SPECTRAL GRAPH BASED APPROACH FOR ANALYSIS OF 3D LIDAR POINT CLOUDS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EDA BAYRAM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

MARCH 2017

Approval of the thesis:

SPECTRAL GRAPH BASED APPROACH FOR ANALYSIS OF 3D LIDAR POINT CLOUDS

submitted by **EDA BAYRAM** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver ______________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tolga Çiloğlu ______________
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. A. Aydın Alatan ______________
Supervisor, **Electrical and Electronics Engineering, METU**

Assist. Prof. Dr. Elif Vural ______________
Co-supervisor, **Electrical and Electronics Engineering, METU**

Examining Committee Members:

Assoc. Prof. Dr. Cüneyt F. Bazlamaçcı
Electrical and Electronics Engineering Department, METU ______________

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Department, METU ______________

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering Department, METU ______________

Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Department, METU ______________

Assoc. Prof. Dr. Selim Aksoy
Computer Engineering Department, Bilkent University ______________

Date: ______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: EDA BAYRAM

Signature :

ABSTRACT

SPECTRAL GRAPH BASED APPROACH FOR ANALYSIS OF 3D LIDAR POINT CLOUDS

Bayram, Eda
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. A. Aydın Alatan
Co-Supervisor : Assist. Prof. Dr. Elif Vural

March 2017, 77 pages

Airborne Laser Scanning is a well-known remote sensing technology, which provides quite dense and highly accurate, yet unorganized, point cloud descriptions of the earth surface. However, processing of such a 3D point cloud is quite challenging due to its irregular structure and 3D geometry. In this thesis, two novel approaches for the analysis of unorganized 3D point cloud data are proposed, specifically the ones that are generated by the airborne mounted LIDAR sensor. These methods rely on the spectral graph based and graph signal processing techniques which gain attention in the recent years. The state-of-the-art techniques addressing the problems of LIDAR point clouds are first examined. Next, the theory presented by the spectral graph based methods is reviewed to analyze their solutions. Since irregular discrete data lying on a high dimensional geometry, such as LIDAR point clouds, can be conveniently represented by weighted graphs, spectral graph methods based on such weighted graphs enable spectral analysis of the data representation, as in classical Fourier analysis for signal processing. In the light of the revisited spectral graph literature, one can examine techniques for clustering as well as edge detection problems by using graph representation of the unorganized 3D point clouds. The graph based representation introduces the opportunity of analysis of the signal over its original input space; therefore, it provides qualified comprehension of the data. Based on simulations, it is shown that the graph spectral solutions can acquire remarkable advance in the analysis of unorganized 3D point clouds and the experimental results indicate the potentials of this new approach.

Keywords: Signal Processing on Graphs, Graph Signal Filtering, Spectral Graph Theory, Spectral Clustering, Unorganized 3D Point Cloud, LIDAR, Airborne Laser Scanning

ÖZ

3 BOYUTLU LIDAR NOKTA BULUTLARININ ANALİZİNDE SPEKTRAL ÇİZGE TEMELLİ YAKLAŞIM

Bayram, Eda
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. A. Aydın Alatan
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Elif Vural

Mart 2017 , 77 sayfa

Havadan lazer tarama, dünya yüzeyinin oldukça yoğun ve yüksek kesinliğe sahip fakat organize olmayan nokta bulutu betimlemesini sağlayan, tanınan bir uzaktan algılama teknolojisidir. Fakat, bu tip bir 3 boyutlu (3B) nokta bulutunun işlenmesi düzensiz yapısı ve 3B geometrisi bakımından zorlayıcıdır. Bu tezde, organize olmayan 3B nokta bulutu verisinin analizi için iki yeni yaklaşım önerilmektedir, özellikle hava araçları üzerindeki LIDAR cihazlarından alınmış veriler üzerine odaklanılmaktadır. Bu yöntemler son yıllarda ilgi kazanan spektral çizge temelli ve çizge sinyal işleme tekniklerine dayanmaktadır. Öncelikle LIDAR nokta bulutu alanındaki problemlere hitap eden güncel teknikler incelenmiştir. Daha sonra, spektral çizge temelli yöntemler tarafından sunulan teori, çözümlerini incelemek için gözden geçirilmiştir. Ağırlıklandırılmış çizgeler, LIDAR nokta bulutları gibi, yüksek boyutlu geometri üzerine oturan, düzensiz, ayrık verilerin ifadesi için çok uygun olduğundan, aynı sinyal işlemede klasik Fourier analiz kullanılarak yapıldığı gibi, spektral çizge yöntemleri ile de veri ifadesinin spektral analizini gerçekleştirmek mümkündür. Tekrar gözden geçirilmiş spektral çizge literatürünün ışığında, 3B nokta bulutunun çizge ifadesi üzerinde kümeleme ve kenar tespiti problemleri incelenebilir. Çizge temelli ifade, sinyali özgün yapısı ile tanımlı olduğu uzay üzerinde analiz etme imkanı sağladığı için, verinin niteikli bir şekilde kavranmasını sağlamaktadır. Simülasyonlar üzerinden spektral çizge temelli çözümlerin, organize olmayan 3B nokta bulutlarının analizinde kayda değer bir avantaj sağladığı gösterilmiş ve deneyler ile bu yeni yaklaşımın potansiyeli ortaya konmuştur.

Anahtar Kelimeler: Çizge Sinyal İşleme, Çizge Sinyal Filtreleme, Spektral Çizge Teori, Spektral Kümeleme, Organize olmayan 3 Boyutlu Nokta Bulutu, LIDAR, Havadan Lazer Tarama

Can we hear the shape of a drum?

ACKNOWLEDGMENTS

Above all, I would like to thank my supervisor Prof. Aydın Alatan for his valuable guidance and tolerance. There is no doubt that his wise questions and advices made my work better. I would like to thank my dear co-supervisor Elif Vural not just for sharing her valuable knowledge on the literature, but for discovering and discussing together. I am very appreciated by her encouragements for me to follow an academic path.

I want to thank all the researchers, whose study is refereed in this work, for enlightening my way to accomplish my thesis. One of them is very special to me; I would like to thank Prof. Pascal Frossard for introducing to me the core theory of this thesis and for his invaluable support. In addition, I want to thank his dear assistant Dorina Thanou for her nice explanations to all my questions during my internship at LTS4.

I would like to thank my former colleagues, Kutalmış İnce, Cem Tarhan and Fatih Erkan, for feeding me socially and intellectually and for the fruitful coffee sessions. Special thanks to Kutalmış, for sharing the most victorious and troublesome moments together during preparation of our thesis.

A huge thanks to my dear friends; Alp Yurtsever for being the one encouraging me to follow an academic path in the first place, Gökçe Öztürk for always being ready to listen me and for our precious conversations about everything since the bachelor years, Elvan Aydemir for her support and for being always sincere in her opinions she shared.

Last but not the least I thank my parents not for bringing me to a life but for giving me the space to explore my own life, which seems to be an important point for accomplishing this study as well.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ALS	Airborne Laser Scanning
DEM	Digital Elevation Model
DSM	Digital Surface Model
GFT	Graph Fourier Transform
IGFT	Inverse Graph Fourier Transform
LIDAR	Light Detection and Ranging
LSC	Landmark Based Spectral Clustering
TIN	Triangulated Irregular Network

CHAPTER 1

INTRODUCTION

1.1 Motivation

Over the past few decades, the multimedia technology has evolved to a state where it is capable of sampling and storing rich environmental information. For example, various active or passive optical devices such as 3D scanners or depth sensors can capture high dimensional or hyperspectral data. 3D data has become more widespread in virtual reality, computer graphics, vision and surveillance systems, furthermore, it is predicted to be more preferable and widely used in the future. As much more effort is devoted on the aforementioned hardware technologies, the research community is expected to attend to the problems emerged by their output data.

One of these technologies is LIDAR (Light Detection And Ranging) which is an optical remote sensing device providing very dense and accurate point samples of the surface scanned in 3D. Therefore, in recent years, airborne laser scanning (ALS) has become one of the most desired technologies, especially in the Geographic Information System (GIS), photogrammetry and cartography projects. However, processing of LIDAR data is a challenging task due to the nonuniform sampling on the 3D geometry. To put it in another way, the solutions to the problems arising in unorganized 3D data are not straight forward as the ones developed for images or videos, which are the 2D data structures having the regular lattice forms. For this purpose, most of the primary studies converted airborne LIDAR point clouds into some other reduced data formats, such as Digital Elevation Model (DEM), Digital Surface Model (DSM) or Triangulated Irregular Network (TIN)[2, 3], in order to use the conventional data processing methods to analyze the low dimensional and structured version of the original data. However, transforming the 3D point cloud into those range images manipulates the data due to interpolation and resampling operations and loss of 3D geometry [4, 5]. Moreover, from a signal processing point of view, we need to designate the raw output data as the signals to be processed, in order to fully exploit the 3D nature of the data and so as to lead to efficient interpretations.

In recent years, many other irregular data formats have been produced together with the 3D point clouds generated by LIDAR. For instance, 3D depth sensors capture 3D models of the environment or there are data sources having network-like structures such as the social networks, transportation networks etc. All the signals animated on those topologically complicated domains can be represented on the vertices of the weighted graphs by their nature [1, 6]. We refer those signals as graph signals and this new research field as Graph Signal Processing, which analyzes the signals on their native structures. It is built upon the well-known spectral graph theory [7], and above that, it accomplishes the harmonic analysis of the graph signals in the adapted graph structure of the data.

In a graph representation, vertices symbolize the samples on the discrete data domain. The connections on the graph topology are represented by the links called edge, and the weight associated with each edge stores the correlation between the two vertices it links. With this in mind, a weighted graph is desirable tool for the acquisition of the underlying data geometry. Building up such a representation, we can transpose the Fourier analysis methods from the classical signal processing settings to the graph settings.

Laplacian based transforms appear in many differential problems in physics and mathematics such as wave equation, heat equation or Schrödinger equation [8], since Laplacian eigenfunctions provide a generalization of Fourier analysis on various domains from time domain signals to manifolds or graph settings. Therefore, the Laplacian operator defined on the graph settings uncovers the spectral identity of the graph structure, just as we can obtain a wave component at a given frequency level on which it vibrates based on the wave equation.

Discovering spectral embedding of a graph structure may serve many pattern recognition problems such as clustering, feature detection and classification. Moreover, having revealed the spectral identity of the underlying graph, harmonic analysis of the graph signal can be accomplished. Within the scope of signal processing of graphs, filtering, compression and multi-resolution analysis applications can be realized [1].

In the light of this background, this study addresses the segmentation problem of LIDAR data first by practicing the spectral clustering. Second, edge features on a LIDAR scene are detected adopting a graph signal filtering based approach. To place this study in the categorical ordering of the LIDAR literature, it can be asserted that we propose new clustering based segmentation and edge-based segmentation methods for unorganized airborne LIDAR point clouds.

1.2 Thesis Outline

The goal of this study is to analyze unorganized 3D airborne LIDAR point clouds by adapting spectral graph based solutions.

For this purpose, we first go over the theory presented by the graph spectral approaches in order to familiarize the problems they address and examine the possible solutions to our subject of interest. In Chapter 2, we review spectral graph theory by introducing the terms; weighted graph representations, Laplacian operator on graphs and eigensystem of graph Laplacian. We explain the significance of spectral graph theory owing to exploring the spectral embedding encoded through the graph representation by conducting an eigenvalue problem. We mention the solutions proposed for dimension reduction and clustering problems by exploiting the uncovered spectral embedding of the data.

In Chapter 3, we go one step further and carry the Fourier analysis methodology to the graph structures. Just as we operate the classical signal processing on two different domains, namely the time (or spatial) domain and the frequency domain, we define the vertex domain and spectral domain on graphs. We introduce the term of graph signal and we analyze this signal through some operations such as graph Fourier transform, filtering, translation and localization.

After we assemble the proposed solutions declared by the spectral graph literature, we adapt some of them to practice on a number of LIDAR point cloud datasets by adopting computationally efficient algorithms. In Chapter 4, we propose a new framework for LIDAR point cloud segmentation based on spectral clustering method. We justify an adequate graph representation for the aerial LIDAR data to improve the performance of spectral clustering. In order to accelerate the segmentation process, we adapt a practical and fast spectral clustering algorithm.

In Chapter 5, we practice graph signal filtering method to detect the edge features on a LIDAR point cloud, which is enlightened by the conventional signal processing approaches. This is a novel framework for the detection of boundary points in a LIDAR object or scene. The boundary points resulted in this framework can be further be assembled to create edge lines those are important for segmentation and reconstruction tasks. In this framework, before the filtering operation for edge detection, we perform a smoothing operation to loose the effect of clutter. We adjust the filter to be employed with respect to the property of the feature points on interest.

Finally in Chapter 6, the thesis is summed up by pointing out the important conclusions given by the experimental results and the issues to be improved for further.

CHAPTER 2

AN OVERVIEW OF SPECTRAL GRAPH BASED METHODS

2.1 Introduction

In recent years, many high dimensional data, those are structured or unstructured, have been emerged by some advanced sensing technologies. The primary studies working on high dimensional data developed the kernel-based methods to combat the curse of dimensionality [9]. Kernel-based methods typically operate on high dimensional or irregular structured data by building correlations on pairs of samples. Similarly, on later studies, those kernel-based relations established on the sensed data are accommodated on a graph and the kernel-matrix is granted as a weight matrix. The weight function is selected to measure the similarities between the data points by exploiting some statistical learning problems [1], which is identical to the use of kernel-function [10].

It is worth noting that graphs are flexible data structures those are capable of dealing with various types of data domains, since they can encode complex relationships between samples on any domain.

In this chapter, we introduce the spectral graph theory which is the scientific field we leverage for significant applications defined on 3D point clouds (i.e., spectral clustering). It also constitutes a basis for the next chapter where we explain how to apply signal processing algorithms on graphs. First, we give the basic definitions on graph based methods. Second, we express their expansion in spectral domains within the scope of the literature on algebraic graph theory [11] and spectral graph theory [12], [7]. Finally, building an analogy between spectral graph based methods and manifold learning methods, we explain utilization of spectral graph theory for the essential problems such as dimensionality reduction and spectral clustering.

2.2 Review of Spectral Graph Theory

2.2.1 Weighted Graphs and Graph Laplacian

Weighted graphs are generally utilized for representing the geometry-based local interactions between inputs. In algebraic graph theory, *adjacency matrix*, A, stores binary relations between the entries, namely $A(x, y) = 1$, if node-x is adjacent to node-y, i.e., $x \sim y$, and $A(x, y) = 0$, otherwise. The adjacency matrix informs only about the topology of the graph, whereas for a weighted and undirected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, *weight matrix*, W stores similarities between nodes. Its element W_{xy} is determined by a weight function $w(x, y) = w(y, x) > 0$ for $x \sim y$, and $w(x, y) = 0$, otherwise. $\mathcal{V}$ and $\mathcal{E}$ stand for the vertex and edge set of the graph with $|\mathcal{V}| = N$.

In most of the practical cases, the vertices of the graph are depicted as the samples in high dimensional Euclidean space and the weight function is formulated as a function of Euclidean distance between the pairs of them.

Laplacian matrix, L, is described as $L = D - W$, where D is diagonal *degree matrix*, whose entry D_{ii} is the sum of the edge weights incident to vertex-i. The entries of the Laplacian matrix are obtained as:

$$L_{ij} = \begin{cases} D_{ii} & \text{if } i = j, \\ -W_{ij} & \text{if } i \neq j. \end{cases} \tag{2.1}$$

The graph Laplacian is simply the discrete analogue of the Laplacian operator Δ in calculus; i.e. the sum of the second partial derivatives, $\Delta f = \sum_i \frac{\partial^2 f}{\partial x_i^2}$. The Laplacian matrix is indeed a differential operator on the graph [1], in which interaction occurs only between the neighboring vertices [13]. Considering a function f defined on the vertices of the graph, $f : \mathcal{V} \to \mathbb{R}$, the discrete model of Δf measures how the function differs at a vertex with respect to its neighbors. By means of 2nd order finite difference method (2.2), Laplacian operator $\Delta f = \Sigma \frac{\partial^2 f}{\partial x_i^2}$ can be approximated for discrete data domains.

$$f^{''}(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{2.2}$$

Therefore, at a vertex of a graph, Laplacian can be computed by taking the difference of neighbors, and calculating the weighted sum of the results as the following,

$$\sum_{j \sim i} W_{ij}[f(i) - f(j)] \tag{2.3}$$

where the sum is calculated over $j \sim i$ which are the vertices connected to vertex i. The above formulation can be rearranged as,

$$\begin{aligned} \sum_{j \sim i} W_{ij} f(i) - \sum_{j \sim i} W_{ij} f(j) \\ D_{ii} f(i) - \sum_{j \sim i} W_{ij} f(j) \end{aligned}$$

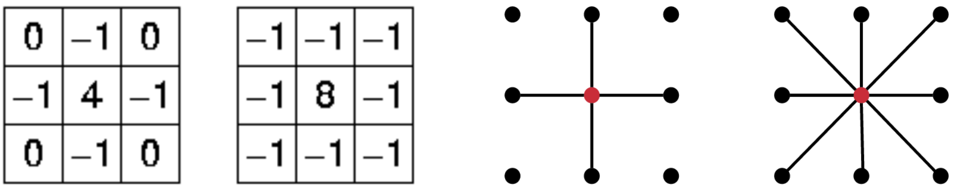
When we show the above rearrangement in terms of matrix-vector multiplication, we obtain,

$$Df_{(i)} - Wf_{(i)} = (D - W)f_{(i)}$$

and since we define graph Laplacian as $L = D - W$, we can compute Laplacian on graphs as follows,

$$\sum_{j \sim i} W_{ij}(f(i) - f(j)) = (Lf)_{(i)} \tag{2.4}$$

Furthermore, one can easily draw an analogy between the aforementioned Laplacian matrix for the graphs and Laplacian kernels for images, which are the discrete approximation of Laplacian operator obtained by finite-difference method and generally applied for edge detection purposes. In Figure 2.1, the Laplacian image kernels are shown with the associated graph topologies. Let us consider an image represented by a 4-connected topology, which is illustrated by the graph on the left in Figure 2.1(b). When the graph Laplacian operates on the center red node, whose degree is 4 and weight of each incident edge is 1, it would be equivalent to filtering the image with a kernel on the left in Figure 2.1(a). For an 8-connected case, one can consider the illustrations on the right in Figure 2.1(a & b). Such an image filtering example would



0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

(a) Laplacian Filters

(b) 4-connected and 8-connected graph

Figure 2.1: Analogy between Laplacian image kernels and Laplacian of graph topologies

be transposed to graph domain by performing the operation in (2.4), where f contains the pixel values.

As a differential operator, the graph Laplacian has a crucial role for encoding the physical properties of the input space, which will be explained in the next section.

