0)=f(p), g(p_1)... g(p_n)=f(q)$, such that p and q are λ-connected with respect to g, then this path is called λ-connectable and p and q are called λ-connectable in S.

Theorem 5.1 Let D be a ε-net of S, $D_0 \subseteq D$ *and* f_0: $D_0 \rightarrow R$. For any two points p, $q \in D_0$, every shortest path between p and q is λ-connectable in D, and there exists a function f that is a normal λ-connected interpolation of f_0, i.e. (D,f) is a normal λ-connected set.

The special cases of this theorem and their proofs were shown in [6,7]. In fact, f in Theorem 5.1 has properties like those of the B-spline method:

Corollary 5.1 An algorithm exists to generate a normal λ-connected interpolation, such that the resulting function is in the convex closure of its guiding points f_0.

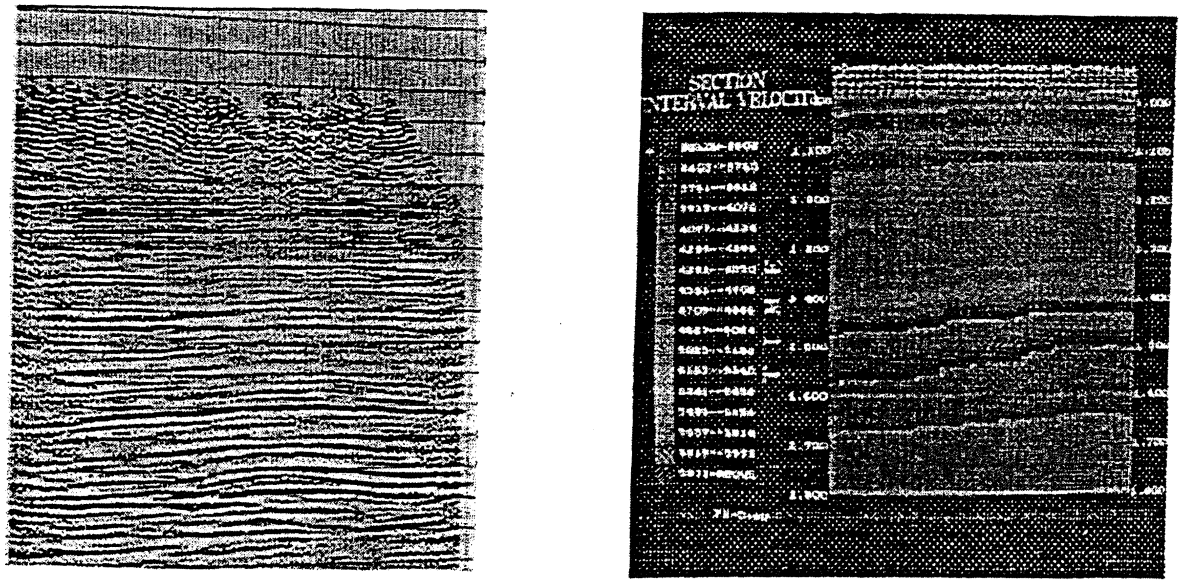
Corollary 5.2 Let X be a connected subset of a n-dimensional grid space and $X_s \subseteq X$ and f: $X_s \rightarrow R$. Assume $\{S_1,...,S_m\}$ is a partition of X_s and $\{R_1,...,R_m\}$ is a partition of X , if

(1) $S_i \subseteq R_i$, $i=1,...,m$ & Boundary(R_i) $\subseteq S_i$, and
(2) For any two points p,q in S_i, each shortest path in R_i between p and q is λ-connectable,

then, there is a λ-interpolation F on X of f, such that $S_\lambda(X, F) = \{R_1,...,R_m\}$.

6. INTELLIGENT DATA FITTING AND VELOCITY RECONSTRUCTION

In this section we describe how to apply intelligent data fitting to 3D seismic data reconstruction. Seismic data has two different forms: waveform seismic data, such as amplitude data, and gray-scale seismic data such as velocity, porosity etc. In this paper, we deal with gray-scale seismic data. The goal of seismic data processing for oil-gas exploration is to find oil-gas traps. Geology interpreters seek to find a region with high oil-gas probability in which to drill. A decision is carefully made based on geologic and geophysical data. The most important geophysical data is seismic data. Oil-gas traps are of two major types. One is the structural type, formed by geometric structure such as a convex region or a fault structure. To recognize these structures one generally uses amplitude waveform data (Figure 6.1 (a)). Another geologic feature is lithologic type, which is affected by differing rock types, e.g. sandstone versus granite. In this case, velocity is the most important factor (Figure 6.1 (b)). For example, one usually tries to find a region with low velocity since that typically indicates high porosity lithology and is much more likely to store oil and/or gas.



(a) (b)

Figure 6.1 (a) a waveform seismic section, and (b) a velocity section

The best way to precisely determine rock velocities is to drill a well. However it is not a realistic way to drill wells everywhere in an area being examined. An effective and more economic method is to drill a few wells in the area, and then use well-log information and seismic data to interpolate determine the velocities for the entire area. Figure 6.2 pictorially represents such a process.

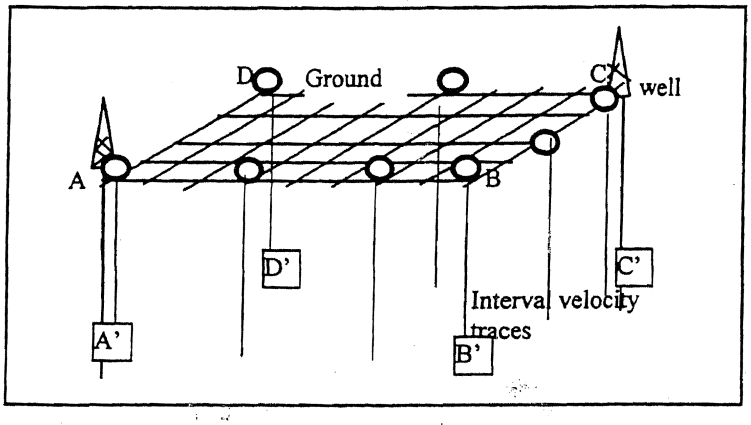


Figure 6.2 An example for petroleum prospecting: two wells, and seismic data on each line.

Note that data may be 2D or 3D.

Although there are many velocity measures for seismic data, in this paper, we use interval velocity data, which indicates the velocity of each small layer in the stratum.

6.1 Fitting a 3-D Interval Velocity Volume

In this subsection, we present a practical method to reconstruct an interval velocity volume from seismic data. In the example, we explain how 3D interval velocity volume data was filled for a region in Sichuan province, China. This interval velocity volume was used to guide drilling in this area .

To get an interval velocity trace on a vertical line is not a simple task. An interpreter need to process seismic data and interact with well-log data many times to complete a trace. In an about 1km x 1km target square area (Figure 6.2), we have a 41 x 20 x 201 3D seismic data, two well-log data for testing use, and 8 interval velocity traces on the edge of 41 x 20 array (see Figure 6.2).

The goal here is to reconstruct a 41 x 20 x 201 interval velocity array (or 41 x 20 interval velocity traces). To do so the data fitting methodolgy just discussed was used to develop the following "Edge First Fitting" algorithm. The following data fitting algorithm is used when most of the guiding points are on the "edge," and the object region is relatively small. The strategy here is to use the data on four edge sections of a volume to fill the entire array or volume. For instance, in Figure 6.2, we use this edge fitting technique on sections [A, A', B',B], [B,B',C',C], [C,C',D',D] and [D,D',A',A]. Commonly edge points are chosen as guiding points in such seismic data applications.

In seismic data processing, each vertical data trace is called a CDP trace. If we assume that there are m CDP traces with known interval velocity in the edge of a region of ground surface, then the distance between each CDP point can vary. In the region in the Sichuan province, we acquired eight such CDP traces, and 201 sample points along a vertical (time) direction.

Step 1: ***Velocity layer classification.*** In this step, one first applies λ-connected segmentation or split-and-merge segmentation (or some other segmentation method) on the m CDP traces. Because of the different distances among CDP traces, the connected algorithm needs to be modified to account for this fact. For example, the adjacency distance between two traces should be greater than 1.

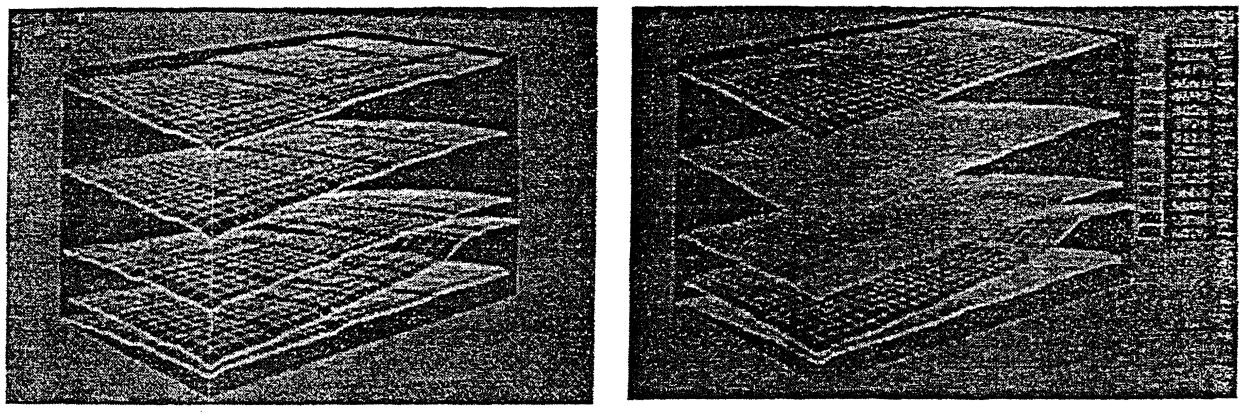
Step 2: ***Boundary curve fitting and interface fitting.*** After separating each layer in the m CDP traces in Step 1, one must extract m points which belong to a single interface in a stratum. The goal of this step is to fit or find every such interface. There are two techniques to reach this goal. First, one can fit the surface directly using the m points. However, for our example, since we had only eight points, the result of such a fitting was not satisfactory. The second method, which was used, was to first fit the boundary of the surface to a rational function using the m points. A rational function gave us a better "looking" answer.

Once the boundary fitting curve was arrived at, one can use a Coons surface fitting or gradually varying surface fitting algorithm [6-7] to obtain the layer interfaces. Figure 6.3 shows the five major layers in our example region. A surface smoothing process can be used to correct minor fitting errors.

Step 3: ***Surface Lofting.*** To give all points the same value in a layer (determined by two interfaces) is the simplest method to reconstruct the volume. However, since a layer is a λ-connected set, i.e. it satisfies some uniformity measure, the difference between the values of two points in the volume may not be small. Also, geologic layers generally have the property of "gradual variation." The use surface lofting technique between two interfaces allows one to arrive at a more accurate reconstruction of the volume. A lofted surface is obtained by interpolation of two interface surfaces. In the application, we arrived at 171 lofted surfaces. Each lofted surface was treated as a small layer.

Step 4: ***Gradually varied fill on lofted surfaces.*** A lofted surface only gives the geometric locations in which each point should have the "same value." Although we knew the values at the intersection of the m CDP traces, the lofted surface were not unique. What was needed was to fit a surface on each lofted surface (See Figure 6.3 (b)). A gradually varied fill technique [6,7] was a good technique because we did not care about the smoothness of the edges only their "continuity." This "continuity" meant discrete "gradual variation." Figure 6.4 shows four gradually varied fitting surfaces.

Step 5: Since some points were not in the lofted surfaces, we simply assigned a value to these points using 1-D linear interpolation. The interval velocity volume in our application is shown in Figure 6.5.



