Graph Embedding and Arbitrarily Shaped Clustering for Unsupervised Image Segmentation

by

Zhiding YU

A Thesis Submitted to

The Hong Kong University of Science and Technology

in Partial Fulfillment of the Requirements for

the Degree of Master of Philosophy

in The Department of Electronic and Computer Engineering

August 2012

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

Zhiding YU

Graph Embedding and Arbitrarily Shaped Clustering for Unsupervised Image Segmentation

by

Zhiding YU

This is to certify that I have examined the above MPhil thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the Thesis Examination Committee have been made.

_____
Prof. Oscar C. Au (ECE), Thesis Supervisor

_____
Prof. Chin-Tau Lea (ECE), Thesis Committee Chairman

_____
Prof. Long Quan (CSE), Thesis Committee Member

_____
Prof. Ross Murch (ECE), Department Head

Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
August 2012

Acknowledgements

Foremost, I would like to express sincere thanks and gratitude to my M.Phil. advisor, Prof. Oscar C. Au, for giving me the opportunity to conduct research study at Multimedia Technology Research Center (MTrec). It becomes one of the most fruitful and treasurable time in my life. Prof. Au not only is a good research advisor, but also in many ways a nice mentor. His kind guidance, patience and encouragement helped to boost my current research. His generosity on research study gave me the freedom to work on my favorite research topic, enrich my academia experience and broaden my research horizon. And his advice on the right attitude towards research and good communication will keep benefitting my prospective career.

Moreover, I wish to give thanks to my research intern advisors, Dr. Chunjing Xu, Prof. Jianzhuang Liu and Prof. Fernando De la Torre. We had wonderful and fruitful collaborations at Shenzhen Institutes of Advanced Technology (SIAT) Chinese Academy of Sciences, and at CMU. It is these experiences that further enhanced my research knowledge significantly. Thanks to Prof. Ming-Ting Sun at University of Washington for the kind advice and recommendation on my graduate school application. My thanks also go to Prof. Bert Shi, for his patient and detailed research advice during my M.Sc. study at HKUST, prior to M.Phil. enrollment.

Thanks to Prof. Chin-Tau Lea and Prof. Long Quan at HKUST, for their time and effort serving as committees, reviewing my thesis and giving insightful comments.

I would also like to thank all my colleagues at MTrec, SIAT and CMU, and my friends Xing Wen, Jiayi Xian and Zhengwu Zhang for their friendship and help.

My special thanks go to my girlfriend, Wenbo Liu, for her love and encouragement which become the most important stimulus for every progress I made. Last, but not least, my deepest gratitude goes to my parents, for their constant, unconditioned love, comfort, and support.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

MST	Minimum Spanning Tree
RAG	Region Adjacency Graph
EGS	Efficient Graph based Image Segmentation

LIST OF FIGURES

LIST OF TABLES

Graph Embedding and Arbitrarily Shaped Clustering for Unsupervised Image Segmentation

by Zhiding YU

Department of Electronic and Computer Engineering

The Hong Kong University of Science and Technology

Abstract

Image segmentation refers to the process of grouping pixels into spatially continuous regions based on certain similarity measure. One could find its considerable applications in computer vision since it plays a key role in bridging low level information to high level semantic information. Segmentation remains one of the fundamental computer vision problems. Depending on whether there is human interaction, image segmentation can be divided into interactive and non-interactive ones. Based on the learning mode, image segmentation can be classified into supervised and unsupervised ones.

In this thesis, we address the problem of non-interactive, unsupervised image segmentation, aiming at grouping perceptually similar pixels or superpixels into regions. It is expected that segmentations can serve as intermediate input for many high level scene understanding tasks, and that better segmentation results potentially lead to more accurate interpretation of objects and scenes. However, the problem remains challenging in the sense that it is very difficult to model and design methods with segmentation performance that matches human level accuracy without any human interaction or strong prior - constraints that are often rare or even unavailable under real circumstances. In addition, finding a good image partitioning often results in the searching in a very complex or high dimensional solution space. So computational complexity becomes another key issue considering the real implementation of image segmentation algorithms.

Regarding these challenges, this research proposes novel models that improve the

state of the art unsupervised segmentation performance within the scope of acceptable computational cost, where segmentation performance is subjectively evaluated. The purpose is to generate image partitioning results that subjectively suffer less from both oversegmentation and overmerging, both of which can seriously degrade the performance of subsequent applications. Our research effort mainly focuses on graph theory and mode seeking, on top of which better boundary estimation and clustering models are proposed for image segmentation. We believe graphs often carry important structural information. And we want to utilize this information to guide data clustering and image segmentation.

The segmentation accuracy depends largely on both the clustering performance and the designed feature. For clustering, we propose a graph-based contour finding method and minimum spanning tree (MST) embedded mode seeking. An MST is a connected graph that preserves intrinsic compact structures of a data set. It is able to find manifold-like structures or elongated clusters since it shares common features with k-nearest neighbor (k-NN) graph - a graph that has been widely used in finding manifold structures. Thus the MST embedded mode seeking shows larger flexibility and tolerance on cluster shapes. On the image domain, MST serves as a good spatial smoothness constraint in image domain. The introduction of MST significantly helps to boost the performance of image segmentation and can be extended to 3D point cloud object segmentation. For segmentation feature, we propose "Bag of Textons" - a histogram feature descriptor that conveys textual information. We further propose a novel mode seeking framework - called convex shift - to perform constrained mode seeking in histogram space. We show that each kernel shift step can be formulated as a constrained convex optimization problem. The proposed framework produces results that match or outperform the state of the art methods and is compatible with graph-embedding. It is expected that in combination with the embedding of an MST, a further boost in the segmentation performance can be achieved.

CHAPTER 1

INTRODUCTION

In computer vision, segmentation refers to the process of partitioning a digital image into multiple segments. It is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. The goal of segmentation is to simplify and change the representation of an image into something that is easier and more meaningful to analyze. By locating objects and boundaries (lines, curves, etc.) in an image, low level features can be grouped or separated such that higher level information can be better extracted. General segmentation proves to be difficult and nontrivial, even though humans are quite able to handle similar tasks very easily. Currently, results obtained by computer segmentation without any human interaction remain far inferior to what human perceives to be a good segmentation with semantic meanings. The underlying reason is that human segmentation by nature is hierarchical, which consists of a presegmentation stage with human perceptual grouping, followed by higher level semantic understanding stages.

Depending on whether there is human interaction, image segmentation can be divided into interactive and non-interactive ones. The purpose of human interaction is to incorporate human knowledge to provide additional segmentation constraints - either soft or hard - so that machines could overcome segmentation errors. Segmentation can also be classified into supervised and unsupervised ones based on the learning mode. Unsupervised segmentation is to segment the image into regions based on Gestalt Laws of perceptual grouping [84–86], whereas supervised segmentation aims at incorporating segmentation with additional priors learned from training data to guide the segmentation. These priors may include object class information and its possible shape, color, or texture. They serve as additional constraints imposed to the segmentation solution space, or help to inference correct object labels in joint recognition and segmentation

of multiple objects. It is anticipated that the incorporation of supervision information or priors, in addition to perceptual similarity, will guide the algorithm towards more accurate segmentation. This is only true, however, provided that the incorporated priors are very accurate, which is not likely in general recognition-segmentation problems. Previous works [80–83] have investigated top-down segmentation which incorporates object class priors, but only on images with class specific knowledge. In other words, one already possesses this information that the input image contains certain class of object, whereas in a more general case this information is often unavailable unless with recognition and detection. Another family of methods [66] address the problem of scene parsing where each object is inferenced with class label. However, the current state of the art performance is also not satisfactory due to the limitation of recognition.

1.1 Previous Methods

On the contrary, unsupervised segmentation can serve as intermediate input for many high level scene understanding tasks, leading to more accurate object interpretation. A good unsupervised segmentation can provide regional support that improves object recognition performance, and consequently benefits supervised image segmentation. In this thesis, we address the problem of non-interactive, unsupervised image segmentation, aiming at grouping perceptually similar pixels or superpixels into regions. In a sense, it is similar to the human perceptual grouping in the early stage of scene understanding. Despite the fact that we only consider Gestalt Laws of perceptual grouping, the problem remains challenging for it is not easy to model and design a similarity measure meeting what humans visually perceive to be similar. Currently our knowledge about human visual perception and brain reaction to some extent still remains partial. Even though current mathematical models could partially fit the observations, further improvements are necessary. In addition, finding a good image partitioning often results in the searching in a very complex or high dimensional solution space. So computational complexity becomes another important issue considering the real implementation of image segmentation algorithms. A number of previous methods

have been proposed to address unsupervised segmentation. Canonical segmentation methods include, but are not limited to the following ones:

1.1.1 Early Segmentation Techniques

The earliest segmentation techniques were based on gray-level similarity, designed to locate simple objects [87]. The task of the algorithm would be to identify contiguous pixels with similar gray-level value, and group them into regions. Several classifications have been proposed for algorithms that are based on gray-level pixel similarity, and the number of existing variations of these methods number well into the hundreds. Comprehensive reviews of early segmentation techniques can be found in [88], and [89]. Among them, there are two important broad classes of segmentation methods: Gray level thresholding, and region growing/merging techniques.

1.1.2 Feature Space Analysis

Image thresholding using histogram can be considered as a special case of feature space analysis. Feature space information, such as density, cluster and large margins contains important information for modeling and generating image segments. Segmentation based on feature space analysis include the segmentation methods based on nonparametric density estimation and mode seeking [24, 32–34, 40, 59, 60], or other clustering methods such as k-means, expectation maximization or maximum margin clustering [90].

1.1.3 Energy minimization

These methods often formulate image segmentation as an energy minimization problem within the framework of Markov Random Field (MRF) [8, 29]. The labels of the image pixels are inferenced such that the cost of certain designed energy function can be minimized. The energy function often consists of a likelihood term that encourages intra-region similarity, and a smoothness term that imposes smoothness on the label and penalizes small, discrete regions.

1.1.4 Graph Theoretic Algorithms

Graph theoretic algorithms all treat image as a graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$, in which $\mathbf{V}$ is a set of vertices corresponding to image elements (which may be pixels, feature descriptors, and so on), and $\mathbf{E}$ is a set of edges linking vertices in the graph together. The weight of an edge $w_{i,j}$ is proportional to the similarity between the vertices v_i and v_j and is usually referred to as the affinity between elements i and j in the image. The goal is to find and remove a set of edges such that the graph is partitioned. The way one chooses to partition the graph include the use of MST [14], Max-Flow Min Cut [16,57], spectral clustering [15] and other methods that find a "cut" of the graph by minimizing certain cost functions [23].

1.1.5 Deformable Contours

Deformable contours were introduced as a technique for interactive image segmentation by Kass et al., [91]. A deformable contour (also known as active contour, or snake) consists of a parametric curve that is attracted to image features such as edges, lines, or corners. This attraction is quantified by an energy function that has low values at places in the image that contain the desired features. The energy function also incorporates a term that depends on the shape of the curve, and can be used to bias the curve to take on smooth shapes (which is something that previously discussed algorithms can not do, as they dont model the shapes of region boundaries explicitly), and a term that depends on user interaction, so that the curve can be interactively pulled toward a desired location.

1.2 Our Methods

Regarding the challenges, the purpose of this research is to propose novel models for this problem and improve the state of the art segmentation performance within the scope of acceptable computational cost. Our research effort mainly focuses on graph theory and mode seeking, on top of which better boundary estimation and cluster-

ing models are proposed for image segmentation. Boundary estimation aims at finding contours (or boundaries) of regions by modeling inter-region difference. While it tends to show better tolerance towards intra-region variation, the drawback is that "region leak" often occurs due to weak boundaries, causing serious overmerging. We address the problem of boundary estimation by formulating it as inter-region contour and intra-region information analysis in the framework of graph-based segmentation. The defined region comparison predicate makes a better boundary estimator than efficient graph-based image segmentation (EGS) [36] - a well known and widely used segmentation method. It will be shown in this thesis that our method better alleviates the "region leak" problem since we used the maximum likelihood inter-region difference, rather than the weakest inter-region difference adopted in EGS. We further illustrate, by making a small relaxation, further improvement of segmentation performance can be achieved. Experimental results have demonstrated the effectiveness of our proposed method.

On the other hand, clustering-based segmentation aims at finding regions through elaborate design of intra-region similarity metrics. Compared with boundary estimation, segmentation by clustering tends to suffer less from overmerging, but often at the cost of generating oversegmentation due to the lack of flexibility with cluster model. Since features of an intra-region components usually show an arbitrary shape in the feature space, the fact that arbitrarily shaped clustering possesses higher cluster flexibility makes it perform better than many other clustering methods assuming regular cluster shapes. In this thesis, we present a novel framework for graph-embedded mode seeking and its fast approximation. The density estimation function is defined as a joint representation of the feature space and the distance domain on the graph's minimum spanning tree. The graph-embedded density estimator endows: 1 Even more flexibility in dealing with arbitrarily shaped or manifold-like data distributions; 2 Better spatial consistency constraint for image segmentation. 3. Strong compatiblity with region-wise operations that allows more elaborate feature descriptors, in addition to pixel-wise operations. The mode seeking method corresponding to the new density

estimator is inferenced and fast approximation is proposed, based on which one could accordingly perform data clustering and image segmentation.

We also introduce transductive distance that projects large margins in the MST space. As an interesting application we present a system that can automatically segment objects in large scale 3D point clouds obtained from urban ranging images. The system consists of three steps: The first one involves a ground detection process that can detect relatively complex terrain and separate it from other objects. The second step superpixelizes the remaining objects to speed up the segmentation process. In the final step, graph-embedded mode seeking method is adopted to segment the point clouds. The projected transductive distance on MST effectively improves the segmentation performance due to the fact that continuous artificial objects often have manifold-like structures.

Selecting good features and distance metric makes another key issue in designing visual similarity measures. In this thesis, we further propose a method based on texton similarity and a modified mode seeking - called convex shift - to group superpixels and generate segments. The distribution of histograms is modeled nonparametrically in the histogram space, using Kullback-Leibler divergence (K-L divergence) and kernel density estimation. We show that each kernel shift step can be formulated as a convex optimization problem with linear constraints. Convex shift can effectively perform mode seeking on an enforced histogram structure, grouping visually similar superpixels and generating good results on natural images with relatively complex contents. It shows significant superiority over traditional mode seeking based segmentation methods, while outperforming or being comparable to the state of the art methods. Convex shift can be extended to other convex distance metrics such as Jeffrey Divergence. In addition, it is compatible with the scheme of graph-embedded mode seeking. With the better spatial constraint imposed by graph-embedded mode seeking, an even larger boost of segmentation performance is expected, which will be included in our future work.

CHAPTER 2

PRELIMINARIES

One major contribution of this thesis is the embedding of a tree structure into density estimation and mode seeking - a novel framework that facilitates the manipulation of mode seeking characteristics, and consequently, improves the segmentation performance. While two methods can find diverse applications in data mining, machine learning and computer vision, the combination of them is seldom reported. Before developing our methods, some related background need to be addressed.

2.1 Graph Theory and Minimum Spanning Tree

A graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$ is composed of a set of nodes, or vertices $\mathbf{V}_i$ connected to each other by edges $\mathbf{E}_{i,j}$, where $\mathbf{V}_i$ and $\mathbf{V}_j$ are the terminal nodes that the edge connects. In a weighted graph the vertices and links have weights associated with them, v_i, and $w_{i,j}$, respectively. Each node need not necessarily be linked to every other, but if they are then the graph is complete. A partial graph has the same number of nodes but only a subset of the edges of the original graph. A subgraph has only a subset of the nodes of the original graph, but contains all the links whose terminal nodes are within this subset.

A "chain" is a list of successive nodes in which each vertex is connected to the next by an edge in the graph. A "cycle" is a chain whose end links meet at the same node. A "tree" is a connected set of chains such that there are no cycles. A 'spanning tree' is a tree which is also a partial graph. A 'shortest spanning tree' of a weighted graph is a spanning tree such that the sum of its edge weights, or some other monotonic function of its edge weights, is a minimum for any possible spanning tree. A "forest" is a set of trees disjointed with each other, and a "spanning forest" is a forest which is also a

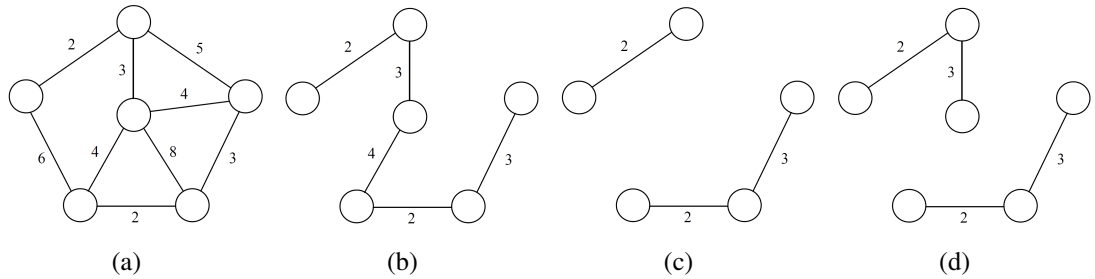


Figure 2.1: Examples of different classes of graphs where circles represent graph nodes, lines denote graph edges and numbers beside edges indicate edge weights. (a) Original graph. (b) The MST of (a). (c) A forest. (d) Spanning forest obtained by cutting one of the tree edges in (b).

partial graph. Figure 2.1 respectively shows an example graph, the MST of this graph, the forest and the spanning forest.

Both Prim Algorithm and Kruskal Algorithm are designed to extract the MST from an input graph, with the execution time proportional to $O(\mathbf{E}\log(\mathbf{V}))$. In this thesis, we re-implemented the Kruskal Algorithm to search for MST. The Kruskal Algorithm can be described as follows:

1. Initially the forest contains no links.
2. Repeat:

   Find the next least-weighted link.

   If The link would not form a cycle with the forest:

   Add the link to the forest.

   Else Discard the link.

   Until the forest becomes a tree spanning the graph.

To carry image segmentation using MST, one first constructs a region adjacency graph (RAG) where each image pixel or superpixels obtained by oversegmentation correspond to graph nodes [14, 25]. Spatial neighboring relationship between pixels or superpixels are represented by edges and the inter-pixel or inter-superpixel dissimilarity correspond to the edge weights. The concept of MST based segmentation is

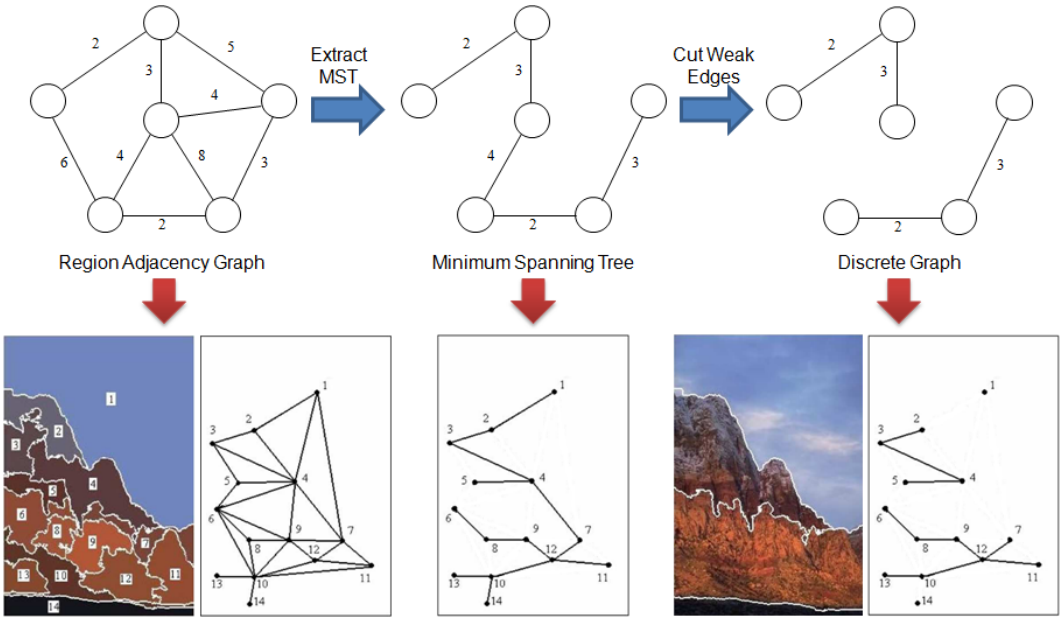


Figure 2.2: An example showing the process of MST based segmentation.

straight forward: after the extraction of MST on an RAG, image pixels/superpixels are only connected with the strongest edges without any circles formed. Any cut of an edge will partition the MST into a forest consisting of two subtrees. Consequently, the image itself is partitioned into segments. In addition, simultaneous K-way cut can be easily carried out by cutting K-1 tree edges and forming a forest of K subtrees. Edges that are relatively weak in the tree are cut with priority since we assume regions merged together should have strong similarity and connection. An example of image segmentation by MST is illustrated in figure 2.2.

2.2 Arbitrarily Shaped Clustering

Mode seeking based clustering can be regarded as one kind of arbitrarily shaped clustering methods since it does not assume the cluster shape, which makes it distinct from methods such as k-means that assume regular cluster shapes. Mode seeking provides a versatile tool for feature space analysis. One can find diverse applications in computer vision problems where feature space analysis plays a crucial and indispensable role. Examples of possible applications include image smoothing, segmentation, object tracking [39, 69, 70] , key frame detection and image clustering.

2.2.1 Nonparametric density estimation

As pointed out by Comaniciu et al., "feature space can be regarded as the empirical probability density function(pdf) of the represented parameter" [24]. Given a set of independent and identically distributed (i.i.d.) data $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ drawn from some unknown distribution, nonparametric density estimation seeks to approximate the probability density function $p(\mathbf{x})$. Instead of representing $p(\mathbf{x})$ by a single parametric model or a mixture model, the method finds a small number of nearest (or most similar) training instances and interpolate from them, which can be represented as follows:

$$\hat{p}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i), \tag{2.1}$$

where $\mathbf{H}$ is a symmetric positive definite $d \times d$ bandwidth matrix and:

$$K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2}\mathbf{x}). \tag{2.2}$$

The d-variate kernel $K(\mathbf{x})$ is a bounded function with compact support satisfying

$$\int_{R^d} K(\mathbf{x})\mathrm{d}\mathbf{x} = 1 \quad \lim_{\|\mathbf{x}\|\to\infty} \|\mathbf{x}\|^d K(\mathbf{x}) = 0$$
$$\int_{R^d} \mathbf{x}K(\mathbf{x})\mathrm{d}\mathbf{x} = 0 \quad \int_{R^d} \mathbf{x}\mathbf{x}^\top K(\mathbf{x})\mathrm{d}\mathbf{x} = c_K\mathbf{I},$$

where c_K is a constant. Equation (2.1) is also known as the Parzen window density estimation. Gaussian kernel is most commonly adopted kernel that generates smooth, differentiable pdf and gives the best performance. In detail, the kernel is mathematically defined as:

$$K_N(\mathbf{x}) = (2\pi)^{-d/2} \exp(-\frac{1}{2}\|\mathbf{x}\|^2) \tag{2.3}$$

An example of density estimation using Gaussian kernel is illustrated is Fig. 2.3:

2.2.2 Mode seeking

Local density maxima are also known as modes. In feature space analysis, one is often interested in finding high density points, while low density points are of less

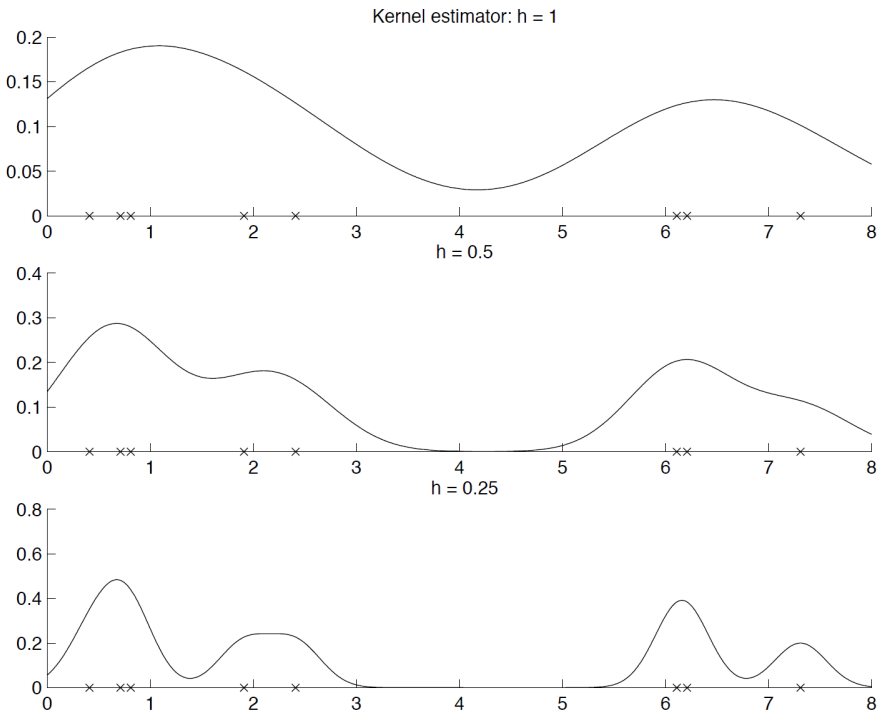


Figure 2.3: Example of density estimation using Gaussian kernel. The vertical axis represents the pdf $p(\mathbf{x})$.

importance. Increasing the density estimation leads to mode seeking, a process exactly like its literal meaning: finding the modes. The intuition is that data belonging to the same cluster are assumed to fall within the same density attraction basin, where the attraction force points to the direction that mostly increases the the estimated density. In other words, the feature space can be partitioned by several clusters or "basins" with density maxima being the cluster centroids or the lowest points of basins. The process is exactly dropping a water (data piece one wants to cluster) into a basin, and the water will surely goes to the basin bottom, attracted by gravity (density). Fig. 2.4 illustrates a vivid example of mode seeking based clustering.