2.2.2 Laplacian Eigensystems and Graph Spectrum

Defining the graph Laplacian is a significant step in practicing the differential geometry on graphs. In this point of view, the analogy between the spectral analysis of the Riemannian manifolds and the spectral graph theory is quite compelling [14]. In certain conditions, the graph representation is considered as the proxy for the manifold underlying the sampled data. There are important studies for the approximation of the Laplace-Beltrami operator on manifolds by the graph Laplacian [15], [16], [17]. Moreover, the eigenvectors of the graph Laplacian are affirmed to be natural discretization of the eigenfunctions of the Laplace-Beltrami operator on manifolds [18]. In this sense, the Laplacian based manifold methods can be transcribed to the graph Laplacian based methods [7], [19], [20]. Since the Laplacian eigensystem unveils the spectral identity of the data manifold, we exploit this information later on inferring coarse-to-fine characteristics thoroughly on the underlying domain. The eigenvalues of the Laplacian operator correspond to the characteristic frequencies of the structure [13], which is manifested by the below eigenvalue problem or the Helmholtz equation:

$$\Delta f = -\lambda f \tag{2.5}$$

Since the Laplacian matrix L is real, symmetric and positive-semidefinite, it has a complete set of orthonormal eigenvectors and non-negative, real eigenvalues. For a connected graph, the eigenvalues of L are sorted as $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$, which is called as the *graph spectrum*.

The non-negativity of the Laplacian operator is expressed by the *graph Laplacian quadratic form* [12] defined as,

$$f^T L f = \sum_{(i,j)\in\mathcal{E}} W_{ij}[f(i) - f(j)]^2 \geq 0 \tag{2.6}$$

The quadratic form of Laplacian matrix, $f^T L f$, holds a kind of global notion of smoothness in a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ [1], which will be exploited in the next sections. One can calculate a global smoothness measure by computing the weighted sum of squared differences of the neighbors defined by the edge set $\mathcal{E}$ of the graph as follows,

$$\sum_{(i,j)\in\mathcal{E}} W_{ij}[f(i) - f(j)]^2 \tag{2.7}$$

Hence, how small the sum results signifies how smoothly the function f is changing on the vertices of the graph. Since the sum is calculated over the edge set of the graph, we can rearrange it as follows,

$$\sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}_i} W_{ij}[f(i) - f(j)]^2 \tag{2.8}$$

If we expand the squared term, we get,

$$\sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}_i} \underbrace{W_{ij}[f(i)(f(i) - f(j)]}_{(1)} + \underbrace{W_{ij}[f(j)(f(j) - f(i))]}_{(2)}$$

Let us analyze term (1) and term (2) separately. Starting with (2), we can divide it into two parts as well,

$$\sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}_i} W_{ij} f(j) f(j) - \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}_i} W_{ij} f(j) f(i)$$

$$\sum_{i\in\mathcal{V}} \underbrace{\sum_{j\in\mathcal{N}_i} W_{ij} f(j)^2}_{[W(f\odot f)]_{(i)}} - \sum_{i\in\mathcal{V}} f(i) \underbrace{\sum_{j\in\mathcal{N}_i} W_{ij} f(j)}_{[Wf]_{(i)}}$$

where $\odot$ stands for elementwise product for a vector. When we analyze them further by knowing that weight matrix, W, is a symmetric matrix, we see that both parts are equal to $f^T W f$, quadratic form of the weight matrix, hence they negate each other.

Now let us analyze term (1). We can displace the multiplier $f(i)$ out of the inner sum since it is constant with respect to the inner sum,

$$\sum_{i \in \mathcal{V}} f(i) \underbrace{\sum_{j \in \mathcal{N}_i} W_{ij}[(f(i) - f(j)]}_{Lf(i)} = f^T(Lf)$$

We already know the expansion stated in (2.4), which is equal to the inner sum in the above equation. Therefore, term (1) can be written in matrix vector multiplication form, which turns out to be the quadratic form of Laplacian matrix.

2.3 Dimension Reduction and Spectral Clustering

The spectral embedding exhibited by the graph representation empowers useful mathematical tools for nonlinear dimensionality reduction and clustering. In this section, we analyze the practice of spectral graph theory for the embodied problems below.

2.3.1 Graph Based Dimensionality Reduction

When data is expected to occupy a manifold of lower dimensionality than the original high dimensional space, it is crucial to explore the low-dimensional embedding of the data set in order to simplify the data analysis process. Graphs are adequate for representing the high dimensional data by defining similarities inside the local neighborhoods of the data geometry and accommodate powerful tools for nonlinear dimensionality reduction methods. During the last decades, important nonlinear dimensionality reduction algorithms are proposed by the help of graph based techniques, such as Isomap and Local Linear Embedding (LLE) [21]. In this chapter, we concentrate on the one proposed by Belkin et al., namely Laplacian Eigenmaps [22].

From a mathematical point of view, the dimensionality reduction problem can be defined as the mapping of the k points $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_k}\}$ in $\mathbb{R}^l$ to a set of points

$\{\boldsymbol{y_1}, \boldsymbol{y_2}, \cdots, \boldsymbol{y_k}\}$ in $\mathbb{R}^m$ where $m \ll l$. Note that, here the bold notation is used for the vectors such as $\boldsymbol{y_1}$ and the regular letters denote the vector elements as y_1 stands for the first element of $\boldsymbol{y}$. To serve the dimensionality reduction problem, after a weighted graph is constructed to represent the high dimensional data set, the Laplacian matrix is computed subsequently by $L = D - W$. It is desired to find the optimal mapping in the lower dimensional space.

Let us start with the one dimensional case where the sample set in $\mathbb{R}^l$ is aimed to be mapped to a set of points on a line $\boldsymbol{y} = [y_1, y_2, \cdots, y_k]^T$. The connected points on the graph are desired to be positioned as close as possible on the lower dimensional mapping. Hence, the following objective function is minimized to obtain the optimal dimensionality reduction into one dimension,

$$\sum_{(i \sim j)} W_{ij}(y_i - y_j)^2 = \boldsymbol{y}^T L \boldsymbol{y} \tag{2.9}$$

which is identical to the quadratic form of Laplacian for the new representation vector $\boldsymbol{y}$. In this regard, we can interpret the graph Laplacian quadratic form as a measure for preservation of the neighborhood. In other words, quadratic form of Laplacian is smaller when the connected points with large weights (i.e. more similar) are mapped closer in this new representation [1]. Such an approach is the main principle of the nonlinear dimensionality reduction problems, known as *preservation of locality*.

In this formulation, a constraint $\boldsymbol{y}^T\boldsymbol{y} = 1$ must be introduced to fix the scale of the vector. As a consequence, the following optimization problem is obtained:

$$\underset{\boldsymbol{y}}{\operatorname{argmin}}\, \boldsymbol{y}^T L \boldsymbol{y} \quad \text{subject to} \quad \boldsymbol{y}^T \boldsymbol{y} = 1$$

One can obtain the same optimization after a conversion by the variational method or the min-max theorem,

$$\underset{\boldsymbol{y}}{\operatorname{argmin}} \left. \frac{\boldsymbol{y}^T L \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{y}} \right|_{\boldsymbol{y}^T \boldsymbol{y} = 1}$$

The above minimized ratio is called *Rayleigh quotient*, which is denoted by $\mathcal{R}(L, \boldsymbol{y})$ and it is bounded with the spectral content of the graph Laplacian $\{0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{k-1}\}$,

$$\lambda_0 \leq \mathcal{R}(L, \boldsymbol{y}) \leq \lambda_{k-1}$$

The Rayleigh quotient of an eigenvector is its associated eigenvalue. As a result, the solution is originated by the eigenvalue problem for the Laplacian operator,

$$L\boldsymbol{y} = \lambda \boldsymbol{y}$$

The trivial solution associated with $\lambda = 0$ is discarded, since it yields a constant vector solution $\boldsymbol{y} = \mathbf{1}$, which maps all the vertices to the same point on 1-D line. Therefore, another constraint is inserted for solution to be orthogonal to the constant vector $\boldsymbol{y}^T\mathbf{1} = 0$.

It can be shown that the solution vector $\boldsymbol{y}$, which minimizes the Rayleigh quotient, is given by the minimum eigenvalue of the eigenvalue problem, where $\lambda \neq 0$. In other words, it will be the eigenvector corresponding to λ_1.

In order to get an m-dimensional mapping, where each sample on the original dataset is represented by an $(m \times 1)$ vector, one has to pick up the eigenvectors associated with the m-lowest eigenvalues. If we denote this embedding with a $(k \times m)$ matrix $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m]$, which is composed by stacking those eigenvectors, then it minimizes the following objective function, similar to the one described in (2.9),

$$\sum_{i,j} \|Y^{(i)} - Y^{(j)}\|^2 W_{ij} = tr(Y^T L Y) \tag{2.10}$$

where i^{th} row of Y is expressed as $Y^{(i)} = [\boldsymbol{y}_1(i), \boldsymbol{y}_2(i), \cdots, \boldsymbol{y}_m(i)]$, and it represents the data sample $\boldsymbol{x}_i$ on the original dataset.

2.3.2 Graph Based Clustering

Prior to examining the spectral clustering technique, the graph theoretical segmentation begins with the basic graph partitioning methods, such as graph cuts. Such problems are stated to be on NP-hard, therefore, in [23], Shi & Malik reformulated the normalized cut criteria by reducing the solution to finding the second eigenvector of the normalized graph Laplacian on a generalized eigenvalue problem $L\boldsymbol{y} = \lambda D\boldsymbol{y}$. In this formulation, the second eigenvector refers to the eigenvector associated with the second smallest eigenvalue, the reader is referred to the previous part in order to understand the reason for eliminating the smallest eigenvalue. They are followed with spectral clustering by allowing a relaxation on the eigenvector space [24].

In the previous formulation of the dimensionality reduction problem, the connected points on the data set are desired to be mapped as close as possible. Therefore, the Laplacian quadratic form on the transformed data is minimized, since it constitutes a measure for preservation of locality. This relation indicates that the graph Laplacian quadratic form gets smaller for the solution vectors corresponding to the smaller eigenvalues on the graph spectrum. Reinterpreting this situation, one can claim that locality is much more preserved by the eigenvectors corresponding to the smaller eigenvalues. In other words, those eigenvectors are varying slowly through the connected vertices of the graph structure, which can also be observed on Figure 2.2.

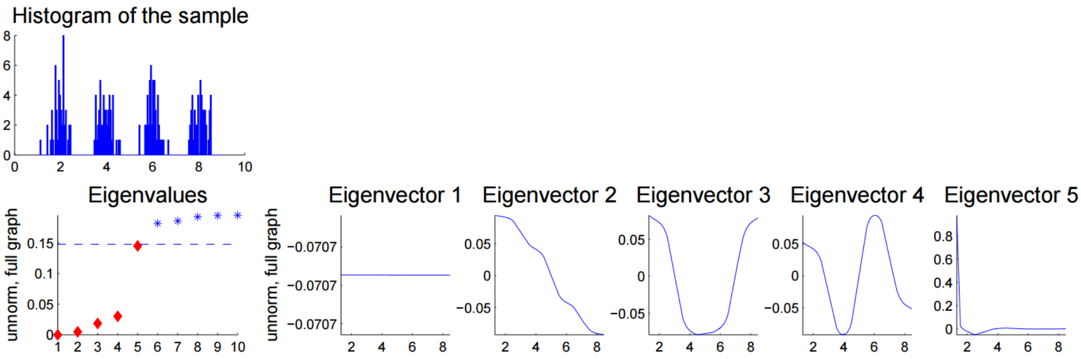


Figure 2.2: First row: Histogram of mixture of Gaussians on $\mathbb{R}$, Second row: Eigenvalues and eigenvectors of unnormalized Laplacian, `Adapted from [25]`

In Figure 2.2, a connected graph is constructed from 200 random samples which are taken from a mixture distribution of four Gaussians. The first four eigenvectors constitute stable information for clustering these four Gaussian dataset since the difference between the fourth and the fifth eigenvalues are relatively large. This heuristic is known eigengap [25, 24], and, the number of clusters and the eigenvectors to be used in clustering procedure can be recognized by searching for the eigengap, as shown in the plot of eigenvalues in Figure 2.2. The assumptions on the cohesiveness of the clusters and the stability of the eigenvector set are explained in [25, 24] in detail to justify the eigengap proposition.

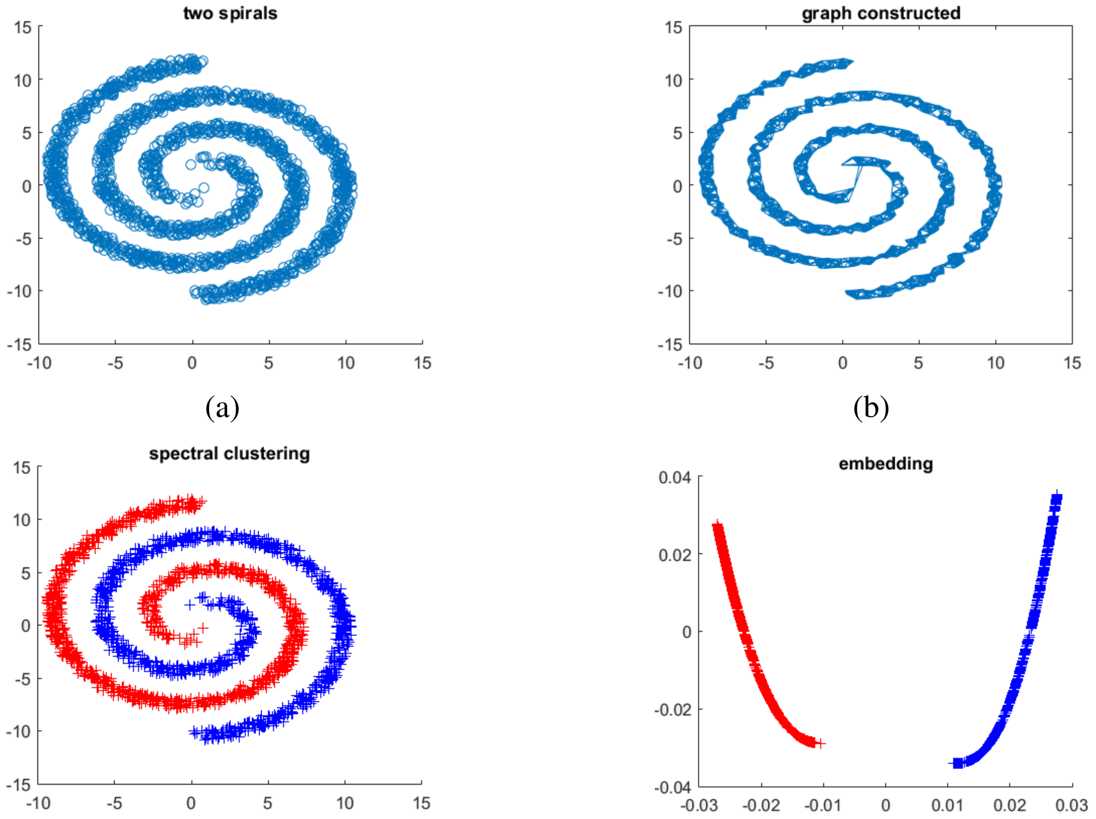
After revealing the spectral identity of the data space, clustering is performed based on the slowly varying components of the graph structure, since they designate a consistent set of characteristics for the embedding. Within the complete graph spectrum, a set of eigen components is favored and the data analysis is accomplished on this small but consistent set.

In [25], an algorithm is presented for the final clustering by using n eigenvectors simultaneously. Based upon the presented dimensionality reduction approach, the following algorithm is given,

```
Algorithm 1 Unnormalized Spectral Clustering [25]
Input: $k$ point dataset $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_k}\}$ in $\mathbb{R}^l$ to be segmented into $n$ clusters
1: Construct weight matrix $W \in \mathbb{R}^{k \times k}$
2: Compute the Laplacian matrix $L = D - W$
3: Compute the first $n$ eigenvectors $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n$ of $L$
4: Let $Y \in \mathbb{R}^{k\text{x}n}$ matrix of stacked eigenvectors
5: Let $u_i \in \mathbb{R}^n$ be $i$-th row of $Y$
6: Cluster the $k$ points $\{u_i\}_{i=1,\cdots,n}$ in $\mathbb{R}^n$ into clusters $\{C_1, \cdots, C_n\}$ via K-means algorithm
7: Assign point $\boldsymbol{x_i}$ to cluster-$j$ if and only if $u_i$ was assigned to $C_j$
```

By maintaining the locality on the new mapping, the graph-based dimensionality reduction methods naturally provide the clustering solutions [22]. Indeed, developing on the dimensionality reduction concept, the given algorithm intends to discover the clusters properly by changing the representation, so that K-means algorithm can detect the clusters in the new representation [25].

When the dataset lies already in a low dimensional space, yet it is complicated in terms of connectivity, the linear dimensionality reduction or discrimination techniques might fail to comprehend the intrinsic data structure. In that case, one may even prefer to augment the dimension of data analysis in order to qualify the spectral reasoning. Figure 2.3 illustrates such an example. In this example, each sample point is taken as a node on the graph, and the weights are determined in terms of a Gaussian function of the Euclidean distance between the samples. The two intertwined spiral arms constitute one complicated structure for any linear partitioning method. On the other hand, with a proper connected graph encoding, as shown in 2.3(b), the connectivity of the dataset can be captured. By this mean, a proper segmentation can be accomplished, as in 2.3(c), through practicing a k-means clustering on the spectral embedding displayed in 2.3(d).



(a) (b) (c) (d)

Figure 2.3: (a) Original "Two Spiral" data set (b) Constructed graph topology (c) Clustering on "Two Spiral" (d) 2-dimensional spectral embedding

2.4 Conclusion

In this chapter, we introduce the spectral graph theory which models the geometrical structure of discrete input spaces. The weighted graph structure automatically captures the intrinsic relationship between the data samples so that it embeds the geometrical domain of the data to the graph representation. We reviewed how one can benefit from the graph based spectral analysis in dimension reduction and clustering problems.

We presented widely accepted dimensionality reduction and clustering frameworks based on the spectral graph theory. Comprehension of the underlying manifold illuminates the methods for processing the signals residing on those manifolds, which is the main focus of Chapter 2. We will mention the details on the construction of the topology and selection of the weight function in Chapter 4, since those procedures vary according to the application. In addition, the theory presented here will be exploited by some clustering experiments on the 3D point cloud data.

CHAPTER 3

AN OVERVIEW OF GRAPH SIGNAL PROCESSING

3.1 Introduction

Graph-like structures may appear in many data forms. Employing machine learning and signal processing techniques on those data forms can be challenging unlike the traditional methods developed for the data structures defined on lattices or other regular domains. Identifying the structure of those data forms could be the first step to achieve such methods, which is accomplished by describing the statistical relationships between the sample points, prompted by the geometrical domain of the data. The second step is to exploit the recognized structure for performing the signal processing operations.

Graph signal processing both explores the intrinsic geometric structure of data space and investigates the discrete data signal on the recognized domain at the same time. Therefore, it combines the expertise on algebraic and spectral graph theory with computational harmonic analysis [1].