(a) (b)

Figure 6.3 (a) Five major interfaces, (b) velocity on five interfaces

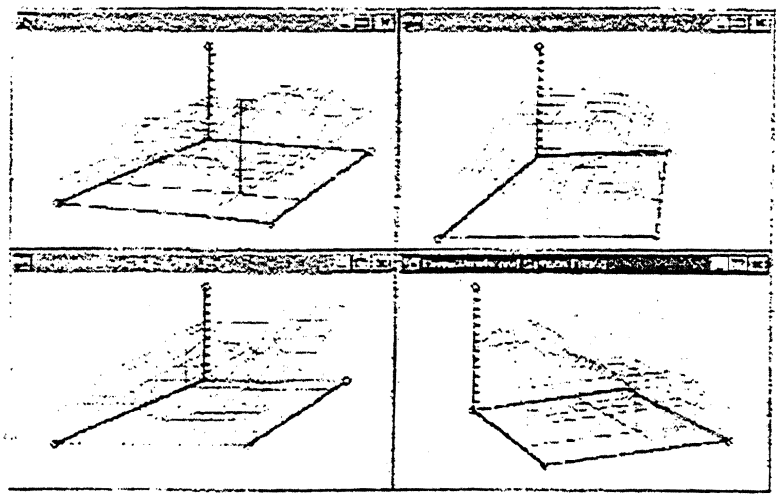


Figure 6.4 Gradually varied fitting surfaces

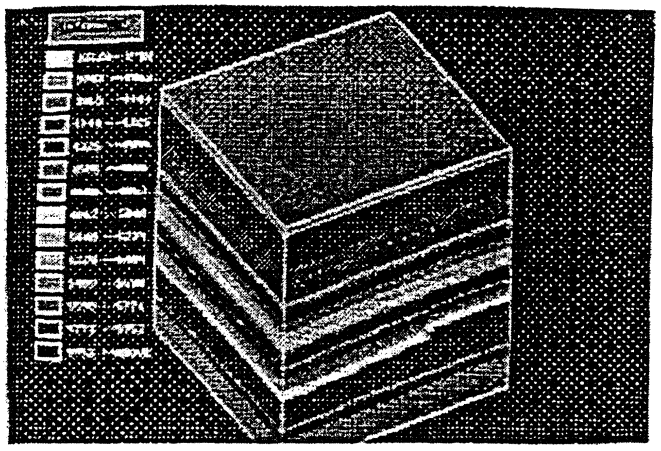


Figure 6.5 The reconstructed velocity volume

7. CONCLUSIONS

In this paper, an intelligent 3D data fitting method was presented, and some of its mathematical properties have been studied. We have used this method to fit interval velocity volumes which are very useful for oil-gas prediction. The method described in this paper has been used successfully in an area in Sichuan province in China.

8. REFERENCES

1. R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.

2. L. Chen, Three-Dimensional Fuzzy Digital Topology and its Application (I-II), *Geophysical Prospecting for Petroleum*, 2 (1985), 2 (1987).

3. L.Chen, The lambda-connected segmentation and the optimal algorithm for split-and-merge segmentation, *Chinese J. Computers* 14:5(1991).

4. L. Chen, H. Cheng, and J. Zhang, Fuzzy Subfiber and its Application to Seismic Lithology Classification, *Information Science: Applications*, Vol. 1, No 2, 1994.

5. A. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

6. L. Chen, The necessary and sufficient condition and the efficient algorithms for gradually varied fill, *Chinese Science Bulletin*, 35:10(1990);

7. L. Chen, Gradually Varied Surface and its Optimal Uniform Approximation, *IS&T/SPIE Symposium on Electronic Imaging*, SPIE Proc. Vol. 2182, 1994

Digital Manifolds: An Intuitive Definition And Some Properties

Li Chen and Jianping Zhang
Department of Computer Science
Utah State University
Logan, UT 84322-4205

1. Introduction

Digital topology deals with the topological properties of digital objects in two or more dimensional discrete arrays. It might provide a theoretical and computational foundation for discrete surface and/or solid design, recognition and representation [1].

Digital curve was first studied by Rosenfeld in early 1970s. A definition of digital curves and the digital Jordan curve theorem were then given by him [1]. The concept of digital surfaces was proposed by Artzy, Frieder, and Herman [2]. They defined a surface as the face of some solid object. Morgenthaler and Rosenfeld later proposed an alternative definition for a digital surface [3]. Khalimsky *et al* tackled the problems of digital topology using pure topological methods [4,5]. In [6], Chen defined the term "digital manifold" as a discretization of a manifold and presented an interpolating algorithm for obtaining gradually varied surfaces on a digital manifold.

In this abstract, we propose a uniform framework for defining and processing digital objects including digital curves, surfaces, solids, and etc. Namely, our framework can handle digital manifolds in a m dimensional space, where m virtually can be any positive integer. Under our framework, the definition of a digital manifold is intuitive as well as simple. The recognition algorithm developed based the definition is tractable. Section 2 reviews some basic digital topology concepts. Section 3 presents the definition and some properties of digital manifolds. The theorems are presented without proofs which can be found in [11].

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

2nd ACM Solid Modeling '93-5/93/Montreal, Canada
© 1993 ACM 0-89791-584-4/93/0005/0459...$1.50

2. Basic Concepts

This section reviews some basic concepts of digital topology. A m dimensional digital space Σ_m is a m dimensional array of integer coordinate points. The points $p=(x_1, x_2, ..., x_m)$ and $q=(y_1, y_2, ..., y_m)$ are two directly adjacent points in Σ_m if

$$d_D(p,q)=\sum_{i=1}^{m} |x_i-y_i| = 1$$

p and q are indirectly adjacent points if

$$d_I(p,q) = MAX_{1<i\le m} |x_i-y_i| = 1$$

In this abstract, only direct adjacency is considered. Let p and q be a pair of (directly) adjacent points of Σ_m, the pair {p,q} is called a line-cell. A surface-cell is a set of 4 points which form a unit square parallel to coordinate planes. A Three dimensional-cell (or 3_D cell) is a unit cube which has 8 points, and so on we can define a k_D-cell.

Two different k_D-cells are k'_D adjacent where $k > k' \ge 0$, if they share a k'_D-cell. A k'_D-connected k_D-cells path is a sequence of k_D-cells $v_0, v_1 \ldots, v_n$, in which v_i and v_{i+1} are k'_D-adjacent for i = 0 to n. Let $\Gamma^{(k)}(S)$ be the set of all k_D-cell of S, where S is a subset of Σ_m where m ≥ k. Two k_D-cells p and q in S are called k'_D-connected if there exists a k'_D-adjacency k_D-cell path $v_0 v_1 \ldots v_n$ such that $v_0 = p$ and $v_n = q$; where $v_0, v_1 \ldots v_n$ are in $\Gamma^{(k)}(S)$. $\Gamma^{(k)}(S)$ is called as k'_D-connected if every pair p, q in $\Gamma^{(k)}(S)$ is k'_D-connected.

3. Digital manifolds

In this section, we propose a mathematical framework of digital manifolds which contain digital curves, digital surfaces, digital solids, and etc. Before the framework is

presented, we first introduce the concept of a parallel-move of a k_D-cell. This concept serves as the basis of our framework.

Definition 3.1 Let A be a subset of Σ_m. A' is called a parallel_move of A if $|A|=|A'|$, $d_D(a,a') \geq 1$ where $a \in A$ and $a' \in A'$, and there exists a biconjunctive mapping f: $A \rightarrow A'$ such that $d_D(a,f(a))=1$ for all $a \in A$.

Given a n_D-cell, the lemma 3.1 and 3.2 can help us obtain any parallel-move of the n_D-cell.

Lemma 3.1 Let $\{p(1),...,p(2^n)\}$ be a n_D-cell in Σ_m; where $n \leq m$ and $p(i) = (p_1(i),...,p_m(i))$ is one point in $\{p(1),...,p(2^n)\}$, $i=1,...2^n$. We have:

1) Among the m components there exist $(m-n)$ components so that the 2^n vectors $p(1)$,.... and $p(2^n)$ have the same values on these $(m-n)$ components.
2) Assume j is one of the (m-n) invariant components. Then, the two n_D-cells
   $\{(p_1(1), ..., p_j(1)+1, ..., p_m(1)), ...,$
   $(p_1(2^n), ...,p_j(2^n)+1, ..., p_m(2^n))\}$ and
   $\{(p_1(1), ..., p_j(1)-1, ..., p_m(1)), ...,$
   $(p_1(2^n), ..., p_j(2^n)-1, ..., p_m(2^n))\}$
   are the two parallel moves of the n_D-cell
   $\{(p_1(1), ..., p_j(1), ..., p_m(1)), ...,$
   $(p_1(2^n), ..., p_j(2^n), ..., p_m(2^n))\}$

Lemma 3.2 Let V be a parallel-move of V', then V' is the parallel-move of V.

Lemma 3.3 Let V be a n_D-cell, then V has exactly $2(m-n)$ parallel-moves.

Lemma 3.4 Let V be a parallel move of V', then $V \cup V'$ is a (n + 1)_D-cell.

Definition 3.2 A subset M of Σ_m is a digital manifold if

0) $M=\cup\{a \mid a \in \Gamma^{(n)}(M)\}$.
1) $\Gamma^{(n)}(M)$ is (n-1)_D-connected.
2) Each element in $\Gamma^{(n-1)}(M)$ has only one or two parallel-moves.
3) $\Gamma^{(n+1)}(M)$ is empty.

Definition 3.3 Let M be a n dimensional digital manifold, the boundary of M, denoted by ∂M, is a set of (n-1)_D-cells which has only one parallel-move in M. M is called n dimensional closed digital manifold if ∂M is empty.

Definition 3.4 Let M be a n dimensional digital manifold, we say M is regular if all of the n_D-cells containing p are $(n-1)_D$- connected for all $p \in M$.

Theorem 3.1 If S is a regular surface in Σ_3, then is regular and the union of several closed digital curves.

Theorem 3.2 The boundary of a regular n dime sional digital manifold is regular and the union of seve $(n-1)_D$ closed digital manifolds.

Theorem 3.3 Given a subset M of Σ_m, there ex $O(m2^n\,|M|)$ time algorithms for all of the decision prc lems in following: Whether M is a n dimensional digi manifold, a closed n dimensional manifold, a regular dimensional digital manifold, and a regular closed n c mensional manifold? Also,there exist $O(m2^n\,|M|)$ time a gorithms to track M's boundary.

References

[1] T.Y.Kong and A.Rosenfeld, Digital topolog Introduction and survey, Computer Vision Graphic Image Process. 48,1989,357-393.

[2] E.Artzy,G.Frieder and G.T.Herman, The theor design, implementation and evaluation of a three dimensional surface detection algorithm, Compute Vision Graphics Image Process. 15 , 1981, 1-24.

[3] D.G.Morgenthaler and A.Rosenfeld, Surfaces in thre dimensional images, Imform. and Control 51, 1981 227-247.

[4] E.Khalimsky, R.D.Kopperman, and P.R.Meye annual. 3,1990, 27-55.

[5] V.A.Kovalevsky, Finite topology as applied to imag analysis, Computer Vision Graphics Image Proces 46, 1989, 141-161.

[6] L.Chen, The necessary and sufficient condition an the efficient algorithms for gradually varied fil Chinese Science Bulletin 35, 1990, 870-87 Germany Zbl. Math. 712-68085(1991).

[7] G.T.Herman and D.Webster, A topological proof of surface tracking algorithm, Computer Vision Graphi Image Process. 23, 1983, 162-177.

[8] L.Chen and J.Zhang, The algorithms for digi surface decision, orientable and tracking problem submitted for publication, 1992.

[9] L.Chen and J. Zhang, Some basic problems of digi surfaces, in preparation, 1993.

[10] D.Gordon and J.K.Udupa, Fast surface tracking three dimensional binary images, Computer Visi Graphics Image Process. 45, 1989, 196-214.

[11] L. Chen and J. Zhang, Digital manifolds: an intuiti definition and some properties, in preparation, 1993.

A Digital Surface Definition and Fast Algorithms for Surface Decision and Tracking

Li Chen and Jianping Zhang

Department of Computer Science
Utah State University
Logan, UT 84322-4205

ABSTRACT

This paper presents a mathematical digital surface definition. This definition is able to deal with boundary points as well as "inner" points. Namely, it is able to distinguish inner points from boundary points. The definition is intuitive and provides a basis for designing fast algorithms for surface decision, boundary search, and surface tracking problems. This definition is equivalent to Morgenthaler and Rosenfeld's definition on simple closed digital surfaces and can be extended to the problem of defining a digital manifold.

1. INTRODUCTION

It has been and continues to be an interesting problem to mathematically define a digital surface in digital geometry and image processing. A clear mathematical digital surface definition could provide a theoretical foundation for some image processing operations such as boundary search and surface tracking. In the past, there were many attempts at defining a digital surface. Among these attempts, the surface definitions given by Artzy et. al[1] and Morgenthaler and Rosenfeld[4] were two of the most notable ones. Artzy and his colleagues[1] defined a digital surface as the face of solid objects, while Morgenthaler and Rosenfeld[4] gave a purely discrete definition of a digital surface. On the basis of the two definitions, many important results were obtained in digital topology. These results in digital topology provided a foundation for many image processing operations[5].

However, the problem of defining a digital surface has not been completely solved. There exists some inconsistency among different digital surface definitions. For example, according to the definition given Artzy et al., the face of the solid object shown in Figure 1 is a surface, but it is not surface under the definition given by Morgenthaler and Rosenfeld. Another example is the Mobuis band shown in Figure 2. The mobuis band is not a surface under both Artzy et al's and Morgenthaler and Rosenfeld's definitions. However, mathematically Mobuis band is an important surface. The reason for this inconsistency is that 1) Mobuis band is not a face of any solid object so Artzy et al's definition fails to recognize it as a surface; 2) Mobuis band is not closed and the definition given by Morgenthaler and Rosenfeld handles only the "inner" points of a surface and fails to recognize the boundary of a surface.