2.2.3 Mode seeking using mean shift

Mean shift is regarded as one of the most canonical mode seeking algorithms with numerous real applications in computer vision. The idea was first proposed in [59] in 1975 and then generalized in [10, 60] in 1995, but has not received wide attention until the publication of [24] in 2002. The algorithm is basically a density gradient ascent

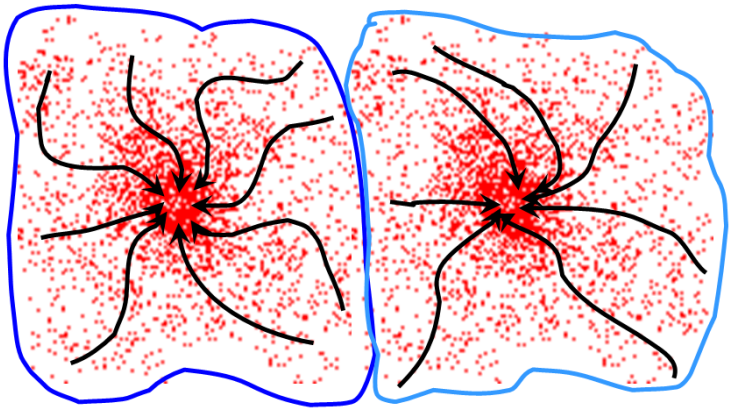


Figure 2.4: Example of the attraction basins.

method that tries to find the modes. Thus the core problem becomes estimating the density gradient. Already one has the estimated density representation described in equation (2.1). Since we are only interested in a special class of radially symmetric kernels satisfying:

$$K(\mathbf{x}) = c_{k,d}k(\|\mathbf{x}\|^2), \tag{2.4}$$

where the function $k(x)$ is called the *profile* of the kernel and $c_{k,d}$ is a normalization constant that makes the $K(\mathbf{x})$ integrates to one. Using a fully parameterized $\mathbf{H}$ increases the complexity and the author only considers the Euclidean metric. The estimated density function thus can be rewritten in the following well-known expression:

$$\hat{p}_{h,K}(\mathbf{x}) = \frac{1}{Nh^d}\sum_{i=1}^{N} K(\frac{\mathbf{x}-\mathbf{x}_i}{h}) = \frac{c_{k,d}}{Nh^d}\sum_{i=1}^{N} k(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2). \tag{2.5}$$

where $K(\cdot)$ is the kernel - a symmetric but not necessarily positive function that integrates to one - and $h > 0$ is a smoothing parameter called the bandwidth. Taking the derivative of $\hat{p}(\mathbf{x})$ with respect to $\mathbf{x}$, one has:

$$\hat{\nabla}p_{h,K}(\mathbf{x}) \equiv \nabla\hat{p}_{h,K}(\mathbf{x}) = \frac{2c_{k,d}}{Nh^{d+2}}\sum_{i=1}^{N}(\mathbf{x}-\mathbf{x}_i)k'(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2). \tag{2.6}$$

Here we define the new kernel profile $g(x) = -k'(x)$ and its related kernel $G(\mathbf{x}) = c_{g,d}g(\|\mathbf{x}\|^2)$, where $c_{g,d}$ is the corresponding normalization constant. Notice that kernel $K(\mathbf{x})$ was called the shadow of the $G(x)$. Introducing $g(x)$ into the estimated density

gradient, the estimated gradient now becomes:

$$\begin{aligned}\nabla\hat{p}_{h,K}(\mathbf{x}) &= \frac{2c_{k,d}}{Nh^{d+2}}\sum_{i=1}^{N}(\mathbf{x}_i-\mathbf{x})g(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2) \\ &= \frac{2c_{k,d}}{Nh^{d+2}}[\sum_{i=1}^{N}g(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2)][\frac{\sum_{i=1}^{N}\mathbf{x}_i g(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2)}{\sum_{i=1}^{N}g(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2)}-\mathbf{x}]\end{aligned} \tag{2.7}$$

Observe that the first term is proportional to the density estimate at $\mathbf{x}$ with the kernel G:

$$\hat{p}_{h,G}(\mathbf{x}) = \frac{c_{g,d}}{Nh^d}\sum_{i=1}^{N}g(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2) \tag{2.8}$$

The second term of equation (2.7) is the *mean shift*

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{N}\mathbf{x}_i g(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2)}{\sum_{i=1}^{N}g(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2)}-\mathbf{x} \tag{2.9}$$

Rewrite equation (2.7) we have:

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{1}{2}\frac{h^2 c_{g,d}}{c_{k,d}}\frac{\nabla\hat{p}_{h,K}(\mathbf{x})}{\hat{p}_{h,G}(\mathbf{x})} \tag{2.10}$$

equation (2.10) shows that at location $\mathbf{x}$, the mean shift vector computed with kernel G is proportional to the normalized density gradient estimate obtained with kernel K. The normalization is by the density estimate in $\mathbf{x}$ computed with the kernel G. This means that the mean shift vector always points toward the direction of maximum increase in the density.

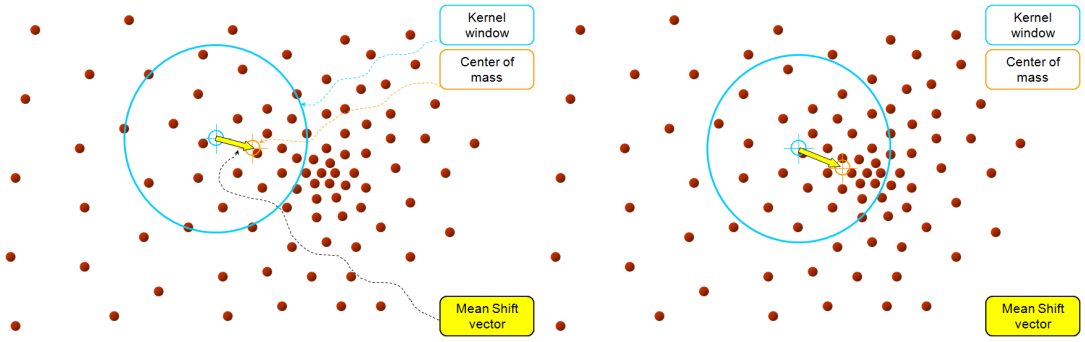
Calculating $\mathbf{m}_{h,G}(\mathbf{x})$ is the most significant step in mean shift since it indicates how one should shift the kernel window in the feature space. The clustering procedure of mean shift is to:

1. For each data piece in the feature space, initialize the kernel window with its center located on the data.

2. Recursively calculate $\mathbf{m}_{h,G}(\mathbf{x})$ and update $G(\mathbf{x})$ with the new kernel window location until convergence. Associate the final kernel position with the data piece.

3. Assign data pieces that have close final kernel positions to the same cluster.

equation (2.9) shows that the second term is actually the difference between the data



(a) (b)

Figure 2.5: Example of the mean shift process.

mean weighted by kernel G and the current kernel window center $\mathbf{x}$. This is intuitively meaningful since it is closely related to the first order estimation of the feature space data. Fig. 2.5 illustrates the procedure of mean shift.

Several important aspects make mean shift the most popular mode seeking method. First, the related theory is very elegant and is intuitively meaningful. An important theorem concerning mean shift is that unlike many traditional gradient descent methods which requires sophisticated step control operations (eg. back tracking line search), there is no such requirement for in mean shift. The mean shift vector $\mathbf{m}_{h,G}(\mathbf{x})$ which is proportional to the estimated density gradient is normalized by the density $\hat{p}_{h,G}(\mathbf{x}$. It is proved that for any kernel K that has a convex and monotonically decreasing profile, the sequence of kernel locations shifted with respect to $\mathbf{m}_{h,G}(\mathbf{x})$ will converge and the estimated density is monotonically increasing. One can prove this theorem utilizing the first order condition of a convex function.

Second, mean shift demonstrated huge application potential and robust performance. Mean shift segmentation has been one of the most canonical segmentation method and is utilized by many researchers as a benchmark for segmentation comparison. Even though from today's perspective, there exist more sophisticated methods that outperforms segmentations produced by mean shift, the produced results at that time showed the state of the art performance.

There are some other nice characteristics as well. For example, mean shift parallel-friendly and can be easily accelerated in real implementation.

2.2.4 Discontinuity preserved smoothing and image segmentation

The fact that mode seeking methods can perform arbitrarily shaped clustering makes it outperform most traditional clustering algorithms assuming regularly shaped clusters in terms of segmentation performance. Traditional mode seeking segmentation is a straightforward extension of the discontinuity preserving smoothing, where each pixel is treated as mode seeking data sample. In addition to the d dimensional feature descriptors $\mathbf{x}_i^f$ of pixels, the 2 dimensional spatial coordinates $\mathbf{x}_i^s$ in the image domain is also considered to impose smoothness constraint. Thus the density estimation function becomes a joint representation of the feature space and the image space:

$$\hat{p}_{h_s,h_f,K}(\mathbf{x}) = \frac{C}{Nh_s^2h_f^d}\sum_{i=1}^{N} k(\|\frac{\mathbf{x}-\mathbf{x}_i^{\mathbf{s}s}}{h_s}\|^2)k(\|\frac{\mathbf{x}^{\mathbf{f}}-\mathbf{x}_i^f}{h_f}\|^2). \tag{2.11}$$

where h_s and h_f respectively correspond to the bandwidth parameters selected for spatial space and feature space. To smooth an image, one initializes a kernel window with the feature descriptor and spatial coordinate of a pixel, and perform mode seeking until convergence. The final converged kernel location in the feature space corresponds to the filtered pixel feature and this is iterated for every pixel in the image. Since mode seeking image smoothing preserves major region boundaries while eliminates intra-region variation, it becomes very easy to group image pixels to form segments.

CHAPTER 3

GRAPH-BASED CONTOUR FINDING

In this chapter, we aim at finding a computationally efficient contour findind method that generates visually good image partitionings. Segmentation often serves as an image simplification stage in the recognition framework. Many higher level tasks such as object recognition and scene understanding typically require regional support. The key issue considered here is: how to maximize the simplification level while preserving its region labeling accuracy. Even though a more simplified image probably conveys a clearer scene structures, one also risks more with false labeling in the segmentation process. Oversegmentation, on the contrary, is an effective strategy to reduce false labeling, as is adopted by a number of literatures. For references we recommend readers look into works concerning superpixel or the watershed algorithm [37, 38], where the methods are intrinsically featured with oversegmentations. The disadvantage with such strategy, however, is that useful information such as shape and object topology are basically discarded. Another issue is the complexity of the designed method. We argue, that the algorithm should be fast, simple to implement and intuitively easy to understand.

Our work is closely related to EGS, a very simple yet effective graph-based image partitioning method proposed by Felzenszwalb et al [36]. For the problem's tractability, the authors defined the least weighted inter-region edge weight as the inter-region difference. This can be problematic, which is pointed out by themselves in the paper. The accompanied problem is oversmoothing due to certain weak object boundaries or camouflage, which happens frequently in real situations. We address this problem by introducing predicate defined with the maximum likelihood (ML) estimation of inter-region dissimilarity and biased thresholding with intra-region variance. We further make a relaxation to balance the segmentation, by introducing mutual volume for the

(a) (b)

Figure 3.1: Segmentations of *Toco Toucan*. (a) Result obtained by EGS. (b) Result obtained by our proposed method

threshold function. Although relaxing the predicate makes the problem intractable, one shall see such relaxation is essential in producing favorable results. Figure 3.1 shows the segmentation examples using EGS and our proposed method.

The following parts of this chapter will give a detailed discussion on the proposed method. In section 3.1, we introduce the ML edge predicate and the biased intra-region variance. We then discuss the properties related to the proposed methods in the next session. Based on the discussion, we also introduce a relaxation to the original problem. Finally, experimental results are illustrated in section 3.3.

3.1 Graph-based Image Segmentation

Our proposed method belongs to the family of graph-based image segmentations. Suppose $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ represents an undirected graph where $\mathbf{V} = \{v_i | i = 1, ..., N\}$ denotes the set of nodes corresponding to image pixels and $\mathbf{E} = \{(v_i, v_j) | (i, j) \in \mathbf{S}_{8-Neigh}\}$ denotes the neighboring node pairs. The set $\mathbf{S}_{8-Neigh}$ indicates the 8 connectivity of pixel pair (i, j) in the image space and each $(v_i, v_j) \in \mathbf{E}$ is related with a non-negative edge weight $w(v_i, v_j)$ which is the dissimilarity measure of the two nodes. The $w(v_i, v_j)$ is chosen as the Euclidean distance between RGB values of neighboring nodes, normalized by $\sqrt{3}$. Our goal is to find $\mathbf{P}$, a partition of $\mathbf{V}$ such that each component (or region) $\mathbf{C} \in \mathbf{P}$ corresponds to a connected component in a graph $\mathbf{G}' = (\mathbf{V}, \mathbf{E}')$, where $\mathbf{E}' \subseteq \mathbf{E}$. One could see the definition of $\mathbf{P}$ is identical to the one in EGS.

3.1.1 Proposed pairwise region comparison predicate

We define a novel pairwise region comparison predicate as the boundary estimator. Instead of choosing the weakest pairwise region difference, one naturally looks into the ML estimation to robustly reject outliers and weak boundary portions. Suppose $\mathbf{C}_1$ and $\mathbf{C}_2 \subseteq \mathbf{V}$ denote components corresponding to two different regions in the image space, the inter-region difference is defined as:

$$Dif(\mathbf{C}_1, \mathbf{C}_2) = \frac{1}{|\mathbf{B}_{C_1,C_2}|} \sum_{(v_i,v_j)\in\mathbf{B}_{C_1,C_2}} w(v_i, v_j), \tag{3.1}$$

where $\mathbf{B}_{C_1,C_2} = \{(v_i, v_j)|v_i \in \mathbf{C}_1, v_j \in \mathbf{C}_2, (v_i, v_j) \in \mathbf{E}\}$ is the set of inter-region links between $\mathbf{C}_1$ and $\mathbf{C}_2$ and $|\mathbf{B}_{C_1,C_2}|$ denotes the cardinality of $\mathbf{B}_{C_1,C_2}$. If $\mathbf{C}_1$ and $\mathbf{C}_2$ are non-adjacent, then $|\mathbf{B}| = 0$ and $Dif(\mathbf{C}_1, \mathbf{C}_2)$ is defined to be ∞.

We also define the intra-region difference as the ML estimation of the intra-region links:

$$Int(\mathbf{C}) = \frac{1}{|\mathbf{E}_C|} \sum_{(v_i,v_j)\in\mathbf{E}_C} w(v_i, v_j), \tag{3.2}$$

where $\mathbf{E}_C = \{(v_i, v_j)|v_i, v_j \in \mathbf{C}, (v_i, v_j) \in \mathbf{E}\}$ corresponds to the set of intra-region links. Intuitively, intra-region difference measures the compactness of a certain region. Our definition here differs from [36] in the sense that such formulation prevents sudden large increase of intra-region difference. In fact, one can also consider defining $Int(\mathbf{C})$ with K-largest $Dif(\mathbf{C}_1, \mathbf{C}_2)$, where $\mathbf{C}_1, \mathbf{C}_2 \subseteq \mathbf{C}$ correspond to regions that are previously merged to form $\mathbf{C}$. Such kind of definition is actually compatible with our framework and is a trade off between [36] and our proposed method. In this chapter, however, we only adopt the ML estimation of all links as the intra-region difference.

To estimate whether there exist a boundary between regions, we define the pairwise region comparison predicate similar to EGS:

$$D(\mathbf{C}_1, \mathbf{C}_2) = \begin{cases} \text{true} & \text{if } Dif(\mathbf{C}_1, \mathbf{C}_2) > Mint(\mathbf{C}_1, \mathbf{C}_2) \\ \text{false} & \text{otherwise} \end{cases}, \tag{3.3}$$

where $Mint(\mathbf{C}_1, \mathbf{C}_2)$ is defined as:

$$\begin{aligned} &Mint(\mathbf{C}_1, \mathbf{C}_2) \\ &= \min(Int(\mathbf{C}_1) + \tau(\mathbf{C}_1), Int(\mathbf{C}_2) + \tau(\mathbf{C}_2)). \end{aligned} \tag{3.4}$$

Here we introduce the notion of biased thresholding with intra-region difference. Instead of simply choosing $\tau(\mathbf{C}) \propto \frac{1}{|\mathbf{C}|}$, we define:

$$\begin{aligned} \tau(\mathbf{C}) &\propto (\frac{Int(\mathbf{C})}{Int(\mathbf{P})})^{\alpha} \frac{1}{f(|\mathbf{C}|)} \\ &= \left[\frac{(\sum_{k=1}^{K} |\mathbf{E}_{C_k}|) \sum_{(v_i,v_j)\in\mathbf{E}_C} w(v_i, v_j)}{|\mathbf{E}_C| \sum_{k=1}^{K} \sum_{(v_i,v_j)\in\mathbf{E}_{C_k}} w(v_i, v_j)} \right]^{\alpha} \frac{1}{f(|\mathbf{C}|)} \end{aligned} \tag{3.5}$$

where α is the parameter controlling the strength of bias and $f(|\mathbf{C}|)$ is a monotonically increasing function of $|\mathbf{C}|$. With the above formulation we add additional adaptivity to the thresholding function with respect to intra-region difference, which is normalized by the weighted average of intra-region difference of all currently formed regions. The intuition here is to introduce merging bias towards textured regions. Indeed, we observe that other than oversmoothing caused by weak object boundary, EGS also tends to oversegment textured regions due to the simple formulation of thresholding function. Thresholding plays a crucial rule in determining the segmentation quality since it is a compensation for the region statistics estimated by $Int(\mathbf{C})$. One could also interpret our formulation of $\tau(\mathbf{C})$ as scaling it with intra-region difference. A region with a large $Int(\mathbf{C})$ is likely to be textured regions. For such kind of regions we encourage the merging of textures by increasing $\tau(\mathbf{C})$, which leads to the potential increase of $Mint$

3.1.2 Segmentation algorithm

Suppose $\mathbf{P}_n = \{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_K\}$ is the current segmentation state and $\mathbf{P}_{n+1}$ is the next state derived from $\mathbf{P}_n$. Given the pairwise region comparison predicate, we define the following algorithm to perform segmentation:

1. Initialize $\mathbf{P}_0$ with each component $\mathbf{C}_k$ representing a single image pixel.

2. Repeat step 3 until all pairwise region adjacency links $(\mathbf{C}_{k_1}, \mathbf{C}_{k_2})$ are considered.

3. Construct $\mathbf{P}_{n+1}$ from $\mathbf{P}_n$ as follows. If there exists any link $(\mathbf{C}_{k_1}, \mathbf{C}_{k_2})$ that has not been considered, find the smallest unconsidered $Dif(\mathbf{C}_{k_1}, \mathbf{C}_{k_2})$. If $Dif(\mathbf{C}_{k_1}, \mathbf{C}_{k_2}) \leq Mint(\mathbf{C}_{k_1}, \mathbf{C}_{k_2})$, merge $\mathbf{C}_{k_1}$ and $\mathbf{C}_{k_2}$. Denote the newly formed region as $\mathbf{C}'$. For any region $\mathbf{C}_{k_3}$ adjacent to $\mathbf{C}_{k_1}$ and $\mathbf{C}_{k_2}$, delete redundant links by merging $(\mathbf{C}_{k_3}, \mathbf{C}_{k_1})$ and $(\mathbf{C}_{k_3}, \mathbf{C}_{k_2})$ into $(\mathbf{C}_{k_3}, \mathbf{C}')$. Update inter-region difference:

$$Dif(\mathbf{C}_{k_3}, \mathbf{C}') = \frac{1}{|\mathbf{B}_{C_{k_3}, C'}|} \sum_{(v_i, v_j) \in \mathbf{B}_{C_{k_3}, C'}} w(v_i, v_j)$$

$$= \frac{Dif(\mathbf{C}_{k_3}, \mathbf{C}_{k_1})|\mathbf{B}_{C_{k_3}, C_{k_1}}| + Dif(\mathbf{C}_{k_3}, \mathbf{C}_{k_2})|\mathbf{B}_{C_{k_3}, C_{k_2}}|}{|\mathbf{B}_{C_{k_3}, C_{k_1}}| + |\mathbf{B}_{C_{k_3}, C_{k_2}}|}$$

Otherwise, $\mathbf{P}_{n+1} = \mathbf{P}_n$. Mark $(\mathbf{C}_{k_1}, \mathbf{C}_{k_2})$ as unconsidered.

4. Return $\mathbf{P}$ and output the segmented image.

3.1.3 Related properties

We analyze some of the properties related to the above algorithm. To tract the segmentation quality, we need to define the fineness and coarseness of the produced result. Here the definition is identical to EGS:

Definition 3.1.1 *A partition $\mathbf{P}$ is too fine if there exist some pair of regions $\mathbf{C}_1, \mathbf{C}_2 \in \mathbf{P}$ for which there is no evidence for a boundary between them.*

Definition 3.1.2 *$\mathbf{P}'$ is a refinement of $\mathbf{P}$ if and only if $\forall \mathbf{C}_i \in \mathbf{P}', \exists \mathbf{C}_j \in \mathbf{P}, \mathbf{C}_i \subseteq \mathbf{C}_j$. $\mathbf{P}'$ is a proper refinement of $\mathbf{P}$ when $\mathbf{P}' \neq \mathbf{P}$.*

Definition 3.1.3 *A partition $\mathbf{P}$ is too coarse when there exists a proper refinement of $\mathbf{P}$ that is not too fine.*

More details and discussions about the definitions can be found in [36] and we will not extend the discussion here. With the above definitions we are now able to evaluate the image partitionings produced by our proposed algorithm:

Lemma 3.1.1 *For any region $\mathbf{C}_{i_m} \in \mathbf{P}_m$, the weakest inter-region difference increases monotonically if $\mathbf{C}_{i_m}$ has not been merged in subsequent operations. In other words,* $\min_{j_n}(Dif(\mathbf{C}_{i_n}, \mathbf{C}_{j_n})) > \min_{j_m}(Dif(\mathbf{C}_{i_m}, \mathbf{C}_{j_m}))$ *if* $n > m$ *and* $\mathbf{C}_{i_n} = \mathbf{C}_{i_m}$.

Proof: Lemma 3.1.1 is a direct result from the inter-region difference updating rule in algorithm step 3. Suppose two regions $\mathbf{C}_{j_1}$ and $\mathbf{C}_{j_2}$ are both adjacent to $\mathbf{C}_i$ and are merged to form $\mathbf{C}_{j'}$. Since the inter-region difference is defined as the set of all the graph edges going across two distinct regions, inter-region difference can be updated by taking the weighted average of $Dif(\mathbf{C}_i, \mathbf{C}_{j_1})$ and $Dif(\mathbf{C}_i, \mathbf{C}_{j_2})$. One thus have: $Dif(\mathbf{C}_i, \mathbf{C}_{j'}) \geq min(Dif(\mathbf{C}_i, \mathbf{C}_{j_1}), Dif(\mathbf{C}_i, \mathbf{C}_{j_2}))$. This leads to the proof of Lemma 3.1.1.

Lemma 3.1.2 *For any considered pair of regions where the regions are not merged, at least one of them will be in the final segmentation.*

Proof: Suppose $Int^{(l)}$ denotes the internal difference of a region in the lth segmentation state. Without loss of generality, also suppose $\mathbf{C}_i$ and $\mathbf{C}_j$ is the pair of regions one is currently considering and $Dif(\mathbf{C}_i, \mathbf{C}_j)$ is the weakest inter-region difference between $\mathbf{C}_i$ and other neighboring regions. Assume:

$$\begin{aligned} Dif(\mathbf{C}_i, \mathbf{C}_j) &> Mint(\mathbf{C}_i, \mathbf{C}_j) \\ &= \min(Int^{(l)}(\mathbf{C}_i + \tau(\mathbf{C}_i)), Int^{(l)}(\mathbf{C}_j + \tau(\mathbf{C}_j))) \\ &= Int^{(l)}(\mathbf{C}_i + \tau(\mathbf{C}_i)). \end{aligned}$$

Suppose $m > l$ and $\mathbf{C}_{j'}$ is the newly merged region in the mth state containing $\mathbf{C}_j$ and another neighboring region. Since inter-region difference is considered in a non-decreasing order and according to Lemma 3.1.1, we have: $Dif(\mathbf{C}_i, \mathbf{C}_{j'}) \geq Dif(\mathbf{C}_i, \mathbf{C}_j), \forall \mathbf{C}_{j'} \neq \mathbf{C}_j$. In addition:

$$\begin{aligned} Mint(\mathbf{C}_i, \mathbf{C}_{j'}) &= \min(Int^{(m)}(\mathbf{C}_i + \tau(\mathbf{C}_i)), Int^{(m)}(\mathbf{C}_{j'} + \tau(\mathbf{C}_{j'}))) \\ &\leq Int^{(l)}(\mathbf{C}_i + \tau(\mathbf{C}_i)) \\ &= Mint(\mathbf{C}_i, \mathbf{C}_j) \end{aligned}$$

In other words, $Dif(\mathbf{C}_i, \mathbf{C}_{j'}) > Mint(\mathbf{C}_i, \mathbf{C}_{j'})$ and no merging will happen to $\mathbf{C}_i$ in subsequent operations. Thus we have proved Lemma 3.1.2

Theorem 3.1.1 *The segmentations produced by the proposed algorithm is always not too fine.*

Proof: The produced segmentation is too fine if there exist some pair of regions, say $\mathbf{C}_i$ and $\mathbf{C}_j$, whose comparison predicate does not hold. Without loss of generality, suppose $\mathbf{C}_{j'} \subseteq \mathbf{C}_j$, $Dif(\mathbf{C}_i, \mathbf{C}_{j'}) = Dif(\mathbf{C}_i, \mathbf{C}_j)$ is considered in step 3 and $Int(\mathbf{C}_i) + \tau(\mathbf{C}_i) \leq Int(\mathbf{C}_{j'}) + \tau(\mathbf{C}_{j'})$. Since $\mathbf{C}_i$ and $\mathbf{C}_j$ are not merged, the corresponding predicate $D(\mathbf{C}_i, \mathbf{C}_{j'})$ considered in step 3 must be true. However, by Lemma 3.1.2 we know at least one of the two regions will be a component of the final segmentation. In other words, $Dif(\mathbf{C}_i, \mathbf{C}_j) > Mint(\mathbf{C}_i, \mathbf{C}_j)$ also holds true, which is a contradiction.