In this chapter, we first review the spectral analysis of weighted graphs for which we define two graph domains: *vertex domain* and *graph spectral domain*. Through those definitions, we give the description of a graph signal and how we perceive it on spectral domain through a new type of Fourier transform on graphs. Building upon the spectral domain representation of a graph signal, we introduce important operations on graph signals, such as filtering and localization, which also gives us the opportunity of performing multi-resolution analysis on graph signal.

3.2 Review of Vertex-Frequency Analysis on Graphs

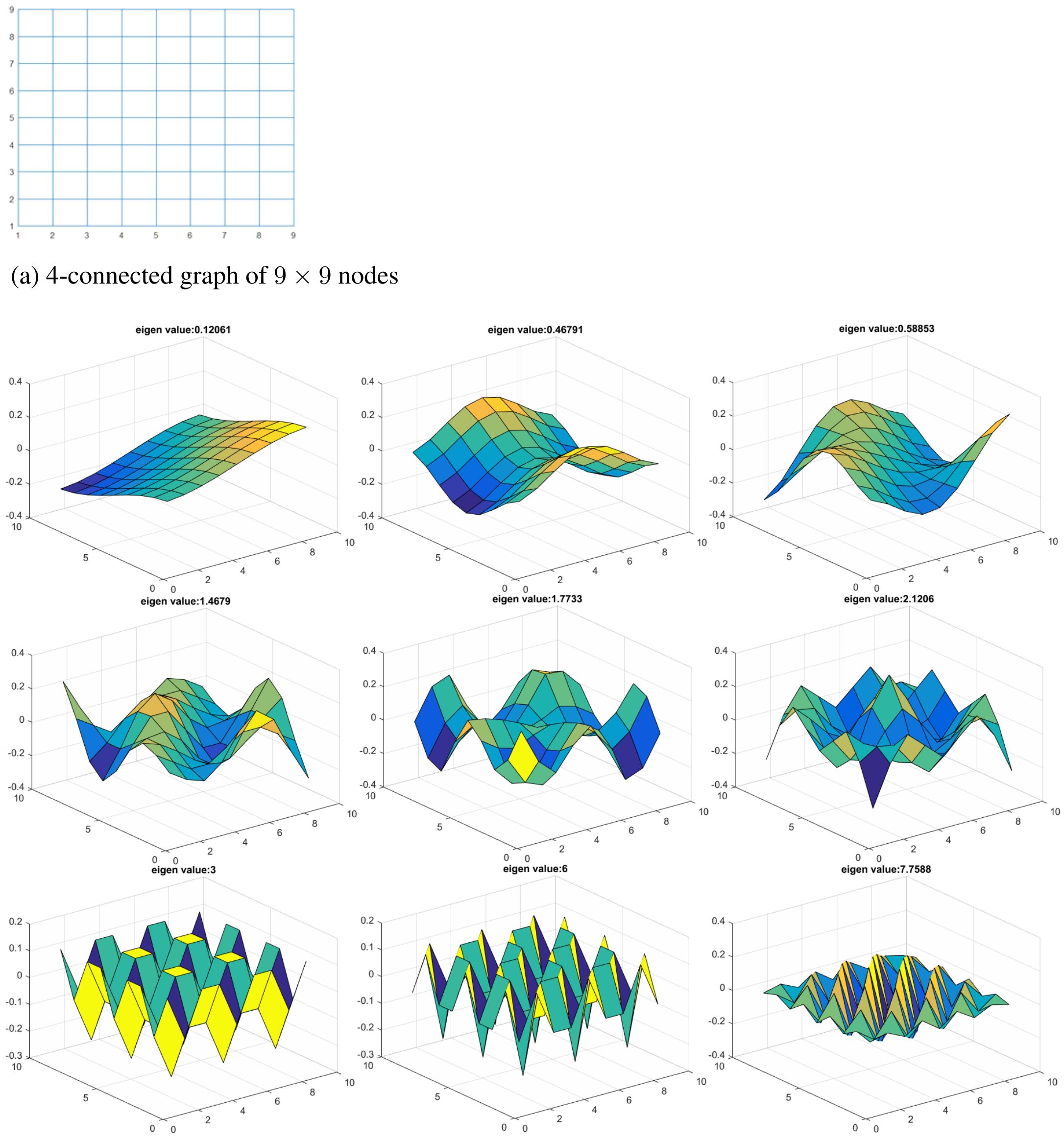
In the previous chapter, we conclude that the spectral decomposition of the graph Laplacian yields the characteristic frequencies of the graph structure. The characteristic frequency components imply that the eigenfunctions associated with the lower eigenvalues are slowly changing and, the ones associated with the higher eigenvalues are changing faster through the graph vertices. In other words, we may have the notion of the frequency components of the underlying geometry.

For a weighted and undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, $\mathcal{V}$ and $\mathcal{E}$ denote the vertex and edge set with $|\mathcal{V}| = N$ and the Laplacian matrix is given by $L = D - W$ where W is weight (or similarity) matrix and D is its degree matrix. Since the Laplacian matrix is real, symmetric and positive-semidefinite, it has a complete set of orthonormal eigenvectors $\{\boldsymbol{e}_l\}_{l=0,1,\cdots,N-1}$ and non-negative, real eigenvalues as explained before. For a connected graph, the eigenvalues of L are sorted as $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$ which is called graph spectrum. The graph Laplacian quadratic form, in 2.6, accommodates a global smoothness measure on a given graph setting $\mathcal{G}$. Moreover, the quadratic form of Laplacian acting on an eigenvector of a graph corresponds to its associated eigenvalue,

$$\lambda_l = \boldsymbol{e}_l{}^T L \boldsymbol{e}_l = \sum_{(i,j)\in\mathcal{E}} W_{ij} [\boldsymbol{e}_l(i) - \boldsymbol{e}_l(j)]^2$$

It should be noted that for the equation above, the smaller the Laplacian quadratic form, the smoother representation it generates due to weighted squared difference between connected elements of the representation. Accordingly, the graph Laplacian eigenvectors corresponding to lower eigenvalues are smoother compared to the ones associated with higher eigenvalues, which can also be seen on Figure 3.1. Therefore, the eigenvalues carry the notion of frequency.

Thus, we can analyze a graph through its spectral decomposition. Indeed, there are two approaches for graph analysis. One approach is applied in the vertex domain, which is interpreted as spatial domain in conventional signal processing sense, and the other is the spectral (or frequency) domain. In the next step, this equivalence is leveraged for the operations on the frequency domain of graph.



(a) 4-connected graph of 9×9 nodes

(b) Eigenvectors in an ascending order of eigenvalues

Figure 3.1: Eigenvectors of a 4-connected graph

3.3 Graph Signals and Graph Fourier Transform

Graph signal can be broadly defined as data existing on graph structures. A graph signal $\boldsymbol{f} \in \mathbb{R}^N$ is described on the vertices of a graph and indicated by a vector $\boldsymbol{f} = (f(v_1), f(v_2), \cdots, f(v_N))^T$. Graphs can be considered as representations of discrete sample points on manifolds [26, 27]. To build an analogy between signals on graphs and functions on manifolds, let us define an integrable function $f(x)$ on a manifold M, where x is a point on M. The function is expressed by $f(x) \in \mathcal{L}^2(M)$, which means the Hilbert space of the orthonormal and complete eigenfunctions $\{\phi_k\}_{k=0}^{\infty}$ of the Laplace-Beltrami operator, Δ_M, on the manifold M. As a result, we can write any function $f(x)$ in terms of the linear combination of the Laplace-Beltrami eigenfunctions, which is called *manifold harmonic transform* [28]. The Laplace-Beltrami eigensystem quite resembles the Fourier basis which is used for the classical Fourier transform of an integrable function $f(t)$,

$$\hat{f}(\xi) = \int\limits_{\mathbb{R}} f(t)\mathrm{e}^{-2\pi i\xi t}dt$$

Similarly, once the eigenfunctions of the graph Laplacian are obtained, any signal residing on the graph can be formulated by the eigensystem of the graph Laplacian, $\{e_k\}_{k=0}^{N-1}$. Then, this transform would be analogues to discrete version of Fourier transform.

In fact, Laplacian eigenfunctions provide a generalization of Fourier analysis on various domains [29]. In the traditional form, the Laplacian eigenfunctions are complex exponentials and their associated eigenvalues are the corresponding frequency values determining their oscillation rate, as explained in Table 3.1.

Table 3.1: Comparison of Fourier analysis in classical and graph settings

	Classical Setting	Graph Setting
Laplacian eigenvalue problem	$\frac{\partial^2}{\partial t^2}\mathrm{e}^{2\pi i\xi t} = (2\pi\xi)^2\mathrm{e}^{2\pi i\xi t}$	$L\boldsymbol{e}_l = \lambda_l\boldsymbol{e}_l$
Eigenfunction	$\mathrm{e}^{2\pi i\xi t}$	$\boldsymbol{e}_l$
Eigenvalue	$(2\pi\xi)^2$	λ_l
A function on the structure	$\boldsymbol{f} \in \mathcal{L}^2(\mathbb{R})$	$f \in \mathcal{L}^2(\mathcal{V})$
Fourier Transfom	$\hat{f}(\xi) = \langle f, \mathrm{e}^{2\pi i\xi t}\rangle$	$\hat{f}(\lambda_l) = \langle \boldsymbol{f}, \boldsymbol{e}_l\rangle$

As explained in the previous section, the graph Laplacian eigensystem carries notion of frequency, just like the Fourier basis. Consequently, the Fourier transform on graph setting is named *Graph Fourier Transform* (GFT) and defined as,

$$\hat{f}(\lambda_l) := \sum_{i=1}^{N} f(i)e_l^*(i) \tag{3.1}$$

By the help of GFT, one can obtain the spectral domain representation of a graph signal $\boldsymbol{g}$ which is already described on the graph vertex domain. There are also signals $\hat{\boldsymbol{g}}$ that are defined directly on the graph spectral domain, which are called *graph kernels*. For example, a *heat kernel* is defined by the relation $\hat{g}(\lambda_l) = C\mathrm{e}^{-t\lambda_l}$ and one can calculate the corresponding vertex domain representation special to a graph structure via inverse GFT. In Figure 3.2, vertex domain representations of two heat kernels with different spreading factor t, are calculated through the inverse GFT. The physical meaning of a heat kernel is explained in Appendix A.2. At the moment, note that as t gets larger, a smoother vertex domain representation is reached in 3.2(b), just as it can be reached by convolving the the initially given distribution in 3.2(a) with a low pass filter.

To be complete, the inverse graph Fourier transform (IGFT) is defined as,

$$f(i) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l) e_l(i) \tag{3.2}$$

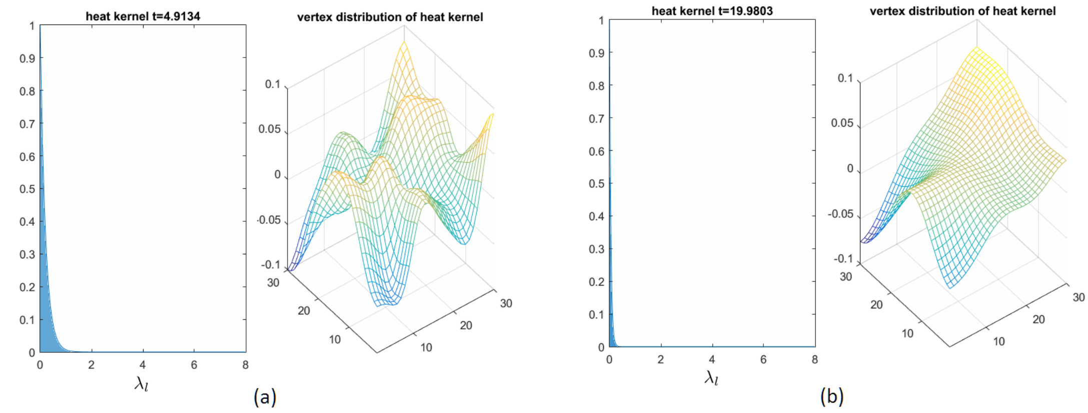


Figure 3.2: Heat kernel $\hat{g}(\lambda_l) = C\mathrm{e}^{-t\lambda_l}$ with two different spreading factor t and their vertex domain distributions computed by IGFT

3.4 Filtering and Convolution on Graphs

In classical settings, for two time domain signals f and g, the convolution operation is defined as,

$$h(t) = (f * g)(t) = \int\limits_{\mathbb{R}} f(\tau) g(t - \tau) d\tau$$

In graph settings, a shifted version of a signal $g(t-\tau)$ can not be defined in a simple way. In order to generalize the convolution in graph settings, frequency domain definition of convolution is employed,

$$h(t) = \int_{\mathbb{R}} \hat{h}(\xi)\mathrm{e}^{2\pi i\xi t} d\xi = \int_{\mathbb{R}} \hat{f}(\xi)\hat{g}(\xi)\mathrm{e}^{2\pi i\xi t} d\xi$$

which applies IGFT on the signal filtered in frequency domain. Similarly, frequency domain filtering in graph settings is expressed as follows and called *graph spectral filtering*,

$$\hat{f}_{out}(\lambda_l) = \hat{f}_{in}(\lambda_l)\hat{h}(\lambda_l) \tag{3.3}$$

Then, simply the inverse graph Fourier transform is applied to obtain the resulting signal in vertex domain,

$$f_{out}(i) = \sum_{l=0}^{N-1} \hat{f}_{in}(\lambda_l)\hat{h}(\lambda_l)e_l(i) \tag{3.4}$$

At this point, one can examine the aforementioned graph operations in terms of the matrix operations. To start with the Laplacian matrix, it is decomposed as $L = E\Lambda E^*$, where $E = [\boldsymbol{e}_0, \boldsymbol{e}_1, \cdots, \boldsymbol{e}_{N-1}]$ is the eigenvector matrix whose columns consist of the eigenvectors of graph Laplacian and Λ is the eigenvalue matrix whose diagonal entries are the corresponding eigenvalues, and then E^* is the Hermitian of E.

Once the graph signal $\boldsymbol{f}$ and its transform $\hat{\boldsymbol{f}}$ are considered as vectors in $\mathbb{R}^N$, the graph Fourier transform and its inverse are given in matrix-vector multiplication form,

$$\hat{f}(\lambda_l) = \sum_{i=1}^{N} f(i)e_l^*(i) \quad \Rightarrow \quad \hat{f} = E^* f$$
$$f(i) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l)e_l(i) \quad \Rightarrow \quad f = E\hat{f}$$

Returning back to the graph spectral filtering, the output signal f_{out} is computed by means of inverse graph Fourier transform of the frequency filtered signal, which can be given in matrix operational form in the following way,

$$f_{out} = \underbrace{E\,\underbrace{(\hat{\boldsymbol{h}} \odot \underbrace{(E^*\boldsymbol{f}_{in})}_{\text{GFT}})}_{\text{frequency filtering}}}_{\text{IGFT}}$$

where $\odot$ stands for the element-wise multiplication operation of the vectors, which is used for pointwise product of frequency components during frequency filtering in (3.3). The element-wise multiplication can be performed in matrix operation by putting the vector $\hat{\boldsymbol{h}}$ in diagonal matrix form $\hat{h}(\Lambda)$ and then taking the product of $\hat{h}(\Lambda)$ and $\hat{\boldsymbol{f}}_{in}$ as follows,

$$\boldsymbol{f}_{out} = E\ (\hat{h}(\Lambda)\ \hat{f}_{in}) = E \begin{bmatrix} \hat{h}(\lambda_0) & & & \mathbf{0} \\ & \hat{h}(\lambda_1) & & \\ & & \ddots & \\ \mathbf{0} & & & \hat{h}(\lambda_{N-1}) \end{bmatrix} E^* \boldsymbol{f}_{in}$$

Since the Laplacian matrix can be transformed into a diagonal form, any function $\hat{h}(\cdot)$ applying to the Laplacian matrix can be written in matrix function form,

$$\hat{h}(L) = E\ \hat{h}(\Lambda)\ E^* \tag{3.5}$$

Therefore, filtering of a graph signal can be given in matrix notation accordingly,

$$\boldsymbol{f}_{out} = \hat{h}(L)\boldsymbol{f}_{in} \tag{3.6}$$

Since graphs are irregular structures, the result of the graph spectral filtering on the vertex domain representation cannot be simply conceived. Assume that the graph spectral filtering operation in (3.6) has only effect on K-hop neighborhood on vertex domain, where the K-hop neighborhood of a vertex i is denoted by $\mathcal{N}(i, K)$ and conveys the reachable vertex set by crossing at most K number of edges starting from vertex i. Such a vertex domain filtering can be formulated as,

$$f_{out}(i) = \sum_{j \in \mathcal{N}(i,K)} \beta_{i,j} f_{in}(j) \tag{3.7}$$

In that case, one can expect that the entries of $\hat{h}(L)$ matrix in (3.6) correspond to the coefficient terms $\{\beta_{i,j}\}$ in the vertex domain filtering operation such as,

$$[\hat{h}(L)]_{ij} = \begin{cases} \beta_{i,j} & \text{if } j \in \mathcal{N}(i, K), \\ 0 & \text{otherwise.} \end{cases}$$

Let L^k denote k^{th} power of Laplacian matrix. Through the graph theory, one can show that $[L^k]_{ij}$ can be nonzero, only if there is a path in graph network that connects the vertex i to vertex j in k step. In other words, if vertex j is not in k-hop neighborhood of vertex i, then $[L^k]_{ij} = 0$ [6]. Exploiting this information, $\hat{h}(L)$ can be written in terms of K degree polynomial expansion of the Laplacian matrix, since it is only effective in K-hop neighborhood,

$$\hat{h}(L) = \sum_{k=0}^{K} \alpha_k L^k \tag{3.8}$$

K-hop filtering of a graph signal in vertex domain can be expressed once again using the polynomial function of graph Laplacian,

$$f_{out}(i) = \sum_{j=1}^{N} f_{in}(j) \sum_{k=0}^{K} \alpha_k [L^k]_{ij} \tag{3.9}$$

which actually accomplishes a localized operation in the vertex domain that is exploited later.

3.5 Localization and Translation on Graphs

Unlike the data structures defined on regular domains, the graph signals lack a shift-invariance notion which makes the translation of a graph signal from a vertex to another ambiguous. However, we can obtain new localized transform of the graph signal by means of centralizing the spectral content of it.