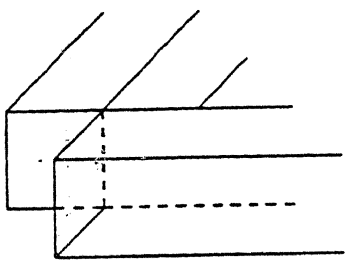


Figure 1. A 3-dimensional solid object whose face is not a digital surface

0-8194-1325-9/93/$6.00

The work reported in this paper can be viewed as a continuation or an improvement of Rosenfeld's definition. We developed a novel mathematical definition of a digital surface in a 3-dimensional space. This definition is able to deal with boundary points as well as "inner" points. Namely, it is able to distinguish inner points from boundary points. The definition is intuitive and provides a basis for designing efficient surface decision, boundary search, and surface tracking algorithms. This definition is equivalent to Morgenthaler and Rosenfeld's definition on simple closed digital surfaces[16] and can be extended to the problem of defining a digital manifold[14].

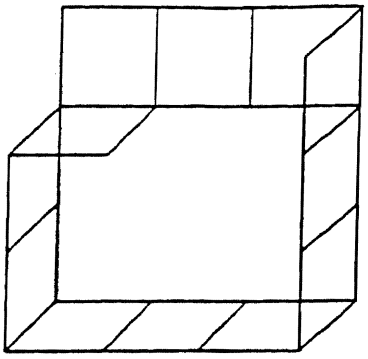


Figure 2. Mobuis band

In section 2, we review some basic concepts in digital surface and digital topology[5]. The digital surface definition developed by us is presented in section 3. Section 4 describes algorithms for surface decision, boundary search and surface tracking problems. These algorithms were developed on the basis of our surface definition. Section 5 includes conclusions.

2. BASIC CONCEPTS

This section briefly reviews some basic concepts in digital surface. Digital surface was first studied by Herman and his colleagues[1,2]. However, the first discrete definition of a closed digital surface was given by Morgenthaler and Rosenfeld[4]. Based on this definition, Rosenfeld and his colleagues derived many important theoretical results, e.g., discrete Jordan theorem. An alternative approach was introduced by Khalimsky and Kovalevsky[6,7]. In their approach, digital images are represented by topological spaces. Recently, Rosenfeld, Kong and Wu[10] discussed and proved some basic properties of digital surfaces. The results reported in this paper were based on these previous work. In this paper, we adopt the terminology used in Kong and Rosenfeld[5].

We use Σ_m for the set of all points in a m-dimensional lattice space. Thus, Σ_3 is a set of all points in a 3-dimensional space. Each point in Σ_3 is represented as 3 integer coordinates. Two points $p = (x_1\ x_2\ x_3)$ and $q = (y_1\ y_2\ y_3)$ in Σ_3 are said to be directly adjacent if

$$D_d(p,q) = \sum_{i=1}^{3} |x_i - y_i| = 1$$

Two points $p = (x_1\ x_2\ x_3)$ and $q = (y_1\ y_2\ y_3)$ in Σ_3 are said to be indirectly adjacent if

$$D_i(p,q) = MAX_{1 \le i \le 3}\ |x_i - y_i| = 1$$

In a three-dimensional space Σ_3, a point has six directly adjacent points and 26 indirectly adjacent points. Therefore, two directly adjacent points in Σ_3 are called 6-adjacent and two indirectly adjacent points are called 26-adjacent. We only consider directly adjacency in this paper. In the rest of the paper, the term adjacency is used for direct adjacency, unless otherwise stated.

A point in Σ_3 is called a point_unit-cell. Two adjacent points p, q in Σ_3 is called a line_unit-cell. A line_unit-cell composed of two points p and q is written as (p, q). A surface_unit-cell is a set of 4 points that forms a unit square parallel to one coordinate plane. A 3-dimensional_unit-cell is a set of 8 points that forms a unit cube parallel to all three coordinate planes. Figure 3 (a)(b)(c) show a line_unit-cell, a surface_unit-cell and a 3-dimensional_unit-cell, respectively.

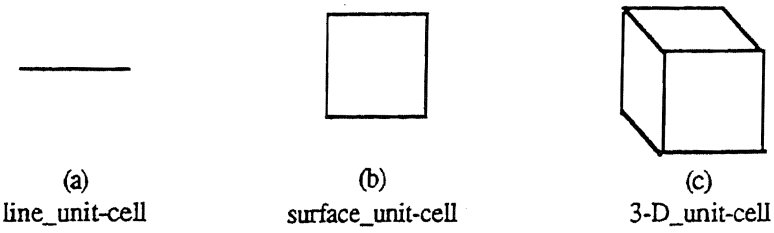


Figure 3. Unit-cells

Two points p and q are connected if there exists a simple path (a list of points) $p_0, p_1, ..., p_n$, where $p_0 = p$, $p_n = q$, and p_i and p_{i+1} are adjacent for $i = 0, 1, ..., n-1$. A subset S of Σ_3 is said to be point connected if all points in S are connected.

Two unit-cells are point_adjacent if they share a point. For example, line_unit-cells l1 and l2 are point_adjacent in Figure 4(a), and surface_unit-cells s1 and s2 in Figure 4(b) are point_adjacent. Two surface_unit-cells are line_adjacent if they share a line_unit-cell. For example, surface_unit-cells s1 and s2 in Figure 4(c) are line_adjacent.

Two line_unit-cells l_j and l_k are point_connected if there exists a sequence of line_unit-cells $l_0, l_1, ..., l_n$ such that $l_0 = l_j$, $l_n = l_k$, and l_i and l_{i+1} are point_adjacent for $i = 0, 1, ..., n-1$. For example, line_unit-cells l1 and l4 in Figure 4(a) are point_connected. Two surface_unit-cells s_j and s_k are line_connected if there exists a sequence of surface_unit-cells $s_0, s_1, ..., s_n$ such that $s_0 = s_j$, $s_n = s_k$, and s_i and s_{i+1} are line_adjacent for $i = 0, 1, ..., n-1$. For example, surface_unit-cells s1 and s3 in Figure 4(c) are line_connected.

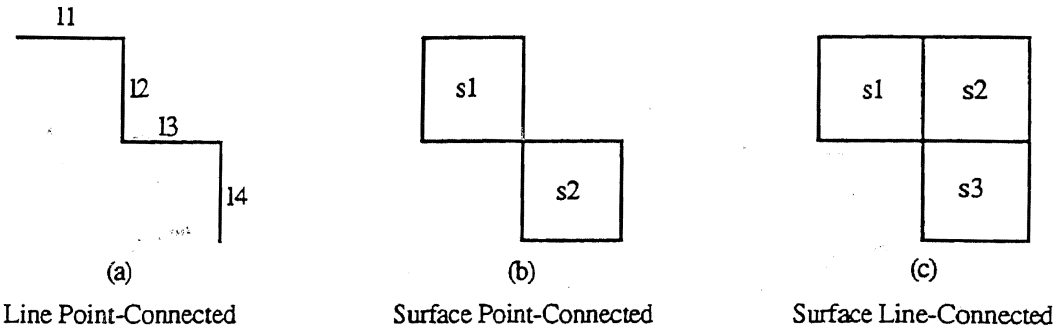


Figure 4. adjacency and connectedness

3. A DIGITAL SURFACE DEFINITION

This section introduces a novel digital surface definition in a three dimensional digital space. Based on this definition, boundary points can be easily recognized. Fast algorithms for surface decision, boundary search and surface tracking can be developed on the basis of this definition. This definition is equivalent to Morgenthaler and

Rosenfeld's definition on simple closed digital surfaces[16]. All lemmas, theorems, and algorithms are presented with no proof in this paper. Proofs will be given in Chen and Zhang [15].

Definition 3.1 Let (p, p') be a line_unit-cell. A line_unit-cell (q, q') in Σ_3 is called a parallel_move of the line_unit-cell (p, p') if p and p' are adjacent to q and q' respectively; but neither p and q' nor p' and q are adjacent. A surface_unit-cell (p_1, p_2, p_3, p_4) is a parallel_move of another surface_unit-cell (q_1, q_2, q_3, q_4) if p_i is adjacent to q_i (i = 1, 2, 3 ,4) and p_i is not adjacent to q_j when $i \neq j$ (i, j = 1, 2, 3, 4).

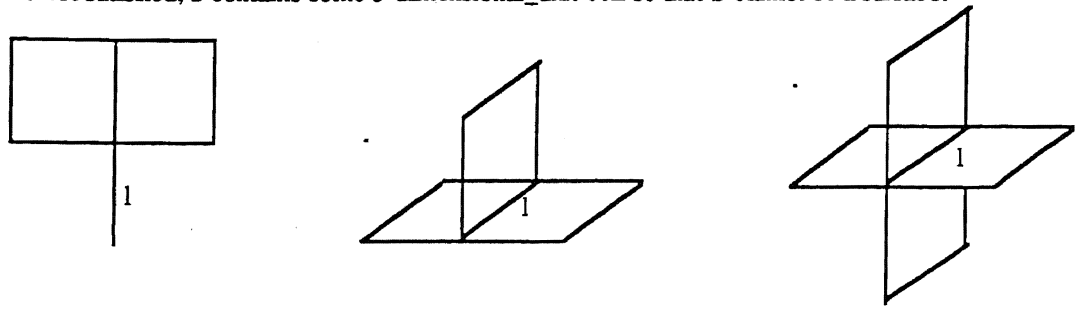
Lemma 3.1 Each line_unit-cell in Σ_3 has 4 parallel-moves in Σ_3.

Lemma 3.2 In Σ_3, a line_unit-cell and one of its four parallel-moves form a surface_unit-cell.

Definition 3.2 A point connected set S in Σ_3 is called a digital surface if

1) all line_unit-cells of S have either one or two parallel_moves.
2) Any two surface_unit-cells of S are line_connected.
3) S does not contain any 3-dimensional_unit-cell.

All three conditions in Definition 3.2 are necessary conditions. If the first condition is not satisfied, then there must exist one line_unit-cell in S which has 0, or 3 or 4 parallel_moves. In these cases S cannot be a surface (See Figure 5). If the second condition is violated, there exist two surface_unit-cells in S are not line_connected. Because S is connected, these two surface_unit-cells must be point_connected, thus S is not surface (see Figure 6). If the last condition is not satisfied, S contains some 3-dimensional_unit-cell so that S cannot be a surface.



(a) A line_cell with 0 parallel_move

(b) A line_cell with 3 parallel_move

(c) A line_cell with 4 parallel_move

Figure 5 A line_unit-cell with 0, or 3, or 4 parallel_moves

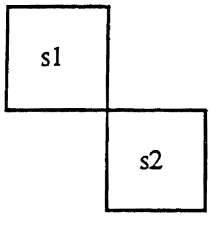


Figure 6 Two point_connected surface_unit-cells

Lemma 3.3 p is an inner point in surface S if every line_unit-cell containing p has exactly 2 parallel_moves in S (See Figure 7(a)). p is a boundary point of S if there exists a line_unit-cell containing p has only one parallel_move (See Figure 7(b)). The set of all boundary points is denoted by ∂S.

Lemma 3.4 The subset S of Σ_3 is a closed surface if all line_unit-cells of S have exactly 2 parallel_moves in S.



(a) Inner point p

(b) Boundary point p

Figure 7 Inner and boundary points

4. ALGORITHMS FOR SURFACE DECISION, BOUNDARY SEARCH, AND SURFACE TRACKING

This section presents algorithms for surface decision, boundary search and surface tracking. These algorithms are developed on the basis of the surface introduced in section 3. The surfaces decision problem was raised by Reed and Rosenfeld [13]. It can be stated as follows:

Given: a subset S of Σ_3
Determine: if S is a surface

The boundary search problem can be defined as below:

Given: a digital surface S in Σ_3
Determine: the boundary of S

The surface tracking problem has been tackled by many researchers including Herman and Webster[2], Gordon and Udupa[8], and Kong and Udupa[9]. The surface tracking problem can be stated as follows:

Given: a solid object,
Determine: its boundary surface

4.1 Data Structures

We use an adjacency-list to represent a subset of Σ_3[11,12]. Let $S = \{p_1, ..., p_n\}$ be a subset of Σ_3. The adjacency-list of S is

$p_1 \rightarrow p_{11},...,p_{1m_1}$
$p_2 \rightarrow p_{21},...,p_{2m_2}$
.....
$p_n \rightarrow p_{n1},...,p_{nm_n}$

$p_{i1},...,p_{im_i}$ (i = 1, ..., n) are all directly adjacent points of p_i in S and are denoted by $ALp(p_i)$. m_i is not larger than 6, because each point has at most 6 directly adjacent points.

$$ALp(p_i) = \{p_{i1},...,p_{im_i}\},\ i = 1, ..., n$$

4.2 Surface Decision Algorithm

The surface decision algorithm consists of three algorithms, each of which checks one of the three conditions in Definition 3.2. Specifically, the three algorithms will respectively perform the following three tasks:

Algorithm 1: Find out all parallel-moves of every line_unit-cell (p_i, p_{ij}) in S and decide how many parallel_moves (p_i, p_{ij}) has. As a side effect, this algorithm constructs the adjacency-list of line_unit-cells that is used in algorithm 3 to decide if a surface_unit-cell has any parallel_move. An adjacency-list of line_unit-cells stores for each line_unit-cell all its parallel_moves.