Theorem 3.1.1 states that the boundaries estimated by our proposed boundary estimator tends to be true object boundaries. In other words, the estimated boundaries are reliable. On the other hand, we are not able to guarantee that the method could find all true object boundaries since segmentations produced by the algorithm is possible to be too coarse.

Does the algorithm produces results with more serious oversmoothing than EGS? Actually not. In fact, our proposed predicate makes a stronger boundary estimator than EGS in the sense that one has the opportunity to correct weak object boundary by averaging it with strong parts of the object boundary. We can interpret the comparison of our method and EGS in the following way. In EGS, the merging predicate always holds since the inter-region difference that causes the merging remains the same no matter how the merged regions grow by merging other regions. What makes a difference is our definition of inter-region difference: the inter-region difference that causes the merging always grows larger as more regions are merged to the previously merged regions. In other words, our definition of inter-region difference makes one "realize" there is a chance that the merged regions should actually be separated. The same error

could probably also happen to EGS but one simply does not ”realize” the previously made error.

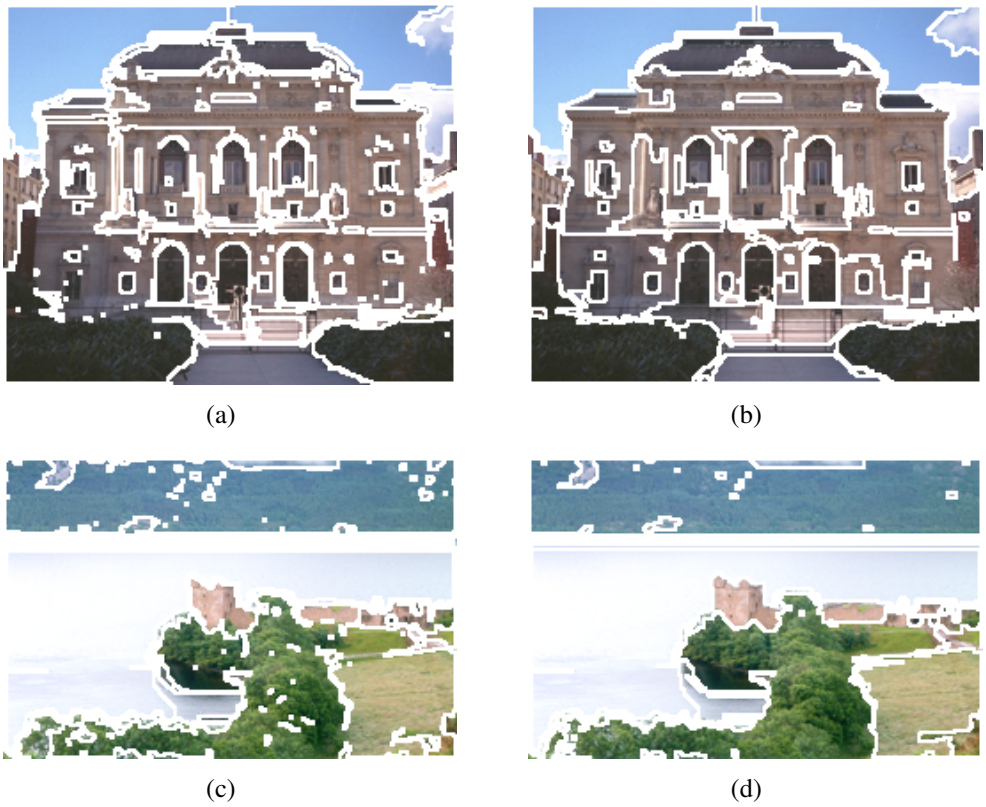
3.2 Relaxation with Mutual Volume

We implement the segmentation algorithm described in section 3.1.2 and compare its segmentation results with segmentations obtained by EGS. Figure3.2 illustrates the original images and the comparison of results obtained by different algorithms. The images are first smoothed by a gaussian kernel with $\sigma = 0.8$ and both algorithms are performed on the grid graph whose definition is described in the beginning of Section 3.1. For EGS we set $\tau(\mathbf{C}) = \frac{300}{|\mathbf{C}|}$, where the parameter $k = 300$ is the recommended value that has been adopted in experiments of [36]. Since our proposed predicate makes a stricter boundary estimator, we relax the constraint by setting a larger k value and suppress the increase of $f(|\mathbf{C}|)$ as $|\mathbf{C}|$ increases. Specifically, we set $k = 400$. $f(|\mathbf{C}|)$ is defined as:

$$f(|\mathbf{C}|) = \begin{cases} \sqrt{|\mathbf{C}|} & \text{if } |\mathbf{C}| < 4000 \\ \sqrt[3]{|\mathbf{C}|} - \sqrt[3]{4000} + \sqrt{4000} & \text{otherwise} \end{cases} .$$

The bias strength controlling parameter α is set to 1.

From figure3.2 we can see our proposed method produces more favorable results by adopting a boundary estimator stricter than EGS. Moreover, adding more adaptivity to the thresholding function which compensates region statistics estimation further helps to improve the segmentation quality. We also observe, however, that the algorithm still tend to oversegment where there exist strong textures and noises. One can infer that the fundamental reason is the formulation of a region's thresholding function being independent with respect to other region volumes. In fact, such formulation is indispensable for the problem's tractability, but is not always necessary in producing good segmentations. Our strategy for penalizing oversegmentation is to introduce pairwise mutual region volume - a value defined as the geometric mean of two adjacent regions' volume - to substitute each single region volume in determining the



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

Figure 3.2: Segmentations of *Opera* and *Loch Ness*. (a)(c) Result obtained by EGS. (b)(d) Result obtained by the proposed algorithm

thresholding value. Even though the relaxation leads to intractability of the segmentation problem, one shall see such approach is actually very powerful in producing good segmentations. To implement the above relaxation, we do not need to change the algorithm except simply making a small modification to the minimum intra-region difference:

$$\begin{aligned} &Mint(\mathbf{C}_1, \mathbf{C}_2) \\ &= \min(Int(\mathbf{C}_1 + \tau(\mathbf{C}_1, \mathbf{C}_2)), Int(\mathbf{C}_2 + \tau(\mathbf{C}_1, \mathbf{C}_2))), \end{aligned} \tag{3.6}$$

where

$$\begin{aligned} &\tau(\mathbf{C}_1, \mathbf{C}_2) \\ &= \left[\frac{(\sum_{k=1}^{K} |\mathbf{E}_{C_k}|) \sum_{(v_i,v_j)\in\mathbf{E}_C} w(v_i, v_j)}{|\mathbf{E}_C| \sum_{k=1}^{K} \sum_{(v_i,v_j)\in\mathbf{E}_{C_k}} w(v_i, v_j)} \right]^{\alpha} \frac{1}{f(\sqrt{|\mathbf{C_1}||\mathbf{C_2}|})} \end{aligned} \tag{3.7}$$

3.3 Experimental Results

We implement the algorithm with the relaxation proposed in section 3.2 to segment a number of test color images and evaluate its performance. The corresponding parameters are identical to those in section 3.2. EGS with parameter k equal to 300 was implemented for the purpose of comparison. We also compare our results with quick shift (QS), a fast and effective mode seeking algorithm recently proposed for applications of image segmentation. We run the quick shift algorithm with the VLFeat Matlab package which is kindly available at http://www.vlfeat.org/. The parameters $ratio$, $kernelsize$ and $maxdist$ are respectively set to 0.5, 10 and 30. Results produced by the above three methods are partially illustrated in figure 3.3. One could observe that our proposed method tend to generate most favorable segmentations with less over-segmentions and more continuously preserved major object boundaries, which can potentially lead to less false labeling and beneficial gains in subsequent operations.

We also perform quantitative evaluation to the produced results. Details about the benchmarks can be found in [27] and the benchmarks have been well designed for evaluating unsupervised segmentations. The evaluation of segmentation results are illustrated in table 3.3. Results indicate our method outperforms the other two methods on the adopted two benchmarks.

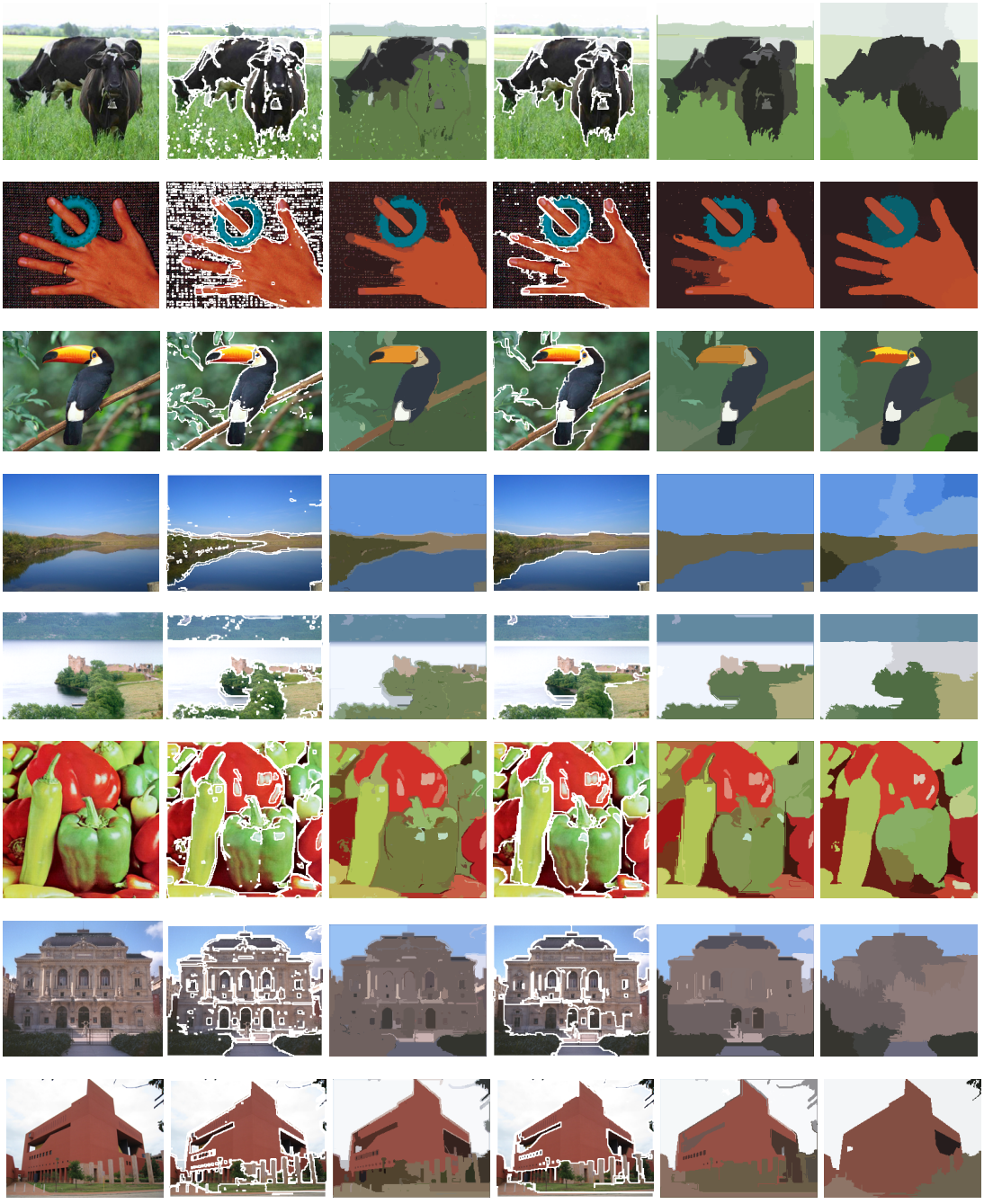


Figure 3.3: Segmentation Results. The first column contains the original test images. The second and third column correspond to results obtained by EGS. The fourth and fifth column are segmentations produced our proposed method. The last column are segmentations produced by quick shift. The corresponding test images from the first row to the last row are respectively *Cow*, *Hand*, *Toco Toucan*, *Lake*, *Loch Ness*, *Peppers*, *Opera* and *Red Building*.

Table 3.1: Quantitative evaluation

Image	***F'(I)***			***Q(I)***		
	EGS	**Ours**	**QS**	**EGS**	**Ours**	**QS**
Brandy Rose	2.71	**0.44**	1.82	5.24	**2.56**	3.12
Butterfly	0.72	**0.08**	0.23	1.13	**0.51**	0.76
Cow	4.37	**0.23**	0.6	6.14	**1.23**	1.75
Flowers	8.33	**0.73**	3.13	8.7	**3.6**	4.53
Frangipani1	2.08	**0.3**	0.62	3.67	2.03	**1.89**
Frangipani2	4.85	**0.64**	1.36	6.4	3.4	**3.1**
Kids	2.68	**0.29**	2.51	5.49	**2.35**	2.86
Lake	0.48	**0.05**	0.05	1.5	0.63	**0.52**
Loch Ness	2.3	**0.12**	0.34	3.18	**0.81**	1.14
Mountain	1.95	**0.18**	0.18	3.25	1.67	**0.64**
Opera	2.44	0.23	**0.16**	3.35	1.19	**1.09**
Peppers	4.48	**0.66**	0.78	8.02	3.55	**2.24**
Red Building	2.19	**0.66**	0.1	2.47	2.27	**0.86**
Skyline Arch	5.13	**0.78**	2.9	6.01	**2.95**	4.2
Toco Toucan	1.88	**0.24**	0.67	3.71	**1.66**	1.8
Water Lilies	10.4	**0.77**	4.59	10.2	**3.68**	4.55
Hand	6.45	**0.36**	1.2	7.58	**1.62**	2.03
Horse	5.55	**0.29**	1.01	7.03	**2.1**	2.56
Average	3.83	**0.39**	1.24	5.17	**2.1**	2.2

CHAPTER 4

GRAPH-EMBEDDED MODE SEEKING

The problem with contour finding or boundary estimation is that it suffers from "region leak" at the presence of weak boundaries since it starts from merging the most similar pixels/superpixels. On the contrary, clustering based segmentation models the intra-region similarity, suffering much less from this problem. In this chapter, we investigate the problem of tree-structure embedded density estimation and mode seeking for data clustering and segmentation. Our work provides a novel angle looking into the mode seeking problem by introducing metrics learned from a spanning tree into mode seeking. In particular, we adopt minimum spanning tree (MST) to learn compact structures in the feature space or on a connected graph. On one hand, the inclusion of MST helps to find manifold structures for feature space analysis and data clustering. On the other hand, the graph-based attribute works compatibly with regional level image operations in computer vision. A wide range of computer vision problems in principle requires regional support, where relation between image regions are typically depicted with a weighted graph and graph-based methods have consequently become a powerful tool. Such characteristic offers several intuitionally reasonable advantages. First, region-wise operation allows one to investigate and design more versatile and powerful features, as a region often contains much more information than a single pixel. Second, adopting region as basic processing unit can largely alleviate the computational burden.

In the chapter, we will illustrate the applications of our method in data clustering and region-based image segmentation. Fig. 4.1 shows one example of data clustering using our proposed method. The potential application of this algorithm, however, is considerable, as mode seeking has diverse applications.

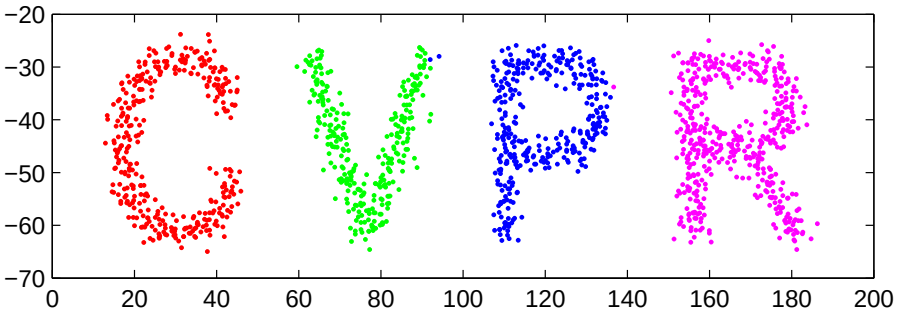


Figure 4.1: Example of data clustering using the proposed mode seeking algorithm with $h_1 = 180$ and $h_2 = 40$.

This chapter is organized as follows: In Section 4.1, we briefly introduce the background and closely related works. Readers already familiar with nonparametric density estimation and mean shift may jump to Section 4.2, where we describe the proposed method and discuss its important properties. Some experimental results regarding clustering and application of our method in image segmentation are illustrated in Section 4.3, showing that the method is an effective one.

4.1 Related Works

The paradigm of density estimation and clustering includes a family of mode seeking algorithms with Parzen density estimation. More recently, several works have explored the improvement of traditional mean shift algorithm. In [39], the author introduced asymmetric kernel to mean shift object tracking. The scale and orientation of the kernel is automatically and adaptively selected, depending on the observations at each iteration. In [32], A new mode seeking algorithm called the medoid shift was proposed. The purpose of medoid shift is to extend mode seeking to general metric spaces. The method, however, requires huge computational load and tends to result in over-fragmentation. It essentially becomes a finite point searching problem and is quite different from our method in terms of both purpose and algorithmic process. In [33], the authors proposed the quick shift algorithm which is considerably faster than mean shift and medoid shift. Their emphasis tends to concentrate on algorithm acceleration while preserving its performance. The GPU implementation of quick shift was dis-

cussed in [34] to further speed up the algorithm from the hardware perspective. There has also been other works trying to improve the efficiency of mode seeking [40].

Considering the nearest neighbor property of MST, our method to some extent are related to previous works that generalize mean shift to non-linear manifolds [41], or introduce nonlinear kernelized or manifold metrics [32, 33]. Our method can achieve some similar goals but the idea remains very different. One also notices that there exist a great many works concerning MST based graph segmentations [14]. Even though the method have also utilized MST, we generally think it belongs to the family of mode seeking methods where the algorithm characteristics are quite different from many graph based segmentation methods. In fact our work presents a general framework of embedding tree structures into the mode seeking process. Therefore it is straight forward for one to plug in many other trees and bring in additional algorithm characteristics.

4.2 Graph-embedded Mode Seeking

We propose to perform density estimation on a joint domain represented by the node feature space and the distance space defined by the minimum spanning tree of that graph. There are several advantages operating on an MST-based structure. First, tree-based structure helps to uniquely define distances for any node pair, as a tree does not have circles. Of course, one could directly define the pairwise node distances in the Euclidean space, resulting in the traditional mean shift. But this basically discards the structural information preserved by a graph. In applications such as image segmentation, spatial information preserved by a graph can be very important. Second, an MST is the connected graph structure where all nodes are connected with least edges numbers and weights. In other words, an MST can be regarded as a "compact" structure that preserves important information about the cluster structure in a feature space. Although the introduction of a tree structure in practice could possibly be problematic - as it faces the risk of large tree structure variation induced by noise points, especially

for those important tree roots - one shall see, the proposed method works pretty well and robustly in real image segmentation tests. In addition, such formulation helps to improve mode seeking performances for many manifold-shaped clusters.

We will define the density function and describe its mode seeking process in the following part of this section.

4.2.1 Proposed density estimator

Given N samples represented by the set $\mathbf{V} = \{\mathbf{v}_i | i = 1, \ldots, N, \mathbf{v}_i \in \mathbf{R}^d\}$ and the undirected weighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, the minimum spanning tree $\mathbf{S} = (\mathbf{V}, \mathbf{E_S})$ is a connected graph of G with $\mathbf{E_S} \subseteq \mathbf{E}, |\mathbf{E_S}| = N - 1$. For any node pair (i, j) where $i \neq j$, there exists a unique path $\mathbf{E}_{ij}$ such that $\mathbf{E}_{ij} \subseteq \mathbf{E_S}$, i and j is connected by $\mathbf{E}_{ij}$ and deleting any element of the set results in the disconnection of i and j. In addition, we define $\mathbf{E}_{ij}$ to be $\emptyset$, if $i = j$.

Property 4.2.1 *For any given node pair (i, j), the set of connecting edges $\mathbf{E}_{ij}$ is unique.*

The above attribute comes directly from the tree structure. The proof is simple: if there is more than one $\mathbf{E_{ij}}$ then there exists at least one circle, which contradicts with the proposition. The unique distance definition on an MST facilitates the definition of density for a given location.

We propose to use a joint representation of the MST distance space (or MST space for short) and the feature space to define the density estimator. Consider the simplest case where the MST space kernel center is located exactly at a tree node $\mathbf{v}_j$, then the density estimator can be written as follows:

$$f(\mathbf{v}) = c_0 \sum_i k\left(\frac{d(\mathbf{v}_j, \mathbf{v}_i)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right), \tag{4.1}$$

where $d(\mathbf{v}_j, \mathbf{v}_i) = \sum_{(\mathbf{v}_{k1}, \mathbf{v}_{k2}) \in \mathbf{E_{ij}}} ||\mathbf{v}_{k1} - \mathbf{v}_{k2}||$ is the cumulative weight of edges that connects the two nodes, $\mathbf{v}$ is the feature space kernel center, h_1 and h_2 are the bandwidth parameters controlling the window size and c_0 is a constant term determined by

the sample size and bandwidth. $k(x)$ is the profile of a normal kernel:

$$k(x) = \exp(-\frac{1}{2}x). \tag{4.2}$$

To define a density estimator for any location on the MST space, we have to first define the branch of an MST node. Here by saying "any location" we actually allow the MST space kernel center to be located on an MST edge between neighboring nodes. In other words, the kernel can shift on the constrained space defined by MST. Suppose $\mathbf{v}_{neigh}$ is a neighboring node of $\mathbf{v}_i$, we have the following definition:

Definition 4.2.1 *The branch of a given tree node* $\mathbf{v}_i$ *with respect to its connected edge* $(\mathbf{v}_i, \mathbf{v}_{neigh})$ *is a set of nodes and edges* $\mathbf{B} = (\mathbf{V_B}, \mathbf{E_B})$*, such that* $\mathbf{V_B} = \{\mathbf{v}_j | j \neq i, (\mathbf{v}_i, \mathbf{v}_{neigh}) \in \mathbf{E_{ij}}\}$, $\mathbf{E_B} = \{(\mathbf{v}_i, \mathbf{v}_j) | i \neq j, (\mathbf{v}_i, \mathbf{v}_{neigh}) \in \mathbf{E}_{ij}\}$.