In fact, the localized graph signal filtering approach in (3.6) is not far from the traditional convolution application which is followed by "flip-shift and inner product". Each column (or row) of matrix $\hat{h}(L) = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_N] \in \mathbb{R}^{N \times N}$ can be considered as a graph signal vector translated on the associated vertex. During the the matrix-vector product, we take the inner product of a row, which is the shifted version of the originating graph signal $\boldsymbol{h}$, with the graph signal vector to be filtered $\boldsymbol{f}_{in}$ and then we write the result to the corresponding index of the output graph signal vector,

$$\boldsymbol{f}_{out}(i) = [\hat{h}(L)\boldsymbol{f}_{in}](i) = \langle \boldsymbol{h}_i, \boldsymbol{f}_{in} \rangle \tag{3.10}$$

To analyze further the columns of matrix $\hat{h}(L)$, let us define an impulse signal on the graph denoted by $\boldsymbol{\delta}_i \in \mathbb{R}^N$, which expresses a graph signal taking a nonzero value only at vertex i. We can obtain a column vector of $\hat{h}(L)$ by simply multiplicating it with an impulse vector as follows,

$$\begin{aligned}\boldsymbol{h}_i &= \hat{h}(L)\boldsymbol{\delta}_i \\ &= E\hat{h}(\Lambda)E^*\boldsymbol{\delta}_i \\ &= \underbrace{E\,\underbrace{(\hat{\boldsymbol{h}} \odot \underbrace{(E^*\boldsymbol{\delta}_i)}_{\text{GFT of impulse}}\ )}_{\text{frequency filtering}}}_{\text{IGFT}}\end{aligned} \tag{3.11}$$

A column vector of $\hat{h}(L)$ appears as a signal derived by the graph spectral filtering of the originating signal $\boldsymbol{h}$ with an (shifted) impulse. Similarly, in classical settings, a time domain signal $f(t)$ to be translated is convolved with an impulse centered at the location of shifting, such as $f(t-u) = (f * \delta_u)(t)$.

As a consequence, to localize a graph kernel $\hat{h}(\lambda)$ at vertex i, one can get the pointwise product of a kernel $\hat{\boldsymbol{h}}$ and i^{th} column of E^*, and then take the inverse graph Fourier transform. A column of matrix E^* is the frequency domain transform of an impulse. Therefore, (3.11) can be written in the summation form in order to define generalized translation operator on graphs, $T_i : \mathbb{R}^N \to \mathbb{R}^N$,

$$(T_i h)(j) = \sqrt{N} \sum_{l=0}^{N-1} \hat{h}(\lambda_l) e_l^*(i) e_l(j) \tag{3.12}$$

where the normalizing constant $\sqrt{N}$ is used for preserving the mean of the signal after the translation operator.

Note that the graphs are irregular structures, which may result in complicated frequency components. Therefore, this method should not be considered as translating a graph signal to a vertex in general, but rather as localizing the spectral content of a graph kernel to a desired vertex. Nonetheless, performing this, we may obtain a shifting effect as illustrated for a heat kernel on a 4-connected graph in Figure 3.3.

In addition to the localization, translation and filtering operations, it is possible to create dilated versions of a graph signal in both graph vertex domain and graph spectral domain, those are analogues of the multi-resolutions applications in conventional signal processing. More on the multi-resolution approaches on graphs can be found in Appendix A.

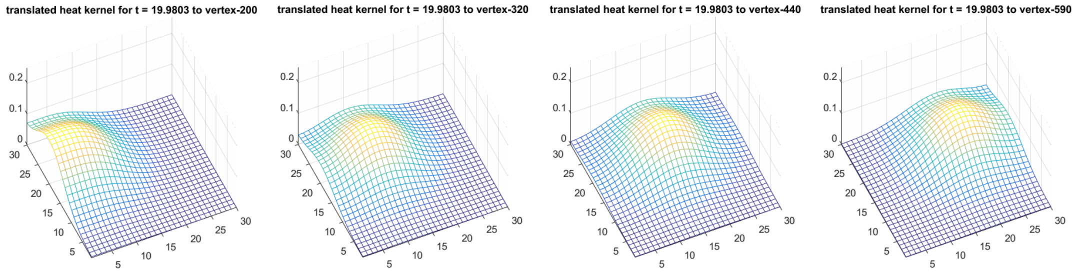


Figure 3.3: Translation of heat kernel $\hat{g}(\lambda_l) = C\mathrm{e}^{-t\lambda_l}$ on a 4-connected graph

3.6 A low-pass filtering application : Tikhonov Regularization [1]

A filtering example for noise reduction in graph signals is introduced in [1] by means of Tikhonov regularization. The theory presented here will be employed in Chapter 5 for the smoothing problem on 3D point clouds.

In Tikhonov regularization, the regularization term is the quadratic form of Laplacian, which penalizes the signal difference of the neighbor vertices with the factor of the weight of the link between them as follows,

$$f^T L f = \sum_{(i,j)\in\mathcal{E}} W_{i,j}[f(i) - f(j)]^2$$

where $f \in \mathbb{R}^N$ is the graph signal. Consequently, the optimization problem is denoted as,

$$\breve{x} = \underset{x}{\operatorname{argmin}}\, ||x - f||_2^2 + \gamma x^T L x \tag{3.13}$$

where we desire to estimate the smoothed signal $\breve{x}$, for a smoothing factor γ. The optimization problem is convex, and the solution can be simply calculated by taking the derivative of the objective function with respect to x and equating it to 0,

$$\breve{x} = (I + \gamma L)^{-1} f \tag{3.14}$$

The graph signal filtering operation is justified under matrix notation in (3.6), which can be shown in this case as follows,

$$\hat{h}(L) = E\hat{h}(\Lambda)E^* = (I + \gamma L)^{-1}$$

Here, recall that the decomposition of Laplacian matrix is given as $L = E\Lambda E^*$. Hence, $\hat{h}(\Lambda) = (I + \gamma\Lambda)^{-1}$ and the optimal reconstruction corresponds to filtering the graph signal with the following low-pass filter,

$$\hat{h}(\lambda) = \frac{1}{1 + \gamma\lambda} \tag{3.15}$$

3.7 Conclusion

In this chapter, the Fourier analysis obtained by the Laplacian operator on graph structures is examined. Graph spectral domain is introduced through GFT and it is explained how it is leveraged for defining filtering and translation operations on graph signals. Finally, a low-pass filtering example based on the Tikhonov regularization is given, which will be employed in Chapter 5 for smoothing of 3D point cloud data. All things considered, this chapter points out the significance of the localized versions of a graph signal in spatial and spectral fashion in terms of manifesting the multi-resolution approaches mentioned in Appendix A. Hence, it stimulates the ideas on some classification problems defined on the 3D point cloud data.

CHAPTER 4

GRAPH REPRESENTATION OF AIRBORNE LIDAR DATA AND ITS SEGMENTATION BY SPECTRAL CLUSTERING

4.1 Introduction

The computer vision problems defined on 3D data are highly challenging when the sampling is irregular. LIDAR (Light Detection and Ranging) instrument is one of the devices which produce unorganized 3D point clouds by scanning the environment. Segmentation is one of the fundamental tasks defined on point clouds acquired by LIDAR and it is significant for locating the point of interest objects on the scene such as man-made objects and separating the scene into complete parts in terms of shape. We address the problem of segmentation of unstructured LIDAR point clouds via spectral clustering on the graph representation.

For irregular data structures, the topology of the data domain and the relationship model between the data points are not directly given things in general, which may be the main limitation in graph representations [1]. First of all, there is no optimal way to decide the connection topology. Second, the weight function and the parameters it acts on might be critical depending on the operations to be deployed on the graph. Therefore, we may need to infer the weights and especially the edges of the graph by exploring physical stance of the data. Nonetheless, the graph based techniques are very convenient to conduct directly on the unstructured data.

Once a proper representation of a point cloud is embedded on a graph structure, we can develop a framework achieving a semantic interpretation of a point cloud scene.

The first step in this framework would probably be the segmentation of the scene. We may employ some clustering and segmentation operations on the graph representation in order to divide the scene into smaller regions which have a notion of interior physical completeness. Moreover, we simplify the complex operations to be executed further by focusing on those small set of regions.

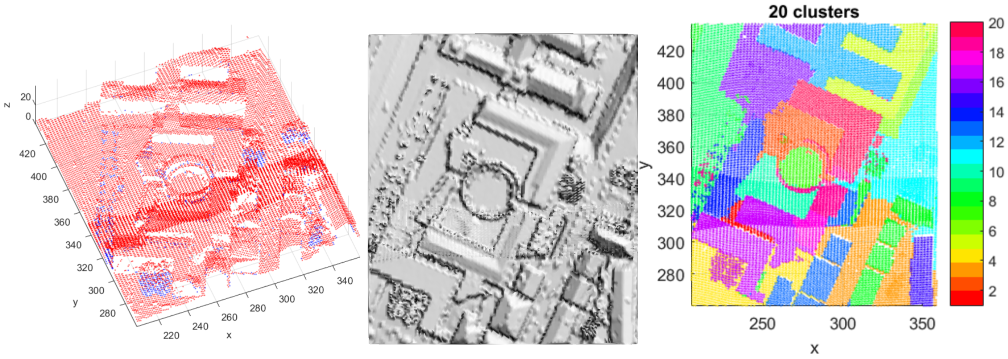
In this chapter, first we explain how one can represent a 3D point cloud in a graph structure and we give the options for parameters to be chosen on the graph structure. Then we experiment the spectral clustering techniques, whose theory is given in Chapter 2, on some LIDAR point cloud datasets.

4.2 Related Work

Airborne Laser Scanning (ALS) systems are very popular in urban applications and geospatial analysis since they provide highly accurate and dense data. An ALS system consists of a LIDAR device mounted on an aircraft and basically emits a laser beam and extracts a 3D point cloud by collecting the reflections back from the ground objects.

Besides the (x, y, z) world coordinates of a point, the second or last return of a laser point may be available. The second return mostly occurs, when a laser point is reflected by the ground underneath a tree by passing through the leaves or branches of the tree as demonstrated in Figure 4.1a. Additionally, some LIDAR data can also contain the reflection intensity of the laser points (More details on airborne LIDAR technology can be found in [30]).

In order to utilize the image processing tools readily, the early studies developed raster image forms of LIDAR data such as DEM (Digital Elevation Model) and DSM (Digital Surface Model) [2] using methods such as TIN (Triangular Irregular Networks)[3]. The range images derived from the LIDAR data are basically generated by resampling regularly over the XY plane which creates the image grid and the interpolated Z values are considered as pixel intensities. The interpolation artifacts occurring during this conversion often mislead the following processing stages, besides, valuable information attached to the 3D shape and geometric layout is lost [4, 5]. In Figure



(a) (b) (c)

Figure 4.1: (a) ALS data, first returns are shown in red and second returns are blue (b) DEM generated with the given ALS data (c) Segmentation result by the proposed spectral clustering algorithm
`Dataset is available on website of ISPRS WG III/3 [31]`
`DEM generated using LasTools [32]`

4.1b, distortions mostly appear around the intersection of two scan stripe, where the density of the samples are mixed up and very obscure. Another approach, whose implementers also aim direct use of image processing tools, is scan-line based approach [33, 34]. They perceive the airborne 3D point cloud as irregularly spaced 2.5D points where elevation values (z) can be defined as a function of x and y [35, 36]. This method first detects the line segments by knowing that a scan line on a 3D plane corresponds to a 3D line. Then it merges the detected scan lines based on a proximity measure in a region growing sense. Therefore, each scan line corresponds to a row in the 2.5D range image. The main disadvantage of this method is its dependency on 2.5D grid model which is not applicable for the unstructured point clouds. In some of ALS data, the scan lines are generated by arranging the points in a tubular volume and it is mostly dependent on the preferred scan line direction and consequently the segmentation result depends on the orientation.

Hence, the main motivation of this study is to fully exploit the 3D geometry of the point cloud by establishing the processing methods directly on the raw data.

Numerous studies have been developed for segmentation of 3D point clouds [5] and substantial amount of work have focused on LIDAR point clouds [37]. First, there are region growing methods which select a number of seed points and enlarge over the neighborhood of points up to a certain criterion [38]. The numerical graph based methods can be adapted to the region growing approach as in [39]. Second, there are attribute based methods which make a selection of parameters for a group of points or an individual point by usually extracting geometric primitives such as surface normals [40], slopes [41], curvatures or eigenvalues of covariance matrix. Then, the clustering is performed on the parameter (or feature) space to obtain a segmentation on the point cloud by defining a similarity and neighborhood measure between the group of points [42, 43]. Moreover, the geometric primitive shapes are employed by the model based approaches to fit mathematical models for set of points [44]. RANSAC (Random Sample Consensus) is the method that is commonly embraced by these approaches, and first developed by Fischler et al. [45]. By combining the intensity or fused RGB values with the 3D point cloud data, the attribute based segmentation methods are maintained adopting the numerical graph based techniques in some studies [46, 47]. In addition, the classical graph partitioning methods are integrated by some studies to segment 3D point clouds, either by constructing the graph directly on 3D point data [48, 49] or on a feature space [50]. In [48], min-cut method is applied for foreground/background segmentation at a given object location and background radius, which is utilized for background penalty term within the cut cost. Furthermore, in [50], graph-cut is employed for labeling 3D points in airborne LIDAR scene as points belonging to a surface or a scatter. Similarly, it is used for categorizing the points in a scene as tree or non-tree points in [49]. That is to say, the graph based methods are observed to be optionally embraced both in the region growing based approaches and the clustering based segmentation approaches in the literature. In terms of the aforementioned categories, our proposed algorithm can be counted under the clustering based segmentation techniques.

There are few studies pointing out the spectral embedding of the 3D point cloud data [51, 52], even though they do not address the exact problem in our case, which is the segmentation of an airborne LIDAR scene. In [53], spectral clustering is employed on 3D point cloud data by determining the neighborhood of points in graph using surface normals to overcome the problem of close-by surface sheets, which could be valid for 360° laser scanners. However, it does not likely occur in airborne LIDAR data which should be rather considered as irregular sampled range data since the scanning is made in downward direction constantly.

In this study, we practice the spectral clustering to obtain the elementary segments on an aerial LIDAR scene by constructing an adequate graph representation of the uneven sampled 3D point data.

4.3 Graph Representation of ALS Data

For the unstructured data, the neighborhood information is not inherent unlike the one presented by the uniform sampled data structures. The performance of the operations on an inferred graph representation may crucially be dependent on the encoded locality. Different heuristics have been adopted for graph formulations depending on the application. Some studies employ the k-nearest neighborhood to determine the edges of graph [48, 54], while some others use ϵ - neighborhood [55], that is the neighborhood constructed within the point volume of ϵ-radius sphere centering the subject point. The former is the standard choice for irregularly spaced data to avoid the disconnected points, nonetheless, disconnected components may still appear. Therefore, k has to be sufficiently large to conform the whole dataset in one connected component, in turn, the choice of k is specific to the dataset.

The weights of a graph store the similarity measures of the connected points, thus it should be inversely related to the distances of the edges. In general, the weight function is chosen as a Gaussian function of the Euclidean distances between the points. It is a standard choice to model the similarities between entities as Gaussian processes.

When we imagine the airborne LIDAR data as non-uniformly sampled range data, then we can adopt an image segmentation approach in terms of its graph representation. In notable graph based image segmentation studies [56, 57], pixels of an image are regarded as the nodes of the graph and the distances of the edges are computed as the pixel differences, where the lattice structure of image grid is leveraged for the graph network. Similarly, many graph based approaches in 3D point cloud field [48, 53], have considered 3D points as the vertices of the graph and they have selected the weights in terms of the Euclidean distance between them. In addition, they have determined the neighborhood connections based on the same measure by embedding the 3D geometry directly to the graph representation. Noting that the ALS data is aerial range data, it is more plausible to encode the relationships in terms of the range difference rather than the Euclidean distance between the 3D feature vectors. This preference encourages a more discriminative characteristic on the graph representation for segmentation application. A similar approach is employed in [47] by weighting the impact of feature elements in the distance metric calculated between the nodes. They determined both neighborhood model and the weights by assigning the highest emphasis on z data and the lowest on the intensity data in their 4-tuple feature vector $[x, y, z, i]$ which constitutes 3D coordinate and intensity value of each point. In our case, the intensity values are disregarded, since they are not discriminative enough because of the noise content, moreover, they may not always be available. In our graph representation, there is equal impact of point vector elements $[x, y, z]$ on the neighborhood model, yet the weight function is purely determined by the z difference of neighboring points. We still prefer the graph connections to depend on the 3D Euclidean locality, in order to retain the 3D topology of the underlying data.

We represent the 3D point cloud data as a weighted and undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V}$ is the set of nodes , $\mathcal{E}$ is the set of edges. The weight matrix is constructed as follows,

$$W_{ij} = \begin{cases} exp(-\frac{(z(i)-z(j))^2}{2\sigma^2}) & \text{if } v_j \in \mathcal{N}_{kNN}(v_i), \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where $\mathcal{N}_{kNN}(v_i)$ stands for k nearest neighborhood of vertex i.

After constructing the graph, the classical spectral clustering algorithm, whose theory is covered in Chapter 2, can be employed. However, we adapt a more efficient spectral clustering algorithm above the classical one for the sake of computational cost. The next section explains the adapted spectral clustering algorithm.

4.4 Spectral Clustering on ALS Data

The spectral clustering is not based on convex partitioning unlike the traditional clustering algorithms [58], rather it depends on the spectral embedding of the sample space. Therefore, it performs better on the high dimensional and complex datasets in comparison to the traditional clustering techniques [25].

It achieves multiple clusters on the data set by employing a relaxation on classical graph partitioning methods [24]. Algorithms, such as min-cut and normalized cut, are equivalent to dividing the data space into two partitions through a linear clustering based on the eigenvector corresponding to the second smallest eigenvalue of Laplacian. In spectral clustering case, which provides multi-clustering, the relaxation on eigenvector space implies a clustering based on k number of eigenvectors corresponding to the k smallest eigenvalue. In graph-cut point of view, a graph is partitioned so that the graph edges linking different groups have lower weights whereas, the ones within a group have higher weights [25]. While spectral clustering tries to achieve the same goal, it determines the groups through a spectral relaxation procedure. In this regard, spectral clustering may end up with segments which are not connected and yet assigned to the same cluster due to their similarity on the spectral embedding. That is why, it is adapted by notable semi-supervised classification studies [59, 60]. All in all, it can be preferable just by the reason that it presents a simple linear algebra problem to solve [25] in comparison to the numerical graph partitioning methods.

The number of partitions is usually decided by resolving the eigengap heuristic introduced in [24]. Eigengap proposition searches for a gap in the trend of the first eigenvalues of Laplacian. To give an illustration, the number of clusters can be determined as l if the first $l-1$ eigenvalues, $\lambda_1, \lambda_2, \cdots, \lambda_{l-1}$, are small and close and λ_l is relatively larger and distant to that set.

4.4.1 Landmark Based Spectral Clustering

Segmentation task can be accomplished through the spectral clustering on the data space, nonetheless, this has some drawbacks to be employed on large scale ones. Despite the superiority of the spectral clustering above the traditional clustering techniques, it has some limitations in terms of computational complexity and memory use. In recent years there have been many efforts to overcome those problems [61, 62, 63, 64]. For instance in [61], the weight matrix is constructed as a sparse matrix by storing only the similarities belonging to k nearest neighbors in each column which effectively handles the memory problem for large datasets. Then, the Laplacian is calculated as a sparse matrix as well which brings the opportunity of employing a sparse eigensolver through Lancoz-Arnoldi factorization to shorten the computational time.