Algorithm 2: Decide if all surface_unit-cells are line-connected.

Algorithm 3: Find out all parallel_moves of each surface_unit-cell in S. Each parallel_move of a surface_unit-cell and the surface_unit-cell itself form a 3-dimensional_unit-cell. Therefore, if there exists any surface_unit-cell in S with a parallel_move, then S contains a 3-dimensional_unit-cell.

Before describing these three algorithms, we first introduce two lemmas that are useful in understanding and analysis of these algorithms.

Assume that $S = \{s_1, s_2, ..., s_{|S|}\}$ is a subset of Σ_3. Let $\Gamma^{(0)}(S)$ be the set of all points in S, $\Gamma^{(1)}(S)$ be the set of all line_unit-cells in S, and $\Gamma^{(2)}(S)$ be the set of surface_unit-cells in S.

Lemma 4.1 Let (p, q) be a line_unit-cell in Σ_3 where $p = (x_1\ x_2\ x_3)$ and $q = (y_1\ y_2\ y_3)$. We have:

1) Two of the three coordinates of q are the same as those of p. Without lose of generality, we assume the first two coordinates of p and q are the same. Namely, $x_1 = y_1$ and $x_2 = y_2$.
2) The following are the four parallel_moves of (p, q):
   $((x_1+1\ x_2\ x_3), (y_1+1\ y_2\ y_3))$
   $((x_1-1\ x_2\ x_3), (y_1-1\ y_2\ y_3))$
   $((x_1\ x_2+1\ x_3), (y_1\ y_2+1\ y_3))$
   $((x_1\ x_2-1\ x_3), (y_1\ y_2-1\ y_3))$

Lemma 4.2 Let P be a parallel_move of P', then P' is a parallel_move of P.

Lemma 4.3 $|\Gamma^{(1)}(S)| = \frac{1}{2}\sum_{i=1}^{|S|} m_i \le 3|S|$, where $S = \{p_1, ..., p_{|S|}\}$ and m_i is the number of adjacent points of the point p_i.

Lemma 4.4 If p, q are two directly adjacent points in Σ_3, then there does not exist a point $r \in \Sigma_3$ so that r is directly adjacent to both p and q.

Algorithm 1: Find out all parallel_moves for all line_unit-cells in S and check if any line_unit-cells has zero or more than two parallel_moves. Algorithm 1 is divided into two phases: the first phase is to find out all parallel_moves for each line_unit-cell in S; the second phase is to check if any line-unit-cell has zero or more than two parallel_moves. The first phase constructs a list that stores all parallel_moves for each line_unit-cell, we call this list as Line Parallel_Move list: LPM. The set of all parallel_moves of a line_unit-cell l is denoted as LPM(l).

Step 1: For each point p_i in S, repeat step 2 to step 6 .
Step 2: For each adjacent point p_{ij} of p_i in $AL_p(p_i)$, repeat step 3 to step 6.
Step 3: Construct a line_unit-cell (p_i, p_{ij})
Step 4: For each adjacent point p_{ijk} (except p_i) of p_{ij} in $AL_p(p_{ij})$, repeat step 5 to step 6.
Step 5: For each adjacent point p_{ijkl} (except p_{ij}) of p_{ijk} in $AL_p(p_{ijk})$, repeat step 6.
Step 6: if p_{ijkl} is in $AL_p(p_i)$, then (p_{ijk}, p_{ijkl}) is a parallel_move of (p_i, p_{ij}). Add the parallel_move (p_{ijk}, p_{ijkl}) to $LPM((p_i, p_{ij}))$. $LPM((p_i, p_{ij}))$ stores all parallel_moves of the line_unit-cell (p_i, p_{ij}).
Step 7: For each line_unit-cell (p_i, p_{ij}), determine the size of $LPM((p_i, p_{ij}))$ (the number of parallel_moves of (p_i, p_{ij}))

From Lemma 4.3 and $AL_p(p_i)$ has at most 6 elements, we could derive Lemma 4.5. Steps 2, 4, and 5 repeats at most six times.

Lemma 4.5 Given the adjacency list of S: $AL_p(pi)$ for all i, the time complexity of Algorithm 1 is $O(|\Gamma^{(1)}(S)|)$, simply O(|S|).

In Algorithm 1, the first 6 steps construct the Line Parallel Move list (LPM) for each line_unit-cell in S. Step 7 and step 8 determine the number of parallel_moves of each line_unit-cell in S using LPM constructed in the first 6 steps.

Algorithm 2 decides if all surface_unit-cells in S are line_connected. This algorithm uses the Line Parallel_Move (LPM) list constructed by Algorithm 1. A surface_unit-cell contains 4 points p_1, p_2, p_3, and p_4. Four line_unit-cells, (p_1, p_2), (p_2, p_3), (p_3, p_4), and (p_4, p_1) are involved in a surface_unit-cell. Actually, a line_unit-cell (p_1, p_2) and its parallel_move (p_3, p_4) determine a surface_unit-cell. Two surface_unit-cells are line_adjacent if and only if they share a line_unit-cell. The two adjacent surface_unit-cells are determined by the shared line_unit-cell and two of its parallel_moves respectively. Thus, we can traverse from one surface_unit-cell to an adjacent surface_unit-cell through the shared line_unit-cell. Let us denote a surface_unit-cell as a pair of line_unit-cells (l_1, l_2) where l_1 is a parallel_move of l_2. That is, l_2 is a member of LPM(l_1) and l_1 is a member of LPM(l_2). The Algorithm 2 tries to traverse all surface_unit-cells in S using the Line Parallel_Move List. If it fails to pass through all surface_unit-cells, then there exists some surface_unit-cell that is not line_connected to some other surface_unit-cell.

Algorithm 2: Determine if all surface_unit-cells in S are line_connected.

Step 1: Let the queue Q be empty. Choose a line_unit-cell l_i in S and mark it.
Step 2: For each unmarked line_unit-cell l_{ij} of LPM(l_i), form a surface_unit-cell s_k using l_{ij} and l_i. s_k involves 4 line_unit-cells, l_i, l_{ij}, l_k, l_l.
Step 3: Add l_{ij}, l_k, and l_l into Q.
Step 4: Remove a line_unit-cell from Q and mark it, then repeat the step 2 and step 3 until Q is empty.
Step 5: If all line_unit-cells in S are marked, then all surface_unit-cells are line_connected.

Because all line_unit-cells are traversed at most twice, so Lemma 4.6 holds.

Lemma 4.6 The time complexity of Algorithm 2 is $O(|\Gamma^{(1)}(S)|)$, simply O(|S|).

Algorithm 3 is to determine if there exists some 3-dimensional_unit-cell in S. S contains a 3-dimensional_unit-cell if and only if S contains some surface_unit-cell that has some parallel_move in S. Algorithm 3 first finds parallel_moves for all surface_unit-cells and then checks if any surface_unit-cell has some parallel_move. Algorithm 3 is the exactly same as Algorithm 1 except that the adjacency list used in Algorithm 3 is the Line Parallel_Move list.

Algorithm 3: Find out all parallel_moves for all surface_unit-cells in S and check if any surface_unit-cell has one or more parallel_moves.

Step 1: For each line_unit-cell l_i in S, repeat step 2 to step 6 .
Step 2: For each parallel_move l_{ij} of l_i in LPM(l_i), repeat step 3 to step 6.
Step 3: Construct a surface_unit-cell (l_i, l_{ij})
Step 4: For each parallel_move l_{ijk} (except l_i) of l_{ij} in LPM(l_{ij}), repeat step 5 to step 6.
Step 5: For each parallel_move l_{ijkl} (except l_{ij}) of l_{ijk} in LPM(l_{ijk}), repeat step 6.
Step 6: if l_{ijkl} is in LMP(l_i), then (l_{ijk}, l_{ijkl}) is a parallel_move of (l_i, l_{ij}). Add the parallel_move (l_{ijk}, l_{ijkl}) to SPM((l_i, l_{ij})). SPM((l_i, l_{ij})) stores all parallel_moves of the surface_unit-cell (l_i, l_{ij}). SPM denotes Surface Parallel_Move list.

Step 7: For each surface_unit-cell (l_i, l_{ij}), determine the size of SPM((l_i, l_{ij})) (the number of parallel_moves of (l_i, l_{ij}))

Step 8: If there exists some SPM((l_i, l_{ij})) with one or more elements, then S contains some 3-dimensional_unit-cell.

Lemma 4.7 $|\Gamma^{(2)}(S)| \leq 2|S|$.

Lemma 4.8 Given Line Parallel_Move lists, the time complexity of Algorithm 3 is $|\Gamma^{(2)}(S)| = O(|S|)$.

Given point adjacency lists ALp(p_i), the algorithm for deciding whether a subset S of Σ_3 is a digital surface is given as below.

Surface Decision Algorithm:

Step 1: Apply Algorithm 1 to determine the number of parallel_moves for all line_unit-cells in S.

Step 2: If there exists some line_unit-cell with zero or more than two parallel_moves, then S is not a surface. Goto step 8.

Step 3: Apply Algorithm 2 to determine if S is line_connected.

Step 4: If S is not line_connected, then S is not a surface. Goto step 8.

Step 5: Apply Algorithm 3 to determine if S contains any 3-dimensional_unit-cell.

Step 6: If S contains some 3-dimensional_unit-cell, then S is not a surface. Goto step 8.

Step 7: Report that S is a digital surface.

Step 8: Report that S is not a digital surface.

Theorem 4.1 Given point adjacency lists ALp(p_i), the time complexity of the Surface Decision Algorithm that determines if a subset S of Σ_3 is a digital surface is O(|S|).

4.3 Boundary Search Algorithm

The Boundary Search Algorithm determines the boundary of a surface S. According to lemma 3.4, if all line_unit-cells of a digital surface have exactly two parallel_moves, then the surface has no boundary. According to lemma 3.3, all points of a digital surface involved in the line_unit-cells that have only one parallel_move are the boundary points of the surface. The Boundary Search Algorithm uses Algorithm 1 of the Surface Decision Algorithm to determine the number of parallel_moves for all line_unit-cells in S.

Given point adjacency lists ALp(p_i), the algorithm for deciding all boundary points is given as below.

Boundary Search Algorithm

Step 1: Let the set B be empty. Apply Algorithm 1 of Surface Decision Algorithm to determine the number of parallel_moves for all line_unit-cells in S.

Step 2: For every line_unit-cell in S, if it has only one parallel_move, add the two points involved in the line_unit-cell into the set B if they are not in B.

Step 3: If B is not empty, output B that contains all boundary points of the surface S. Otherwise, report that S has no boundary and is a closed surface.

Theorem 4.2 Given point adjacency lists ALp(p_i), the time complexity of the Boundary Search Algorithm is O(|S|).

4.4 Surface Tracking Algorithm

Given a solid digital object, the Surface Tracking Algorithm determines the boundary surface of the solid object. Here, the boundary surface of a solid object is not the surface defined by the definition 3.2 in this paper. Before presenting the Surface Tracking Algorithm, we first introduce a solid object in Σ_3. A solid object in Σ_3 is a point-connected subset of Σ_3 and consists of a number of 3-dimenssional_unit-cells. A surface_unit-cell (l_1, l_2) and

its parallel_move (l_3, l_4) determine a 3-dimensional_unit-cell. For a formal definition of a solid object (3-complex), see [2].

Lemma 4.9 Let M be a solid object of Σ_3. The boundary surface of M: ∂M is all surface_unit-cells of M which have only one parallel_move.

Given point adjacency lists $ALp(p_i)$, the algorithm for determining the boundary surface of a solid object M is given as below.

Surface Tracking Algorithm

Step 1: Apply Algorithm 1 of the Surface Decision Algorithm to construct Line Parallel_Move lists $LPM(l_i)$ for all line_unit-cells in M.

Step 2: Apply Algorithm 3 of the Surface Decision Algorithm to determine the number of parallel_moves of each surface_unit-cell in M.

Step 3: Let the set B empty.

Step 4: For every surface_unit-cell in M, if it has only one parallel_move, add the four points involved in the surface_unit-cell into the set B if they are not in B.

Step 5: Output B that contains all points of the boundary surface of M.

Theorem 4.3 The time complexity of the Surface Tracking Algorithm is O|M|.

5. CONCLUSIONS

A mathematical digital surface definition is given in this paper. This definition is easy to understand and very intuitive. Simple algorithms for surface decision, boundary search, and surface tracking were developed on the basis of this definition. The time complexities of these three algorithms are O(|S|), where |S| is the size of the subset S of Σ_3, when S is given as an adjacency list.