The branch of a node is an "induced subgraph" rooted at $\mathbf{v}_i$, and descending from its referenced connected edge. There exist at least one corresponding MST edge - denoted as e_{ref} - where the MST space kernel center is located on. If the center is located exactly on a tree node, then one may choose any edge connecting this node to one of its neighboring nodes as e_{ref}. Suppose that the two nodes connected by e_{ref} are respectively $\mathbf{v}_{ref1}$ and $\mathbf{v}_{ref2}$, and that the distances from the kernel center to $\mathbf{v}_{ref1}$ and $\mathbf{v}_{ref2}$ are respectively x_1 and x_2 ($x_1 + x_2 = d(\mathbf{v}_{ref1} - \mathbf{v}_{ref2}) = ||\mathbf{v}_{ref1} - \mathbf{v}_{ref2}||$), then the density estimator defined with respect to $\mathbf{v}_{ref1}$ can be written as:

$$\begin{aligned}
&\hat{f}_{eref,vref1}(\mathbf{v}, x_1) = \\
&c_0 \sum_{i, \mathbf{v}_i \in \mathbf{V}_{ref1}} k\left(\frac{(d(\mathbf{v}_{ref1}, \mathbf{v}_i) - x_1)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right) + \\
&c_0 \sum_{i, \mathbf{v}_i \notin \mathbf{V}_{ref1}} k\left(\frac{(d(\mathbf{v}_{ref1}, \mathbf{v}_i) + x_1)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right).
\end{aligned} \tag{4.3}$$

where $\mathbf{V}_{ref1}$ is the set of branch nodes with respect to $\mathbf{v}_{ref1}$ and e_{ref}. Similarly, we can define the density estimator with respect to $\mathbf{v}_{ref2}$:

$$\begin{aligned}
&\hat{f}_{eref,vref2}(\mathbf{v}, x_2) = \\
&c_0 \sum_{i,\mathbf{v}_i \in \mathbf{V}_{ref2}} k\left(\frac{(d(\mathbf{v}_{ref2}, \mathbf{v}_i) - x_2)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right) + \\
&c_0 \sum_{i,\mathbf{v}_i \notin \mathbf{V}_{ref2}} k\left(\frac{(d(\mathbf{v}_{ref2}, \mathbf{v}_i) + x_2)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right).
\end{aligned} \tag{4.4}$$

where $\mathbf{V}_{ref2}$ is defined in a similar way. Associated with the above density estimator are some good properties that facilitates the mode seeking process:

Property 4.2.2 $\hat{f}_{eref,vref1} = \hat{f}_{eref,vref2}, \forall e_{ref} \in \mathbf{E}$

The above equality holds in the sense that $\mathbf{V}_{ref1} \cup \mathbf{V}_{ref2} = \mathbf{V}$ and $\mathbf{V}_{ref1} \cap \mathbf{V}_{ref2} = \emptyset$, which indicates $\{\mathbf{v}_i | \mathbf{v}_i \in \mathbf{V}_{ref1}\} = \{\mathbf{v}_i | \mathbf{v}_i \notin \mathbf{V}_{ref2}\}$. In addition, since $d(\mathbf{v}_{ref1}, \mathbf{v}_i) - x_1 = d(\mathbf{v}_{ref1}, \mathbf{v}_{ref2}) + d(\mathbf{v}_{ref2}, \mathbf{v}_i) - x_1 = d(\mathbf{v}_{ref2}, \mathbf{v}_i) + x_2$ when $\mathbf{v}_i \in \mathbf{V}_{ref1}$, we obtain the following equality:

$$\begin{aligned}
&\sum_{i,\mathbf{v}_i \in \mathbf{V}_{ref1}} k\left(\frac{(d(\mathbf{v}_{ref1}, \mathbf{v}_i) - x_1)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right) \\
&= \sum_{i,\mathbf{v}_i \notin \mathbf{V}_{ref2}} k\left(\frac{(d(\mathbf{v}_{ref2}, \mathbf{v}_i) + x_2)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right).
\end{aligned}$$

The equality relation between the second term of (4.3) and the first term of (4.4) can be proved similarly. Property 4.2.2 states that the estimated density does not depend on the choice of reference point.

Property 4.2.3 *If e_{ref1} and e_{ref2} are two edges that connects the same node $\mathbf{v}_{ref}$,* $\hat{f}_{eref1,vref}(\mathbf{v}, 0) = \hat{f}_{eref2,vref}(\mathbf{v}, 0), \forall \mathbf{v}_{ref} \in \mathbf{V}$.

Property 4.2.3 states that the estimated density does not depend on the choice of reference edge when the MST space kernel is located on a tree node. Here we consider the special situation where the MST space kernel is shifting from one edge to another.

When the kernel is located on $\mathbf{v}_{ref}$, the density estimator degenerates to (4.1), as $x = 0$. The same condition also holds when we define the density estimator with respect to any other edge connecting to $\mathbf{v}_{ref}$, which indicates the above property.

Property 4.2.4 *The kernel defined on the MST distance space is continuous and is piecewise differentiable.*

According to the definition of density estimator, one is easy to verify the piecewise continuity and differentiability given the MST space kernel is located on the same edge. Together with Property 4.2.3, we can obtain Property 4.2.4. The above property also infers the continuity and piecewise differentiability of the density estimator since it is a linear combination of continuous and piecewise differentiable kernels.

4.2.2 Mode seeking with force competition

We seek the mode by maximizing the density estimator with respect to $\mathbf{v}$ and x simultaneously. The step is to piecewisely estimate the density gradient, which is similar to mean shift. Taking the derivative of the density estimator with respect to $\mathbf{v}$, one get the estimated density gradient:

$$\begin{aligned}
&\frac{\partial \hat{f}_{eref,vref}(\mathbf{v}, x)}{\partial \mathbf{v}} \\
&= \frac{2c_0}{h_2^2} \sum_i (\mathbf{v}_i - \mathbf{v}) K_i g\left(\left\| \frac{\mathbf{v} - \mathbf{v}_i}{h_2} \right\|^2 \right) \\
&= \frac{2c_0}{h_2^2} \left[\sum_i K_i g\left(\left\| \frac{\mathbf{v} - \mathbf{v}_i}{h_2} \right\|^2 \right) \right] \left[\frac{\sum_i K_i g\left(\left\| \frac{\mathbf{v}-\mathbf{v}_i}{h_2} \right\|^2 \right) \mathbf{v}_i}{\sum_i K_i g\left(\left\| \frac{\mathbf{v}-\mathbf{v}_i}{h_2} \right\|^2 \right)} - \mathbf{v} \right]
\end{aligned} \tag{4.5}$$

where $g(x) = -k'(x)$, K_i is the MST space kernel function:

$$K_i = \begin{cases} k((d(\mathbf{v}_{ref}, \mathbf{v}_i) - x)^2 / h_1^2) & \text{if } \mathbf{v}_i \in \mathbf{V}_{ref1} \\ k((d(\mathbf{v}_{ref}, \mathbf{v}_i) + x)^2 / h_1^2) & \text{otherwise} \end{cases}$$

The second term in (4.5) is the well known mean shift vector for the feature space kernel center $\mathbf{v}$:

$$\mathbf{m}(\mathbf{v}) = \frac{\sum_i K_i g\left(\left\| \frac{\mathbf{v}-\mathbf{v}_i}{h_2} \right\|^2 \right) \mathbf{v}_i}{\sum_i K_i g\left(\left\| \frac{\mathbf{v}-\mathbf{v}_i}{h_2} \right\|^2 \right)} - \mathbf{v}. \tag{4.6}$$

[24] has already developed a sound theoretical basis for mean shift algorithm concerning its physical meaning, convergence analysis and relation to other feature space analysis methods. Here we will not extend the discussion. Now consider the second variable. Taking the derivative of $\hat{f}_{eref,vref}(\mathbf{v}, x)$ with respect to x, we have:

$$\begin{aligned}
&\frac{\partial \hat{f}_{eref,vref}(\mathbf{v}, x)}{\partial x} = \\
&\frac{2c_0}{h_1^2} \sum_{i,\mathbf{v}_i \in \mathbf{V}_{ref}} (d(\mathbf{v}_{ref}, \mathbf{v}_i) - x) K_{joint,i} \\
&+ \frac{2c_0}{h_1^2} \sum_{i,\mathbf{v}_i \notin \mathbf{V}_{ref}} (-d(\mathbf{v}_{ref}, \mathbf{v}_i) - x) K_{joint,i},
\end{aligned} \tag{4.7}$$

where $K_{joint,i}$ is the product of the feature space kernel and the negative derivative of the MST space kernel profile:

$$K_{joint,i} = \begin{cases} -k'\left(\frac{(d(\mathbf{v}_{ref},\mathbf{v}_i)-x)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v}-\mathbf{v}_i}{h_2}\right\|^2\right) & \text{if } \mathbf{v}_i \in \mathbf{V}_{ref} \\ -k'\left(\frac{(d(\mathbf{v}_{ref},\mathbf{v}_i)+x)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v}-\mathbf{v}_i}{h_2}\right\|^2\right) & \text{otherwise} \end{cases}$$

Equation (4.7) can be further rewritten as:

$$\begin{aligned}
&\frac{\partial \hat{f}_{eref,vref}(\mathbf{v}, x)}{\partial x} = \\
&\frac{2c_0}{h_1^2} \left[\sum_i K_{joint,i} \right] \left[\left(\sum_{i,\mathbf{v}_i \in \mathbf{V}_{ref}} K_{joint,i} d(\mathbf{v}_{ref}, \mathbf{v}_i) \right. \right. \\
&\left. \left. - \sum_{i,\mathbf{v}_i \notin \mathbf{V}_{ref}} K_{joint,i} d(\mathbf{v}_{ref}, \mathbf{v}_i) \right) \Big/ \sum_i K_{joint,i} - x \right]
\end{aligned} \tag{4.8}$$

The last term of (4.8) results in the displacement of the MST space kernel, which is the so called *force competition*. Force competition can also be regarded as a special case of univariate mean shift with $\mathbf{v}_{ref}$ representing the origin. One could imagine it as a tug of war where data points weighted by K_{joint} are tugging along each side of $\mathbf{v}_{ref}$. The shifting step size, however, should be chosen carefully since $\hat{f}_{eref,vref}$ is only piecewise differentiable. Suppose we use the ms to denote the last term of (4.8), the displacement of the MST space kernel is defined as:

$$\mathbf{m}(x) = \max(-x, \min(|e_{ref}| - x, ms)) \tag{4.9}$$

The above term generantees that the MST space kernel is always shifted along the same reference edge. Here we seek to provide more intuition by discussing some properties of the density gradient estimation:

Property 4.2.5 *The estimation of density gradient does not depend on the choice of reference node* $\mathbf{v}_{ref}$.

Since the density estimator is piecewise differentiable on the edge, according to Property 4.2.2 we can verify the above property. The estimated density gradient, however, does depend on the choice of reference edge when the MST space kernel reaches a tree node with more than two connecting edges. Difference in the choice of the reference edge results in the following inequality:

$$\mathbf{V}_{vref,eref1} \cup \mathbf{V}_{vref,eref2} \neq \mathbf{V},$$

where $\mathbf{V}_{vref,eref1}$ is the branch node set with respect to node $\mathbf{v}_{ref}$ and its connecting edge e_{ref}, and similar for $\mathbf{V}_{vref,eref2}$. Such inequality leads to the sudden jump of estimated density gradient at some tree nodes.

Theorem 4.2.1 *Given any node* $\mathbf{v}_{ref}$ *where the MST space kernel is located and there are more than two connecting edges, the number of reference edge* e_{ref} *with positive MST space kernel displacement is no more than 1.*

Proof: Without loss of generality, suppose the MST space kernel is located on node $\mathbf{v}_{ref}$ with three connecting edges e_{ref1}, e_{ref2} and e_{ref3}, and $D_{eref1} > D_{eref2} > D_{eref3}$, where D_{eref} is defined as follows:

$$D_{eref} = \sum_{i,\mathbf{v}_i \in \mathbf{V}_{vref,eref}} -k'\left(\frac{d(\mathbf{v}_{ref},\mathbf{v}_i)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v}-\mathbf{v}_i}{h_2}\right\|^2\right) d(\mathbf{v}_{ref},\mathbf{v}_i).$$

The force competition term $ms_{vref,eref}$ equals to the estimated density gradient with

respect to $\mathbf{v}_{ref}$ and e_{ref} times a positive scalar:

$$ms_{vref,eref1} = c\frac{\partial \hat{f}_{eref,vref}(\mathbf{v}, x)}{\partial x}\bigg|_{x=0}$$
$$= D_{ref1} - D_{ref2} - D_{ref3}.$$

Similarly, we have $ms_{vref,eref2} = D_{ref2} - D_{ref1} - D_{ref3}$ and $ms_{vref,eref3} = D_{ref3} - D_{ref1} - D_{ref2}$. Since $D_{eref1} > D_{eref2} > D_{eref3}$ and $D_{eref} > 0$, $ms_{vref,eref2}$ and $ms_{vref,eref3}$ can not possibly be larger than 0. The only positive $ms_{vref,eref}$ comes when $D_{ref1} > D_{ref2} + D_{ref3}$ and the above proof can be easily extended to nodes with multiple edges. Thus we have proved the above Theorem.

4.2.3 Algorithmic description

Theorem 4.2.1 states that when the MST space kernel is located on any tree node, either this node is a local maxima, or there is only one edge to which shifting the kernel results in the increase of the density. The conveyed intuition here is important: each time the MST space kernel is shifting from one edge to another, one does not face the problem of multiple selectable paths since there is at most one edge that increases the estimated density. Such property leads to the basis of our implemented algorithm and its fast approximation method. The mode seeking algorithm is a step size controlled gradient ascent:

1. For each data point $\mathbf{v}_i, i = 1, 2, ..., N$, initialize the its feature space kernel position as the data point itself. Select $\mathbf{v}_i$ as $\mathbf{v}_{ref}$ and initialize the MST space kernel on the reference node.

2. Compute the MST space kernel shift with the following rules:

   If the MST space is exactly located on any tree node, calculate $\mathbf{m}_j(x)|_{x=0}$ with respect to all its connecting edges e_j.

   If There exists one positive $\mathbf{m}_j$, select the corresponding edge e_j as the reference edge e_{ref}. $\mathbf{m}(x) = \mathbf{m}_j$ as the MST space kernel shift.

Else $\mathbf{m}(x) = 0$.

Else calculate $\mathbf{m}(x)$ with respect to $\mathbf{v}_{ref}$ and e_{ref}.

3. Calculate the step control factor α:

   If $\mathbf{m}(x) = 0$, $\alpha = 1$.

   Else $\alpha = |\mathbf{m}(x)|/|ms|$.

4. Compute the feature space kernel shift and scale it with α: $\mathbf{m}'(\mathbf{v}) = \alpha\mathbf{m}(\mathbf{v})$.

5. Simultaneously shift the MST space kernel and the feature space kernel with respect to the kernel shifts calculated in Step 2 and Step 4. The MST space kernel is shifted with the following rule:

   If the MST space kernel is exactly located on a node

      If $\mathbf{m}(x) = |e_{ref}|$, shift the MST space kernel to the neighboring node connected by e_{ref} and select the neighboring node as the new reference node.

      Elseif $\mathbf{m}(x) = 0$, the MST space kernel stays on the current node.

      Else update the kernel position on the edge: $x = \mathbf{m}(x)$.

   Elseif the MST space kernel is located on an edge

      If $\mathbf{m}(x) == -x$, shift the MST space kernel to the reference node.

      Elseif $\mathbf{m}(x) = e_{ref} - x$, shift the MST space kernel to the neighboring node connected by e_{ref} and select the neighboring node as the new reference node.

   Else update the kernel position on the edge: $x = \mathbf{m}(x) + x$.

6. Repeat Step 2 to Step 5 until convergence.

4.2.4 Fast approximation

Due to the piecewise differentiability and step control, the above algorithm gives the best mode seeking performance but requires more iterations before convergence. In ad-

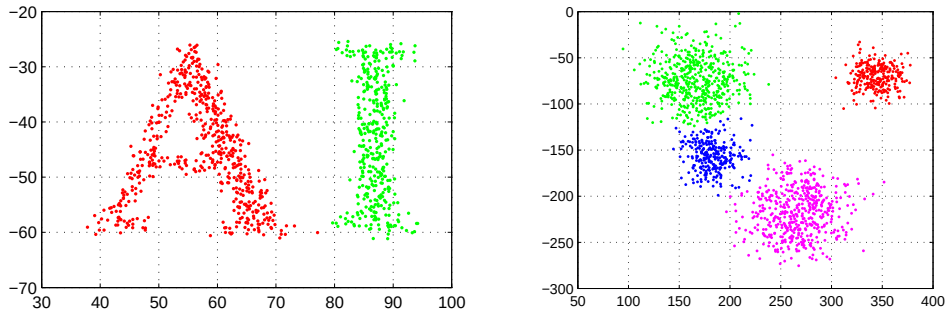
dition, the algorithm contains numerous "if-then-else" conditions, which is not friendly to hardware implementation. Here we also propose a fast approximation to the original algorithm by iteratively shifting the MST space kernel and the feature space kernel. The method is straight forward:

1. For each data point, initialize the MST space kernel and the feature space kernel.
2. Shift the feature space kernel according to (6).
3. If there exist neighboring nodes that increase the estimated density, shift the MST space kernel to the nearest one. Otherwise, stop shifting.
4. Repeat Step 2 and 3 until convergence.

In all of the following experiments, we only implement the above fast algorithm.

4.3 Experimental Results

We show three sets of experiments using our proposed algorithm. The first set of experiments demonstrates the performance of the method in the task of data clustering. Fig. 4.2(a) shows a character shaped distribution containing 934 data points and its clustering result. The bandwidth parameters h_1 and h_2 were respectively set to 150 and 40 for this experiment. Fig. 4.2(b) shows the mixture of 4 gaussian distributions with a total of 1500 data points. Here we set h_1 to 700 and h_2 to 150. From the two experiments one could observe that the method works reasonably well for both arbitrarily shaped and regularly shaped cluster of data. The real challenge comes when we want to cluster the spiral-like data distribution with highly nonlinear cluster separation boundaries. The example of spiral-like data given in [32] was reproduced with the Matlab code kindly available at http://www.cs.cmu.edu/∼new_medoid.htm. In this experiment h_1 and h_2 are respectively set to 150 and 300. Note that we have achieved the clustering performance that approximates the one given in [32] without using any non-Euclidean metric, while mean shift or Euclidean medoid shift usually will fail on such task.



(a) (b)

Figure 4.2: Data clustering using the proposed method. (a) Clustering with linearly separable data. (b) Clustering with mixture of gaussians

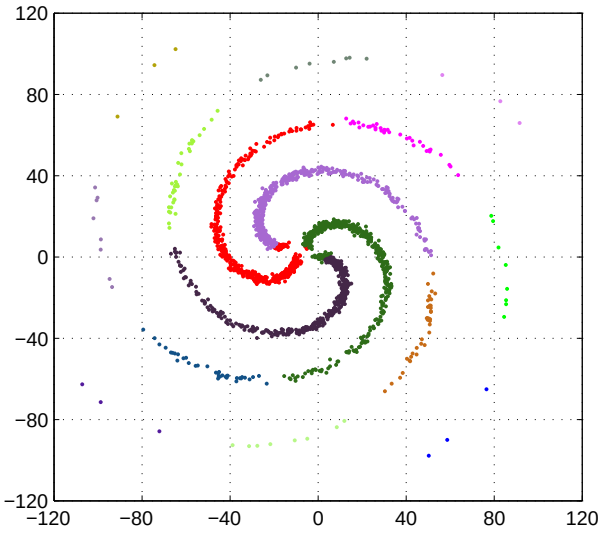


Figure 4.3: Clustering with spiral-like cluster of data using the proposed method

The second set of experiments address the problem discontinuity preserved smoothing with superpixelized images. As discussed in previous section, region-wise operation significantly reduces the required computation power, thus greatly accelerates the image smoothing and segmentation process. The introduction of MST space kernel works in compatible with the region adjacency graph and in addition, further improves the smoothing and segmentation performance. Figure 4.4 and figure 4.5 shows the images and their smoothing results using different methods in the RGB color space. The images are first superpixelized using normalized cut [15, 38]. The corresponding Matlab code is kindly provided at http://www.cs.sfu.ca/∼mori/research/superpixels/. We set the number of coarse superpixels N_sp to 200, the number of fine superpixels

N_sp2 to 400 and the number of eigenvectors N_ev to 40. Each superpixel is then represented by the mean RGB value and the whole image is mapped to an undirected, weighted region adjacency graph where edges corresponds to the eight-connectivities of two regions and edge weights are defined as the Euclidean distances between the region means. We extract the minimum spanning tree from the region adjacency graph using Kruskal's Algorithm and perform mode seeking using our proposed method. Here we fixed h_1 as 30 and h_2 as 50 for all the test images. The obtained results are illustrated in the second column of figure 4.4 and 4.5. To demonstrate the improvement of algorithm performance by introducing the MST space kernel, we compare the results with medoid shift smoothing where each super pixel is represented by the 5D joint representation of the RGB mean and spatial coordinate mean. The distance matrix is obtained by calculating the Euclidean distances between each pair of super pixels and the parameter $Sigma$ is set to 2000. We also compare our results with quick shift which is a fast mode seeking algorithm. We run the quick shift algorithm with the VLFeat Matlab package which is publicly available at http://www.vlfeat.org/. The parameters $ratio$, $kernelsize$ and $maxdist$ are respectively set to 0.3, 12 and 30. The results illustrated in figure 4.4 and figure 4.5 indicates the advantage of using our proposed method for image smoothing.

We illustrate the potential application of image segmentation using our method in the last set of experiments. Note that the segmentation performance depends largely on the defined feature. With superpixelized images, the definition of image feature becomes much more versatile than pixel based methods. Such framework allows one to improve the segmentation performance by defining the feature in a sophisticated way, using textons, texture detectors or other region statistics. For simplicity we only adopt region color histogram. Each region is represented by a 24-D concatenated histogram with each RGB channel returning a histogram of 8 bins. We then use principal component analysis (PCA) to perform dimensionality reduction on the obtained histograms. The percentage of preserved variance for PCA is set to 0.9, a typical rule of thumb value for PCA. For most of the images, the reduced dimension after performing PCA

often lies in between 4-8, which is much smaller than the original dimension number. By running PCA we reduces the computational complexity and effectively avoids from suffering the ”curse of dimensionality”. The segmentation results are shown in figure 4.6. One could observe that the proposed method is effective and produces reasonably good segmentations.

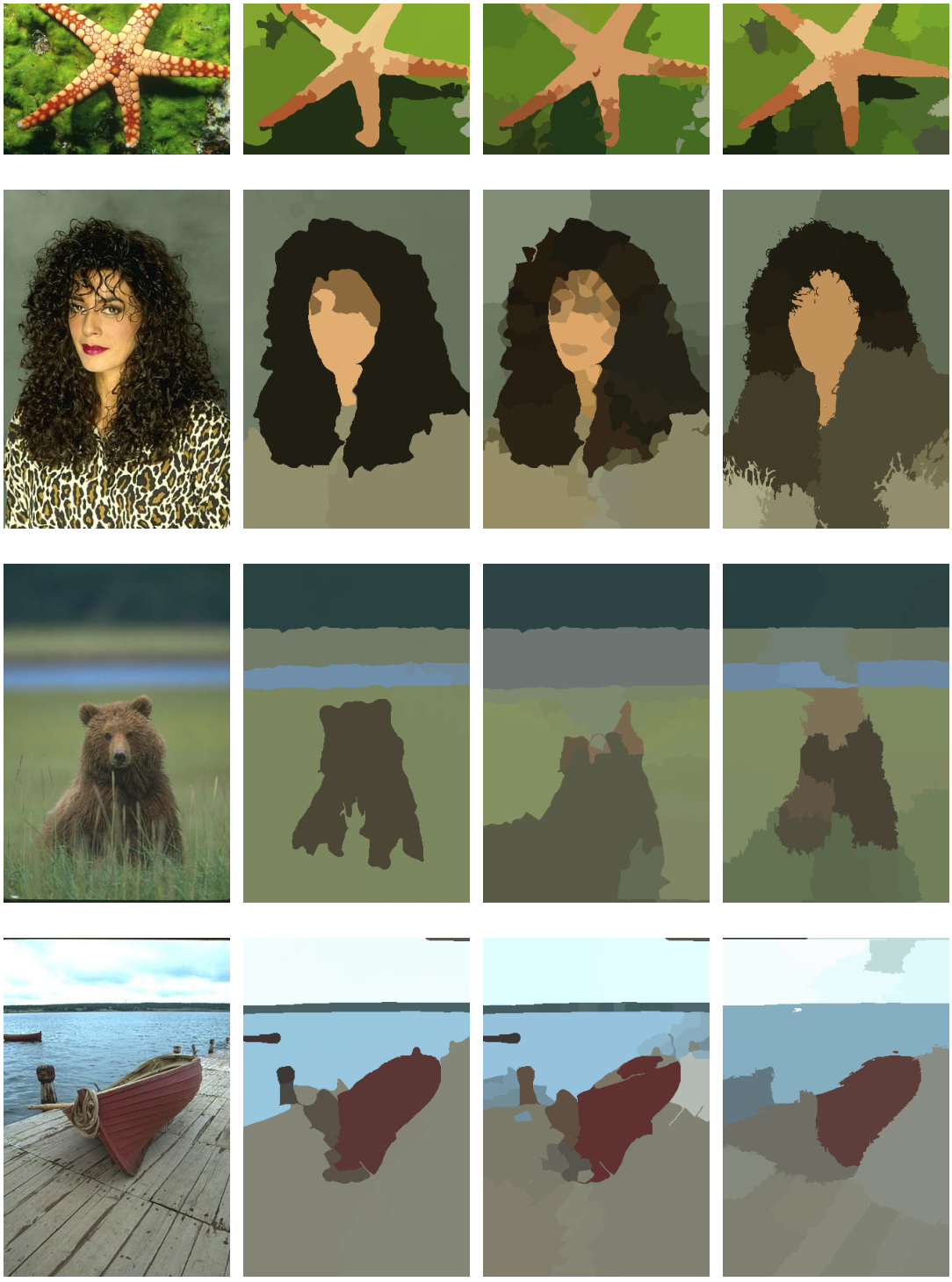


Figure 4.4: Discontinuity preserved smoothing with superpixelized images: The four columns correspond to the original images and the smoothed results by the proposed method, medoid shift and quick shift respectively.