Others preferred to approximate the graph similarity matrix which degrades memory usage substantially and simplifies the eigen-decomposition problem at the same time [63, 64]. We adopt a landmark based spectral clustering (LSC) [63], which picks p number of sample points, those are called landmark points, and estimates the weight matrix using the relations between the entire dataset and the landmarks. For an N sample dataset, a $(p \times N)$ similarity matrix Z is created to accommodate the similarities between landmark points and the dataset. This intends to represent all the samples in terms of linear combinations of landmarks, which follows a sparse coding concept to compress original dataset. However, instead of directly coding the samples in terms of landmarks, it is claimed to be more practical achieving that representation directly based on the relationships with landmarks.

In the end, the weight matrix W is approximated as,

$$W = \hat{Z}^T \hat{Z} \tag{4.2}$$

where $\hat{Z}$ is calculated via a sequence of normalizations on Z, to equate the degree matrix of W to identity matrix I. By this means, the Laplacian matrix can be directly calculated as $I - W$. Consequently, retrieving the eigenvectors for l smallest eigenvalues of L will be equivalent to obtaining the ones corresponding to l largest eigenvalue of W [24], which means eigen-decomposition can be directly performed on W.

The ultimate goal is to perform the eigen-decomposition on $\hat{Z}\hat{Z}^T$, whose size is $(p \times p)$, instead of dealing with the eigen-problem of $(N \times N)$ matrix W. The key point lies in the singular value decomposition of $\hat{Z}$,

$$\hat{Z} = A \Sigma B^T \tag{4.3}$$

Here, each column of $(p \times p)$ matrix A consists of the eigenvectors of $\hat{Z}\hat{Z}^T$ and diagonal elements of Σ are the corresponding eigenvalues, those are obtained through the eigen-decomposition of $\hat{Z}\hat{Z}^T$. The columns of matrix B give us the eigenvectors of

$\hat{Z}^T\hat{Z}$ in other words W. Hence, after calculating A and Σ, matrix B can be achieved as follows,

$$B^T = \Sigma^{-1} A^T \hat{Z} \tag{4.4}$$

See Algorithm 2 for details on the flow and adaptation to our ALS data application.

Algorithm 2 Landmark based spectral clustering on ALS data [63]

```
Input: ALS point cloud with $N$ samples to be segmented into $l$ clusters
1: Generate $p$ landmark points with random selection
2: Construct $Z \in \mathbb{R}^{p \times N}$ by finding $r$ number of nearest landmarks to each data point, store their similarities and zero out the other relationships
3: Obtain matrix $\bar{Z}$ by normalizing the columns of $Z$
4: Compute $\hat{Z}$ by dividing each element in a row of $\bar{Z}$ to the square-root of row sum.
5: Compute $A = [a_1, \cdots a_l]$ and $\Sigma$ based on first $l$ eigenvectors of $\hat{Z}\hat{Z}^T$
6: Calculate $B^T = [b_1, \cdots, b_l]$ by $B^T = \Sigma^{-1} A^T \hat{Z}$
7: Employ k-means to get clusters considering each row of $B^T$ as a vector (Classical Spectral Clustering Finale)
```

As expressed in the previous section, the nearest landmarks are discovered based on the 3D Euclidean distance, however, the similarities calculated with the Gaussian function of range difference between nearest landmarks and data samples.

Classical spectral clustering for n data points has the computational complexity of $O(n^2)$ in graph construction part and $O(n^3)$ in eigen-decomposition part, whereas LSC requires $O(np)$ for building the graph and $O(p^3 + p^2 n)$ to calculate the eigenvectors of Laplacian. Accuracy of LSC is lower than the classical spectral clustering in general due to the approximation. As p approaches the number of whole samples, the algorithm approaches the classical spectral clustering, hence accuracy can be raised by incrementing p with a trade in timing. Similarly to the classical spectral clustering algorithm, the clusters can be achieved by employing the k-means method on the eigenvector set yielded by the approximated weight matrix.

4.5 Experimental Results

4.5.1 Utilized Datasets

First of all, we have experimented classical Spectral clustering on a sample data "France" presented by LasLab [32] which is demonstrated on Figure 4.2.

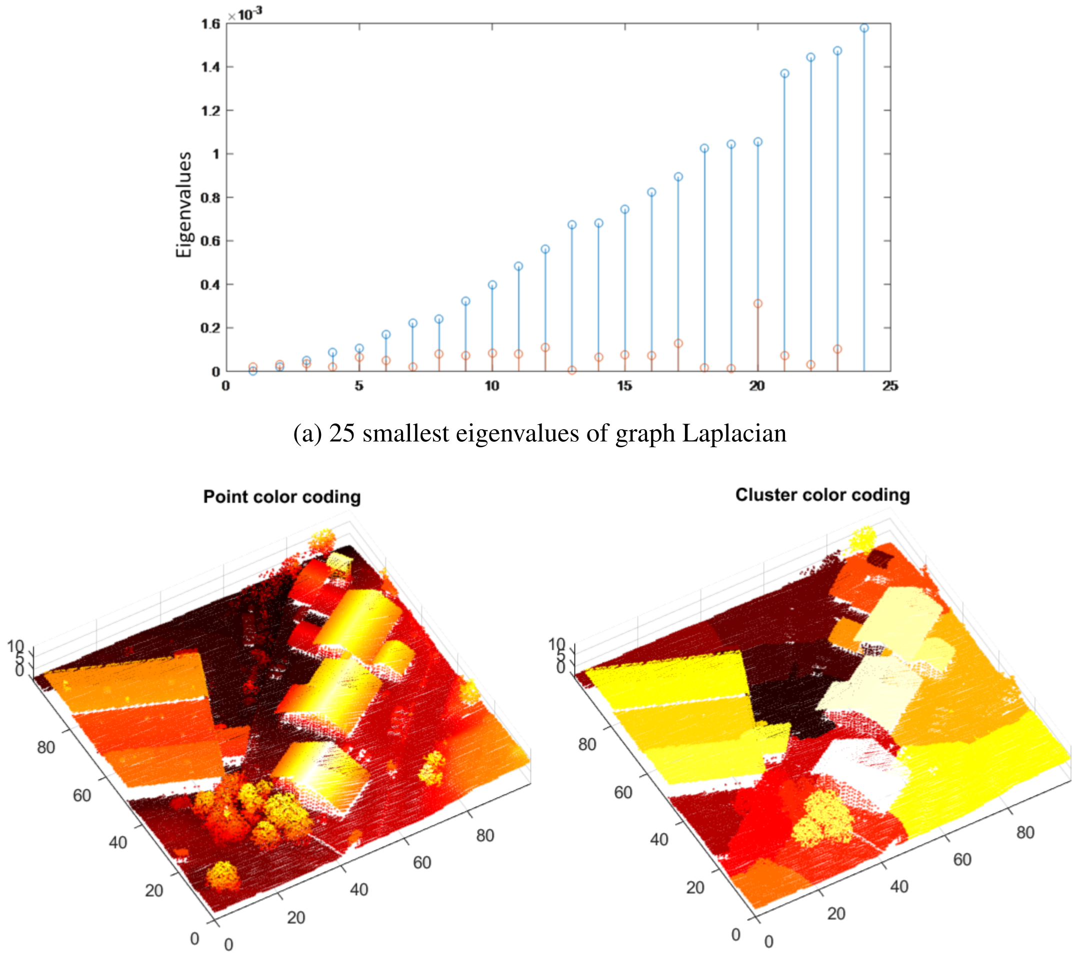
We tested the proposed segmentation framework adapting Algorithm 2 on eight sites of publicly available dataset by ISPRS Working Group III/3 [31]. A result on "City Site2" is demonstrated on Figure 4.5, where the total number of points is 243,400 with a density of 0.67 points per square meter. In this dataset, second returns are mostly present around the tree regions where the laser beams are reflected by the ground after the tree scatters. We just focused on the first returns in order to segment the structures above the ground level, including the tree objects. In studies concentrating on the ground plane extraction, those second returns are very valuable. The point cloud is denser on the parts when two stripes of scanning intersect.

Another ALS dataset "Vaihingen" is presented by ISPRS Commission WG III/4 [65]. We selected a part of this dataset in Figure 4.6, which is dominated by irregular shaped buildings and trees.

4.5.2 Experiments on Spectral Clustering using Eigengap Proposition

In this section, we have experimented the classical spectral clustering algorithm based on Algorithm 1, introduced in Chapter 2. We have adapted the eigengap heuristic in the determination of number of clusters and the eigenvectors to be used in the clustering procedure.

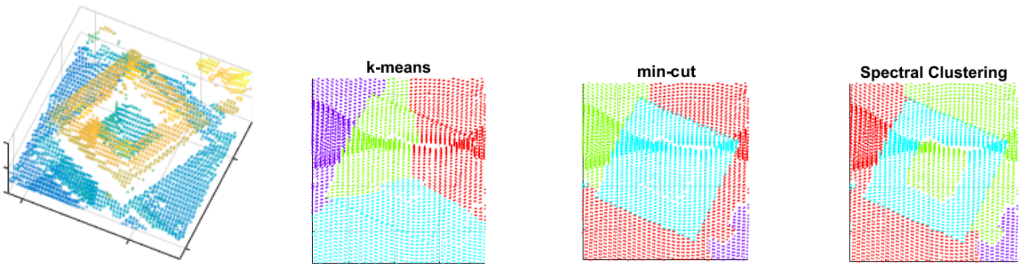
In the classical approach, we have constructed the graph using 8 nearest neighbors. After calculating a number of smallest eigenvalues of Laplacian, the number of clusters is chosen based on the eigengap proposition. For instance, in France point cloud, an eigengap exists after the 20th eigenvalue since the smallest eigenvalues are plotted in blue stems in 4.2a and the red stems stand for the difference with the next larger eigenvalue, those make the peak at 20.



(a) 25 smallest eigenvalues of graph Laplacian

(b) 3D view with elevation color coding using "hot" colormap, pointwise on the left and clusterwise on the right

Figure 4.2: Segmentation of "France" point cloud by spectral clustering



(a) A complex structure in 3D view

(b) k-Means Clustering in 2D view

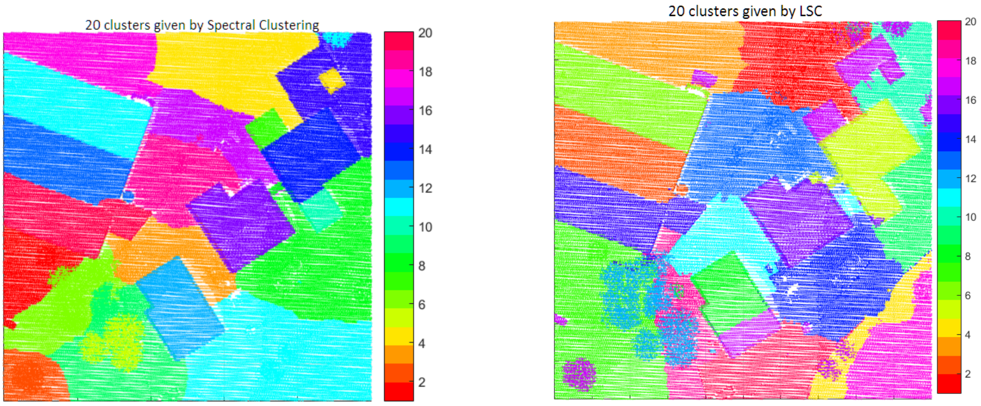
(c) Min-cut Clustering in 2D view

(d) Spectral Clustering in 2D view

Figure 4.3: Segmentation of a complex building by different clustering methods

Recalling the "intertwined spiral" demonstration in Chapter 2, we focus on a segmentation of a building on "City Site2" which exhibits a complex structure. We conducted k-means, min-cut and spectral clustering algorithm on this example. It is declared that k-means clustering tends to find convex sets [25], and min-cut can manage to partition simple convex distributions as well [58]. On the other hand, spectral clustering respects the inner connectivity of the data space owing to the encoded graph representation and the spectral relaxation. As seen on Figure 4.3, k-means clustering on 3D points only gives a correct segmentation at one side of the building, whereas min-cut gives a better result by discriminating four sides of the building from the background owing to the graph representation developed specially for the airborne LIDAR scenes. Nonetheless, only the spectral clustering algorithm can handle the shape of the building having a yard structure and gives a correct segmentation.

4.5.3 Experiments on LSC Algorithm



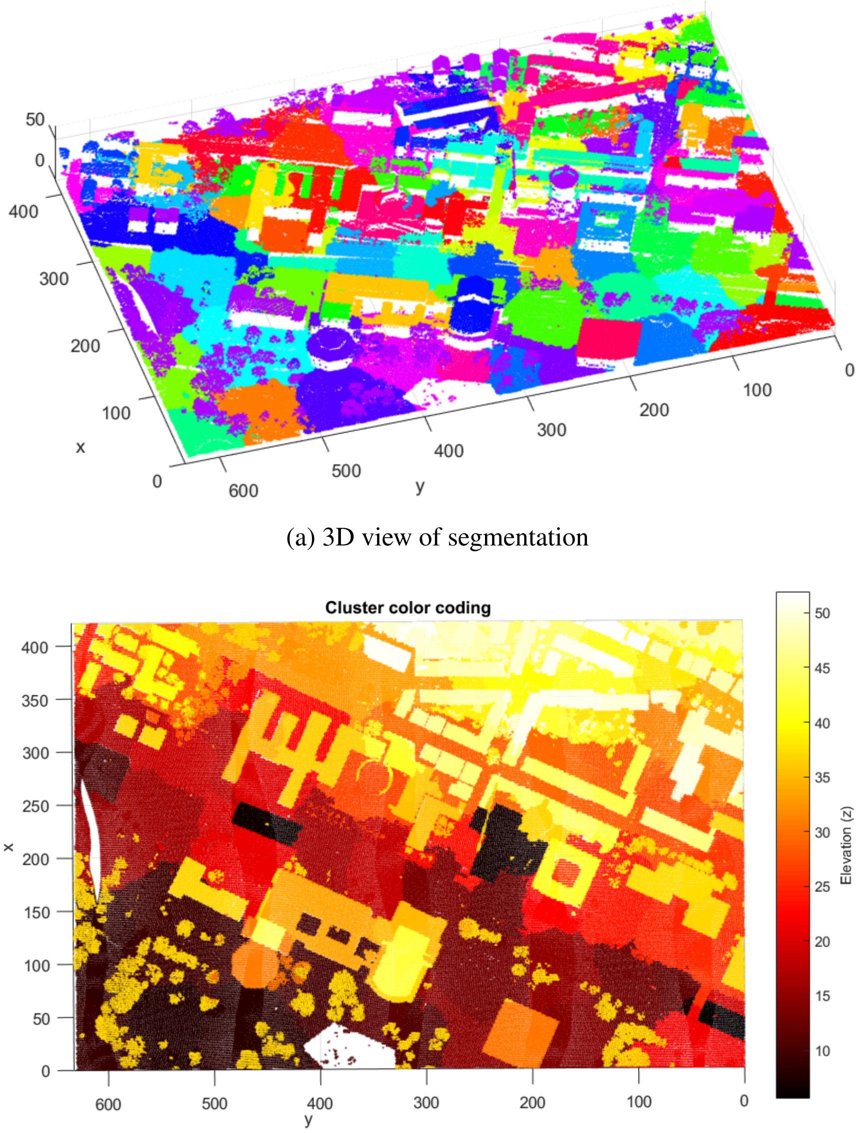
(a) Algorithm 1

(b) Algorithm 2

Figure 4.4: Segmentation of "France" point cloud on 2D view
Comparison of Algorithm 1 and Algorithm 2

In the experiments of LSC, based on Algorithm 2, the number of landmarks p is chosen as 1000 and r is taken as 8, which must be chosen with respect to the scale of dataset.

First of all, we present the segmentation results on "France" point cloud given by Algorithm 1 and Algorithm 2 for the purpose of visual comparison. As seen on Figure 4.4, performance of LSC algorithm is close to the one given by classical spectral clustering, except some of wrongly segmented points belonging to a roof. In terms of computational time, LSC quite accelerates the segmentation process. For "France" point cloud, where more than 100,000 points exist, it resulted 56 times faster than Algorithm 1. Note that, eigengap proposition is not adapted in LSC algorithm since the eigen spectrum of Laplacian can not be analyzed explicitly in LSC owing to the approximations pursued.



(a) 3D view of segmentation

(b) 2D view with elevation

Figure 4.5: Segmentation of City Site2 by LSC

Then, LSC framework is practiced on "City Site2" and "Vaihingen" datasets, whose segmentation results are given on Figure 4.5 and Figure 4.6 respectively. In the results, we observe the effect of two major principle of spectral clustering, which are very advantageous for segmentation task. First, it estimates the clusters by evaluating their spectral characteristics rather than spatial closeness. For this reason, two

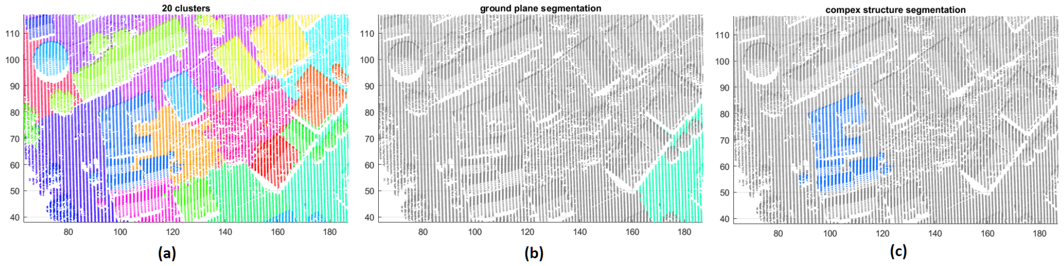


Figure 4.6: Segmentation of Vaihingen point cloud by LSC in 2D view
(a) overall segmentation (b) ground plane segmentation (c) complex building segmentation

LIDAR objects can be assigned to the same cluster when they enclose similar spectral identities in their localities even if they are not literally neighbors in the graph. As it is seen from Figure 4.5, the trees in the same region are assigned to one cluster, since they all have scattered forms which presents repeated localized patterns. In this regard, spectral clustering fulfils a kind of classification task together with the segmentation. Second, spectral clustering favors the spatial connectedness of the segment, and it is not simply based on convex partitioning, which is demonstrated by Figure 4.3. Therefore, it can be preferred to discriminate architectures with yard structure as it is the case for some buildings in Figure 4.5. Both "City Site2" and "Vaihingen" dataset contain irregularly shaped and connected large buildings which can be challenging for model based algorithms. The spectral clustering does not require a previous knowledge on the shapes to be discriminated, thus it can be used to segment any kind of shape in general. For this purpose, in Figure 4.6, we present the segmentation result of a scene which is part of "Vaihingen" dataset. Together with the overall segmentation of this scene, we highlight a cluster on Figure 4.6 (b & c), which belongs to a region on ground plane and a complex building respectively.