6. REFERENCES

1. E. Artzy, G. Frieder and G.T. Herman, "The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm," Computer Vision Graphics Image Process, 15,1981, 1-24.

2. G.T. Herman and D. Webster, "A topological proof of a surface tracking algorithm," Computer Vision Graphics Image Process, 23, 1983, 162-177.

3. A. Rosenfeld, "Three-dimensional digital topology," Information and Control, 50, 1981, 119-127.

4. D.G. Morgenthaler and A. Rosenfeld, "Surfaces in three -dimensional images," Information and Control, 51, 1981, 227-247.

5. T.Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," Computer Vision Graphics Image Process, 48,1989,357-393.

6. E. Khalimsky, R.D. Kopperman, and P.R.Meyer, "Boundaries in digital planes," Journal of Applied Math.Stochastic annal. 3,1990, 27-55.

7. V.A. Kovalevsky, "Finite topology as applied to image analysis," Computer Vision Graphics Image Process, 46, 1989, 141-161.

8. D. Gordon and J.K. Udupa, "Fast surface tracking in three -dimensional binary images," Computer Vision Graphics Image Process, 45, 1989, 196-214.

9. T.Y. Kong and J.K. Udupa, "A justification of a fast surface tracking algorithm," CVGIP: Graphical Models and Image Processing, 54, 1992, 162-170.

10. A. Rosenfeld, T.Y. Kong, and A.Y. Wu, "Digital surface," CVGIP: Graphical Models and Image Processing 53, 1991, 305-312.

11. T. Pavlidis, Algorithms for Graphics and Image Processing, Computer Science, Rockville, MD. 1982.

12. A. Aho, J.E. Hopcroft, and J.D. Ullman, The Design and Anlysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

13. G.Read and A.Rosenfeld, "Recognition of surfaces in three-dimensional digital images," Information and Control 53, 1982, 108-120.

14. L. Chen, and J. Zhang, "Digital Manifolds: An Intuitive Definition And Some Properties," Proceedings of the Second ACM/SIGGRAPH Symposium on Solid Modeling and Applications, 1993.

15. L. Chen and J. Zhang, "A Digital Surface Definition and Surface Decision and Tracking Algorithms," in preparation.

16. L. Chen and J. Zhang, "Some Concepts And Problems on Digital Surfaces", in preparation.

Classification of Simple Surface Points and A Global Theorem for Simple Closed Surfaces in Three Dimensional Digital Spaces

Li Chen Jianping Zhang
Department of Computer Science
Utah State University
Logan, UT 84322-4205

ABSTRACT

In this paper, we present two theorems: Classification theorem and corner point theorem for closed digital surfaces. The classification theorem deals with the categorization of simple surface points and states that there are exactly six different types of simple surface points. On the basis of the classification theorem and Euler formula on planar graph, we have proved the corner point theorem: Any simple closed surface has at least eight corner points, where a corner point of a closed surface is a point in the surface which has exactly three adjacent points in the closed surface. Another result reported in this paper is that any simple closed surface has at least fourteen points.

1. INTRODUCTION

Digital geometry and topology study the geometric and topological properties of digital images and objects. It provides a mathematical foundation for important image processing and computer vision operators such as connected component labeling, border following and contour filling[1]. The study of digital curves and surfaces is one of the most important aspects in digital geometry and topology. In addition, the topological and geometric properties of curves and surfaces in digital spaces can be applied to design fast algorithms for image processing and computer vision[2,3,4].

In the past, some important global results have been obtained in digital topology. Rosenfeld proved an important theorem: Any simple closed curve must have five points in a two dimensional digital plane, where a curve is a directly connected (or 4-connected) path[5]. Kong, Khalimsky and Kopperman gave a topological proof for this theorem[6]. Some improvements of the Rosenfeld's theorem have been achieved. For example, any simple closed curve must have eight points in the sense of 4-connectedness given by Abelson and diSessa. On the other hand, Rosenfeld, Morgenthaler and Reed showed that the Jordan curve theorem and Jordan surface theorem[5,7,8]. Recently, Herman showed several results on multidimensional Jordan surfaces[9].

This paper presents some new results on simple closed surface defined by Morgenthaler and Rosenfeld[8]. In order to discuss the properties of a simple closed surface, we prove a theorem for classification of simple surface points in Section 3 . This theorem states that there are exactly six types of simple surface points in simple closed surfaces. We call this theorem as classification theorem.

Based on classification theorem and Euler's Formula on planar graph[10], we proved a global theorem: Any simple closed surface at least has eight corner simple surface points. Its proof is given in Section 4. In addition, we show that any simple closed surface at least has 14 points in section 4.

0-8194-1325-9/93/$6.00

2. BASIC CONCEPTS AND PROPERTIES

In this section, we review some basic concepts of digital spaces and digital surfaces, and two discrete definitions of digital surfaces given by Morgenthaler and Rosenfeld[7], and Chen and Zhang respectively[4,11].

A three dimensional digital space Σ_3 is a three dimensional lattice points' array, where each point has integer coordinates. Two points $P = (i, j, k)$ and $P' = (i', j', k')$ are 6-adjacent if

$$|i - i'| + |j - j'| + |k - k'| = 1.$$

They are 26-adjacent if

$$max\{|i - i'|, |j - j'|, |k - k'|\} = 1.$$

Let S be a subset of Σ_3. The 6- and 26-connectedness can be defined by these two kinds of adjacencies respectively. Two points p and q are 6- (or 26-) connected if there exists a path $p_0, p_1, ..., p_n$, where $p_0 = p$, $p_n = q$, p_i and p_{i+1} are 6- (or 26-) adjacent.

Morgenthaler and Rosenfeld gave a definition of simple surface points; we now introduce their definition as follows[8]:

Let S be a subset of Σ_3 and p be a point in S. Let $N_{27}(p)$ be the set of p's 26 26-adjacent points. Use the 6-connectivity and 26-connectivity to S and $\bar{S}$ respectively. p is called as a simple surface point of S if

1) $S \cap N_{27}(p)$ has exactly one component adjacent to p (in the S sense), denote this component A_p.

2) $\bar{S} \cap N_{27}(p)$ has exactly two components,C_1 and C_2, adjacent to p (in the $\bar{S}$ sense).

3) If $q \in S$ and q is adjacent to p (in the S sense), then q is adjacent (in the $\bar{S}$ sense) to both C_1 and C_2.

S is called a simple closed surface when each point of S is a simple surface point. A surface satisfying the definition is called Morgenthaler-Rosenfeld surface in this paper.

Because Morgenthaler-Rosenfeld surface cannot handle the surfaces with boundaries, the authors gave another definition of digital surfaces in reference 4, 11. We now describe this definition below.

Definition 2.1 A line_cell $(q, q')'$ is called a parallel-move of (p, p') if p and p' are adjacent to q and q' respectively; but neither p and q' nor p' and q are adjacent.

Definition 2.2 Two surface_cells are line_adjacent if they share a line_cell. Two surface_cells are line_connected if there is a path of surface_cells connected with these two surface-cells, and each adjacent pair in the path are line_adjacent.

Definition 2.3 A connected set $S \subset \Sigma_3$ is a digital surface if

(1) Each line_cell of S has one and two parallel_moves in S.

(2) Any two surface_cells of S are line_connected.

(3) S does not contain any 3_dimensional unit cell.

From now, a "surface" or "digital surface" means that it satisfies Definition 2.3 in this paper. p is an inner point in surface S if every line_cell containing p has exactly 2 parallel-moves in S. p is a boundary point of S if there exists a line_cell containing p which has only one parallel_move. The set of all boundary points is denoted by ∂S.

Definition 2.4 Let p be a point of a surface S, p is a regular point if all surface_cells including p in S are line_connected in $N(27,p) = N_{27}(p) \cup p$. If p is both an inner and a regular point, then p is called a regular inner surface point.

Theorem 2.1[12] A Morgenthaler-Rosenfeld simple surface point is a regular inner surface point. That is, a regular closed surface is a Morgenthaler-Rosenfeld surface.

This theorem will be used in proving classification theorem in this paper.

3. GEOMETRIC EQUIVALENCE AND CLASSIFICATION OF SIMPLE SURFACE POINTS

In order to explore the structure of simple closed surfaces, we should know the number of different types of simple surface points. Classification theorem presented in this section states that there are exactly six different types of simple surface points.

Definition 3.1 Let S and R be two subsets in $N(27,p)$ and p be a point in both S and R. S and R are geometric equivalent if and only if there is an one to one mapping $f : S \rightarrow R$ which satisfies:

(1) $f(p) = p$,

(2) $d(x,y) = d(f(x), f(y))$, d is the distance for 6-connectedness.

(3) $D(x,y) = D(f(x), f(y))$, D is the distance for 26-connectedness.
Where x and y in S and the distance means the length of the shortest path.

It is easy to see that the conditions (1) and (2) are necessary in Definition 3.1 for this equivalence. It is not obvious that the condition (3) is necessary. However, without the condition (3), S and R in Figure 3.1 are equivalent.

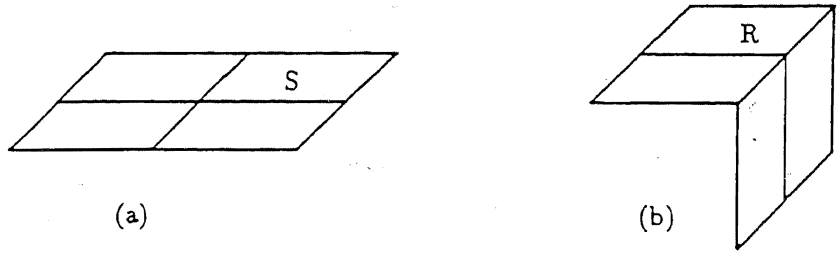


Figure 3.1 Two simple surface points

Lemma 3.1 The geometric equivalence relation is a mathematical equivalence relation.

A simple surface point p concerns with the point p and its surrounding points. The geometric equivalence

relation classifies all $N(27,p)$'s subsets with point p into a number of geometric equivalence classes. Among thes classes, only a few of them make p as a simple surface point. Only 6-adjacent simple surface points are considere in this paper. We will classify all simple surface points based on the geometric equivalence.

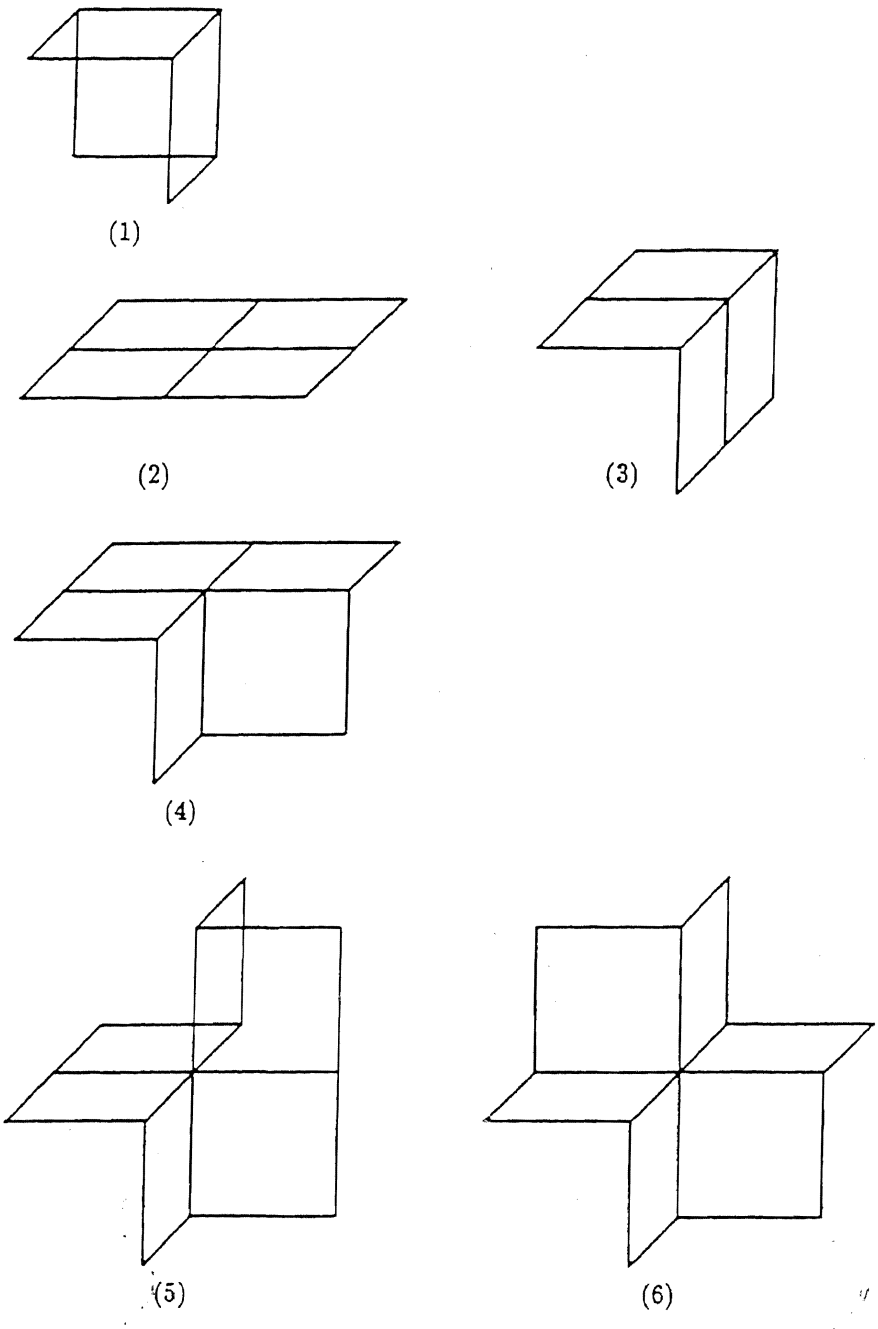


Figure 3.2 All types of simple surface points

Lemma 3.2[12] If p is a simple surface point, then each line_cell containing p in $N(27,p)$ has exactly two parallel-moves; any two surface_cells are line-connected in $N(27,p)$, and there is no 3D_cell in $N(27,p)$.