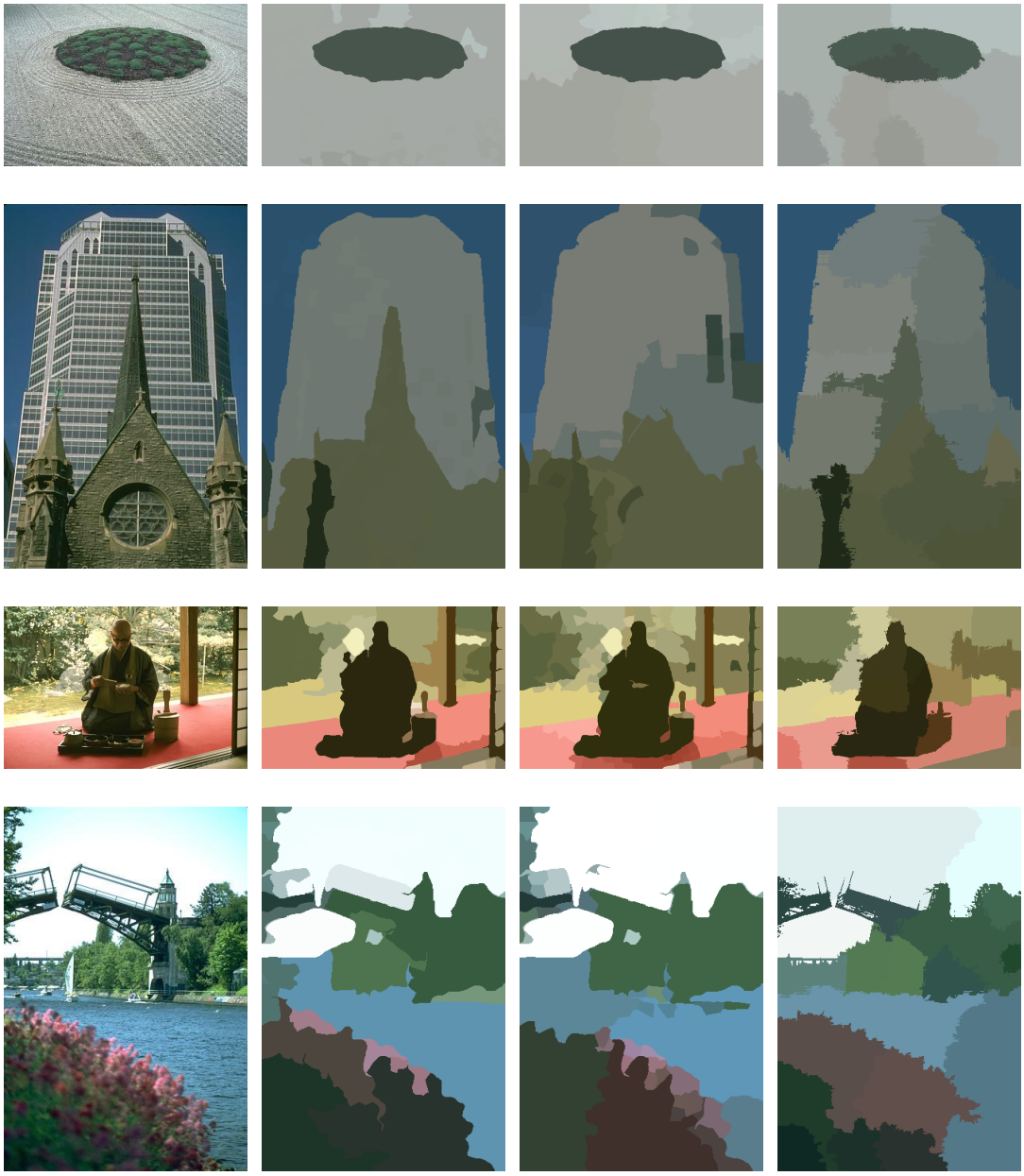


Figure 4.5: Discontinuity preserved smoothing with superpixelized images: The four columns correspond to the original images and the smoothed results by the proposed method, medoid shift and quick shift respectively.

Figure 4.6: Image segmentation experiments with region histogram

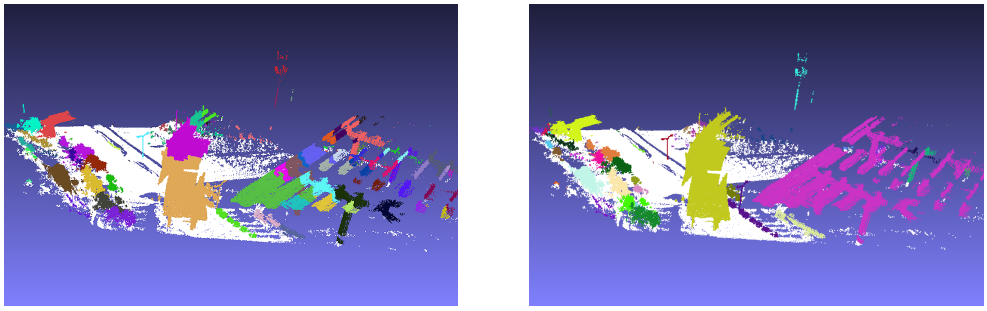
CHAPTER 5

3D POINT CLOUD SEGMENTATION

As an extension to chapter 4, we further investigate the real application of graph-embedded mode seeking in 3D point cloud object segmentation. We also introduce transductive distance on the MST space and several new features regarding this specific task, such as ground detection and bandwidth prior. 3D models with geographical locations and semantical labels are the key for urban modeling that is widely used in a variety of applications, such as urban planning, simulation and visualization. Compared to modeling with traditional tedious and time-consuming CAD tools to create and visualize 3D digital urban models, many methods have been developed in recent years to obtain 3D information automatically. With these methods, it becomes possible that large scale 3D data can be obtained and processed.

As a common method to obtain 3D information, systems equipped with LIDAR (light detection and ranging) sensors are widely used. The obtained data, also called ranging images, can be represented by 3D point clouds. However, a cloud as a whole reveals only limited structure of the urban scene and is far from being an informative visualization. To further utilize it, a necessary step is to label and categorize the points in the cloud object by object.

Object segmentation plays a crucial role in the point cloud processing routine and is often considered as one of the major processing steps. By segmentation, the points composing an object is extracted from the scene for recognition and a semantical tag is then attached to it. Despite the abundant previous works, only a few are related to large scale urban scene object localization [43]. Most methods are designed for a much confined scenario, such as focusing on a specific class of objects like roads [44], vehicles [44, 45], trees [47, 48, 51], and buildings [49, 50, 52]. In these cases, either the scenario is relatively simple that contains only a few objects [53], or the input objects



(a) (b)

Figure 5.1: An example of urban object segmentation. (a) Mean shift segmentation. (b) Segmentation obtained by our system with manifold embedded mode seeking, which improves the result in (a) significantly.

have been segmented from the scene. The theme of these papers is mainly related to recognition and reconstruction of a specific class of objects where object recognition (or classification) techniques play a central role.

The urban object recognition system proposed in [43] uses normalized cuts [15] and min cut [57] to localize and partition point cloud objects. Other graph partitioning methods such as graph cut [8] are also useful tools for partitioning a set of points into subgroups. Our data, however, consist of much more complicated urban scenes, which render these methods fail in our experiments. The major challenge is basically twofold: First, point clouds typically consist of complicated, densely aligned objects with large size variation. Second, the inherent computation consuming nature of segmentation further leads to difficulties in dealing with large scale problems, reducing the practicality of a method. Such characteristic, to a large extent, hinders one from obtaining accurate segmentations with available clustering algorithms.

In this chapter, we develop a system that can automatically segment objects in a 3D cloud of a large scale urban scene that contains billions of points. An example of the segmentation using our system is illustrated in Fig. 5.1(b). It is worth mentioning that Golovinskiy et al. [43] and Lim et al. [53] also developed a system for similar purpose. The differences between our work and theirs include that 1) our system can extract complicated terrain maps, while [43] and [53] assume relatively simple, flat terrains. Without accurate terrain extraction, segmentation of some objects is easy to fail, 2) buildings that later can be used for further mesh reconstruction are included and

segmented, 3) our data are highly noisy and heterogeneous due to the data acquisition and preprocessing method, and 4) the objects in our data are highly occluded since the data are generated by one time scan.

5.1 Our 3D Point Cloud Data Set

The range images obtained are generated by scanning along a major street in Shenzhen city. The street view contains objects of varying sizes ranging from high rise buildings and overpasses to trees, pylons and road signs. Some data also consist of complicated grounds which are difficult to extract accurately. Relatively small objects such as trees and pylons are densely aligned and often overlap with each other, while buildings typically have much larger sizes. The ranging images are obtained by driving a vehicle loaded with LIDAR censors on the street. Due to the local traffic, the objects in the scene, such as vehicles and working people, are not stationary, causing some objects distorted. Another reason for low quality of the data is that our scene covers a much larger urban area, with the spanning about 10 kilometers. So the density of the data at many locations is low where it is hard to define objects. All these factors cause considerable difficulties in separating the objects to obtain accurate segmentation. One of a close-up images can be seen in Figure 5.2.

5.2 The Proposed Method

Fig. 5.3 shows how our method works. The three main steps, ground (terrain) detection, superpixelization, and object segmentation, are described as follows.

5.2.1 Terrain detection

The first step of our system is to detect terrain from a point cloud. The terrain of a scene is related to the resolution we consider. For example, a coarse terrain map of a city can be obtained by airborne scanning, in which we can tell the hills but cannot find small objects:

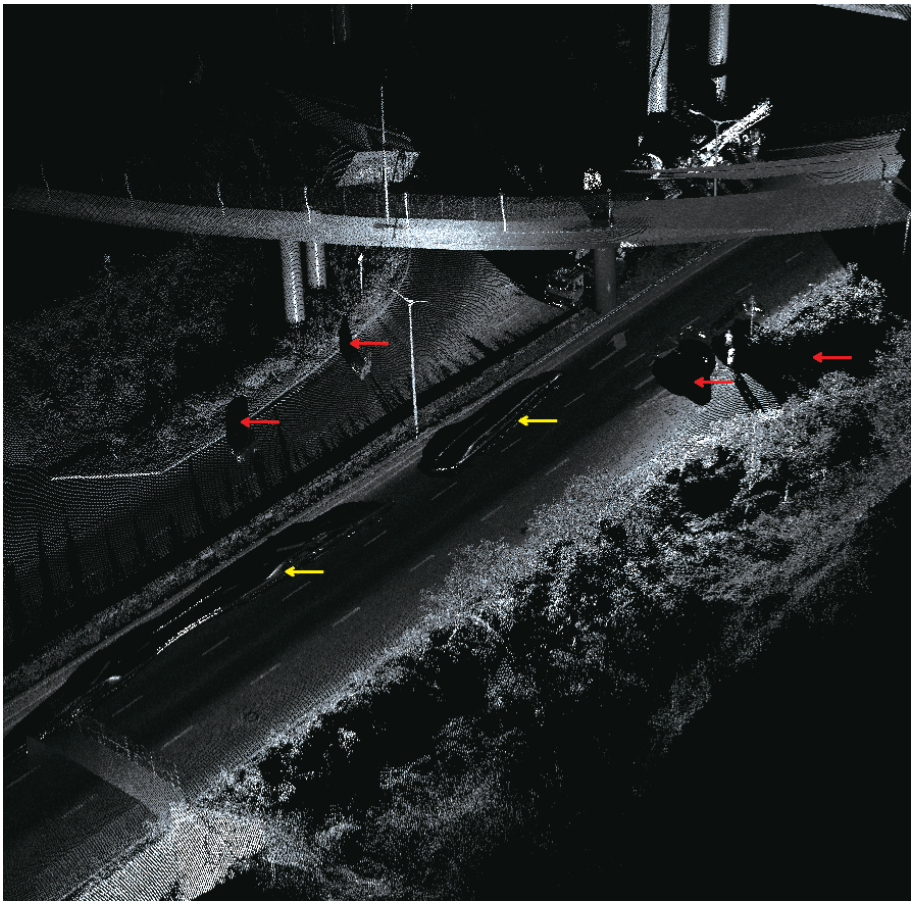
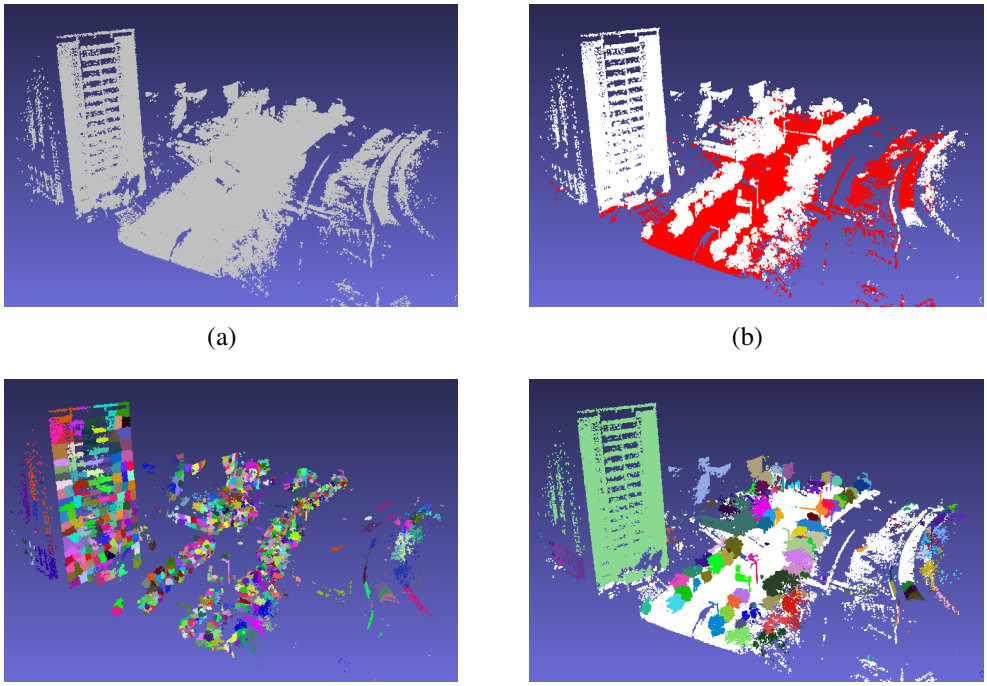


Figure 5.2: A close-up scene extracted from our urban model. Please note that the occluded areas denoted by red markers and the distorted cars due to their high speed movement denoted by yellow markers.



(a) (b)

(c) (d)

Figure 5.3: An example illustrating the segmentation process. From (a) to (d) are respectively figures of the original point cloud, ground (in red) detection, superpixelization and object segmentation.

Definition 5.2.1 *Let* $C = \{p_i = (x_i, y_i, z_i) | i = 1...N\}$ *be a perfectly captured scene without occlusion and noise where* N *is the number of points in* C*, and* Δ *be the resolution with which each pixel on a terrain map represents a* $\Delta \times \Delta$ *square in the real scene. The terrain map* T^{Δ} *with resolution* Δ *is defined as:*

$$T^{\Delta}(k,l) = \min_{\{p \in C | (x_{\Delta}(p), y_{\Delta}(p)) = (k,l)\}} z(p), \tag{5.1}$$

where $z(p)$ is the function to have the z-coordinate, along with $x_{\Delta}(p)$ and $y_{\Delta}(p)$ being the functions that retrieve the grid indices along the x-axis and the y-axis respectively when the xy-plane is masked with a $\Delta \times \Delta$ grid.

It is not difficult to develop a simple algorithm to generate the terrain map based on the definition if a perfectly captured scene is given. In real cases, however, occlusion and noise are ubiquitous. When occlusions occur (note that occlusions often cause abrupt changes), parts of the ground hidden under cars cannot be detected. We use the following techniques to overcome this problem: 1) a maximum threshold is set to detect an abrupt change of the local minimum of heights for points, 2) paved road

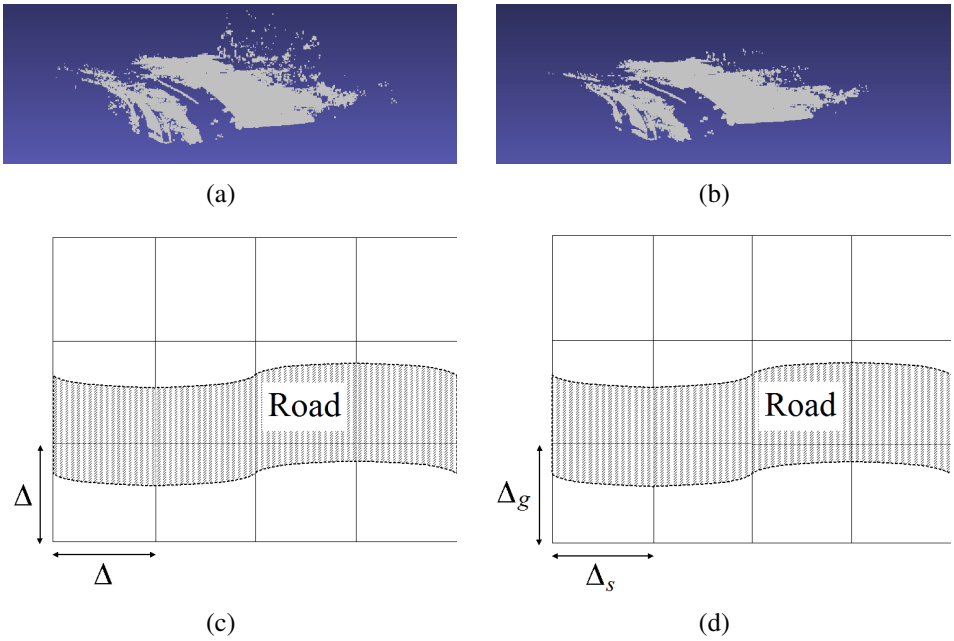


Figure 5.4: An example illustrating the segmentation process. In this figure, (a) and (b) respectively illustrate the coarse extraction of terrain and its further refinement, while (c) and (d) show the corresponding algorithmic interpretation of (a) and (b).

detection, and 3) expansion from paved road to obtain a better terrain.

The ground extraction algorithm is a two-pass process consisting of rough ground extraction and further refinement. In the first step, we divide the xy-plane into regular grids of size $\Delta \times \Delta$. For each grid, we eliminate points higher than the lowest point with a height difference larger than $Thre_1$. Mathematically, we define the roughly extracted ground as a set of extracted points satisfying the following condition:

$$\begin{aligned} T_1^{\Delta}(k,l) = \{z(q) | q \in C, (x_{\Delta}(q), y_{\Delta}(q)) = (k,l), \\ z(q) - \min_{\{p \in C | (x_{\Delta}(p), y_{\Delta}(p)) = (k,l)\}} z(p) \leq Thre_1\}, \end{aligned} \tag{5.2}$$

The intuition is that objects such as trees, overpasses, road signs and pylons typically have multiple-layered structures along the z direction within a small grid where the structures consist of the ground and the upper portion of the objects. The first step can effectively remove the majority of the objects while preserving uneven portions of the ground. We set Δ to 1.5 and $Thre_1$ to 1.7. Our system can tolerate inclination up to approximately 40 degrees. The method can also eliminate vertical structures with abrupt rise. Most of these structures correspond to building facades, walls, overpass pillars and trunks and should be considered as objects.

The consequent problem with the first step is that for large objects, the ground under them is not detected due to occlusion, and the upper portions of the objects are recognized as the ground. The second step aims at eliminating undesired object residues. Since the detected point clouds show an elongated shape along the road, we perform principal component analysis (PCA) with the cloud points on the xy-plane and select the eigenvector corresponding to the largest eigenvalue to indicate the road direction. We then divide the points into stripes of width Δ_s along the road direction, using multiple hyperplanes perpendicular to the road. The divided stripes are thus also approximately perpendicular to the road direction. Within each stripe, the points are further divided into grids of size $\Delta_g \times \Delta_s$. Fig. 5.4 illustrates the two-pass terrain extraction process.

Suppose we use l_k to indicate the l_kth grid in the k stripe, T_1^k to indicate the point set in the kth stripe, and T_1^{k,l_k} to indicate the point set corresponding to the l_kth grid. To locate the expansion starting point, the grid containing the paved road in the kth stripe is detected as:

$$l_{Road} = \arg\min_{l_k} \frac{1}{|T_1^{k,l_k}|} \sum_{p \in T_1^{k,l_k}} (z(p) - z_{Road})^2, \tag{5.3}$$

where the road height z_{Road} is defined as:

$$z_{Road} = \text{mean}(\{z(q) | q \in T_1^k, z(q) \in Bin_{max}(q)\}) \tag{5.4}$$

and the maximum bin Bin_{max} is defined as the largest bin of the histogram of point heights ranging from $\min_{p \in T_1^k}(z(p))$ to
$\min_{p \in T_1^k}(z(p)) + Thre_2$. Experiments show that most road points are well detected, with similar heights densely concentrated in the height histogram to form the largest bin. Thus it is reasonable to estimate the road height by calculating the mean of the largest bin.

The detected grid is taken as the starting point. We eliminate points within this grid that are higher than the lowest point with height difference more than $Thre_3$ and mark this grid as "refined". We then propagate the refinement from the starting grid in

a spatially continuous manner along the stripe. For each unrefined grid, we refer to the highest point that is identified as the ground in the previously refined grid and denote it as the reference point. Each unidentified point in the unrefined grid is compared with the reference point. Those higher with height differences larger than $Thre_3$ are eliminated and the rest are identified as ground. The highest among these points is selected as the new reference point. If there are no newly identified ground points, the reference point is not changed. The above process is repeated until all grids are refined in the map. We select Δ_s, Δ_g, $Thre_2$ and $Thre_3$ respectively as 10, 1, 10 and 1 and set the bin size as 1.

5.2.2 Point cloud superpixelization

The residual $C - T$ by removing the detected terrain T from a given cloud C is to be segmented. In order to solve large scale segmentation problem, a necessary step prior to the segmentation of objects is superpixelization. We first use mean shift oversegmentation with a bandwidth 5. The obtained clusters are further refined by recursively performing cluster split using 2-cluster k-means clustering, until each cluster contains less than 300 points.

5.2.3 Segmentation of Objects

Since the number of objects in a given urban scene is unknown, a clustering algorithm that needs this number known is not suitable for such a task. Mode seeking methods such as mean shift can serve as an appropriate tool. Besides the ability to automatically determine the cluster number and the potential for parallelization, mode seeking can accurately detect many objects (such as trees) even though they are densely aligned. We observe that these objects tend to have high point density at their centers while showing low density at their boundaries, which particularly favors such method.

Our method is closely related to mean shift but has many additional features that prove to be essential for obtaining good segmentations: 1) a directional biased kernel bandwidth with z axis suppression, 2) minimum panning tree (MST) embedded

mode seeking [56] that can detect compact structures and favor point connectivity with manifold-like point clouds, and 3) a large size prior for buildings with adaptive kernel size.

The observation for introducing the first feature is that points tend to have larger correlation along the vertical direction than horizontal directions. It inspires us to increase the kernel bandwidth along the vertical direction to strengthen point connectivity. This is equivalent to suppressing the z-axis coordinates for all the points and performing subsequent operations including MST construction and mode seeking in the z axis suppressed coordinate space. We suppress the z coordinates with a ratio of 0.5.

Suppose the superpixels of a point cloud are represented by a set $\mathbf{V} = \{\mathbf{v}_i | i = 1, \ldots, N, \mathbf{v}_i \in \mathbf{R}^d\}$, where N is the total number of superpixels. The dimensionality d equals 3 in our case and $\mathbf{v}_i$ is the mean z-suppressed space coordinate of the points belonging to the ith superpixel. For a given set of superpixels, we construct its full connection graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where $\mathbf{E} = \{e_{i,j} | i, j = 1, ..., N, i \neq j\}$ and $|e_{i,j}| = \|\mathbf{v}_i - \mathbf{v}_j\|$. Using Kruskal algorithm we are able to extract the minimum spanning tree (MST) $\mathbf{S} = (\mathbf{V}, \mathbf{E_S})$, which is a connected graph of $\mathbf{G}$ with $\mathbf{E_S} \subseteq \mathbf{E}, |\mathbf{E_S}| = N - 1$. For any node pair (i, j), $i \neq j$, there exists a unique path $\mathbf{E}_{ij}$ such that $\mathbf{E}_{ij} \subseteq \mathbf{E_S}$, i and j are connected sequentially by elements of $\mathbf{E}_{ij}$ and deleting any one of the elements results in the disconnection of i and j. In addition, we define $\mathbf{E}_{ij}$ to be $\emptyset$, if $i = j$.