4.6 Conclusion

In this chapter, we have proposed a special graph construction strategy to be adopted for representing the aerial LIDAR data and conducted clustering experiments on some ALS datasets, those are unstructured 3D point clouds. Due to the flexibility of graph representations, we have accomplished the clustering operations directly on the unstructured 3D data. We have presented a visual comparison between spectral clustering and k-means, since k-means is a traditional clustering method that is applicable to 3D data as well. On this comparison, we have pointed out the superiority of spectral clustering to k-means in terms of consideration of connectivity of data space.

In classical spectral clustering framework, eigengap heuristic can be used as a hint for the number of clusters. However, LSC framework does not allow this.

The LSC algorithm intends to accelerate the classical spectral clustering through some approximations. It provides sufficient results with a quite improvement in the computational complexity.

All in all, spectral clustering gives adequate results for the post processing operations such as recognition, 3D modeling and reconstruction. In this point of view, the proposed approach can be considered as the first step in a framework reaching a semantic description of a LIDAR scene. After employing the clustering operation, the next step would be to formulate some feature vectors on the segmented sets in terms of shape and context.

CHAPTER 5

GRAPH SIGNAL FILTERING BASED EDGE DETECTION FOR AIRBORNE LIDAR DATA

5.1 Introduction

In this chapter, we address the problem of edge detection on 3D LIDAR point clouds. Edge features carry valuable information for segmentation and reconstruction tasks and for the detection of man-made structures. In order to enhance the overall performance of the proposed framework, outlier removal and smoothing operations are highly recommended to be applied on the aerial LIDAR data as the pre-processing steps.

Although LIDAR systems provide highly accurate and dense data, it may also contain spikes and noise due to the measurement errors or complexity of the object space to be scanned as it is the case for many signal forms captured by various sensors. The outliers should be removed and the noisy points should be refined with a de-noising stage for the purpose of preventing any misconception in the following processing stages. By defining a smoothing stage prior to the edge detection procedure, we intend to eliminate the distortions and the clutter in the scene, in addition to loosening the effect of errors. While doing that, we avoid distorting the important features such as discontinuities at the object boundaries. The graph based representation may provide an edge aware smoothing solution [26, 1] depending on the construction of the graph and choice of the graph signal.

In this chapter, we first review the related work on edge based segmentation and filtering methods on LIDAR data. Then, we explain how to build the graph representation of ALS point cloud properly for the operations to be conducted. Then, we present the algorithms for smoothing and edge detection tasks. Finally, we provide experiments on some ALS data and give the discussions on the results.

5.2 Related Work

In order to acquire semantic analysis of a LIDAR scene, segmentation is a fundamental task, which aids to focus on objects of interest by separating the entire data set into regions. For example, building extraction and reconstruction are among the most essential LIDAR applications requiring segmentation information. In some studies, the segmentation problem is addressed by adopting edge-based approaches which desire to determine the edges on a data set to obtain the segments. The early ones mostly preferred to detect edges on the range image formats of LIDAR data [66]. In the previous chapter, we mentioned the possible information loss owing to converting the 3D data to 2D. Especially for retrieving the important features such as edges, the assessment of 3D geometry is vital. Furthermore, the studies operating directly on 3D data mostly conduct a local spatial analysis for each point to determine the points on object boundaries. Among those studies, there are the ones calculating the local curvatures [67], normals [68] and slopes or the ones computing the local convex hull [69, 4] and local eigenanalysis [70].

Calculating all those local spatial primitives and grouping the points based on them can be difficult. On the other hand, we propose to find the boundaries on a LIDAR scene by adopting well-known signal processing filtering approaches. We intend to detect the edge features on 3D point cloud data by employing high-pass and band-pass filters via the graph spectral filtering. In other words, the classical image processing tools can be transposed to weighted graphs [54] which is a practical way to process unorganized 3D point clouds.

In LIDAR literature, filtering terminology is usually ascribed to applying some order of thresholds based on elevation, slope or other spatial primitives for separation of ground and non-ground points [71, 72, 73] rather than a spectral meaning as it is generally accustomed by the signal processing communities. In this study, we will address the problem of edge detection abiding by the later convention and practice band-pass and high-pass filtering operations on the graph representation of ALS data.

5.3 Outlier Removal and Graph Spectral Smoothing on ALS Data

The first step of the pre-processing stage is outlier removal. Outlier points are designated as the unexpected speckles on the elevation data. They can be distinguished by investigating the isolated z values on the whole range of the point cloud without focusing on any local part. Nonetheless, for the data collected from the very large fields or the ones having above a million points and steep elevation range, can be divided into several to be analyzed separately for capturing the outliers efficiently. Those points are omitted from the processing in order to prevent any misinterpretation that might affect the next stages. In [74], histogram of elevation data is analyzed to reveal the obvious outliers. We adopt this approach and simply discard the points having an isolated z value with respect to the elevation range of entire point cloud.

The second step is the refinement of the elevation data. This step intends to raise the integrity of the edge detection stage rather than de-noising of the 3D data on its own. Indeed, it diminishes the small high frequency regions such as low vegetation and little objects on the ground which are not in the point of interest, consequently helps the edge detection result to be clearer. Moreover, the proposed smoothing operation does not distort the object boundaries owing to the edge aware graph structure.

The most conventional studies approach the smoothing problem by convolving the signal with a low pass filter, for instance a Gaussian filter. In fact, for the uniform encoded signals, this is equivalent to create an isotropic diffusion across the signal space using the heat equation, which is explained in Appendix A.2. In [75], a series of regularization operators on graphs are introduced, including the diffusion kernels, by defining them as functions of graph Laplacian. In [26], Zhang et al. exploit graph representations for the anisotropic image smoothing application which preserves the discontinuities on boundaries. By embedding the edge information into the weights of the graph, they accomplish controlling the smoothing rate in the direction of edges to the contrary of smoothing the signal in each direction equally. For this purpose, they utilize heat diffusion kernels described by the graph Laplacian and adjust the heat flow from one vertex to another by determining edge weights between them in terms of pixel intensity differences.

Others prefer to leverage p-Laplace regularization on weighted graphs, where p-Dirichlet energy of a function defined on the vertices of the graph is affiliated as the notion of smoothness on the graph [76]. In [27], *graph signal* terminology is embraced instead of the function definition, and p-Laplace regularization is practiced for de-noising of 3D point clouds, where their 3-tuple graph signal consists of the 3D coordinates of each point.

In our smoothing application on LIDAR point clouds, we employ only the z values of the point cloud as the graph signal, since airborne LIDAR signal constitutes an irregular elevation data. We embed the 3D geometry directly into a weighted and undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V}$ is vertex set whose size $|\mathcal{V}| = N$ is equal to the number of points in the point cloud. The edge set, $\mathcal{E}$, of the graph is built on the k-nearest neighbor of points based on the 3D Euclidean distance between them. Additionally, the elements of weight matrix, W, are calculated by weighting the 3D Euclidean distances between points with a Gaussian function as follows,

$$W_{i,j} = \begin{cases} exp(-\frac{d(i,j)^2}{2\sigma^2}) & \text{if } v_j \in \mathcal{N}_{kNN}(v_i), \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where $\mathcal{N}_{kNN}(v_i)$ stands for k nearest neighborhood of point i and $d(i,j)$ is the distance between point i and point j. Here, σ is simply chosen as the mean distance between the neighbors. Then, we can compute the unnormalized Laplacian matrix, $L = D - W$, where D is diagonal degree matrix whose elements are the row-sum (or column-sum) of W.

We adapt the Tikhonov regularization on graphs, that is presented in Chapter 3. In Tikhonov regularization, the regularization term is the quadratic form of Laplacian, which corresponds to p-Laplace regularization where p = 2. Tikhonov regularization results in the following filtering operation,

$$\check{x} = (I + \gamma L)^{-1} f \tag{5.2}$$

where we desire to estimate the smoothed signal $\tilde{x}$ for the given graph signal f, with a smoothing factor γ. Recall that in Chapter 3, graph signal filtering operation is explained under matrix notation as $\boldsymbol{f}_{out} = \hat{h}(L)\boldsymbol{f}_{in}$. Hence, in this case, the optimal reconstruction is given by filtering the graph signal with the following low-pass filter, as expected for the smoothing problem,

$$\hat{h}(\lambda) = \frac{1}{1 + \gamma\lambda}$$

Note that, in this graph signal filtering based smoothing solution, the graph signal is chosen as the z values on $[x, y, z]$ coordinates of 3D point cloud, which appreciates the elevation information enclosed in aerial LIDAR data. Another key thing to remember is that z data acts upon the graph structure as well. To put it another way, graph signal has influence on the weights of the graph, which accommodates an edge aware smoothing procedure. To give an illustration, the neighboring points on the boundaries of the point cloud are connected with small weights or not connected at all due to the k-NN structure. Therefore, the edge features given by graph signal are also emphasized in graph structure. The smoothing term, recalled at (3.6), requires a graph signal that changes slowly over the graph representation, through the optimization problem. Although elevation changes rapidly on the discontinuities, the solution allows them to maintain on the reconstructed signal due to the low link weight between the points on those discontinuities. In other words, smoothing is less effectively performed across the edge features compared to the noisy or cluttered regions, which prevents distorting the edge features.

5.4 Edge Detection Algorithm for ALS Data

In this section, we desire to find the edge points on the point cloud by performing a graph signal filtering operation as well, where the signal is again the elevation values. However, we need a slight change in the graph construction part special to the edge detection application. Here, the graph structure is totally isolated from the graph signal, z values of the point cloud. The graph signal filtering in this part serves to find the signal differences between the neighboring points by the factor of the weights between them. Hence the graph weights should be powerful to boost the differences at the discontinuities to the contrary of the smoothing application, where the information flow across the discontinuities are highly discouraged.

For this purpose, the connections of the graph are determined with respect to k nearest neighbors on xy plane. In addition, their weights are computed in terms of the 2D Euclidean distances as well, that is only dependent on the x and y coordinates of the 3D points.

In conventional image processing applications, robust features are detected by the well-known operations such as difference of Gaussians (DoG) and Laplacian of Gaussians (LoG). Similarly, we employ a Mexican hat kernel for the detection of edge points, denoted as follows,

$$\hat{g}(\lambda) = \lambda \text{exp}(-t\lambda) \tag{5.3}$$

The Mexican hat kernel is actually derived from the heat kernel $\hat{h}(\lambda) = \mathrm{e}^{-t\lambda}$. It is the negative first derivative of the heat kernel with respect to diffusion factor t, which behaves as a band-pass filter [77]. As we all know, in spatial domain it corresponds to the negative normalized second derivative of the Gaussian function $G_t(x)$ with respect to x, that is mentioned in Appendix A.2. By adjusting the parameter t acting on Mexican hat kernel, $\hat{g}(\lambda)$, it is possible to generate a series of filters from band-pass to high-pass.

Eventually, we can detect the edge features on the point cloud signal using a Mexican hat kernel exhibiting much of a high-pass filter property. After the filtering operation $f_{out} = \hat{g}(L)f$, the edge points can be found with a proper thresholding on the absolute value of the output signal f_{out}.

Note that, if one prefer to apply the smoothing procedure explained in the previous section prior to the edge detection procedure in this section, it will not be equivalent to application of one filter produced by the integration a low-pass kernel and a high-pass kernel. Here the reason lies in the adoption of different graph representations for these two filtering operations, where the first one does not smooth the edge points, whereas the second one responses at the edge points at most.

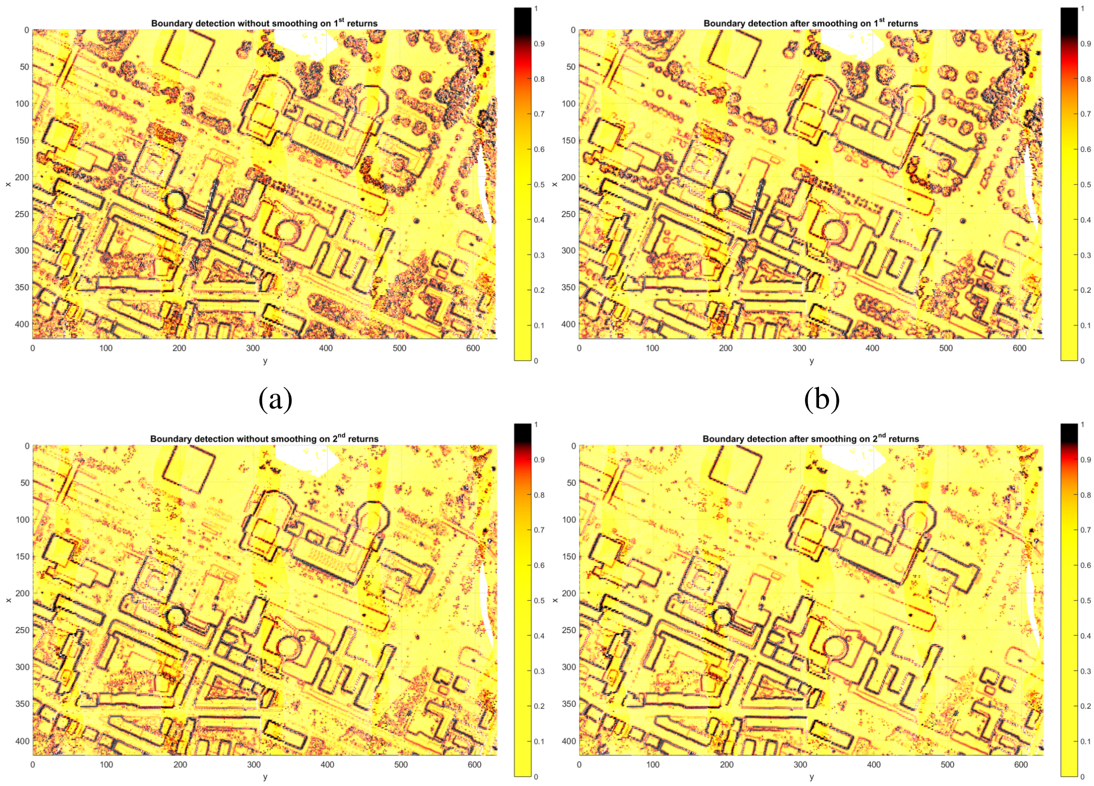
Performing a smoothing operation prior to the edge detection stage, eliminates both the distortions aroused by noise content and clutter, and deduces a much more clear picture of object boundaries. Nonetheless, highly scattered objects such as trees are still included in the detected high frequency features. For this purpose, in [4], the edge points belonging to the tree objects and the ones of buildings are separated using the correlation of the outlines where the former exhibits a 3D volume characteristics and the later is much like 2D. Another option to discard the trees from this picture could be exploiting the second returns in the LIDAR data, which aids to concentrate on only the man-made objects in the scene. We will simply adopt the this approach to remove the high frequency content belonging to the trees.

In [78], Bhandarkar et al. introduce two types of edges mostly encountered in range imagery data, namely *jump (step) edge* and *crease (roof) edge*. The jump edges are described by the discontinuities in elevation which correspond to the exterior boundaries of the 3D structures and they are leveraged for segmenting and detecting the objects. On the other hand, the crease edges occur where two surfaces intersect and they constitute vital information for roof/building reconstruction. The proposed edge detection algorithm can also be employed for detecting the crease edges of an object for building reconstruction and modeling purposes. Using a band-pass filter will directly give the interior breakline points of a LIDAR object, no matter how complex shape it presents (convex, non-convex, polyhedral). Moreover, our method is much more practical for extracting the planar and non-planar points than dealing with the surface normals and curvatures etc.

5.5 Experimental Results

In this chapter, we worked on the same datasets, "Vaihingen" and "City Site2", those have been utilized in the previous one.

In the experiments, we constructed the graph representations using a k nearest neighbor approach where $k = 8$. Computation of the neighborhoods can be accomplished by applying the k-d tree search methods for nearest neighbors in an efficient way [79, 80]. Moreover, computing the aforementioned graph kernel operators, $\hat{h}(L)$ and $\hat{g}(L)$ may be problematic as the number of nodes in the graph increases. To overcome this problem, Hammond et al. exploit polynomial expansion of a filtering operation explained in Section 3.4 and introduce the Chebyshev polynomial approximation of the graph kernels, which approximates graph kernel operators in terms of low order polynomial of L [6]. This avoids fully decomposition of L and computes the filtered signal through repeated matrix-vector multiplications in a fast and efficient way. For this purpose, we have benefited from the SGWT toolbox (`https://wiki.epfl.ch/sgwt`) which includes the implementation of the Chebyshev polynomial approximation of graph kernels. In the experiments, we approximated the kernels as polynomials having a degree of 25.



(a) (b)

(c) (d)

Figure 5.1: Discovered boundaries on 'City Site2'(a) without smoothing on 1st returns(b) after smoothing on 1st returns (c) without smoothing on 2nd returns(d) after smoothing on 2nd returns
The filter output is shown in "hot" colormap where the edge points are color in red to black region

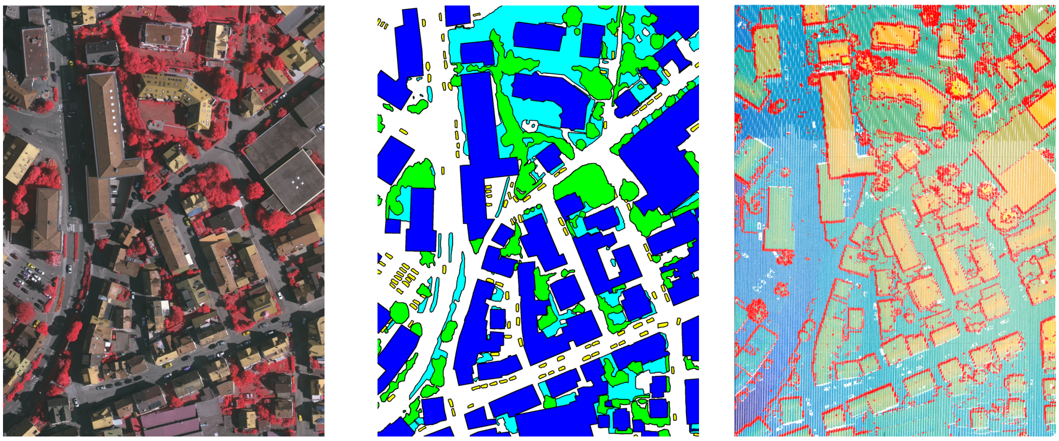
5.5.1 Edge Detection on some LIDAR scenes

Unlike the clustering experiments conducted in the previous chapter, in this chapter we worked on both first and last echoes in "City Site2" dataset and in Figure 5.1 we present both of them for a visual comparison. Furthermore, we show the edge detection results those are obtained with a pre-processing stage and without it in order to asses the significance of smoothing operation on the performance of edge detection.

In pre-processing stage, we first removed the outliers and then smoothed the graph signal adapting Tikhonov regularization, where $\gamma = 10$ based on kernel in (3.15). Moreover, in the filtering procedure for edge detection, the Mexican hat kernel is created for $t = 0.1$ based on (5.3).