Therefore, p is a regular inner surface point when p is a simple surface point. According to the two lemmas above, we can obtain:

Theorem 4.1 There are only 6 types of simple surface points that are not geometrically equivalent to each other (see Figure 3.2).

Proof We can start with a point p and one of its directly adjacent points p' because a isolated point can not be a simple surface point. According to Lemma 3.2 and geometric equivalence, the line_cell (p, p') must have two parallel-moves. So, they derive two cases in Figure 3.3, and nothing else.

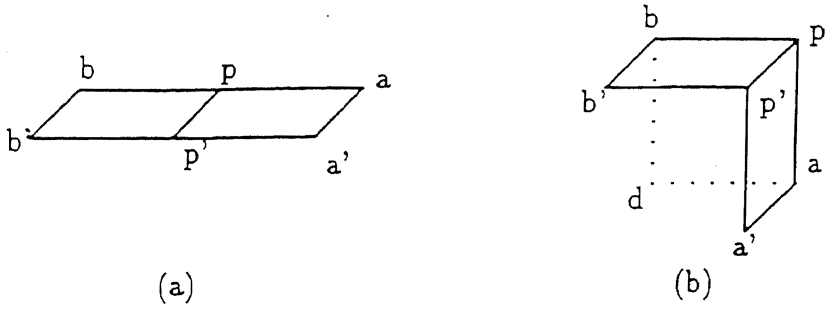


Figure 3.3 Two cases derived by line-cell (p, p')

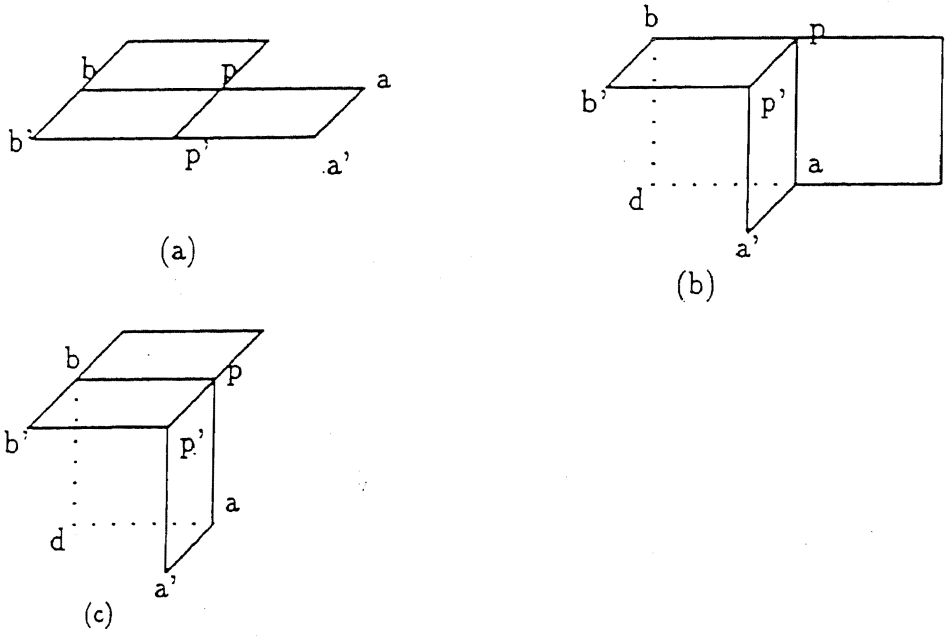


Figure 3.4 Three cases derived by Figure 3.3

Thus, the case (b) in Figure 3.3 can derive a simple surface point with three surface_cells in $N(27, p)$. It is the case (1) in Figure 3.2. Because each line_cell must have exactly two parallel-moves, from Figure 3.3 we can develop the three cases with 3 surface_cells in Figure 3.4, which are possible to be simple surface points without duplicate based on the geometric equivalence.

Continue the derivation, we can develop the 6 cases with 4 surface_cells in Figure 3.5 from Figure 3.4.

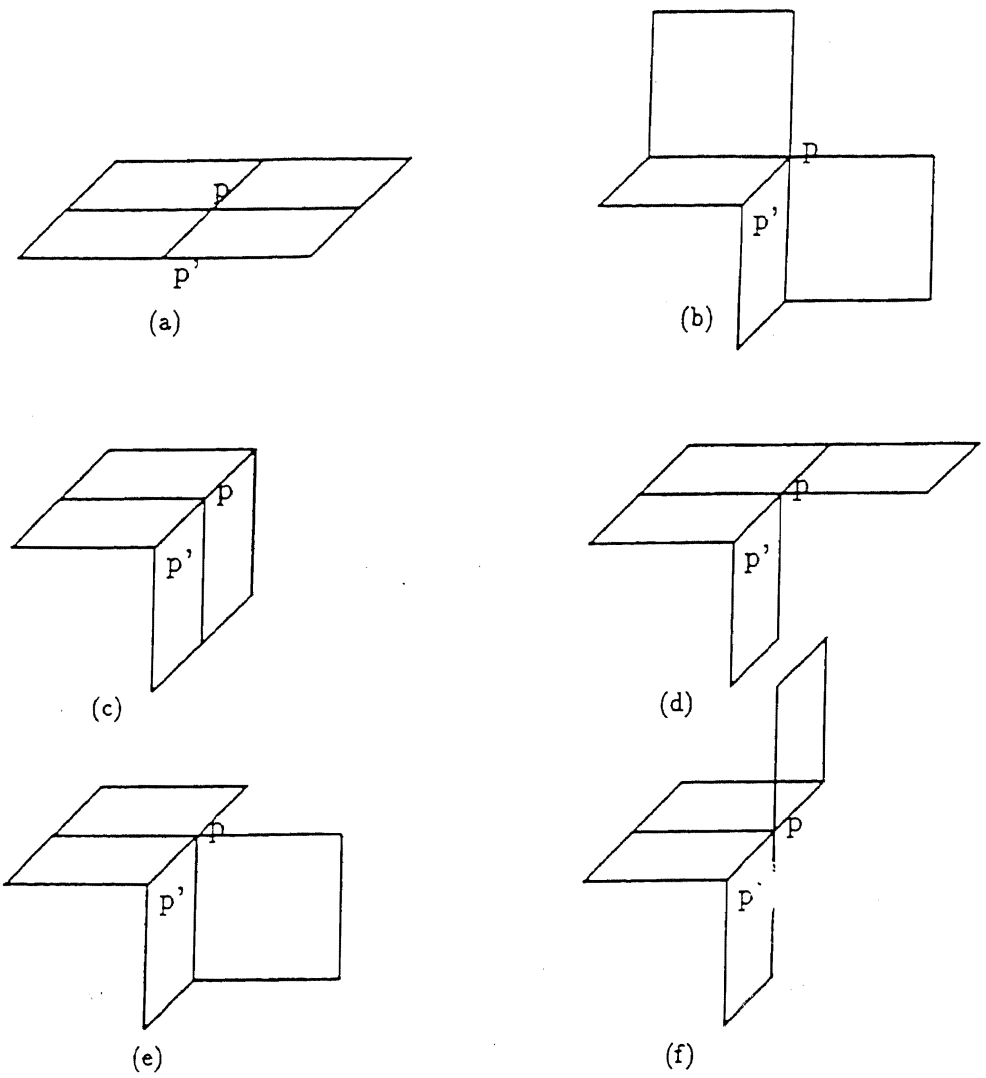


Figure 3.5 Three cases derived by Figure 3.4

Therefore, we get case (2) and (3) in Figure 3.2 from case (a) and (c) in Figure 3.5. Again, we can develop the 6 cases with 5 surface_cells in Figure 3.6 from Figure 3.5.

We get case (4) in Figure 3.2 from case (e) of Figure 3.6. We can see that point p in case (d) can not be a simple surface point, and case (a),(b) (c) and (f) in Figure 3.6 have only one choice to be a simple surface point. When we add a surface_cell to case (a), (c) or (f), we get the simple surface point that is the same as case (5) in Figure 3.2. When we add a surface_cell to case (b), it becomes case (6) in Figure 3.2.

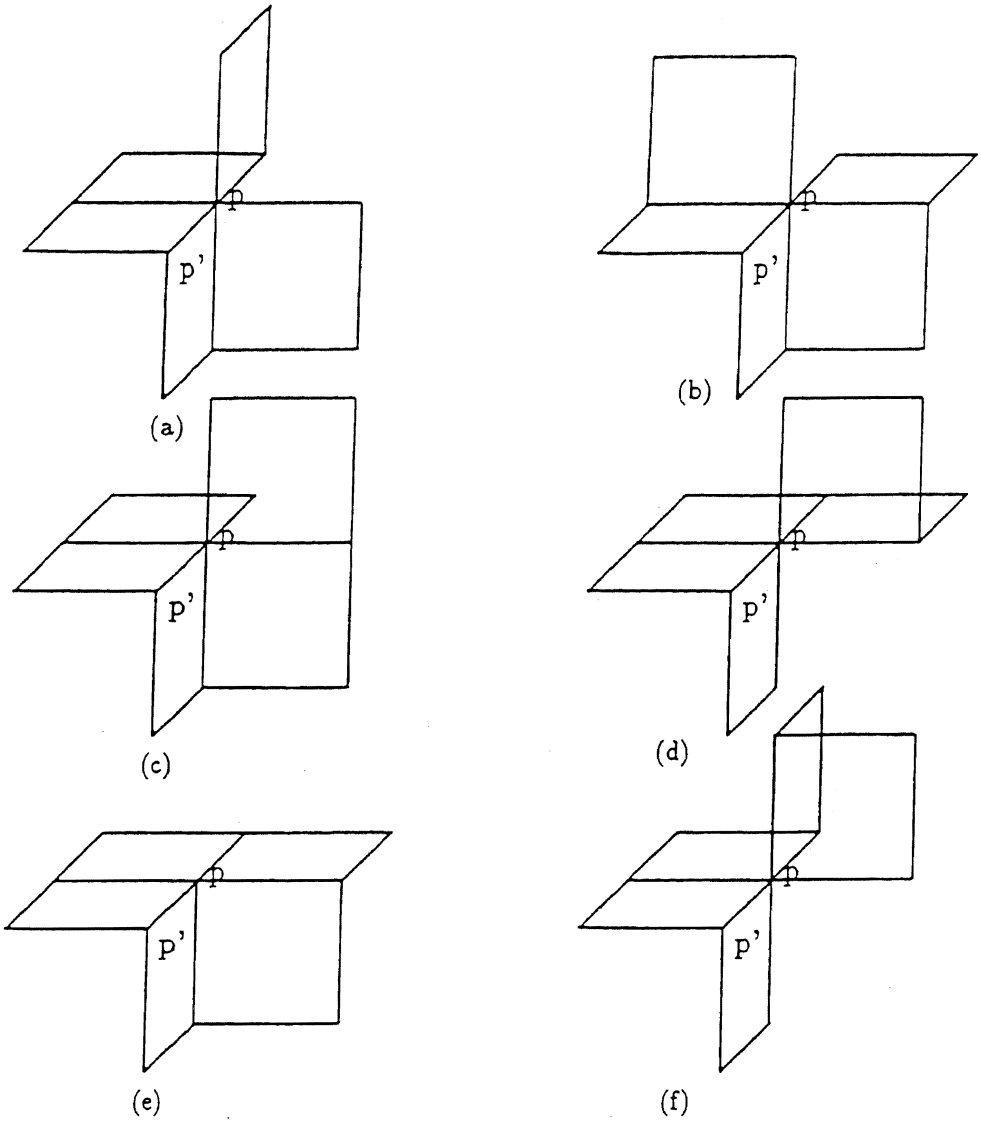


Figure 3.6 Three cases derived by Figure 3.5

4. THE CORNER POINT THEOREM FOR SIMPLE CLOSED SURFACES

This section shows a global theorem for simple closed surfaces in three dimensional digital spaces. The Euler's formula on the planar graph will be used to prove our theorem.