We propose to use a joint representation of the MST distance space ("MST space" for short) and the feature space (the coordinate space) to define the density estimator. Consider the simplest case where the MST space kernel center is located exactly at a tree node $\mathbf{v}_j$. Then the density estimator can be written as follows:

$$f(\mathbf{v}) = c_0 \sum_i k\left(\frac{d(\mathbf{v}_j, \mathbf{v}_i)^2}{h_1^2}\right) k\left(\left\|\frac{\mathbf{v} - \mathbf{v}_i}{h_2}\right\|^2\right), \tag{5.5}$$

where $d(\mathbf{v}_j, \mathbf{v}_i)$ is the transductive distance of the path connecting node $\mathbf{v}_i$ and $\mathbf{v}_j$,

projected on MST:

$$d(\mathbf{v}_j, \mathbf{v}_i) = \sum_{e_{m,n} \in \mathbf{E_{ij}}} \min_{p_k \in \mathbf{V}_m, p_l \in \mathbf{V}_n} ||p_k - p_l||. \tag{5.6}$$

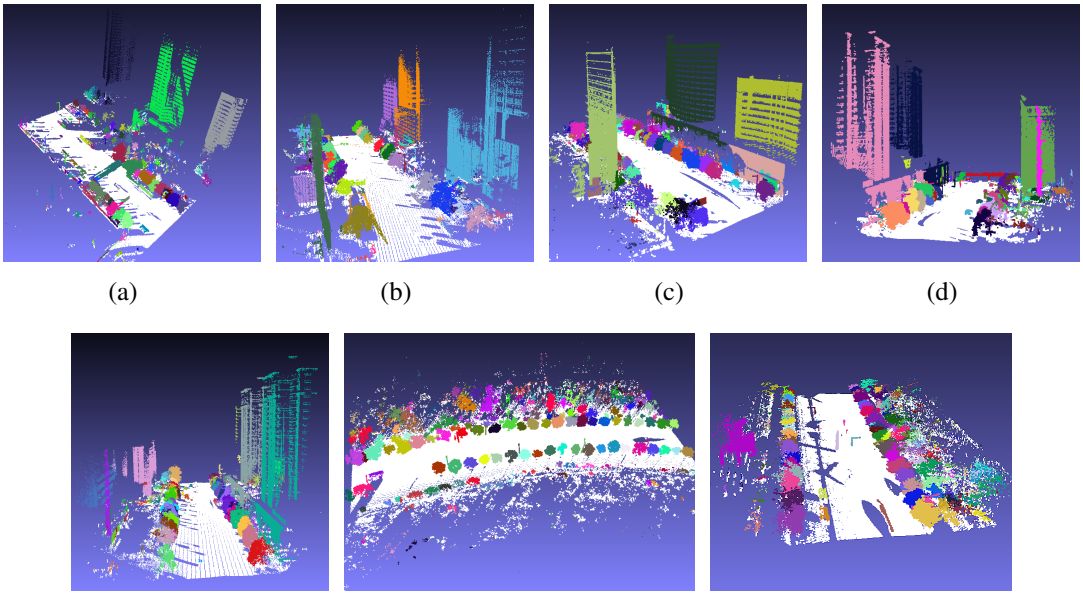
In 5.6, p_k is the single point coordinate of the kth point and $\mathbf{V}_m$ represents the set of points belonging to the mth superpixel. We adopt the above transductive distance instead of $d(\mathbf{v}_j, \mathbf{v}_i) = \sum_{e_{m,n} \in \mathbf{E_{ij}}} ||v_m - v_n||$ defined in [56] so that only effective margins are considered and more attraction is rewarded along manifolds. In (5), $\mathbf{v}$ is the feature space kernel center, h_1 and h_2 are the bandwidth parameters controlling the window size, c_0 is a constant determined by the sample size and bandwidth, and $k(x) = \exp(-\frac{1}{2}x)$ is the profile of a normal kernel.

The inference of mode seeking for the tree-embedded density estimator is well described in [56]. We employ the fast approximation of the mode seeking process by iteratively shifting the MST space kernel and the feature space kernel, which is described in [56]. We also employ an adaptive bandwidth, utilizing a large size prior for buildings. We re-cluster superpixels from clusters containing at least one superpixel higher than the road height for 7.5 in the z axis suppressed space, with the same superpixel set and MST structure as those in the previous step, but the bandwidths for re-clustered superpixels are enlarged to 7.2 and 12, which are six times the original bandwidths. Combining the cluster label for low altitude superpixels together with the re-clustered result, we can obtain the final cluster label for each point in the point clouds.

It should be emphasized again that the method can automatically label different objects without object recognition after the terrain detection. One example is given in Fig. 5.3(d), and more can be seen in the next session.

5.3 Experiments

We conduct comprehensive experiments to test the proposed system on a data set containing 176 images obtained by scanning some streets of our city. To better illustrate



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

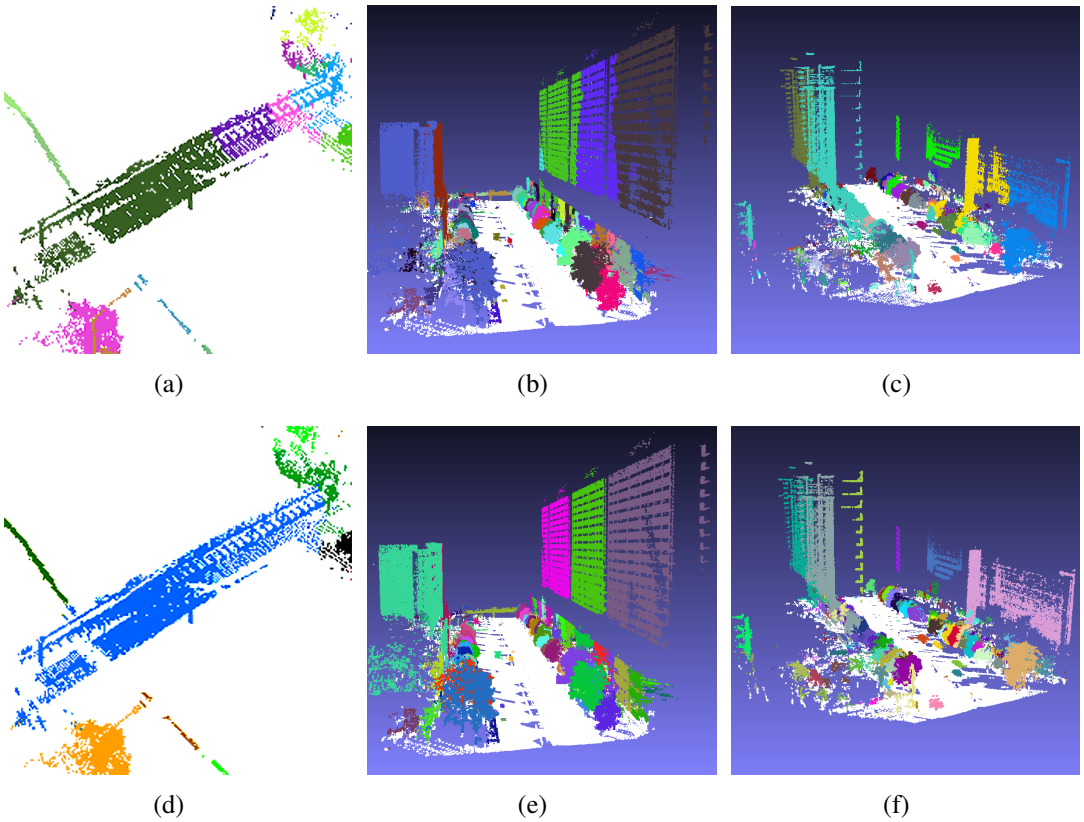
(e) (f) (g)

Figure 5.5: A portion of the segmentation results obtained by our system. The test data contains typical urban scenes.

its performance, the results are compared with results obtained by mean shift, a standard segmentation method widely adopted in the literature for point cloud segmentation [43, 58]. The mean shift algorithm is operated on the same object superpixels obtained through the first and second steps of our method. It also uses adaptive bandwidth in the our experiments, with the same parameter adaptivity settings to our method.

5.3.1 Qualitative Evaluation

We randomly illustrate some ranging images with representative urban scenes and perform ground detection and segmentation. The results indicate our system can generate very good segmentation under a variety of complicated scenes. The scenes illustrated in Figs. 5.1 and Fig. 5.3 contain complex terrain which is hard to extract. Our system can accurately extract the majority of the ground. It is also worth mentioning that the embedding manifold structure helps to improve segmentation on spanning objects, particularly overpasses and buildings, as illustrated in Fig. 5.1, Fig. 5.6(a) and Fig. 5.6(d). Notice that for the data illustrated in Fig. 5.1, there is no way for mean shift to segment the whole bridge without erroneously including nearby objects (the pylon on the right). Fig. 5.6(e) illustrates our segmentation result on a challenging task where



(a) (b) (c)

(d) (e) (f)

Figure 5.6: A comparison of segmentations obtained by our system and mean shift. The second row contains better results obtained by our system, while results obtained by mean shift contain both serious oversegmentation and oversmoothing.

large buildings are densely aligned while the scanned points are sparse on these buildings. The buildings are difficult to separate but our system can generate very accurate segmentation. Fig. 5.5(f) and Fig. 5.5(g) contains densely aligned trees difficult to separate. Our method works well on such kind of data. Tested on the whole data set containing hundreds of scenes similar to the above illustrated ones, the visual examination suggests that our system has similar performance to the illustrated ones. In all our experiments, the bandwidth for mean shift is set to 4 and the two bandwidths h_1 and h_2 for our method are respectively set to 1.2 and 2. These parameters are empirically selected to optimize the performance of the two methods.

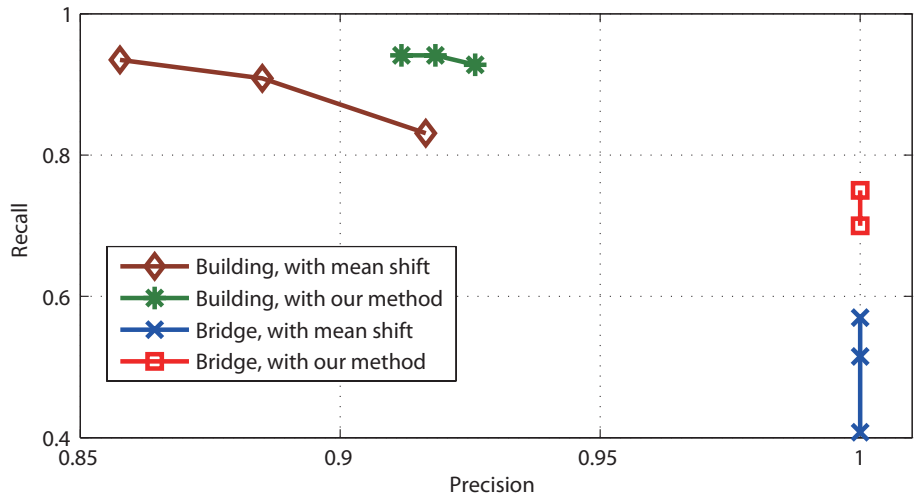


Figure 5.7: A quantitative comparison of segmentations obtained by our system and mean shift.

5.3.2 Quantitative Evaluation

We compare each automatic segmentation against the ground truth [1] segmentation by finding (a) how much of the automatic segmentation contains the whole object (precision), and (b) how much of the object is not oversmoothed with other objects (recall). The result is illustrated in Fig. 5.7. Results show that our method significantly outperforms mean shift.

[1] Since the work load is huge, we only label a portion of the data set and select objects that are easy to label.

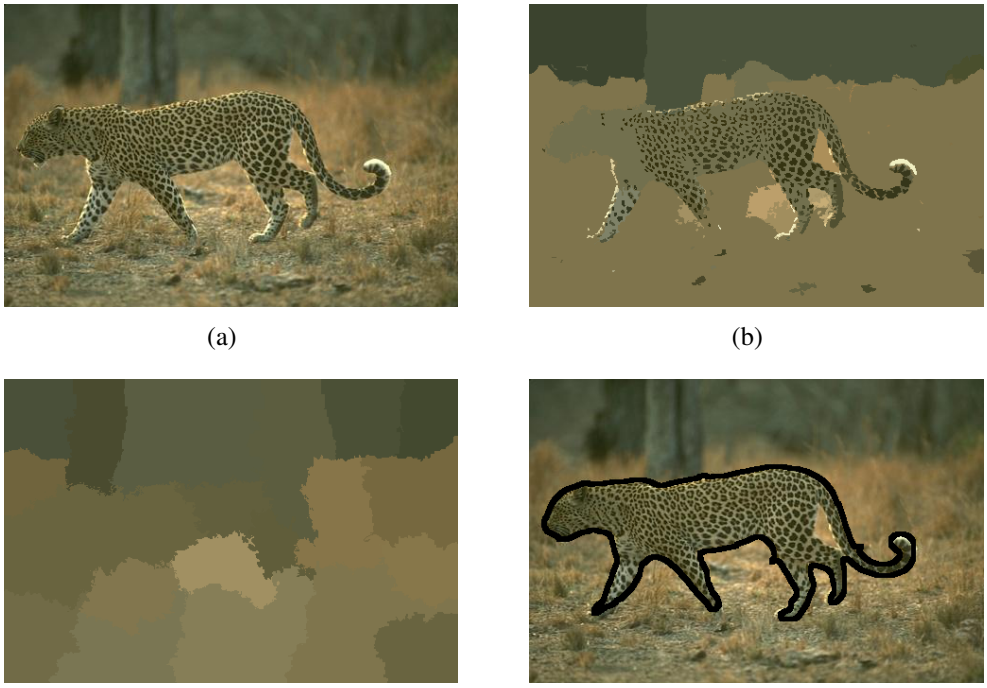
CHAPTER 6

BAG OF TEXTONS AND CONVEX SHIFT

Before the introduction of "Bag of words model" (BoW) into computer vision, one could find the early applications of BoW in natural language processing (NLP) [61]. The BoW in NLP is a popular method that ignores the word orders for representing documents. The BoW model allows a dictionary-based modeling, and each document looks like a "bag" which contains some words from the dictionary. Computer vision researchers use a similar idea for image representation. To represent an image using BoW model, an image can be treated as a document. And features extracted from the image are considered as the "words". Extraction of words often includes following three steps: feature detection, feature description and codebook generation. [62] A definition of the BoW model can be the "histogram representation based on independent features" [63]. It is a widely used basic element for further processing in computer vision, especially in object categorization. Content based image indexing and retrieval (CBIR) is also an early adopter of this image representation technique [64].

Our method shares similar idea with BoW except that the "word" we extract is textual information. What we need is a compact representation for the range of different appearances of an object and this representation should be congruous with human perception of similarity. Texton have been proven effective in categorizing materials [67] as well as generic object classes [68]. Here we use textons [65] for describing human textual and color perception. To establish a metric for region similarity and dissimilarity, we construct a histogram for each superpixel region to quantitatively indicate the proportion of contribution from a specific texton. The texton thus plays a similar role as a "codeword".

The essence of segmentation can be regarded as clustering with elaborately designed pixel features and inter-pixel distance measures that tries to approximate hu-



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

Figure 6.1: Segmentations of an image from the Berkeley Segmentation Dataset. (a) The original image. (b) Segmentation generated by mean shift. (c) Segmentation generated by quick shift. (d) Result obtained by the proposed algorithm, showing considerable improvement in terms of segmentation quality. Notice that although there is no human interaction, the produced foreground object segment highly overlaps the groundtruth.

mans visual perception of similarity. In the feature space, the cluster shape of features belonging to an image segment is often irregular. The fact that mode seeking methods can perform arbitrary shaped clustering makes it superior than many traditional clustering algorithms assuming regular shaped clusters in terms of segmentation performance. Despite the considerable literatures on mode seeking, we observe that many emphasize their applications in image segmentation while potentially posing them as low level preprocessing oriented [24] [32–34, 40]. Due to the pixel-wise operation, there has not been much fundamental improvement in terms of segmentation quality, as illustrated in Fig.6.1(b) and Fig.6.1(c). This tends to generate inferior segmentation when dealing with complex images, while image scenes often do contain abundant artificial or natural textural information. The concept of histogram based mode seeking have been introduced in mean shift tracking [39] [69–71], yet few have explored its application in image segmentation.

Our method combines the advantage of both mode seeking clustering and superpixel textual content which is far more informative than pixel-wise color. Instead of operating with pixels, we propose region-wise operations and we formulate the mode seeking problem into a constrained optimization problem for each kernel shift step. Region-wise operation allows one to investigate and design features much more versatile and powerful. Our method thus possesses the potential to outperform the segmentations produced by traditional mode seeking methods where simple pixel-wise features are not able to adequately describe the visual similarity. Such scheme also considerably alleviates the computational power required. Without loss of generality, suppose the complexity of a mode seeking algorithm is $O(N^2)$ where N is the total number of pixels. Consider the superpixelized image with N' regions (or superpixels). If $N = 100N'$, then for the same mode seeking algorithm the complexity has been reduced to 1/10000 of the original complexity. In practice, the overall algorithm complexity might not be considered such ideally. It does not hinder us, however, to show the potential of complexity reduction by region-wise operation.

6.1 Related Works

There exist considerable previous literatures related to our method concerning the aspects of texton representation and mode seeking. In review of these methods, most of them can be categorized into the following categories:

6.1.1 Relation with Texton Segmentation

In [65], Malik et al. proposed a image segmentation method based on normalized cuts with contour and texture analysis. A 40-D filter bank is used to convolve with the input image and to produce the response image. They also construct texton histogram for overlapped dense regions and χ^2 is adopted as the distance metric between two histograms. K-means is used to generate textons, turning the voting for histogram construction into hard decisions. Works in [38, 66] adopted similar strategies for texture

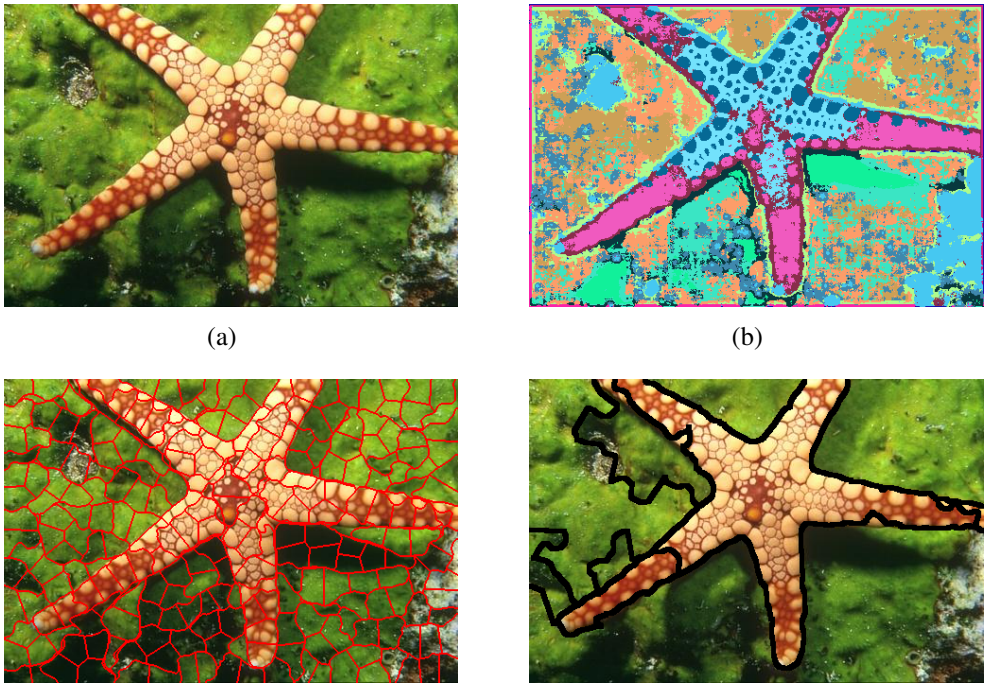


Figure 6.2: Algorithmic flow of the proposed method. Image (a) to (d) respectively correspond to the original image, texton map (each pixel assigned to the most probable texton), superpixelized image and the final segmentation result. The histogram bandwidth and spatial bandwidth are respectively set to 1.2 and 60.

similarity analysis. Different from their method, our method adopts EM soft clustering, which, in comparison with k-means, models the distribution much better since k-means only assumes spherical, uniform cluster shapes. Accordingly, we observe a boost in segmentation performance using textons generated by EM. In addition, the posterior probabilities of belonging to textons returned by EM enable one to adopt soft voting. Histograms constructed by soft clustering tend to reflect region similarity more accurately and the performance are less dependent on the number of textons.

6.1.2 Relation with Non-Euclidean Mode Seeking

Several works tried to introduce more versatile distance metrics into mode seeking. Zhao et al. [70] proposed a differentiable Earth Mover's Distance (EMD) that can be used as a distance metric for mean shift tracking. Leichter [71] proposed an alternative trackers that employ cross-bin metrics based on Mean Shift iterations. Both methods, however, only aim at tracking problem.

There have been interesting efforts that generalize mean shift to non-linear manifolds and intrinsically model curved mean shift space [23, 24]. In contrast, we enforce the structure of the mean histogram directly as an explicit constraint. While intrinsic formulation is of great theory interest, our primary objective is to effectively perform mode seeking given the problem setting for certain task.

Sheikh et al. [32] proposed medoid shift, a mode seeking method that is able to adopt arbitrary, non-differentiable distance metrics. The method essentially transforms the mode seeking problem into a finite point searching problem. The shifted kernel location can appear at limited locations where there are data, thus only pair-wise data distance is needed and no metric differentiability is required. As is reported by [33], however, medoid shift is prone to over-fragmentation when data is sparse. On the other hand, the computation complexity of medoid shift increases significantly with respect to the increase of data size. Only simple, small scale (with respect to image size and number) image segmentation experiments were tested in [32]. Our method does not find approximate shifting locations but seeks an exact, optimal location for each kernel shift step. The proposed method thus works better with relatively sparse data, while its computational complexity increases relatively slower with the increase of data size.

6.2 The Proposed Image Segmentation Method

Our segmentation method consists of three major steps to perform segmentation. For any input image, the algorithm automatically decides the segment number with no human interaction. An algorithmic flow is illustrated in Fig.6.1.

6.2.1 Representation by Textons

Using raw pixel-wise features is not be robust to noise and is difficult to extract invariant properties from the images. We convolve the image with a set of 17 filters (filter bank) to generate 17 response images, constructing a compact pixel-level image representation. In detail, we adopt a bank of 17 filters of size 15×15 which is composed

of Gaussians with 3 different scales (1, 2, 4) applied to LAB channels, Lapacians of Gaussians with 4 different scales (1, 2, 4, 8) and the derivatives of Gaussians with two different scales (2, 4) for each axis (x and y). The filter bank we adopted is exactly the same as that adopted by [66, 68].

The obtained 17-D response image pixels are to be clustered to generate textons. Unlike popular texton generation schemes which commonly use k-means as the clustering method, we adopt K cluster Expectation-Maximization to softly cluster response image pixels and generate K textons. Since k-means only assumes spherical cluster shape which can be far from real data distribution, its texton representation and the region texton statics are far inferior than EM. Using EM we are also able to obtain the K posterior probabilities of belonging to the K textons for each pixel.

6.2.2 Superpixelization and Local Bag of Textons

To reduce computational complexity, we use the method proposed by X. Ren et al. [38, 72, 73] to generate superpixelized images[1]. The parameters N_sp, N_sp2 and N_ev corresponding to the number of superpixels coarse/fine and the number of eigenvalues are first respectively set to 200, 1000 and 40. By this set of parameters we are able to obtain coarsely and finely superpixelized images with more than 200 and 1000 superpixels respectively.

For each coarse superpixel and fine superpixel, we softly vote its texton frequency by averaging posterior texton probabilities over all member pixels and construct a histogram for each superpixel. We call this method "Bag of textons" since there is no constraint on the textons sequence. And just like BoW where frequency of words characterizes the document type, the frequency of textons here characterizes the region appearance and defines the similarities between any two regions.

The set of coarse superpixels are the basic units we want to cluster to generate final segmentations, while the set of fine superpixels serve as mode seeking samples for pdf estimation. For each coarse superpixel, a histogram kernel and a spatial kernel

[1]The corresponding Matlab code is kindly available at http://www.cs.sfu.ca/∼mori/research/superpixels/

are initialized with respect to the superpixel histogram and superpixel spatial location. Mode seeking is then performed for each coarse superpixel based on samples (fine superpixel histograms and spatial locations). The advantage of such strategy is that larger coarse superpixels speeds up the algorithm and contain more region information, while the larger number of fine superpixels give adequate sampling support to estimate a better pdf.

6.2.3 Proposed Convex Shift Algorithm

For traditional mean shift algorithm, suppose $\mathbf{x}^r$ and $\mathbf{x}^s$ respectively represents the d dimensional feature space vector and 2 dimensional spatial coordinate of an image pixel. For mean shift based image segmentation, one adopts the following multivariate kernel density estimator:

$$\hat{f}_{h_r,h_s}(\mathbf{x}^r,\mathbf{x}^s) = \frac{C}{Nh_r^d h_s^2}\sum_{i=1}^{N} k(\|\frac{\mathbf{x}^r-\mathbf{x}_i^r}{h_r}\|^2)k(\|\frac{\mathbf{x}^s-\mathbf{x}_i^s}{h_s}\|^2). \tag{6.1}$$

where the function $k(x)$ is the *profile* of a kernel and C is a normalization constant that makes the above multivariate kernel integrates to one. $h_r > 0$ and $h_s > 0$ are the smoothing parameters called the bandwidth. Taking the derivative of $\hat{f}(\mathbf{x}^r,\mathbf{x}^s)$ with respect to $\mathbf{x}^r$ and defining the new kernel profile $g(x) = -k'(x)$, one has:

$$\begin{aligned}
&\frac{\partial f_{h_r,h_s}(\mathbf{x}^r,\mathbf{x}^s)}{\partial \mathbf{x}^r} \\
&= \frac{2C}{Nh_r^{d+2}h_s^2}\sum_{i=1}^{N}(\mathbf{x}_i^r-\mathbf{x}^r)g(\|\frac{\mathbf{x}^r-\mathbf{x}_i^r}{h_r}\|^2)k(\|\frac{\mathbf{x}^s-\mathbf{x}_i^s}{h_s}\|^2) \\
&= \frac{2C}{Nh_r^{d+2}h_s^2}[\sum_{i=1}^{N} g(\|\frac{\mathbf{x}^r-\mathbf{x}_i^r}{h_r}\|^2)k(\|\frac{\mathbf{x}^s-\mathbf{x}_i^s}{h_s}\|^2)] \\
&\quad [\frac{\sum_{i=1}^{N}\mathbf{x}_i^r g(\|\frac{\mathbf{x}^r-\mathbf{x}_i^r}{h_r}\|^2)k(\|\frac{\mathbf{x}^s-\mathbf{x}_i^s}{h_s}\|^2)}{\sum_{i=1}^{N} g(\|\frac{\mathbf{x}^r-\mathbf{x}_i^r}{h_r}\|^2)k(\|\frac{\mathbf{x}^s-\mathbf{x}_i^s}{h_s}\|^2)} - \mathbf{x}^r]
\end{aligned} \tag{6.2}$$

The last term of equation (6.2) is the *mean shift* for the feature space kernel.

$$\mathbf{m}_{h_r,h_s}(\mathbf{x}^r) = \frac{\sum_{i=1}^{N} \mathbf{x}_i^r g(\|\frac{\mathbf{x}^r - \mathbf{x}_i^r}{h_r}\|^2) k(\|\frac{\mathbf{x}^s - \mathbf{x}_i^s}{h_s}\|^2)}{\sum_{i=1}^{N} g(\|\frac{\mathbf{x}^r - \mathbf{x}_i^r}{h_r}\|^2) k(\|\frac{\mathbf{x}^s - \mathbf{x}_i^s}{h_s}\|^2)} - \mathbf{x}^r \tag{6.3}$$

The mean shift vector for the spatial kernel can be obtained similarly.