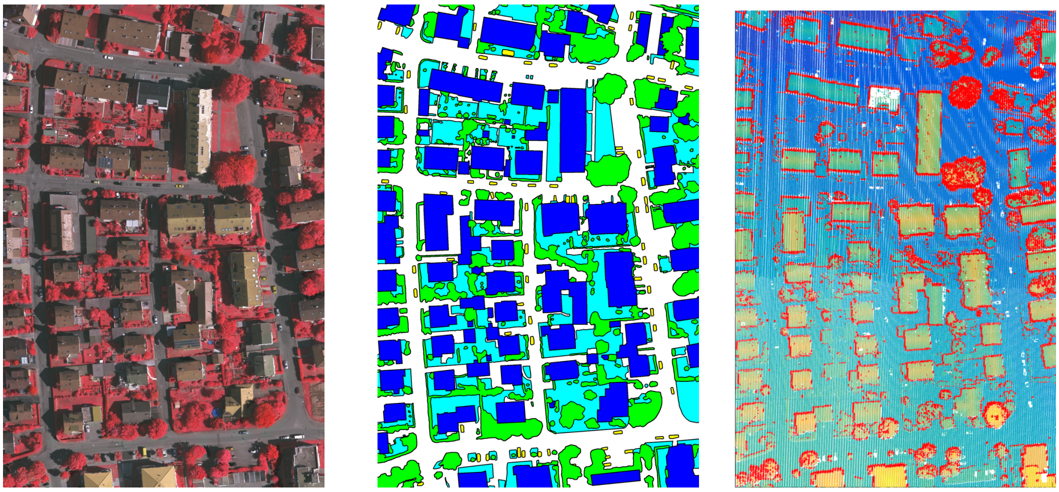
The first column of Figure 5.1 shows the object boundaries without applying a smoothing process prior to the edge detection, on 'City Site2'. The clutter content is more apparent on the results displayed in the first column compared to the ones in the second column where smoothing is applied before the edge detection operation. Therefore, we can claim that a prior edge-aware smoothing stage enhances the edge-detection process, that is akin to the experience gained in the image processing practices. In addition, the boundary detection results on the second row are based on the 2$^{\text{nd}}$ laser returns and we observe that most of the edge features originated by the trees are eliminated on those results.

Additionally, we experimented our edge detection algorithm in the test sites of Vaihingen city: Area 1 & Area 3, those are presented for benchmarking of tree and building extraction studies. The scene is dominated by the trees and complex buildings where the edge points are highly detected. Due to low rate of multiple echoes in this dataset, the scattered points, belonging to the trees, can not by simply separated from the ground objects, which requires further attention following to the edge detection stage. Despite the lack of 3D reference data, the 2D reference data, in which buildings, trees and cars are manually labeled and outlined, is provided in Figure 5.2(b) and 5.3(b), for a qualitative assessment.



(a) (b) (c)

Figure 5.2: Vaihingen Test Site Area 1
(a)Digital area image (b)2D Reference data (c)Edge Detection in 2D view

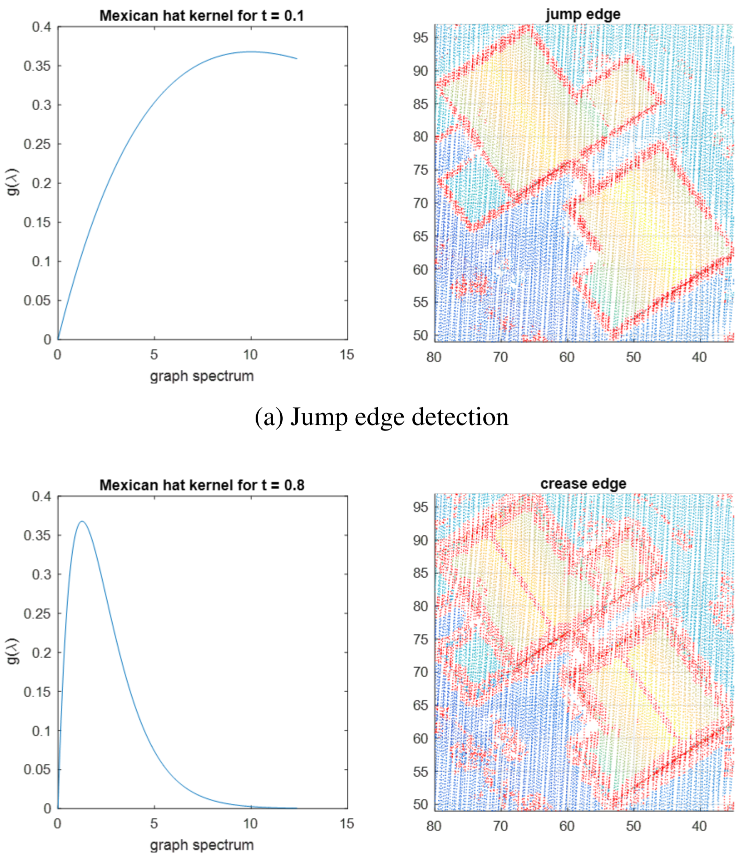


(a) (b) (c)

Figure 5.3: Vaihingen Test Site Area 3
(a)Digital area image (b)2D Reference data (c)Edge Detection in 2D view
The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [81]:
`http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html`

5.5.2 Application of Filtering for different Edge Types

In Figure 5.4, the crease and jump edges are demonstrated on the LIDAR data of two houses. In order to detect the jump edges, we adjusted the Mexican hat kernel to have a high-pass characteristic as shown in 5.4(a). As mentioned in the previous section, the jump edge generally occurs at the borders of the 3D objects where the height changes discontinuously, accordingly they are very valuable for segmentation information. Therefore, the detected edge points, which are produced by the high-pass filter, outline the roof of the houses in 5.4(a). On the other hand, in 5.4(b), we observe that the points on the rooftop, where the two sides of the roof are met, also appear as edge points. These are the crease edges which are not described by the discontinuity in elevation but the discontinuity in surface normal. Consequently, they can be emerged by a Mexican hat kernel with a band-pass characteristic as seen on 5.4(b).



(a) Jump edge detection

(b) Crease edge detection

Figure 5.4: Crease and jump edges and the filters employed to detect them
(Point cloud is shown in 2D view and it is colored with "parula" colormap with respect to elevation values, except the red edge points.)

Note that in the detection of crease edges, low-pass filtering is not involved in the pre-processing stage, since it smooths out crease edges to the contrary of the jump edges, which are protected by the edge aware graph structure.

Sometimes, the jump edges can arise on the inner sections of a LIDAR object which

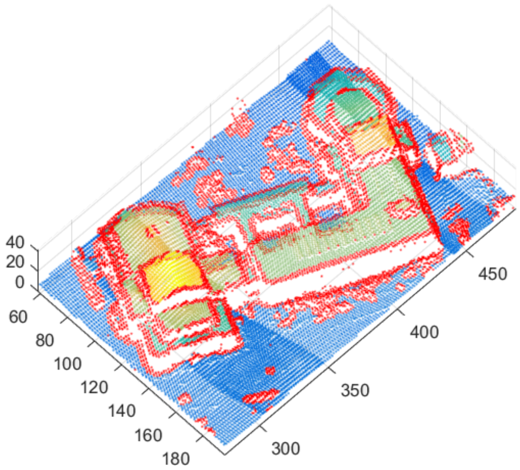


Figure 5.5: Jump edges detected on a multi-layer building
(Point cloud is colored with "parula" colormap with respect to elevation values, except the red edge points.)

is the case for the multi-layer buildings [4]. In Figure 5.5, the ALS data taken on "City Site2" constitutes such an example.

5.6 Conclusion

In this chapter, we have proposed a novel edge detection approach for segmentation and reconstruction of ALS data. Our algorithm is based on graph signal filtering which is a rising field of signal processing for irregular structured data types. Due to the convenient use of graph signal processing on unstructured LIDAR point clouds, we have accomplished to detect the edge points directly on 3D data using a graph structure which is constructed exclusively to answer the needs of the application. Moreover, considering the elevation data as the graph signal, we have leveraged aerial characteristic of the airborne LIDAR data.

We have applied outlier removal and smoothing operations as pre-processing stages in order to raise the performance of overall edge detection framework. We have observed that smoothing procedure quite helps to obtain better edge detection results by eliminating noise and clutter on a scene. In order to prevent any misconception about the overall edge detection procedure, it should not be considered as direct application of a band-pass filter following a low-pass filter. The result is not equal to the direct integration of these filters since the graph representation is different in those two filtering operations. Therefore, while we did not distort the edge points in low-pass filtering due to the edge aware representation, with another graph representation, the edge points are detected by the band-pass filtering.

Our proposed method can be employed both for discovering the jump edges on a segmentation problem and for exploring the crease edges on a LIDAR object on a reconstruction/modeling problem by only adjusting the filter characteristics, yet the former was the main focus of this chapter.

As a framework that is applicable directly on unstructured 3D data, it is theoretically superior to the edge-based methods on 2.5D range images. Furthermore, it is much more practical than calculating surface normals or other geometric primitives which are the approaches adopted by most of the studies working directly on 3D LIDAR data.

In the experiments, we observe that the high frequency features are assembled mainly in the tree objects and boundaries of buildings. As we detected those edge points using one high-pass filter, they can be further classified by extracting multi-spectral features on them. In this point of view, Spectral Graph Wavelet Transform (SGWT), which is mentioned in Appendix A.3, is quite suitable. In future studies, we aim to focus on classification of the LIDAR objects based on the wavelet features on graph representations.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Summary

In this thesis, we have addressed the problem of segmentation and edge detection on 3D LIDAR point clouds by a signal processing approach. In order to process those unorganized point cloud data as signals, we have leveraged graph representations, since graphs are very convenient tools for encoding both irregular data, especially lying on high dimensional geometry. In particular, we have exploited the fields of spectral graph theory and graph signal processing to retrieve information via the graph representations of 3D point clouds. Such approaches support analyzing the data of interest not only in spatial fashion but also in a spectral manner.

Graph spectral approaches are separated from widely known numerical graph algorithms or algebraic graph methods, since they are able to merge the graph theoretic concepts with Fourier Analysis. To that end, we have reviewed the Spectral Graph Theory and Graph Signal Processing in the first chapters. In those chapters, we introduce the *graph Laplacian* and indicate that it is essentially important for the interpretation of the spectral information on graphs. We explain the notion of smoothness in graphs by the relation of *graph Laplacian quadratic form.* With this in mind, we present an analogy between Laplacian eigensystems on graph settings and Fourier basis in classical signal processing settings. In the light of this theoretical background, we aim to employ the operations that are originated by the graph spectral studies, such as spectral clustering and graph signal filtering, as tools for segmentation and edge detection tasks on LIDAR data.

Throughout the remaining chapters, we review the studies that are dedicated for the segmentation of LIDAR data, especially on ALS data, which are mostly addressed by the geo-information and remote sensing literature. We intend to bring solutions to LIDAR problems with a signal processing point of view. In fact, there are many studies which have practiced the well-known image processing tools on the raster image forms of LIDAR data. Yet, to the best of our knowledge, this study is novel in practicing the graph spectral approaches for segmenting directly on the 3D ALS data.

Airborne LIDAR data is different from other types of 3D point cloud data, such as the ones obtained by the terrestrial or mobile applications or 3D models captured by other 3D depth sensors. It constitutes much of the property of non-uniformly sampled elevation data. Therefore, graph representations should be constructed considering this fact. For instance, in spectral clustering application, we construct the graph topology based on 3D geometry, while determining the weights of the graph based on the elevation values only. This approach advances the performance of spectral clustering operation by transposing the graph spectral based image segmentation approaches to the 3D aerial LIDAR data. Moreover, in the following chapter, we utilize the elevation values as graph signal and adopt different graph representations for smoothing and edge detection operations. In smoothing application, edge features are preserved due to the anisotropic regularization resulted with the graph structure. On the other hand, in edge detection application, graph structure is adjusted so that the filtering operation is carried out most effectively at the edge points.

Spectral analysis of a weighted graph requires eigen decomposition of the graph Laplacian matrix. In spectral clustering, it is enough to extract a few of the smallest eigenvalues. Nevertheless, such an approach can be problematic for large scale datasets. For this purpose, we adapt LSC algorithm, which selects some landmark points on the dataset, and approximates a weight matrix which contains the relation of landmark points with the whole dataset, and in turn simplifies the eigen decomposition process. In the edge detection chapter, we also avoid the entire eigen decomposition of the graph Laplacian by considering the filtering operator as a polynomial expansion of Laplacian and then adopting the Chebyshev approximation method for the graph kernels.

Experimental results are given on some datasets provided by the ISPRS commissions, which have been tested in many other studies. The captured scenes for these datasets also constitute difficult scenarios accommodating connected buildings, complex objects having unusual shape, etc. Those objects are not detached and discriminated as a whole by the tested spectral clustering algorithm, since it does not simply realize a convex partitioning, but recognizes the connectivity due to the weighted graph encoding. It should be noted that the tested approaches do not require previous assumptions on the shape of the objects; therefore, they are able to handle those complex shapes successfully. During the background review of ALS research, we have observed that most of the previous approaches are based on extracting the spatial primitives which can be exhausting to compute for each point. On the other hand, we have developed a graph spectral filtering algorithm that finds the points on the boundaries automatically. Moreover, the filtering operation can be adjusted according to the type of the edges to be detected.

As a concluding remark for this summary, we can state that the state-of-the-art techniques addressed on LIDAR segmentation problem are extensively reviewed; next, new approaches are considered during the solution of the problem. Moreover, the algorithms are also extended to be able to utilize them in computationally efficient ways. Experimental results indicate that the graph spectral studies have a promising future, not only on the LIDAR point cloud area, but also in other data fields having high dimensional geometry and irregular structure.

6.2 Conclusions

In this thesis, we present two application chapters one of which practices segmentation of ALS data by spectral clustering. It contains a novel segmentation framework for ALS data that is based on the graph based image segmentation studies. Moreover, the graph representation adopted in this chapter highlights aerial characteristic of the ALS data, therefore it provides adequate segmentation results. In comparison to the linear clustering methods, such as K-means, or basic graph partitioning methods, such as min-cut, spectral clustering yields better partitioning on complex structures, where the connectivity of the samples and the spectral embedding of the dataset carry significant information. Based on the experimental results on classical spectral clustering, the eigengap (i.e. relatively large difference between consecutive eigenvalues) can be a useful indication for the eigenvectors to be used during clustering and the number of clusters. It should also be noted that for the segmentation of large datasets, we tested LSC algorithm, which approximates the classical spectral clustering in a computationally efficient way. It accelerates the segmentation process by 56 times in 'France' point cloud without losing much from the segmentation performance.

In the other chapter, graph signal filtering is employed for detection of edge points on a LIDAR scene or on a LIDAR object. We observed that a smoothing stage employed prior to the edge detection stage successfully suppresses the small distortions on the dataset, and consequently improves the edge detection result. Since we construct the graph representation considering the target of filtering operation, in the smoothing stage low-pass filter does not restrain the edge features, whereas in the edge detection stage, high-pass filter has a larger response on the edge features. Furthermore, on the edge detection part, we have seen that employing a high-pass filter detects the jump edges, which are important for segmentation information, whereas employing a band-pass filter detects the crease edges, which are significant for modeling a LIDAR object.

One can finally conclude that in this thesis simple solutions are presented for some of the problems in LIDAR field that are mostly based on fundamentals of linear algebra. The presented solutions are computationally efficient and quite practical to implement compared to some other approaches calculating the geometric primitives of the samples in a dataset, which could be exhausting.

6.3 Future Directions

Segmentation is important in terms of splitting the scene into internally complete regions and separating the objects such as buildings, trees, roads. Since spectral clustering grasps the shape content by means of spectral analysis of input space, the objects or regions existing on similar spectral ranges, which are encoded by the locality of graph representation, might be assigned to the same cluster. For this reason, the presented spectral clustering techniques can be improved towards a classification problem for the scenes where the objects present dominant shape characteristics, for instance, to classify similar houses in a residential area, or trees from the ground plane in a wooded region.

Another advantage of spectral clustering is that it considers the internal connectivity of the objects which is embedded on the graph representation. Therefore, it can also be employed for discriminating an object having higher inner complexity such as tangled buildings, yard structures etc.

The segmented regions emerged by the spectral clustering can be further analyzed independently for the purpose of recognition and reconstruction. For those problems, spectral graph based solutions can be adopted as well. Recently, some studies belonging to the computer graphics communities leverage spectral geometry for shape retrieval on unstructured 3D models by considering a manifold learning approach [82]. The same approach can be transposed to the discrete data spaces using graph representations as practiced in [83]. Some other studies compute shape signatures based on heat kernel or wavelets [84, 85]. Furthermore, the wavelets described on 3D data gain popularity lately to accomplish multi-resolution analysis for geometric descriptions, in particular, diffusion wavelets are preferred in notable studies [77, 86]. Finally, in [87], local descriptors are generated using spectral graph wavelet transform, then the extracted features are combined to obtain a global shape descriptor. All of these approaches could be adjusted on 3D LIDAR data by adopting the graph based representations for the classification and reconstruction tasks.

For the problem of edge detection on aerial LIDAR data, we locate edge features by adjusting a filter (band-pass or high-pass) according to the property of the edges to be detected. However, using several filters simultaneously and obtaining local descriptors for each point might enrich the solution comparing to employing only one filter with roughly thresholding the result. Spectral graph wavelets can be employed for this purpose, since the wavelets generated by this method localize on different frequency ranges on graph spectrum and present an overcomplete transform at the end. It is possible to realize a classification algorithm based on those local spectral features.

For all the aforementioned problems, better comprehension of 3D data can be attainable through multi-resolution approaches adapted on the original geometry of the data. Indeed, the experience achieved by the classical signal processing literature indicates the promising potential of multi-resolution information on many applications.

REFERENCES

[1] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. Signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular data domains. *CoRR*, abs/1211.0053, 2012.

[2] Franz Rottensteiner and Christian Briese. A new method for building extraction in urban areas from high-resolution lidar data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):295–301, 2002.

[3] Peter Axelsson. Dem generation from laser scanner data using adaptive tin models. *International Archives of Photogrammetry and Remote Sensing*, 33(B4/1; PART 4):111–118, 2000.

[4] Jun Wang and Jie Shan. Segmentation of lidar point clouds for building extraction. In *American Society for Photogramm. Remote Sens. Annual Conference, Baltimore, MD*, pages 9–13, 2009.

[5] Anh Nguyen and Bac Le. 3D point cloud segmentation: A survey. In *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*, pages 225–230. IEEE, 2013.

[6] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[7] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[8] Gerald B Folland. *Fourier analysis and its applications*, volume 4. American Mathematical Soc., 1992.

[9] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[10] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002.

[11] Chris Godsil and Gordon F Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2013.

[12] Daniel Spielman. Spectral graph theory. *Lecture Notes, Yale University*, pages 740–0776, 2009.

[13] Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The Laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.

[14] Steven Rosenberg. *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*. Number 31. Cambridge University Press, 1997.

[15] Amit Singer. From graph to manifold Laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006.