Euler's formula[10] of the planar graph is: Suppose $G = (V, E)$ is a planar graph; if F is the set of faces in G, then $|F| + |V| = |E| + 2$. Let S be a simple closed surface. Because there is no complete bipartite graph $K_{3,3}$ and complete graph K_5 in S, so S is a planar graph. Thus,

Lemma 4.1 S is a planar graph if we view S as a graph $G = (V, E)$, where $V = S$ and $(p, p') \in E$ iff p and p' are adjacent in S.

Let p be a point of S. p is called i-point if p is 6-adjacent to i points in $S \cap N(27, p)$. Let M(3), M(4), M(5),and M(6) be the sets of the 3-point (also called corner point), 4-point, 5-point, and 6-point, respectively. There is no

7-point in a simple surface S [12].

Lemma 4.2 (1) $|V| = |M_3(S)| + |M_4(S)| + |M_5(S)| + |M_6(S)|$;

(2) $|E| = (3|M_3(S)| + 4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)|)/2$;

(3) $|F| = (3|M_3(S)| + 4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)|)/4$.

Proof For (1), according to Theorem 4.1, $M_0(S), M_1(S)M_2(S)$ and $M_7(S)$ etc. can not be the simple surface points. Therefore (1) is ture.

For (2), we just consider each vertex's contribution to the edges E in planar graph G. In $G = (V, E)$, every point in $M_3(S)$ is connected with 3 edges, every point in $M_4(S)$ is connected with 4 edges, every point in $M_5(S)$ connected with 5 edges, and every point in $M_6(S)$ is connected with 6 edges. On the other hand, every edge has two end points. So,

$$|E| = (3|M_3(S)| + 4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)|)/2.$$

For (3), we also consider each vertex's contribution to the faces F in G. In $G = (V, E)$, every point in $M_3(S)$ is connected with 3 faces, every point in $M_4(S)$ is connected with 4 faces, every point in $M_5(S)$ is connected with 5 faces, and every point in $M_6(S)$ is connected with 6 faces. On the other hand, every face has four end points. So,

$$|F| = (3|M_3(S)| + 4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)|)/4.$$

Lemma 4.3 $|M_3(S)| = 8 + |M_5(S)| + 2|M_6(S)|$.

Proof According to the Euler's formula, we have:

$$|M_3(S)| + |M_4(S)| + |M_5(S)| + |M_6(S)| +$$
$$(3|M_3(S)| + 4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)|)/4 =$$
$$(3|M_3(S)| + 4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)|)/2 + 2.$$

Thus,

$$|M_3(S)| + |M_4(S)| + |M_5(S)| + |M_6(S)| =$$
$$(3|M_3(S)| + 4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)|)/4 + 2.$$

Then, $|M_3(S)| = 8 + |M_5(S)| + 2|M_6(S)|$.

According Lemma 4.3, we have:

Theorem 4.1 Any simple closed surface at least has eight corner surface points.

Theorem 4.2 Any simple closed surface at least has 14 points.

Proof First, any two corner surface points cannot be adjacent; otherwise, there would be a three-dimensional unit cell in the surface. According to Theorem 4.1, each simple closed surface has at least eight corner points,

each of which is connected to three edges, which can not connect with corner surface points. Therefore, there are $24 = (3 \times 8)$ points, each of which is not a corner surface points. So, the number of the points in a closed surface is:

$$|M_3(S)| + |M_4(S)| + |M_5(S)| + |M_6(S)| =$$
$$8 + |M_5(S)| + 2|M_6(S)| + |M_4(S)| + |M_5(S)| + |M_6(S)| =$$
$$8 + |M_4(S)| + 2|M_5(S)| + 3|M_6(S)|$$

and

$$4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)| \geq 24.$$

According to Lemma 4.3, we consider the following integer programming problem:

$$8 + |M_4(S)| + 2|M_5(S)| + 3|M_6(S)| = min$$
$$4|M_4(S)| + 5|M_5(S)| + 6|M_6(S)| \geq 24;$$

we have got $|M_4(S)| = 6$ and $|M_5(S)| = |M_6(S)| = 0$. That is, any simple closed surface at least has 14 points.

REFERENCES

1. T. Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," Comput. Vision Graphics Image Process. 48,1989,357-393.

2. G. T. Herman and D. Webster, "A topological proof of a surface tracking algorithm," Comput. Vision Graphics Image Process. 23, 1983, 162-177.

3. T. Y. Kong and J. K. Udupa, "A justification of a fast surface tracking algorithm," CVGIP: Graphical Models and Image Processing 54, 1992, 162-170.

4. L. Chen and J. Zhang, "A Digital Surfaces Definition and Fast Algorithms for Surface Decision and Tracking," Proceedings of SPIE Conference on Vision Geometry II, 1993.

5. A. Rosenfeld, "Digital topology," Amer. Math. Monthly, 86(1979),621-630 .

6. T. Y. Kong, E. Khalimsky and R. D. Kopperman, "A topological approach to digital topology," Amer. Math. Monthly, 98(1991), 901-917.

7. G. Read and A. Rosenfeld, "Recognition of surfaces in three-dimensional digital images," Imform. and Control 53, 1982, 108-120.

8. D. G. Morgenthaler and A. Rosenfeld, "Surfaces in three -dimensional images," Imform. and Control 51, 1981, 227-247.

9. G. T. Herman, "Discrete multidimensional Jordan surfaces," CVGIP: Graphical Models and Image Processing 54, 1992, 507-515.

10. B. Bollobas, "Graph Theory," Springer-Verlag, New York, 1979.

11. L. Chen and J. Zhang, "Digital manifolds: A Intuitive Definition and Some Properties," Proceedings of the Second ACM/SIGGRAPH Symposium on Solid Modeling and Applications, Montreal, 1993.

12. L. Chen and J. Zhang, "Some basic concepts and problems on digital surfaces," SIAM Annual Meeting, July, 1993.

Regular Polygons and Their Applications to Digital Curves

Li Chen Jianping Zhang Donald H. Cooley
Department of Computer Science
Utah State University
Logan, UT 84322-4205
chenl@slow.cs.usu.edu

ABSTRACT

In this note, we discuss various kinds of two dimensional unit cells, or surface-unit cells, made by regular polygons (simplexes) in the plane $R \times R$. Mathematically, the plane can be divided by simplexes or regular polygons (decomposition). If we only allow one kind of surface-unit in the plane, there are only three possible choices: regular triangle (3-regular-polygon), square (4-regular-polygon), or 6-regular-polygon. Using Euler's formula for planar graphs, we give a type of topological proofs to that a closed digital curve has at least 6 points in a 3-regular-polygon decomposition plane, has at least 8 points in a 4-regular-polygon decomposition plane, and has at least 12 points in a 6-regular-polygon decomposition plane, respectively. On the other hand, a plane can also be divided by combinations of two kinds of regular polygons. We have obtained two types of {3, 6}-regular-polygon combinations, two types of {4, 8}-regular-polygon combinations, and one type of {3, 12}-regular-polygon combination. We also discuss the applications of polygons or closed paths to digital surfaces in three dimensional digital spaces.

1. INTRODUCTION

The topological and geometric properties of objects of digital spaces are quite useful for image processing and computer vision. Rosenfeld proved the discrete Jordan curve theorem and another important theorem: Any simple closed curve must have five points in a two dimensional digital plane, where the curve is a directly connected (or 4-connected) path [1]. On the other hand, in differential geometry there is a famous four vertex theorem: Every simple closed convex curve in a plane has at least four vertices, where the vertex is a point whose curvature has a local maximum or minimum.

In this note, we discuss various kinds of two dimensional unit cells, or surface-unit cells, made by regular polygons (simplexes) in the plane $R \times R$. Mathematically, the plane can be divided by simplexes or regular polygons (decomposition). If we only allow one kind of surface-unit in the plane, there are only three possible choices: regular triangle (3-regular-polygon), square (4-regular-polygon), or 6-regular-polygon. However, a plane can also be divided by combinations of two kinds of regular polygons. We have obtained two types of {3, 6}-regular-polygon combinations, two types of {4, 8}-regular-polygon combinations, and one type of {3, 12}-regular-polygon combination.

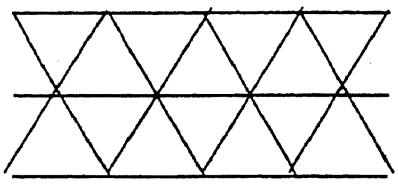
We also study the properties of the topology under these surface-unit cell partitions. Surface-unit cells are used to partition a plane or to form a surface.

0-8194-2214-2/96/$6.00

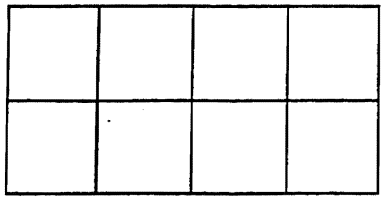
2. REGULAR POLYGONS

There are many ways to divide a plane into simplex pieces. The most popular one is to use regular 4-polygons. Its lattice space is Σ_2 where each point has integer coordinates. Regular triangles (3-polygon) are also used to represent wired frame surfaces. Although many applications need more flexible polygons not only regular polygons, in this note we concentrate on the simplest case of regular polygons.

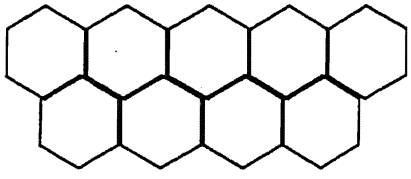
How many kinds of regular-polygons can be used for the purpose of filling a plane? If one only allows one kind of regular n-polygon, then there are only 3 types of discrete planes: 3-polygon plane, 4-polygon plane, and 6-polygon plane (see Figure 2.1). The reason is simple because for an inner angle (A) in Figure 2.2 of a regular n-polygon: $A = (n-2)180/n = 180 - 360/n$, so $B = 360 - A = 180 + 360/n$. In addition, assume there are m regular n-polygons to share a point, then $n \cdot m = 360$. We have $A|360$. Thus $(n-2)|2n$, so n only can be 3, 4, and 6.



(a) 3-polygon-unit cells



(b) 4-polygon-unit cells



(c) 6-polygon-unit cells

Figure 2.1 Planes made by regular polygon cells

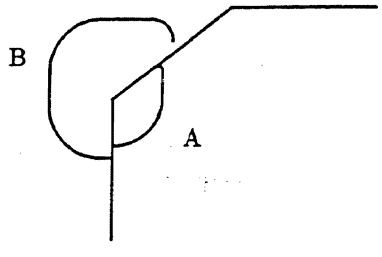
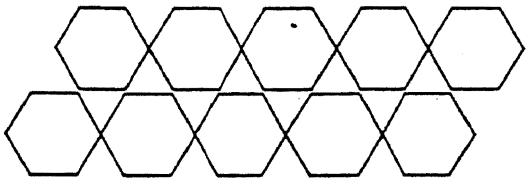
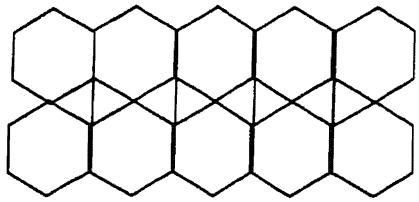


Figure 2.2 An angle in a regular polygon

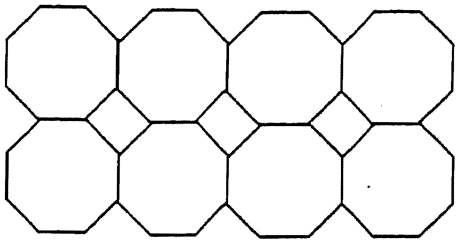
If one allows more than one kind of regular polygon; we have two types of $\{3,6\}$-polygon plane, two types of $\{4,8\}$-polygon plane, and a $\{3,12\}$-polygon plane. See Figure 2.3



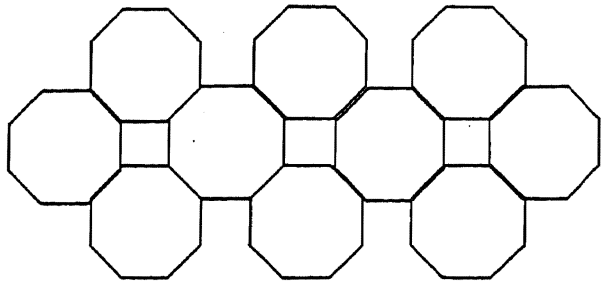
(a) 3-6-polygon-unit cells



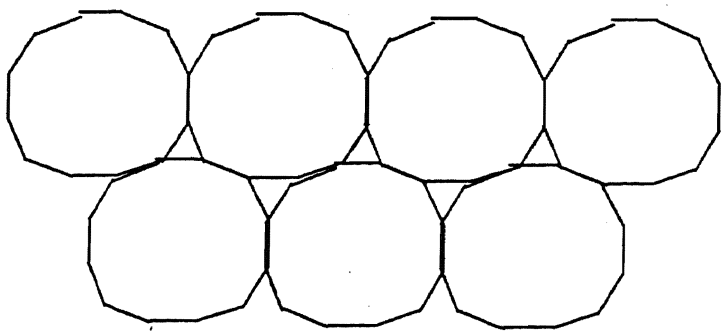
(b) 3-6-polygon-unit cells



(c) 4-8-polygon-unit cells



(d) 4-8-polygon-unit cells



(e) 3-12-polygon-unit cells

Figure 2.3 Planes made by two kinds of regular polygon cells

3. SIMPLE CLOSED DIGITAL CURVES ON PLANES

We know that a 3-regular-polygon has three edges, a 4-regular-polygon has four edges and so forth. Only these edges can be used for vertices connection. For example, for a 4-regular-polygon we use 4-adjacency. A digital curve (or curve) is made by a simple path of vertices. Intuitively, a closed curve is a curve and each vertex has exactly two adjacent vertices in the curve. We know such closed curves have a Jordan property: assume the plane P and the closed curve C, $P - C$ has two nonempty components (a component means a connected part), IN_C and OUT_C, which are disconnected.