Since we use histograms to model region statistics, we adopt K-L divergence to measure the distance between two histograms:

$$d_{KL}(H, K) = \sum_{p=1}^{d} h_p \log \frac{h_p}{k_p}.$$

where d is the histogram dimension, $H = [h_1, h_2, ..., h_d]^\top$ and $K = [k_1, k_2, ..., k_d]^\top$ are two histograms with the constraints $\sum_{i=1}^{d} h_p = \sum_{i=1}^{d} k_p = 1$. Suppose the histogram kernel center is denoted as $\mathbf{x}^h = [x_{(1)}^h, x_{(2)}^h, ..., x_{(d)}^h]^\top$ and the histogram of the ith sample (fine superpixel) is denoted as $\mathbf{x}_i^h = [x_{i,(1)}^h, x_{i,(2)}^h, ..., x_{i,(d)}^h]^\top$. Plugging in the K-L divergence distance measure, we have the following density estimator:

$$\hat{f}_{h_h,h_s}(\mathbf{x}^h, \mathbf{x}^s) = \frac{C}{N h_h^d h_s^2} \sum_{i=1}^{N} k(\frac{d_{KL}(\mathbf{x}^h, \mathbf{x}_i^h)}{h_h^2}) k(\|\frac{\mathbf{x}^s - \mathbf{x}_i^s}{h_s}\|^2). \tag{6.4}$$

The mode seeking problem thus becomes increasing the estimated density subject to the sum of histogram bins in each color channel equals to 1, which is a constrained gradient ascent problem. For histogram kernel, we introduce linear relaxation using K-L divergence kernel with a linear profile, while for spatial kernel, the normal kernel is adopted. Notice that the K-L divergence kernel is meaningful only when the histogram structure is preserved. The density estimator thus becomes:

$$\hat{f}_{h_h,h_s}(\mathbf{x}^h, \mathbf{x}^s) = \frac{C}{N h_h^d h_s^2} \sum_{i=1}^{N} k_{KL}(\frac{d_{KL}(\mathbf{x}^h, \mathbf{x}_i^h)}{h_h^2}) k_N(\|\frac{\mathbf{x}^s - \mathbf{x}_i^s}{h_s}\|^2). \tag{6.5}$$

$$k_{KL}(x) = \begin{cases} 1 - x & 0 \le x \le 1 \\ 0 & x > 1 \end{cases}. \tag{6.6}$$

$$k_N(x) = \exp(-\frac{1}{2}x) \tag{6.7}$$

To solve the problem, rewrite (6.12) into the following form:

$$\begin{aligned}\mathbf{x}^{r(l+1)} =& \frac{\sum_{i=1}^{N} \mathbf{x}_i^r g(\|\frac{\mathbf{x}^{r(l)}-\mathbf{x}_i^r}{h_r}\|^2) k(\|\frac{\mathbf{x}^{s(l)}-\mathbf{x}_i^s}{h_s}\|^2)}{\sum_{i=1}^{N} g(\|\frac{\mathbf{x}^{r(l)}-\mathbf{x}_i^r}{h_r}\|^2) k(\|\frac{\mathbf{x}^{s(l)}-\mathbf{x}_i^s}{h_s}\|^2)} \\ &= \arg\min_{\mathbf{x}^r} \sum_{i=1}^{N} \|\mathbf{x}_i^r - \mathbf{x}^r\|^2 \\ & \quad g(\|\frac{\mathbf{x}^{r(l)} - \mathbf{x}_i^r}{h_r}\|^2) k(\|\frac{\mathbf{x}^{s(l)} - \mathbf{x}_i^s}{h_s}\|^2).\end{aligned} \tag{6.8}$$

where $\mathbf{x}^{r(l)}$ denotes the color space kernel location in the lth iteration. Recall the K-L divergence kernel we introduced. The linear kernel profile yields:

$$g_{KL}(x) = \begin{cases} 1 & 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} . \tag{6.9}$$

Plugging the kernel profile $g_{KL}(x)$ and the K-L divergence measure in equation (6.8), and changing the feature space kernel into the histogram kernel, the histogram kernel shift can be formulated as solving the following convex problem:

$$\begin{aligned} \min_{\mathbf{x}^h} \quad & \sum_{i,\mathbf{x}_i^h \in \mathbf{S}^{(l)}} k_N(\|\frac{\mathbf{x}^{s(l)} - \mathbf{x}_i^s}{h_s}\|^2) \sum_{p=1}^{d} x_{(p)}^h \log \frac{x_{(p)}^h}{x_{i,(p)}^h} \\ s.t. \quad & \mathbf{x}^h \succeq 0 \\ & ||\mathbf{x}^h(1:K)||_1 = 1 \end{aligned}$$

where $\mathbf{S}^{(l)} = \{\mathbf{x}_i^h | d_{KL}(\mathbf{x}^{h(l)}, \mathbf{x}_i^h) \le h_h^2\}$. One can verify the equivalence between solving the above problem and increasing the density estimator in equation (6.11). We will verify this property in the next subsection. For the spatial kernel, the strategy for calculating the spatial kernel shift is identical to that in mean shift:

$$\begin{aligned} &\mathbf{m}_{h_r,h_s}(\mathbf{x}^s) = \\ &\frac{\sum_{i=1}^{N} \mathbf{x}_i^s g_N(\|\frac{\mathbf{x}^s-\mathbf{x}_i^s}{h_s}\|^2) k_{KL}(\frac{d_{KL}(\mathbf{x}^h,\mathbf{x}_i^h)}{h_h^2})}{\sum_{i=1}^{N} g_N(\|\frac{\mathbf{x}^r-\mathbf{x}_i^r}{h_s}\|^2) k_{KL}(\frac{d_{KL}(\mathbf{x}^h,\mathbf{x}_i^h)}{h_h^2})} - \mathbf{x}^s \end{aligned} \tag{6.10}$$

The segmentation algorithm can be described as follows:

1. For each coarse superpixel, initialize its histogram kernel and spatial kernel according to the region color statistics and the mean spatial coordinate of the contained pixels.

2. Recursively shift the histogram kernel by solving the above convex problem using a convex solver and shift the spatial kernel according equation (6.10) until convergence.

3. Group the set of coarse superpixels that share similar histogram kernel locations.

6.2.4 Algorithm Convergence

Definition 6.2.1 *For any sequential step l and $l+1$ and the corresponding histogram kernel location $\mathbf{x}^h(l)$ and $\mathbf{x}^h(l+1)$, the transitory density estimation is defined as:*

$$\hat{ft}_{h_h,h_s}(\mathbf{x}^{h(l+1)},\mathbf{x}^s) = C_1 - C_2 \sum_{i,\mathbf{x}_i^h \in \mathbf{S}^{(l)}} k_N(\|\frac{\mathbf{x}^{s(l)}-\mathbf{x}_i^s}{h_s}\|^2) \sum_{p=1}^{d} x_{(p)}^{h(l+1)} \log \frac{x_{(p)}^{h(l+1)}}{x_{i,(p)}^h} \tag{6.11}$$

where $C_1 = \frac{C}{Nh_h^d h_s^2} \sum_{i=1}^{N} k_N(\|\frac{\mathbf{x}^s-\mathbf{x}_i^s}{h_s}\|^2)$, $C_2 = \frac{C}{Nh_h^d h_s^2 h_h^2}$.

Lemma 6.2.1 $\hat{ft}_{h_h,h_s}(\mathbf{x}^{h(l+1)},\mathbf{x}^s) \geq \hat{f}_{h_h,h_s}(\mathbf{x}^{h(l)},\mathbf{x}^s)$

Proof: According to equation (6.4), we have:

$$\hat{f}_{h_h,h_s}(\mathbf{x}^{h(l)},\mathbf{x}^s) = C_1 - C_2 \sum_{i,\mathbf{x}_i^h \in \mathbf{S}^{(l)}} k_N(\|\frac{\mathbf{x}^{s(l)}-\mathbf{x}_i^s}{h_s}\|^2) \sum_{p=1}^{d} x_{(p)}^{h(l)} \log \frac{x_{(p)}^{h(l)}}{x_{i,(p)}^h} \tag{6.12}$$

According to convex shift which is in the form of constrained minimization, $\mathbf{x}^{h(l+1)}$ is obtained through minimizing $\sum_{i,\mathbf{x}_i^h \in \mathbf{S}^{(l)}} k_N(\|\frac{\mathbf{x}^{s(l)}-\mathbf{x}_i^s}{h_s}\|^2) \sum_{p=1}^{d} x_{(p)}^h \log \frac{x_{(p)}^h}{x_{i,(p)}^h}$ over $\mathbf{x}^h$, we thus directly proved Lemma 6.2.1.

Lemma 6.2.2 $\hat{f}_{h_h,h_s}(\mathbf{x}^{h(l+1)},\mathbf{x}^s) \geq= \hat{ft}_{h_h,h_s}(\mathbf{x}^{h(l+1)},\mathbf{x}^s)$

Proof: Since some of the samples belonging to the lth kernel may go out of the range of $l+1$th kernel, these samples contributes negative values to $\hat{ft}_{h_h,h_s}(\mathbf{x}^{h(l+1)},\mathbf{x}^s)$ while their contribution to $\hat{f}_{h_h,h_s}(\mathbf{x}^{h(l+1)},\mathbf{x}^s)$ is 0. In addition, new samples may come within the range of the $l+1$th kernel, which contributes nonnegative values to $\hat{f}_{h_h,h_s}(\mathbf{x}^{h(l+1)},\mathbf{x}^s)$. Thus, we have the above lemma.

Theorem 6.2.1 *The estimated density monotonically increases with each convex shift step and the algorithm converges.*

Proof: Spatial kernel is independent with histogram kernel and spatial kernel shift also increases the estimated density [2]. According to Lemma 6.2.1 and Lemma 6.2.2, the estimated density thus monotonically increases with iterative histogram and spatial kernel shift. Since the estimated density is upper bounded, the algorithm is guaranteed to converge.

6.3 Experimental Results

We perform segmentation test on a number of natural images selected from the Berkeley Segmentation Dataset. For convex shift, a simple postprocessing is used to eliminate single superpixels by merging them into the most similar neighboring regions. The bandwidth parameters h_h, h_s are respectively set to 1.2 and 60. Our segmentation results are compared with segmentations obtained by quick shift and mean shift. We also compare our method with state of the art segmentation methods such as the Fusion of Clustering Results (FCR) method [74], the Probabilistic Rand Index Fusion (PRIF) method [76] and *gPb*-owt-ucm [79]. We use the VLFeat Matlab package [24] to implement quick shift. The parameters $ratio$, $kernelsize$ and $maxdist$ are respectively set to 0.5, 12 and 30, which is observed to be the best trade off to avoid both oversmoothing and oversegmentation. For the majority of mean shift experiments, we set h_s, h_r and minimum region size M respectively to be 8, 7 and 100 - the set of segmentation parameters adopted in [1]. Smaller bandwidth parameters are chosen for image 4, 19, 23, 24 in order to prevent serious over-merging. We adopt a unified UCM threshold for *gPb*-owt-ucm and tune it to visually optimize its segmentation. The comparison of segmentation results is illustrated in Fig.6.3 to Fig.6.10. Experimental results indicate the superiority of using the proposed method, especially on those images being more complex and textured. Under the scheme of "Bag of textons", our method significantly outperforms quick shift and mean shift for incorporating abundant textual information.

Our method also slightly outperforms FCR and PRIF - which are well-designed state of the art segmentation methods - and is comparable with *gPb*-owt-ucm. We observe that *gPb*-owt-ucm indeed is very powerful but does suffer from over-merging through weak boundary and over-segmentation caused by strong intra-region variation (common problems with contour finding methods). Notice that in contrast to *gPb*-owt-ucm, we have not even elaborately design the spatial constraint and local discontinuity rule to obtain better segmentations. Previous works such as [56, 75] allow one to plug in spatial consistency information on mode seeking in a way far better than current scheme. With better spatial consistency information, a further boost of the segmentation quality is expected.

Figure 6.3: Comparison of segmentation results obtained by different methods. Each row respectively corresponds to the original images and results obtained by quick shift, mean shift.

Figure 6.4: Comparison of segmentation results obtained by different methods. Each row respectively corresponds to the results obtained by FCR, PRIF, *gPb*-owt-ucm and the proposed method.

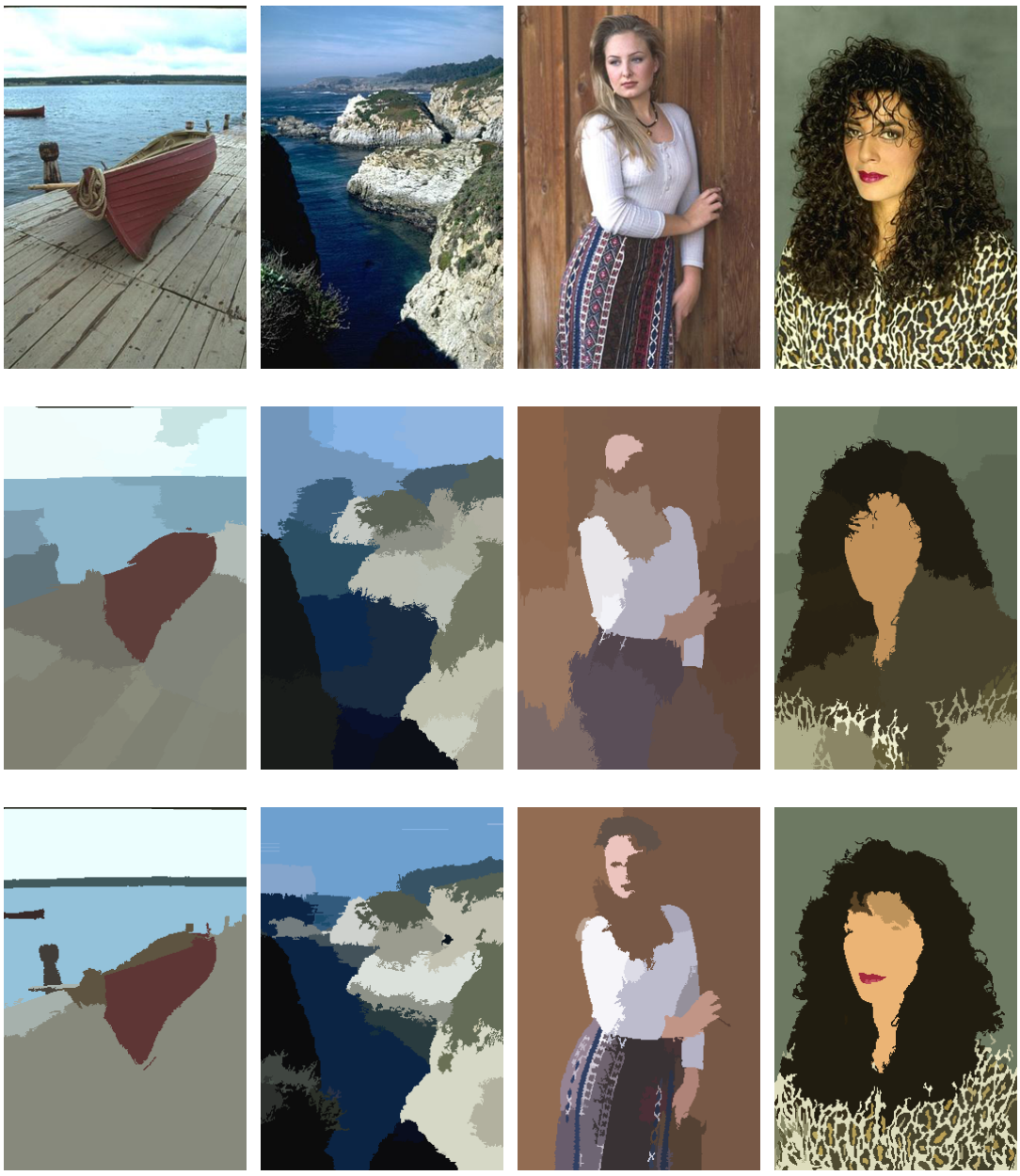


Figure 6.5: Comparison of segmentation results obtained by different methods. Each row respectively correspond to the original images and results obtained by quick shift, mean shift.

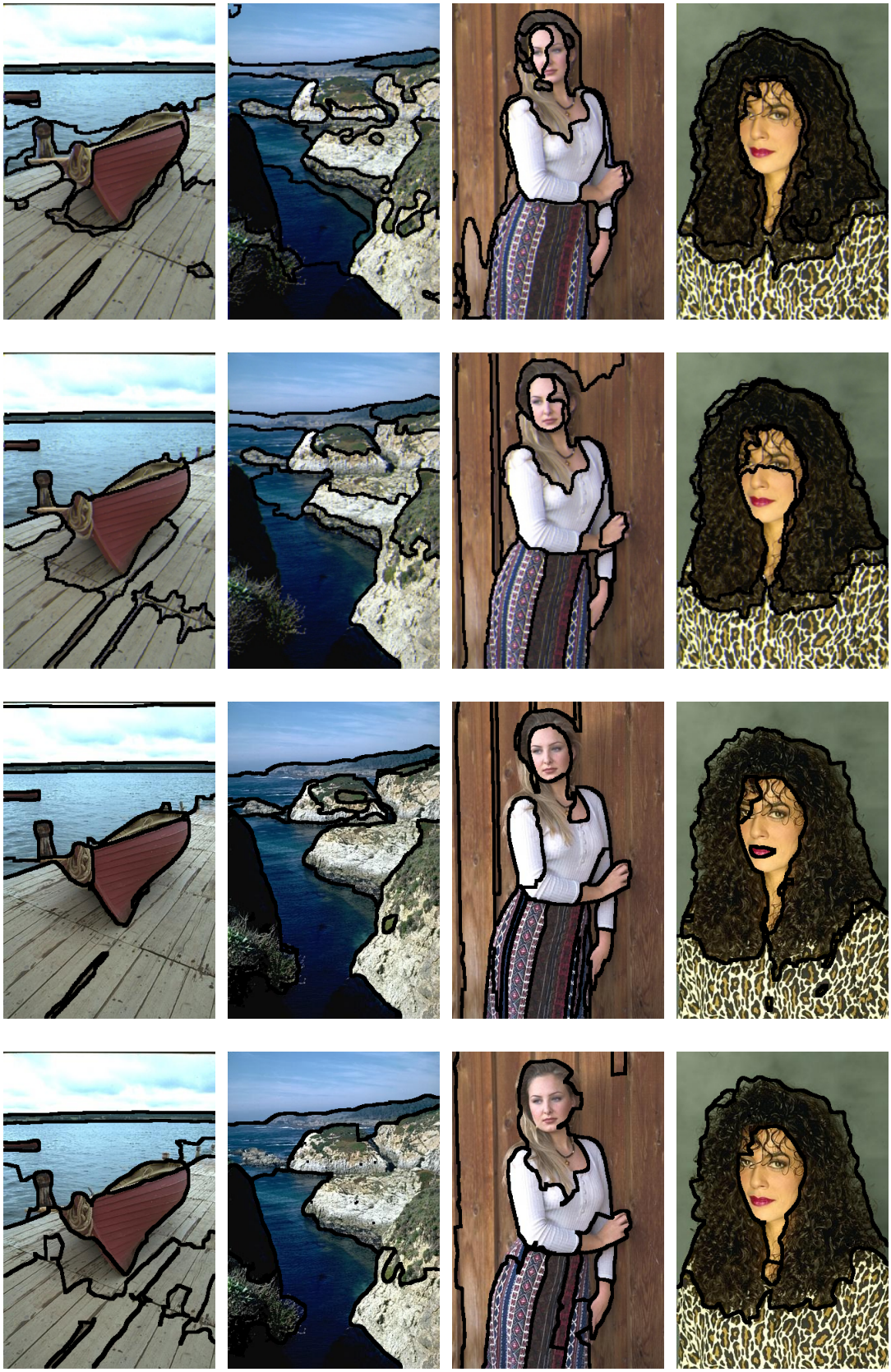


Figure 6.6: Comparison of segmentation results obtained by different methods. Each row respectively corresponds to the results obtained by FCR, PRIF, *gPb*-owt-ucm and the proposed method.

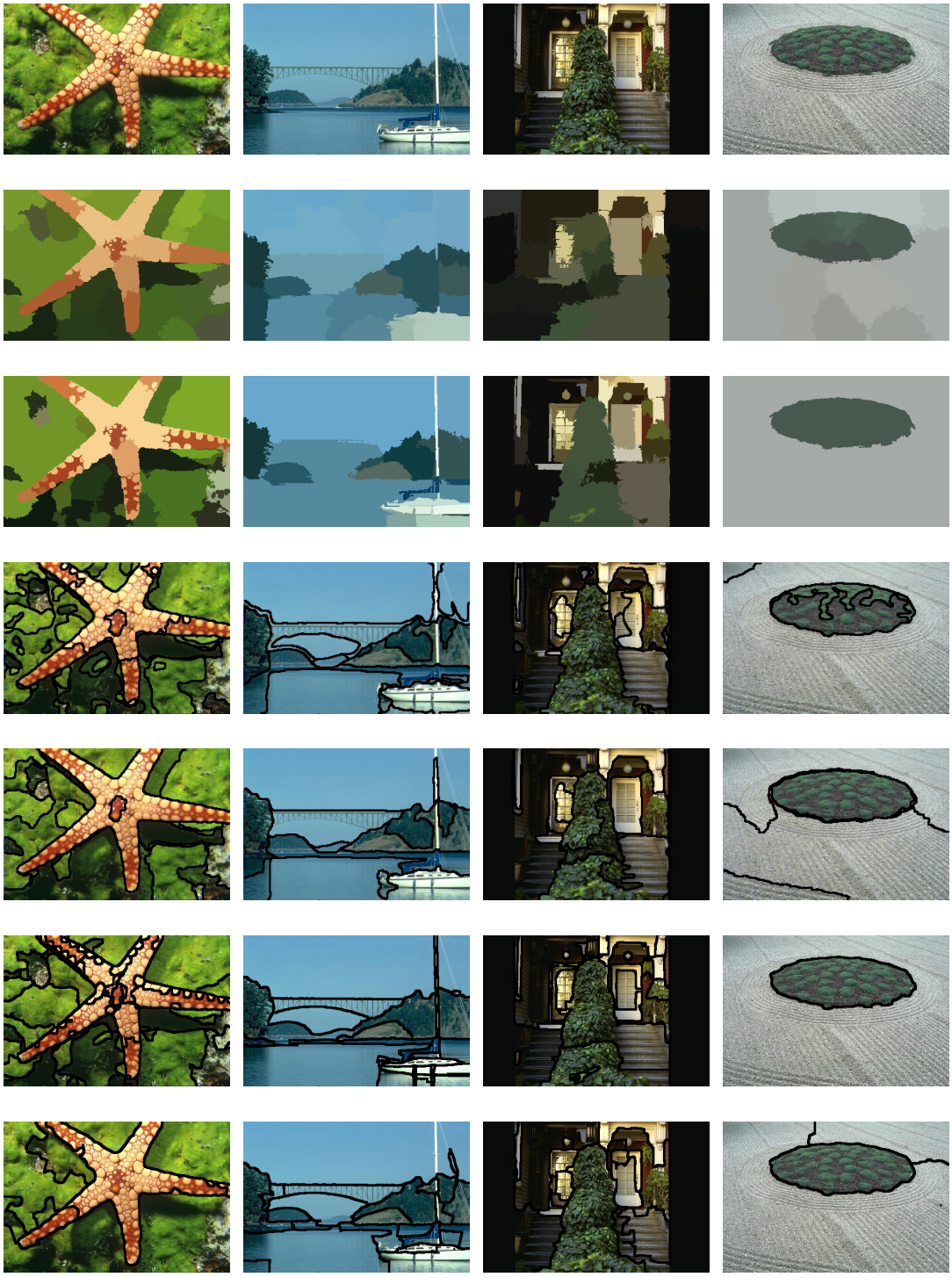


Figure 6.7: Comparison of segmentation results obtained by different methods. Each row respectively corresponds to the original images and results obtained by quick shift, mean shift, FCR, PRIF, *gPb*-owt-ucm and the proposed method.