[16] Mikhail Belkin and Partha Niyogi. Convergence of Laplacian eigenmaps. *Advances in Neural Information Processing Systems*, 19:129, 2007.

[17] Daniel Ting, Ling Huang, and Michael Jordan. An analysis of the convergence of graph Laplacians. *arXiv preprint arXiv:1101.5435*, 2011.

[18] Türker Bıyıkoglu, Josef Leydold, and Peter F Stadler. Laplacian eigenvectors of graphs. *Lecture notes in mathematics*, 1915, 2007.

[19] Matthias Hein, Jean-Yves Audibert, and Ulrike Von Luxburg. From graphs to manifolds–weak and strong pointwise consistency of graph Laplacians. In *International Conference on Computational Learning Theory*, pages 470–485. Springer, 2005.

[20] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. In *International Conference on Computational Learning Theory*, pages 486–500. Springer, 2005.

[21] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16:177–184, 2004.

[22] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[23] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 731–737. IEEE, 1997.

[24] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[25] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[26] Fan Zhang and Edwin R Hancock. Graph spectral image smoothing using the heat kernel. *Pattern Recognition*, 41(11):3328–3342, 2008.

[27] Yann Schoenenberger, Johan Paratte, and Pierre Vandergheynst. Graph-based denoising for time-varying point clouds. In *2015 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2015.

[28] Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Wiley Online Library, 2008.

[29] David I Shuman, Benjamin Ricaud, and Pierre Vandergheynst. Vertex-frequency analysis on graphs. *Applied and Computational Harmonic Analysis*, 40(2):260–291, 2016.

[30] Aloysius Wehr and Uwe Lohr. Airborne laser scanning—an introduction and overview. *ISPRS Journal of photogrammetry and remote sensing*, 54(2):68–82, 1999.

[31] G Vosselman. Isprs test on extracting dems from point clouds: A comparison of existing automatic filters. *ISPRS Commission III, http://www. itc. nl/isprswgIII-3/filtertest*, 2003.

[32] Martin Isenburg. Lastools—efficient tools for lidar processing (version 111216), 2008.

[33] Xiaoyi Y Jiang, Urs Meier, and Horst Bunke. Fast range image segmentation using high-level segmentation primitives. In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*, pages 83–88. IEEE, 1996.

[34] George Sithole and George Vosselman. Automatic structure detection in a point-cloud of an urban landscape. In *Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on*, pages 67–71. IEEE, 2003.

[35] Oliver Wang, Suresh K Lodha, and David P Helmbold. A bayesian approach to building footprint extraction from aerial lidar data. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 192–199. IEEE, 2006.

[36] George Vosselman, Ben GH Gorte, George Sithole, and Tahir Rabbani. Recognising structure in laser scanner point clouds. *International archives of photogrammetry, remote sensing and spatial information sciences*, 46(8):33–38, 2004.

[37] Bertrand Douillard, James Underwood, Noah Kuntz, Vsevolod Vlaskine, Alastair Quadros, Peter Morton, and Alon Frenkel. On the segmentation of 3D lidar point clouds. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2798–2805. IEEE, 2011.

[38] Tahir Rabbani, Frank Van Den Heuvel, and George Vosselmann. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36(5):248–253, 2006.

[39] Frank Moosmann, Oliver Pink, and Christoph Stiller. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 215–220. IEEE, 2009.

[40] Richard Hoffman and Anil K Jain. Segmentation and classification of range images. *IEEE transactions on pattern analysis and machine intelligence*, (5):608–620, 1987.

[41] George Vosselman, Sander Dijkman, et al. 3D building model reconstruction from point clouds and ground plans. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W4):37–44, 2001.

[42] Z Lari, A Habib, and E Kwak. An adaptive approach for segmentation of 3D laser point cloud. In *ISPRS Workshop Laser Scanning*, pages 29–31, 2011.

[43] Sagi Filin and Norbert Pfeifer. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS journal of Photogrammetry and Remote Sensing*, 60(2):71–80, 2006.

[44] Fayez Tarsha-Kurdi, Tania Landes, Pierre Grussenmeyer, et al. Hough-transform and extended ransac algorithms for automatic detection of 3D building roof planes from lidar data. In *Proceedings of the ISPRS Workshop on Laser Scanning*, volume 36, pages 407–412, 2007.

[45] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[46] Johannes Strom, Andrew Richardson, and Edwin Olson. Graph-based segmentation for colored 3D laser point clouds. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2131–2136. IEEE, 2010.

[47] David L Vilariño, Jorge Martínez, Francisco F Rivera, José C Cabaleiro, and Tomás F Pena. Graph-based segmentation of airborne lidar point clouds. In *SPIE Remote Sensing*, pages 100040I–100040I. International Society for Optics and Photonics, 2016.

[48] Aleksey Golovinskiy and Thomas Funkhouser. Min-cut based segmentation of point clouds. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 39–46. IEEE, 2009.

[49] Shaohui Sun and Carl Salvaggio. Aerial 3d building detection and modeling from airborne lidar point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(3):1440–1449, 2013.

[50] Serkan Ural and Jie Shan. Min-cut based segmentation of airborne lidar point clouds. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, 1:167–172, 2012.

[51] Yuxiang He, Chunsun Zhang, and C Fraser. A line-based spectral clustering method for efficient planar structure extraction from lidar data. *Proceedings of ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Antalya, Turkey*, 1113:103108, 2013.

[52] Franck Hétroy-Wheeler, Eric Casella, and Dobrina Boltcheva. Segmentation of tree seedling point clouds into elementary units. *International Journal of Remote Sensing*, 37(13):2881–2907, 2016.

[53] Teng Ma, Zhuangzhi Wu, Lu Feng, Pei Luo, and Xiang Long. Point cloud segmentation through spectral clustering. In *Information Science and Engineering (ICISE), 2010 2nd International Conference on*, pages 1–4. IEEE, 2010.

[54] François Lozes, Abderrahim Elmoataz, and Olivier Lézoray. Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *IEEE Transactions on Image Processing*, 23(9):3896–3909, 2014.

[55] Klaas Klasing, Dirk Wollherr, and Martin Buss. A clustering method for efficient segmentation of 3D laser data. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 4043–4048. IEEE, 2008.

[56] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[57] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[58] Shifei Ding, Liwen Zhang, and Yu Zhang. Research on spectral clustering algorithms and prospects. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 6, pages V6–149. IEEE, 2010.

[59] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. *Advances in neural information processing systems*, pages 601–608, 2003.

[60] Mikhail Belkin and Partha Niyogi. Using manifold stucture for partially labeled classification. *Advances in neural information processing systems*, pages 953–960, 2003.

[61] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y Chang. Parallel spectral clustering in distributed systems. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):568–586, 2011.

[62] Yeqing Li, Junzhou Huang, and Wei Liu. Scalable sequential spectral clustering. In *AAAI*, pages 1809–1815, 2016.

[63] Xinlei Chen and Deng Cai. Large scale spectral clustering with landmark-based representation. In *AAAI*, volume 5, page 14, 2011.

[64] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *IEEE transactions on pattern analysis and machine intelligence*, 26(2):214–225, 2004.

[65] Franz Rottensteiner, Gunho Sohn, Markus Gerke, and Jan Dirk Wegner. Isprs test project on urban classification and 3D building reconstruction. *Commission III-Photogrammetric Computer Vision and Image Analysis, Working Group III/4-3D Scene Analysis*, pages 1–17, 2013.

[66] Xiaoyi Jiang and Horst Bunke. Edge detection in range images based on scan line approximation. *Computer vision and image understanding*, 73(2):183–199, 1999.

[67] Ting-Jun Fan, Gerard Medioni, and Ramakant Nevatia. Segmented descriptions of 3-d surfaces. *IEEE Journal on Robotics and Automation*, 3(6):527–538, 1987.

[68] H Woo, E Kang, Semyung Wang, and Kwan H Lee. A new segmentation method for point cloud data. *International Journal of Machine Tools and Manufacture*, 42(2):167–178, 2002.

[69] Aparajithan Sampath and Jie Shan. Building boundary tracing and regularization from airborne lidar point clouds. *Photogrammetric Engineering & Remote Sensing*, 73(7):805–812, 2007.

[70] Aparajithan Sampath and Jie Shan. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. *IEEE Transactions on geoscience and remote sensing*, 48(3):1554–1567, 2010.

[71] Xuelian Meng, Nate Currit, and Kaiguang Zhao. Ground filtering algorithms for airborne lidar data: A review of critical issues. *Remote Sensing*, 2(3):833–860, 2010.

[72] George Sithole and George Vosselman. Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS journal of photogrammetry and remote sensing*, 59(1):85–101, 2004.

[73] George Vosselman. Slope based filtering of laser altimetry data. *International Archives of Photogrammetry and Remote Sensing*, 33(B3/2; PART 3):935–942, 2000.

[74] Jing Shen, Jiping Liu, Rong Zhao, and Xiangguo Lin. A kd-tree-based outlier detection method for airborne lidar point clouds. In *Image and Data Fusion (ISIDF), 2011 International Symposium on*, pages 1–4. IEEE, 2011.

[75] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[76] Abderrahim Elmoataz, Olivier Lezoray, and Sébastien Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE transactions on Image Processing*, 17(7):1047–1060, 2008.

[77] Tingbo Hou and Hong Qin. *Diffusion-driven Wavelet Design for Shape Analysis*. CRC Press, 2014.

[78] Suchendra M Bhandarkar and Andreas Siebert. Integrating edge and surface information for range image segmentation. *Pattern Recognition*, 25(9):947–962, 1992.

[79] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[80] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.

[81] Michael Cramer. The dgpf-test on digital airborne camera evaluation–overview and test design. *Photogrammetrie-Fernerkundung-Geoinformation*, 2010(2):73–82, 2010.

[82] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace–Beltrami spectra as 'shape-dna'of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.

[83] A Ben Hamza. A graph-theoretic approach to 3D shape classification. *Neurocomputing*, 211:11–21, 2016.

[84] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1704–1711. IEEE, 2010.

[85] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.

[86] Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006.

[87] Majid Masoumi, Chunyuan Li, and A Ben Hamza. A spectral graph wavelet approach for nonrigid 3D shape retrieval. *Pattern Recognition Letters*, 83:339–348, 2016.

[88] S. K. Narang, Y. H. Chao, and A. Ortega. Graph-wavelet filterbanks for edge-aware image processing. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 141–144, Aug 2012.

[89] Philippe Blanchard and Dimitri Volchenkov. *Random Walks and Diffusions on Graphs and Databases: An Introduction*, volume 10. Springer Science & Business Media, 2011.

[90] Dorina Thanou, Xiaowen Dong, Daniel Kressner, and Pascal Frossard. Learning heat diffusion graphs. *arXiv preprint arXiv:1611.01456*, 2016.

APPENDIX A

SUPPLEMENTARY MATERIAL ON MULTI-RESOLUTION APPROACHES ON GRAPHS

Introducing the graph signals yields a requirement of effective description of this new signal form. In particular, multi-resolution representation techniques on vertex and graph spectral domain could give the desired local and global information for the signal. Wavelet transform is one of the traditional and successful approaches in the literature, since it has the property of localizing the signal both in space and frequency at the same time. Therefore, the pioneering studies in graph signal processing have transposed this approach to graph domain by means of windowed graph Fourier transform [29], graph wavelet filterbanks [88] and spectral graph wavelet transform (SGWT) [6].

There are other multiresolution approaches extended to general space of domains, such as graphs, manifolds or meshes. For instance, diffusion wavelets developed by Coifman et al. focus on diffusion operators for multi-resolution analysis [86]. Diffusion operators are emerged from the heat diffusion process defined on various structures. Therefore, there are some studies employing heat kernel based filters for multi-scale representations as well as smoothing purposes [26].

All of these approaches generate various representations of signal in terms of spread and localization on both spatial and frequency domain which might lead to multi-scale and localized descriptors of a graph signal and in turn answers the needs of many pattern recognition problems.

A.1 Diffusion on Graph

Let T denote the diffusion operator acting upon the functions defined on a graph where the impact of diffusion can be extended by means of application of its dyadic powers. The diffusion process can be considered as a stochastic process modeled by the graph Laplacian. Then, one can adjust the quantity of information flow across the graph structure using operator T and create multi-scale representations. In (3.8), we represent a k-hop filtering operation in terms of k degree polynomial of the Laplacian matrix. Similarly, we can dilate the diffusion operator T in vertex domain by taking its powers.

An example of such a diffusion operator is given below, which is stimulated by the transition of random walks on graphs.

$$T = I - \beta L \tag{A.1}$$

for $0 < \beta \leq 1$, T describes a diffusion process on the graph [89]. Moreover, this diffusion operator is associated to the Markov matrix which explains the stochastic transitions on the graph [86].

Another example of T quantifies a dynamical system on a manifold to the discrete input spaces.

$$T = \mathrm{e}^{-\beta L} \tag{A.2}$$

These operators can dilate a signal through the diffusion process depending on its underlying geometry [77]. Higher powers of the operator yield lower numerical rank, which proposes a compression on the function or geometry. Therefore, coarse to fine representations can be expressed via the powers of diffusion operator. Diffusion mechanism is also employed for regularization of graph signals by exploiting the anisotropic diffusion property supported by the graph structure, which will be mentioned in Chapter 5.

When the distribution of the data set is modeled in Gaussian form, then the natural diffusion process appears as a heat flow [10], which will be clarified in the next section.

A.2 Heat Kernel on Graph

The natural diffusion processes induced on graphs are emerged from the heat equation below,

$$\frac{\partial H_t}{\partial t} + LH_t = 0 \tag{A.3}$$

This differential equation presents a solution family of the form $H_t = \mathrm{e}^{-tL}$ where $t > 0$ can be considered as time instants of the heat flow process. H_t is an $N \times N$ operator and can be viewed as the matrix notation of the heat kernel on graph, $H_t = \hat{h}_t(L)$. Accordingly, we can define a heat distribution function $h(x,t)$ representing the amount of heat at time t and on vertex x on a graph $\mathcal{G}$. Then the spectral decomposition of the heat distribution function can be written in terms of the eigensystem of graph Laplacian as follows,

$$h_t \in \mathcal{L}^2(\mathcal{V}) \quad \Rightarrow \quad h(x,t) = \sum_{l=0}^{N-1} \mathrm{e}^{-t\lambda_l} e_l(x) \tag{A.4}$$

Consider a heat transfer function $u_t(x,y)$ which represents the heat flow from point x to point y. Reinterpreting the heat distribution function in terms of the heat transfer function, we obtain,

$$h_t(x) = \sum_{y \sim x} u_t(x,y)$$

Assume that the initial condition for heat distribution is given as dirac delta function, $h_0(x) = \delta(x)$, and associated heat flow $u_0(x,y) = \delta(x,y)$. Recall that the graph Laplacian has an orthornormal set of basis then the initial heat transfer can be expressed as $\delta(x,y) = \sum_{l=0}^{N-1} e_l(x)e_l(y)$. Therefore, the spectral domain expansion of the heat flow is given as follows,

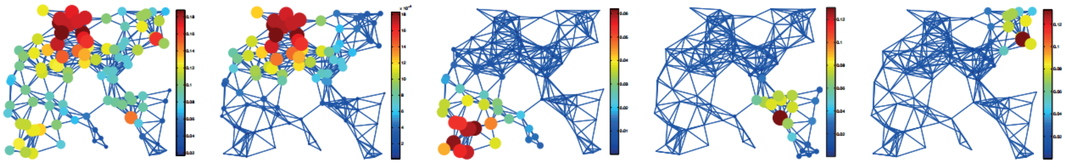
$$u_t(x,y) = \sum_{l=0}^{N-1} \mathrm{e}^{-t\lambda_l} e_l(x)e_l(y)$$

Setting these initial conditions, the heat transfer has an explicit space domain expression as a Gaussian function [10].

$$h_t(x,y) = \frac{1}{4\pi t}\exp(-\frac{d^2_{\mathcal{G}}(x,y)}{4t})$$

where $d_G(x,y)$ is the graph distance from vertex x to vertex y. Thus we can infer that the vertex domain distribution of the heat kernel $\hat{h}(\lambda) = \mathrm{e}^{-t\lambda}$ yields a Gaussian relationship between the data points.

At $t = 0$, the heat distribution is like an impulse and as t increases, heat spreads through the graph structure. Furthermore, heat diffusion can be initiated at different locations of the graph. A column vector of the heat diffusion operator H_t, reveals the vertex domain distribution of a localized kernel as indicated in Equation (3.11). For example, in Figure A.1, a network data signal is approximated by sum of different scale of heat kernels emerged from distinct points of the graph. In [90], this technique is employed for recovering the graph structure by simulating a graph signal in terms of multi-scale and localized heat kernels. Similarly, heat kernels are widely used by manifold learning and computer graphics communities for shape retrieval purposes [77].



(a) Graph signal (b) $e^{-15\mathcal{L}}\delta_4$ (c) $e^{-15\mathcal{L}}\delta_{30}$ (d) $e^{-5\mathcal{L}}\delta_{80}$ (e) $e^{-5\mathcal{L}}\delta_{50}$

Figure A.1: Decomposition of a graph signal (a) in terms of localized heat kernels (b), (c), (d), (e) `Adapted from [90]`

A.3 Spectral Graph Wavelet Transform

As explained in the previous sections, diffusion or heat kernel based operators intend to produce multi-scale transforms through the dilation on vertex domain. On the other hand, the wavelet designs based on the spectral domain of the graph are determined by the dilations of a graph kernel on the graph spectrum. Different scale of wavelets generated on the graph Fourier domain are then localized on any vertex of the graph just as we perform for any graph kernel (3.11).

The spectral graph wavelet transform (SGWT) proposed by Hammond et al. [6] employs a band-pass kernel $\hat{g}(\lambda)$ as a wavelet generating function and a low-pass kernel $\hat{h}(\lambda)$ as a scaling function. Thus, it covers all the spectrum content of graph, accordingly all the frequency components are represented in the transform at some level so that it yields a redundant and invertible transform.

Let the wavelet operator determined by the band-pass filter be denoted by $\Psi_g = \hat{g}(L)$.

By scaling it on the graph spectral domain, the wavelet operator at scale t is given by $\Psi_g^t = \hat{g}(tL)$ for scales $\{t_1, t_2, \cdots, t_S\}$. Likewise, the operator determined by the low-pass filter is designated by $\Psi_h = \hat{h}(L)$.

A wavelet localized on vertex i is computed by applying the wavelet operator to an impulse on vertex i as follows,

$$\psi_{t,i} = \Psi_g^t \delta_i$$

As a result, the entire transform consists of S level wavelet operator and an operator produced by the scaling function, that is indicated as $\Psi^{\text{SGWT}} : \mathbb{R}^N \to \mathbb{R}^{N(S+1)}$ and given by,

$$\Psi^{\text{SGWT}} = [\Psi_h; \Psi_g^{t_1}; \cdots; \Psi_g^{t_S}]$$

The SGWT presents multispectral, localized and overcomplete representation of a graph signal. Hence it constitutes a powerful tool for compression problems in addition to providing effective multiresolution feature descriptions on graph.