There is a only confusion here. The Jordan property holds because we consider the space P is a plane, which is

a continuous space. If we only consider P, a set of vertices (all conner points of polygons), we must perform some changes in order to maintain Jordan property. For example, a curve does not contain any subset which is just all points of a polygon. Otherwise, all vertices of a polygon form a curve C but $P - C$ is just a component. In this paper, we consider that P is a discrete plane.

Some topological properties can be obtained by 3-regular-polygon, 4-regular-polygon and 6-regular-polygon decompositions:

(1) 3-regular-polygon Let the surface-unit cell consist of a 3-regular-polygon. If C is a closed curve then there are five kinds of simple curve points in $IN_C \cup C$. We call a point p on C a CP_i point if p has i adjacent points in $IN_C \cup C$.

Proposition 2.1 $|CP_3| = |CP_5| + 6$. Furthermore, a closed curve has at least 6 points.

Proof. According to Figure 3.1, we know $|CP_1| = 0$ and $|CP_i| = 0$ if $i > 6$. In fact $CP_2 = 0$ (Point A in Figure 3.1) and $CP_6 = 0$ (Point E in Figure 3.1) because C cannot contain any 3-regular-polygon. We call that the points of CP_4 are straight line points, point C in Figure 3.1, as well as outward-conner points for CP_3, point B in Figure 3.1, and inward-conner points for CP_5, point D in Figure 3.1 .

Since the vertices and edges generated by these 3-regular polygons form a planer graph, we can use Euler planer graph theorem to find the relationship between the CP_3 and CP_5. This theorem says: In a planer graph, if V, E, and F represent the set of vertices, edges and faces respectively, then $|E| = |F| + |V| - 2$. In our case

$$|V| = |CP_5| + |CP_3| + |CP_4| + |IN_C|,$$

$$|E| = (6 \cdot IN_C + 5 \cdot |CP_5| + 4 \cdot |CP_4| + 3 \cdot |CP_3|)/2,$$

and

$$|F| = (6 \cdot IN_C + 4 \cdot |CP_5| + 3 \cdot |CP_4| + 2 \cdot |CP_3|)/3 + 1.$$

Therefore,

$$|CP_5| + |CP_3| + |CP_4| + |IN_C| + (6 \cdot IN_C + 4 \cdot |CP_5| + 3 \cdot |CP_4| + 2 \cdot |CP_3|)/3 + 1 - 2$$

$$= (6 \cdot IN_C + 5 \cdot |CP_5| + 4 \cdot |CP_4| + 3 \cdot |CP_3|)/2,$$

we have $|CP_3| = |CP_5| + 6$. □

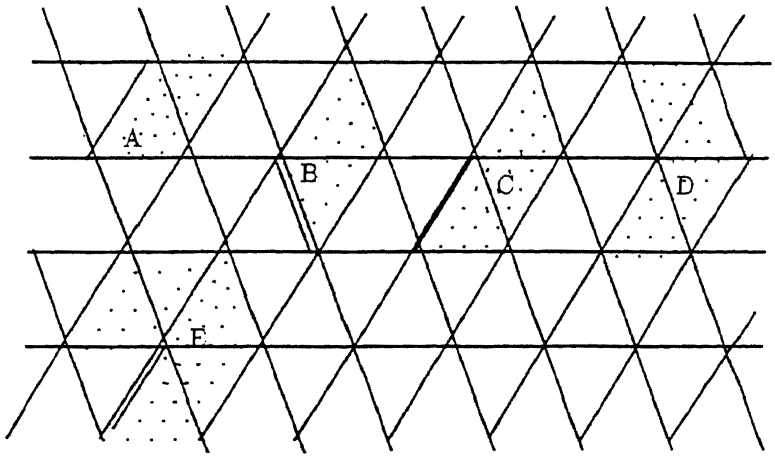


Figure 3.1 Vertices classification of regular polygon cells

(2) 4-regular-polygon Let the surface-unit cell be only made of 4-regular-polygons i.e. $Z \times Z$. This is the most popular case, considerable research work has dealt with this case. If C is a closed curve then we will see that there are three kinds of simple curve points respecting to $IN_C \cup C$. We also call a point p on C is a CP_i point if p has i adjacent points in $IN_C \cup C$. In fact, $|CP_1| = 0$ and $|CP_i| = 0$ if $i > 4$. CP_2 contains outward corner points, CP_3 contains straight-line points, and CP_4 contains inward corner points. Rosenfeld, Kong and etc. obtained that "a closed curve has at least 8 points." We give a different of proof in this note.

Proposition 2.2 $CP_3 = CP_4 + 4$. Furthermore, a closed curve has at least 8 points.

Proof. For this proof, we use the same technique as previously. Since the vertices and edges generated by these 4-regular polygons form a planer graph, we can use Euler planer graph theorem to find the relationship between the CP_2 and CP_4. In this case,

$$|V| = |CP_4| + |CP_3| + |CP_2| + |IN_C|,$$

$$|E| = (4 \cdot IN_C + 4 \cdot |CP_4| + 3 \cdot |CP_3| + 2 \cdot |CP_2|)/2,$$

and

$$|F| = (4 \cdot IN_C + 3 \cdot |CP_4| + 2 \cdot |CP_3| + |CP_2|)/4 + 1.$$

Therefore,

$$|CP_4| + |CP_3| + |CP_2| + |IN_C| + (4 \cdot IN_C + 3 \cdot |CP_4| + 2 \cdot |CP_3| + |CP_2|)/4 + 1 - 2$$

$$= (4 \cdot IN_C + 4 \cdot |CP_4| + 3 \cdot |CP_3| + 2 \cdot |CP_2|)/2,$$

we have $|CP_2| = |CP_4| + 4$.

On the other hand, C does not contain any 4-regular polygon, so any two outward corner points cannot be adjacent. There are at least 4 outward corner points. The positions between two outward corner points must be filled by the points in CP_3 or CP_4. Thus, $|C|$ must be greater than or equal to 8. □

(3) 6-regular-polygon Let the surface-unit be only made of 6-regular-polygons (see Figure 2.1 (c)). If C is a

closed curve then there are two kinds of simple curve points respecting to $IN_C \cup C$. We also call a point p on C a CP_i point if p has i adjacent points in $IN_C \cup C$. In fact, Figure 2.1 (c), $|CP_1| = 0$ and $|CP_i| = 0$ if $i > 3$. CP_2 contains outward corner points, and CP_3 contains inward corner points.

Proposition 2.3 $CP_2 = CP_3 + 6$, $|CP3| \geq 3$. Thus, a closed curve has at least 12 points.

Proof. We use the same technique as we did above. Since the vertices and edges generated by these 6-regular polygons form a planer graph, we can use the Euler planer graph theorem to find the relationship between CP_2 and CP_3. In this case,

$$|V| = |C| + |IN_C| = |CP_3| + |CP_2| + |IN_C|,$$

$$|E| = (3 \cdot IN_C + 3 \cdot |CP_3| + 2 \cdot |CP_2|)/2,$$

and

$$|F| = (3 \cdot IN_C + 2 \cdot |CP_3| + |CP_2|)/6 + 1.$$

Therefore,

$$|CP_3| + |CP_2| + |IN_C| + (3 \cdot IN_C + 2 \cdot |CP_3| + |CP_2|)/6 + 1 - 2$$

$$= (3 \cdot IN_C + 3 \cdot |CP_3| + 2 \cdot |CP_2|)/2,$$

we have $|CP_2| = |CP_3| + 6$.

If $|CP_3| = 0$, then C is not a Jordan curve. Therefore $|CP_3| >= 1$. Let $c \in CP_3$, then there is a point p is in IN_C and adjacent to c. c has two other adjacent points r_1, r_2. If both r_1 and r_2 are in C, then they must be in CP_3 by the definition of CP_3. So, $|CP_3| >= 3$. If r_1 is in CP_3 but r_2 is not in CP_3, then $r_2 \in IN_C$, and r_2 has two adjacent points r_{21} and r_{22}. Neither $r_{21} \in \{c, p, r_1\}$ nor $r_{22} \in \{c, p, r_1\}$. Because In_C is finite, so if r_{21} or r_{21} is not in CP_3 we can find their adjacent points, until we reach the boundary C. Thus, $|CP_3| >= 3$. If neither r_1 nor r_2 is in C, there must be two out off the four adjacent points of them are never tending to be adjacent. So, they will reach different boundary points. Thus, we still have $|CP_3| >= 3$. Since $|C| = |CP_2| + |CP_3| = 2 \cdot |CP_3| + 6 \geq 12$. □

4. AN INTUITIVE CONSIDERATION

Using Euler theorem to prove these propositions is probably too complicated; however, this method at least provides global topological proofs for these properties. A very simple strategy is to consider the 360-degree trace of a simple closed curve. One can get the three propositions very easily. Let us consider the 4-regular-polygon. Let a point p be on closed C, find p's anti-clock adjacent point q. Suppose that C is the path $q = q_0, q_1, ..., q_{n-1} = p$. Start at q and we have (intuitively)

$$360 = \sum_{i=0}^{n-1} angle((p_{i-1}, p_i), (p_i, p_{i+1}))$$

where $p_{-1} = p_{n-1}$ and $p_n = p_0$. A straight-line point has no contribution to the above summation. An outward point has 90 degrees contribution, and an inward point has -90 degrees contribution. Therefore, $|CP_2| = |CP_4| - 4$. Also, if a surface S on a plane has a hole and C' is the boundary of the hole, then $|CP_4| = |CP_2| + 4$ in C'.

Similarly, $|CP_3| = |CP_5| + 6$ for 3-regular-polygon plane closed curve, and $|CP_2| = |CP_3| + 6$ for 6-regular-polygon.

Also using this method, if S is a surface with a hole on a plane, and C is the boundary of S, then the number of inward points equals the number of outward points.

This method has a disadvantage, that is it can not be extended to solve the surface problem in three dimensional digital spaces [2].

5. DISCUSSION RELATED TO 3D SPACES

Some applications need more generalized curves and surfaces, such as a mesh for surface fitting. A mesh of a continuous surface is a "discrete surface" which is mostly like a graph. Giving a graph-based discrete surface definition was first suggested and studied by Herman[3].

Polygons can be defined as surface cells; however, in 3D spaces, a surface cell may not be on a plane. To extend the concept of polygons in use of general surface-cells is necessary. We can define a surface cell as a minimum closed path in a graph. Let $G = (V, E)$ be a simple graph. A subgraph C of G is called a (discrete) curve iff each vertex v in C is included in one or two edges in C and C does not contain any surface-cell. A subgraph S of G is called a (discrete) surface iff each edge e in S is included in one or two surface-cells in S and S does not contain any solid-cell; and so forth.

Based upon this definition, we will see that a simplex decomposition of a continuous surface is a discrete surface if each triangle is treated as a surface-cell. In addition, For a two binary image, we can use a quadtree to represent the image. Each leaf of the quadtree can be used as a surface-cell, then we will be able to develop a very easy to search the boundary of the image.

REFERENCES

1. T.Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," Comput. Vision Graphics Image Process, Vol 48, pp. 357-393, 1989.

2. L. Chen and J. Zhang, "Classification of Simple digital Surface Points and A Global Theorem for Simple Closed Surfaces," The Proc. of SPIE on Vision Geometry II, Vol 2060, pp. 179-188, 1993.

3. G.T. Herman, "Discrete multidimensional Jordan surfaces," CVGIP: Graphical Model and Image Processing, Vol 54, 507-515, 1992.