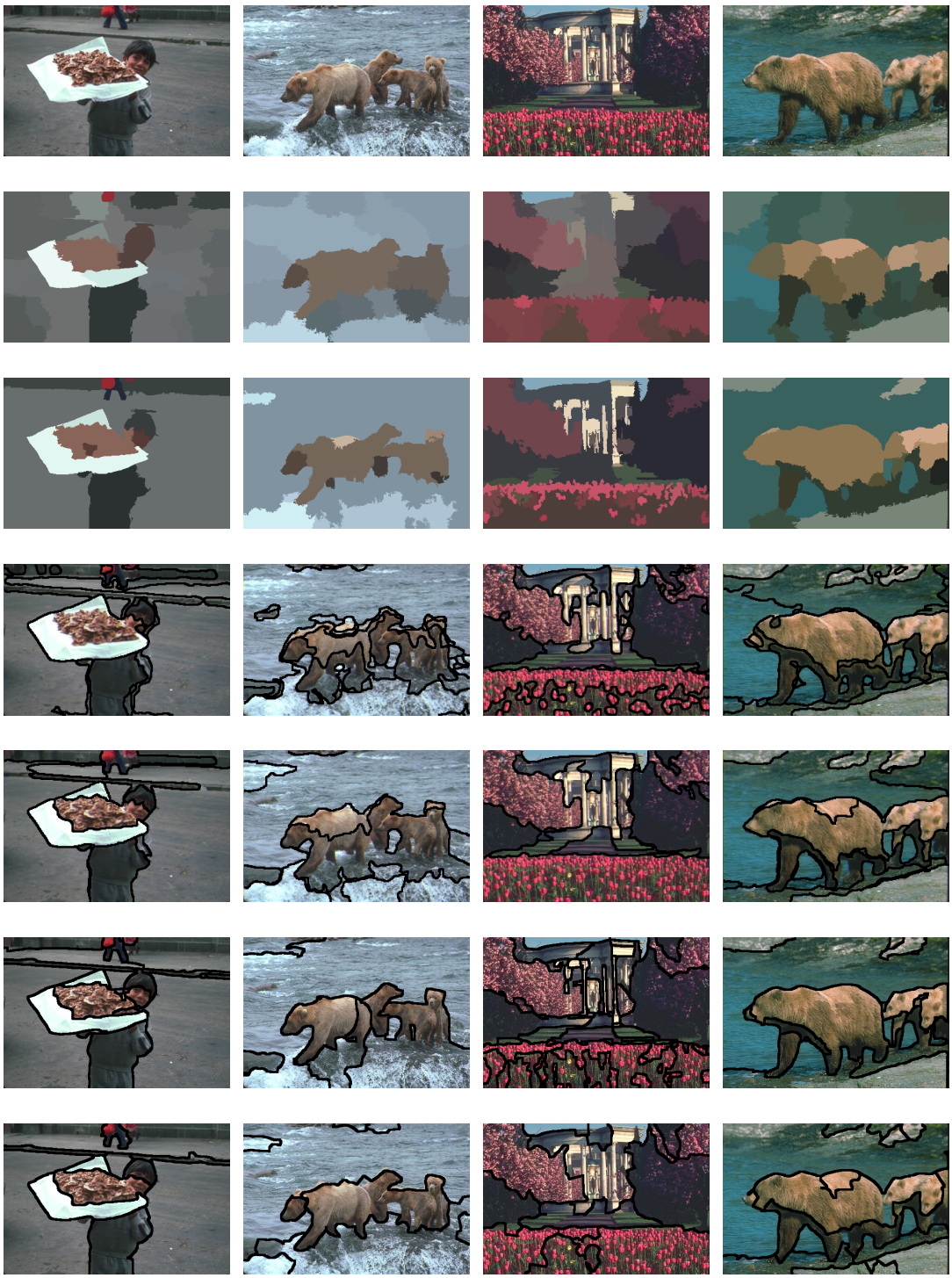


Figure 6.8: Comparison of segmentation results obtained by different methods. Each row respectively corresponds to the original images and results obtained by quick shift, mean shift, FCR, PRIF, *gPb*-owt-ucm and the proposed method.

Figure 6.9: Comparison of segmentation results obtained by different methods. Each row respectively corresponds to the original images and results obtained by quick shift, mean shift, FCR, PRIF, *gPb*-owt-ucm and the proposed method.

Figure 6.10: Comparison of segmentation results obtained by different methods. Each row respectively corresponds to the original images and results obtained by quick shift, mean shift, FCR, PRIF, *gPb*-owt-ucm and the proposed method.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

In this thesis, we proposed novel methods for unsupervised image segmentation and clustering of data/3D point cloud in an attempt to improve the state of the art segmentation performance and reduce both oversegmentation and overmerging regarding human perception. In particular, we looked into graph theoretic approach and arbitrarily shaped clustering, for graph carries significant inter-region structural information while arbitrarily shaped clustering provides better flexibility to model irregular clusters and handle intra-region variation. Our major contribution lies in the proposing of a complete mode seeking framework where any tree structure carrying compact structural information can be embedded and any histogram feature descriptors can be plugged in. More detailedly, we have contributed in the following several aspects:

In chapter 3, we proposed an effective contour finding method that generates favorable partitioned results with less overmerging in the presence of weak boundaries. Our approach is closely related to efficient graph based image segmentation, an MST graph partitioning method both computationally efficient and intuitively simple, and yet surprisingly effective. Although the formulated graph partitioning problem in EGS is tractable and its generated segmentation proved to be "neither too coarse nor too fine", our research show that it can be too coarse under better definition of "coarse" and "fine", which is adopted by our proposed method. Regarding this issue, we proposed a novel boundary predicate that serves as a better contour estimator by reducing overmerging. The method can also be easily extended to the superpixel framework where feature descriptors can be designed in a much more versatile and sophisticated way, leading to better segmentation results or more favorable biases in specific tasks.

While overmerging problem is often associated contour finding methods, clustering based segmentation suffers much less. Chapter 4 detailedly developed the proposed graph-embedded mode seeking framework which belongs to this category. By introducing the MST space kernel, we proposed a novel mode seeking method that incorporates more cluster structure flexibility which improves mode seeking performance on manifold-structured data. We achieved good algorithm performance in clustering data with highly nonlinear separation boundaries without using any manifold distance or some other non Euclidean metrics, which is of considerable challenge. In addition, the embedding of tree works in compatible with region-wise operations and serves as a good spatial smoothness constraint. The advantage of using the proposed method for image smoothing and segmentation is supported by our experiments.

Chapter 5 demonstrated another real application of graph-embedded mode seeking, by developing a system that can automatically segment objects in complicated urban scenes. The system can separate single, relatively small objects while preserving the connectivity of large, spanning ones. It provides good initial interpretations of the urban scenes, based on which 3D object reconstruction, visualization and recognition can be carried out. Additional features also include the concept of transductive distance projected on the MST, and the prior for bandwidth selection in this particular application.

Selecting good feature descriptor is another key issue in segmentation. In chapter 6 we proposed a modified mode seeking method in which any histogram feature descriptors can be plugged in. The proposed method significantly outperforms traditional mode seeking based image segmentation. It tends to produce excellent segmentations that are perceptually congruous with human perception of similarity on complex, textured images, which is comparable with state of the art segmentation methods.

7.2 Future works

Our imminent effort would be focused on the combination of graph-embedded mode seeking with better feature descriptors. The convex shift algorithm proposed in chapter 6 works in compatible with graph-embedded mode seeking. It is expected that with better feature descriptors and smoothness constraint, more accurate segmentations can be obtained.

Another future work would investigate recursive boundary estimation and region splitting such that overmerging can be much better alleviated. It is shown in chapter 3 that although our method suffers less from "region leak", it is not guaranteed that the produced result will never be "too coarse". Different from boundary estimation which intrinsically suffers from overmerging, region splitting through clustering or finding a "cut" of the graph can effectively avoid this problem and search boundaries giving maximum inter-region difference, even if parts of the boundary is weak. With the definition of maximum likelihood inter-region difference and intra-region similarity, exactly finding results that are neither "too coarse" nor "too fine" probably would not be feasible. However, recursive merging and splitting with boundary estimation and region splitting may be able to search results that is approximately neither "too coarse" nor "too fine".

Finally, a long-term work to be done is the incorporation of higher level information into the proposed frameworks so that one can achieve more intelligent and semantically meaningful segmentation results.

REFERENCES

[1] G. Economou, A. Fotinos, S. Makrogiannis and S. Fotopoulos, "Color image edge detection based on nonparametric estimation," In *Proc. Int. Conf. Image Processing*, 2001, pp. 922-925.

[2] K. Haris, S. N. Efstratiadis, N. Maglaveras, A. K. Katsaggelos, "Hybrid image segmentation using watershed and fast region merging," *IEEE Trans. Image Process.*, vol. 7, pp. 1684-1699, Dec. 1998.

[3] O. Lezoray, H. Cardot, "Cooperation of color pixel classification schemes and color watershed: A study for microscopic images," *IEEE Trans. Image Process.*, vol. 11, pp. 783-789, 2002

[4] E. J. Pauwels, G. Frederix, "Finding salient regions in images," *Computer Vision and Image Understanding*, vol. 75, pp. 73-85, 1999

[5] Y. Qian, R. Zhao, "Image segmentation based on combination of global and local information," In *Proc. Int. Conf. Image Processing*, Santa Barbara, CA., 1997, pp. 204-207.

[6] Y. H. Yang, J. Liu, "Multiresolution image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, pp. 689-700, 1994.

[7] M. Zhang, L. O. Hall, D. B. Goldgof, "A generic knowledge-guided image segmentation and labeling system using fuzzy clustering algorithms," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 32, pp. 571-582, 2002.

[8] Y. Boykov, O. Veksler, R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23 (11) (2001) 1222-1239.

[9] P. F. Felzenszwalb, D. P. Huttenlocher, "Image segmentation using local variation," In *CVPR*, Santa Barbara, CA., 1998, pp. 98-103.

[10] Y. Gdalyahu, D. Weinshall, M. Werman, "Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Trans. Image Process.*, vol. 23, pp. 1053-1074, 2001.

[11] D. P. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, "Computing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 850-863, 1993.

[12] D. W. Jacobs, D. Weinshall, Y. Gdalyahu, "Classification with nonmetric distances: Image retrieval and class representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 583-600, 2000.

[13] A. K. Jain, D. Zongker, "Representation and recognition of hand-written digits using deformable templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 1386-1391, 1997.

[14] O. J. Morris, J. Lee, A. G. Constantinides, "Graph theory for image analysis: An approach based on the shortest spanning tree," *IEE Proceedings, Part F, Communication, Radar and Signal Processing,* vol. 133, pp. 146-152, 1986.

[15] J. Shi, J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 888-905, 2000.

[16] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 1101-1113, 1993.

[17] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput. C-20*, pp. 68-86, 1971.

[18] S. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, pp. 267-278, 1994.

[19] P. Corsini, B. Lazzerini, F. Marcelloni, "A fuzzy relational clustering algorithm based on a dissimilarity measure extracted from data," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 34, pp. 775-782, 2004.

[20] S. J. Roberts, "Parametric and nonparametric unsupervised cluster analysis," *Pattern Recognition*, vol. 30, pp. 261-272, 1997.

[21] R. Schumeyer, K. Barner, "A color-based classifier for region identification in video," In *Proc. Visual Communications and Image Processing*, vol. 3309, pp. 189-200, 1998.

[22] R. Yager, D. filev, "Generation of fuzzy rules by mountain clustering," *J. Intell. Fuzzy Syst.*, vol. 2, pp. 209-219, 1994.

[23] S. Wang, J. M. Siskind, "Image segmentation with ratio cut," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 675-690, Jun. 2003.

[24] D. Comaniciu, P. Meer, "Mean shift: A robust approach toward feature space analysis," I*EEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 603-619, May 2002.

[25] W. B. Tao, H. Jin, Y. M. Zhang, "Color image segmentation based on Mean Shift and Normalized Cuts," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 37, pp. 1382-1389, 2007.

[26] S. Makrogiannis, G. Economou, S. Fotopoulos, "A region dissimilarity relation that combines feature-space and spatial information for color image segmentation," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 35, pp. 44-53, 2005.

[27] V. Grau, A. U. J. Mewes, et al, "Improved watershed transform for medical image segmentation using prior information," *IEEE Trans. Med. Imag.*, vol. 23, pp. 447-458, 2004.

[28] R. van den Boomgaard and J. van de Weijer, "On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks,"

In *Proc. Int. Conf. Pattern Recog.*, 2002, pp. 30927-30930.

[29] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." *IEEE Trans. PAMI*, 1984.

[30] H. Zhang, J. E. Fritts and S. A. Goldman. A co-evaluation framework for improving segmentation. In *Proceedings of the SPIE*, vol. 5809, pp.420-430, 2005.

[31] F.R.K. Chung. "Spectral graph theory." Am. Math. Soc., 1997.

[32] Y. A. Sheikh, E. A. Khan and T. Kanade. "Mode-seeking by Medoidshifts." In *ICCV*, 2007.

[33] A. Vedaldi and S. Soatto. "Quick shift and kernel methods for mode seeking." In *ECCV*, 2008.

[34] A. Vedaldi and S. Soatto. "Really quick shift: Image segmentation on a GPU." In *Workshop on Computer Vision using GPUs, held with the ECCV*, 2010.

[35] R. Nock and F. Nielsen. "Statistical region merging." *IEEE Trans. PAMI*, 2004.

[36] P. Felzenszwalb. "Efficient graph-based image segmentation." *IJCV*, 2004.

[37] S. Beucher and F. Meyer, "The morphological approach to segmentation: The watershed transformation." In *Mathematical Morphology in Image Processing*, 1993.

[38] X. Ren and J. Malik. "Learning a classification model for segmentation." In *ICCV*, 2003.

[39] A. Yilmaz, "Object tracking by Asymmetric kernel mean shift with automatic scale and orientation selection." In *CVPR*, 2007.

[40] K. Zhang, J. T. Kwok and M. Tang. "Accelerated convergence using dynamic mean shift." In *ECCV*, 2006.

[41] R. Subbarao and P. Meer. "Nonlinear mean shift for clustering over analytic manifolds." In *CVPR*, 2006.

[42] J. Hu, S. You and U. Neumann, "Approaches to large-scale urban modeling," *Computer Graphics and Applications, IEEE*, vol. 23, pp. 62-69, 2003.

[43] A. Golovinskiy, V.G. Kim and T. Funkhouser, "Shape-based recognition of 3d point clouds in urban environments," In *ICCV*, 2009.

[44] A. Frome, D. Huber, R. Kolluri, T. Bulow and J. Malik, "Recognizing objects in range data using regional point descriptors," In*ECCV*, 2004.

[45] E.B. Meier and F. Ade, "Object detection and tracking in range image sequences by separation of image features," In*IEEE International Conference on Intelligent Vehicles*, 1998.

[46] A. Jaakkola, J. Hyyppa, H. Hyyppa and A. Kukko, "Retrieval algorithms for road surface modelling using laser-based mobile mapping," *Sensors*, vol. 8, no. 9, pp. 5238-5249, 2008.

[47] H. Xu, N. Gossett and B. Chen, "Knowledge and heuristic-based modeling of laser-scanned trees," *ACM Transactions on Graphics*, vol. 26, no. 4, pp. 19, 2007.

[48] Y. Wang, H. Weinacker and B. Koch, "A lidar point cloud based procedure for vertical canopy structure analysis and 3D single tree modelling in forest," *Sensors*, vol. 8, pp. 3938-3951, 2008.

[49] F. Lafarge, X. Descombes, J. Zerubia and M. Pierrot-Deseilligny, "Building reconstruction from a single DEM," *CVPR*, 2008.

[50] J. Chen and B. Chen, "Architectural modeling from sparsely scanned range data," *IJCV*, vol. 78, no. 2, pp. 223-236, 2008.

[51] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang and J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Transactions on Graphics*, vol. 29, no.5, 2010.

[52] L. Nan, A. Sharf, H. Zhang, D. Cohen-Or and B. Chen, "SmartBoxes for interactive urban reconstruction," In *ACM SIGGRAPH*, 2010.

[53] E.H. Lim and D. Suter, "Conditional random field for 3D point clouds with adaptive data reduction," In *Int'l. Conf. on Cyberworlds*, 2007.

[54] A. Frome, D. Huber, R. Kolluri, T. B
"ulow and J. Malik, "Recognizing objects in range data using regional point descriptors," In *ECCV*, 2004.

[55] A.E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE T-PAMI*, vol. 21, no. 5, pp. 433-449, 2002.

[56] Z. Yu, O. Au, K. Tang and C. Xu, "Nonparametric Density Estimation on A Graph: Learning Framework, Fast Approximation and Application in Image Segmentation," In *CVPR*, 2011.

[57] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *IJCV*, vol. 70, no. 2, pp. 109-131, 2006.

[58] X. Shao, K. Katabira, R. Shibasaki and H. Zhao, "Multiple people extraction using 3D range sensor," In *SMC*, 2010.

[59] K. Fukunaga and L. Hostetler. "The estimation of the gradient of a density function with application in pattern recognition." *IEEE Trans. Info. Theory*, 1975.

[60] Y. Cheng. "Mean shift, mode seeking and clustering." *IEEE Trans. PAMI*, 1995.

[61] D. Lewis. "Naive (Bayes) at forty: The independence assumption in information retrieval". In *ECML*, 1998.

[62] FF. Li and P. Perona. "A Bayesian hierarchical model for learning natural scene categories". In *CVPR*, 2005.

[63] FF. Li, R. Fergus and A. Torralba. "Recognizing and learning object categories". In *CVPR 2007 short course*, 2007.

[64] G. Qiu. "Indexing chromatic and achromatic patterns for content-based colour image retrieval". *Pattern Recognition*, 2002.

[65] J. Malik, S. Belongie, T. Leung and J. Shi. “Contour and texture analysis for image segmentation”. *IJCV*, 2001.

[66] J. Shotton. “TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context”. *IJCV*, 2007.

[67] M. Varma and A. Zisserman. “A statistical approach to texture classification from single images”. *IJCV*, 2005.

[68] J. Winn, A. Criminisi and T. Minka. “Categorization by learned universal visual dictionary”. In *ICCV*, 2005.

[69] D. Comaniciu, V. Ramesh and P. Meer. “Real-time tracking of non-rigid objects using mean shift”. In *CVPR*, 2000.

[70] Q. Zhao, Z. Yang, H. Tao. “Differential Earth Movers Distance with its applications to visual tracking”. *IEEE Trans. PAMI*, 2010.

[71] I. Leichter. “Mean shift trackers with cross-bin metrics”. Accpted to *IEEE Trans. PAMI*, 2011.

[72] G. Mori, X. Ren, A. Efros, and J. Malik. “Recovering human body configurations: Combining segmentation and recognition”. In *CVPR*, 2004.

[73] G. Mori. “Guiding model search using segmentation”. In *ICCV*, 2005.

[74] M. Mignotte. “Segmentation by fusion of histogram-based k-means clusters in different color spaces”. *IEEE Trans. Image Proc.*, 2008.

[75] H. Liu and S. Yan. “Robust graph mode seeking by graph shift”. In *ICML*, 2010.

[76] M. Mignotte. “A label field fusion Bayesian model and its penalized maximum rand estimator for image segmentation”. *IEEE Trans. on Image Proc.*, 2010.

[77] H.E. Cetingul and R. Vidal. “Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds”. In *CVPR*, 2009.

[78] A. Vedaldi and B. Fulkerson. “VLFeat - an open and portable library of computer vision algorithms”. http://www.vlfeat.org/, 2008.

[79] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik. “Contour Detection and Hierarchical Image Segmentation”. *IEEE Trans. PAMI*, 2010.

[80] E. Borenstein and S. Ullman, “Class-specific, top-down segmentation,” In *ECCV*, 2002.

[81] E. Borenstein, E Sharon and S. Ullman, “Combining Top-down and Bottom-up Segmentation,” In *CVPR*, 2004.

[82] M. Vasconcelos, G. Carneiro and N. Vasconcelos, “Weakly Supervised Top-down Image Segmentation,” In *CVPR*, 2006.

[83] “Class Segmentation and Object Localization with Superpixel Neighborhoods,” *ICCV*, 2009.

[84] K. Koffka, *Principles of Gestalt Psychology*, Harcourt Brace, New York, 1935.

[85] M. Wertheimer, “Laws of Organization in Perceptual Forms,” *A Source Book of Gestalt Psychology*, W. D. Ellis (ed), pp. 71-88, Harcourt Brace, 1938.

[86] D. G. Lowe, *Perceptual Organization and Visual Recognition*, Kluwer Academics, Boston, 1985.

[87] F. J. Estrada, “Advances in Computational Image Segmentation and Perceptual Grouping,” Ph.D. Thesis, Department of Computer Science, University of Toronto, 2005.

[88] R. M. Haralick and L. G. Shapiro, “Image Segmentation Techniques,” *Computer Vision, Graphics and Image Processing*, vol. 29, no. 1, pp. 100-132, 1985.

[89] N. R. Pal and S. K. Pal, “A Review on Image Segmentation Techniques,” *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1993.

[90] K. Zhang, I. W. Tsang and J. T. Kwok, "Maximum Margin Clustering Made Practical," *IEEE Trans. Neural Networks*, vol. 20, no. 4, pp. 583-596, 2009.

[91] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active contour models." *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321-331, 1988.

List of Publications

Journal Publications

1. **Zhiding Yu**, Oscar C. Au and Chunjing Xu, "Graph-Embedded Mode Seeking for Manifold Structured Data Clustering," to be submitted to IEEE Trans. Pattern Anal. Mach. Intell.

2. **Zhiding Yu** and Oscar C. Au, Ruobing Zou, Weiyu Yu and Jing Tian, "An Adaptive Unsupervised Approach toward Pixel Clustering and Color Image Segmentation," Pattern Recognition, 43, 2010.

Conference Publications

1. **Zhiding Yu**, Ang Li, Oscar C. Au and Chunjing Xu, "Bag of Textons for Image Segmentation via Soft Clustering and Convex Shift," *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) 2012, Providence, Rhode Island, USA. (24.1%)

2. **Zhiding Yu**, Chunjing Xu, Jianzhuang Liu, Oscar C. Au, Xiaoou Tang, "Automatic Object Segmentation from Large Scale 3D Urban Point Clouds through Manifold Embedded Mode Seeking," *ACM Multimedia* (ACM-MM) 2011, Scottsdale, USA. (30%)

3. **Zhiding Yu**, Oscar C. Au, Ketan Tang, Chunjing Xu, "Nonparametric Density Estimation on A Graph: Learning Framework, Fast Approximation and Application in Image Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) 2011, Colorado Springs, USA. (22.5%)

4. **Zhiding Yu**, Oscar C. Au, et al., "Towards Robust and Efficient Segmentation: An Approach based on Inter-Region Contour and Intra-Region Content Analysis," *IEEE International Conference on Multimedia and Expo* (ICME) 2011, Barcelona, Spain. (Top 15%)

5. **Zhiding Yu**, Oscar C. Au, et al., "Graph segmentation revisited: detailed analysis and density learning based implementation," *IEEE International Conference on Multimedia and Expo* (ICME) 2010, Singapore. (30%)

6. Wenxiu Sun, Oscar C. Au, Lingfeng Xu, Yujun Li, Wei Hu and **Zhiding Yu**, "Texture Optimization for Seamless View Synthesis Through Energy Minimization," to appear in *ACM Multimedia* (ACM-MM), Nara, Japan, 2012. (30%)

7. Ketan Tang, Oscar C. Au, Lu Fang, **Zhiding Yu**, Yuanfang Guo, "Multi-scale Analysis of Color and Texture for Salient Object Detection," *IEEE International Conference of Image Processing* (ICIP) 2011, Brussels, Belgium.

8. Ketan Tang, Lu Fang, **Zhiding Yu**, Yuanfang Guo, Oscar C. Au, "How Anti-Aliasing Filter Affects Image Contrast: An Analysis from Majorization Theory Perspective," *IEEE International Conference on Multimedia and Expo* (ICME) 2011, Barcelona, Spain. (30%)

9. Wenxiu Sun, Oscar C. Au, Lingfeng Xu, **Zhiding Yu**, "Adaptive Depth Map Assisted Matting in 3D Video," *IEEE International Conference on Multimedia and Expo* (ICME) 2011, Barcelona, Spain. (30%)

10. Chi Ho Yeung, Oscar C. Au, Ketan Tang, **Zhiding Yu**, "Compressing Similar Image Sets using Low Frequency Template Prediction," *IEEE International Conference on Multimedia and Expo* (ICME) 2011, Barcelona, Spain. (Top 15%)

11. Yuanfang Guo, Oscar C. Au, Ketan Tang, Lu Fang, **Zhiding Yu**, "Data Hiding in Dot Diffused Halftone Images," *International Workshop on Content Protection and Forensics (CPAF)*, held in conjunction with *ICME* 2011, Barcelona, Spain.

12. Ketan Tang, Oscar C. Au, Lu Fang, **Zhiding Yu** and Yuanfang Guo, "Image Interpolation Using Autoregressive Model and Gauss-Seidel Optimization," *International Conference on Image and Graphics* (ICIG) 2011, USTC, Hefei, China.