Graph based approaches for image segmentation and object tracking

Xiaofang Wang

THESE

pour obtenir le grade de
DOCTEUR DE L'ECOLE CENTRALE DE LYON
Spécialité: Informatique

dans le cadre de l'Ecole Doctorale InfoMaths
présentée et soutenue

Graph Based Approaches for: Image Segmentation and Object Tracking

---

Xiaofang Wang
Septembre 2014

Directeur de thèse: Simon MASNOU
Co-directeur de thèse: Liming CHEN

JURY

Prof. Ruan Su	Université de Rouen	Rapporteur
Prof. Patrick Bouthemy	INRIA Rennes	Rapporteur
Prof. Christine Fernandez-Maloigne	Université de Poitiers	Examinateur
Prof. Anuj Srivastava	Florida State University	Examinateur
Prof. Simon Masnou	Université Lyon 1	Directeur de thèse
Prof. Liming Chen	Ecole Centrale de Lyon	Co-directeur de thèse

Acknowledgments

Many people have contributed to make this thesis a reality, by providing me with their guidance, friendship, love, money, code and data.

I would like to start by thanking my advisor, Prof. Simon Masnou, for teaching me a great deal about research. He has been very supportive and given me the freedom to pursue various projects without objection. He is patient, knowledgeable, smart, funny and easy going. We worked together closely, this thesis would not have been possible without his help and his passion for tackling hard problems. Meanwhile, I would like to express my deeply-felt thanks to my co-advisor, Prof. Liming Chen. He has always been extremely generous with his time, ideas and resources, for his attitude and passion of doing research.

I would also want to thank the members of my doctoral committee, Profs. Su Ruan, Patrick Bouthemy, Christine Fernandez-Maloigne and Anuj Srivastava for their time, feedback and interest in my work. I must also acknowledge the important contributions of my other co-authors Dr. Chao Zhu, Dr. Huibin Li, Boyang Gao, Yuxing Tang and Dongming Chen. Their singular abilities, expertise and creativity were fundamental to this thesis and it was great fun to work and travel with both.

I truly enjoyed the time spent doing research in the Lab LIRIS of Ecole Centrale de Lyon in France. I would like to acknowledge my friend and colleagues, for their countless help, thoughtful suggestions, warm and valuable friendship, and contributions to my research. They are Dr. Chao Zhu, Dr. Huibin Li, Boyang Gao, Ningning Liu, Kai Wang, Yuxing Tang, Dr.Dongming Chen, Huiliang Jin, Wei Chen, Wuming Zhang, Huaxiong Ding, Yihang Tang, Ying Lv, Chen Wang. My special thanks are due to Dr. Huibin Li. He discussed with me about my experiment and provided me thoughtful suggestions on applying sparse representation theory. I have learned so much from him, ranging from solid and beautiful academic paper writing to every useful detail for happy life, which are unforgettable and the most precious time in my life. I am also grateful and very glad to work and discuss with Boyang Gao (extremely knowledgeable in just about everything, helpful, and friendly), for his scientific advice on my project of object tracking. Very pleasant corporation with Yuxing Tang on the object detection domain, and with Dongming Chen on the stereo matching domain. I also thank my friends (too many to list here but you know who you are!) for providing support and friendship that I needed. I would like to thank Joy (I think of her as a big sister.) and Guangyu Zhou for being supportive throughout my time here.

I would like to dedicate this thesis to my family, in particular to my parents, my uncle, brothers and sister. My hard-working parents have deep love to people and the circumstances, they are excellent role model for my life. My brothers and their wives, Qianjin Wang and Chunyan Zhang, Xingfu Wang and Zhenhua Wang, my sister and her husband, they are great man for their support to my academic career for many years, and taking care of our parent with great consideration and love. I am very grateful for having such big, warm and harmony family for their unconditional love and care. I love them so much, and I would not have made it this far without them. I would like particularly acknowledge to my friend, Qian Zhang for his selfless and endless support. Finally, I want to express my acknowledge to

Romain Le Goc. He gives me warm support and love.

During my thesis work, I am fully funded by a PhD scholarship from the Chinese Government and the China Scholarship Council. My research in this PhD thesis is also supported by the Freanch research agency, Agence National de Recherche (ANR), through the Visen project within the ERA-NET CHIST-ERA program, under the grant ANR-12-CHIRI-0002-04, and VideoSense project under the grant 2009 CORD 026 02.

Abstract

Image segmentation is a fundamental problem in computer vision. In particular, unsupervised image segmentation is an important component in many high-level algorithms and practical vision systems. In this dissertation, we propose three methods that approach image segmentation from different angles of graph based methods and are proved powerful to address these problems.

Our first method develops an original graph construction method. We also analyze different types of graph construction method as well as the influence of various feature descriptors. The proposed graph, called a local/global graph, encodes adaptively the local and global image structure information. In addition, we realize global grouping using a sparse representation of superpixels' features over the dictionary of all features by solving a ℓ_0-minimization problem. Extensive experiments are conducted on the Berkeley Segmentation Database, and the proposed method is compared with classical benchmark algorithms. The results demonstrate that our method can generate visually meaningful partitions, but also that very competitive quantitative results are achieved compared with state-of-the-art algorithms.

Our second method derives a discriminative affinity graph that plays an essential role in graph-based image segmentation. A new feature descriptor, called weighted color patch, is developed to compute the weight of edges in an affinity graph. This new feature is able to incorporate both color and neighborhood information by representing pixels with color patches. Furthermore, we assign both local and global weights adaptively for each pixel in a patch in order to alleviate the over-smooth effect of using patches. The extensive experiments show that our method is competitive compared to the other standard methods with multiple evaluation metrics.

The third approach combines superpixels, sparse representation, and a new mid-level feature to describe superpixels. The new mid-level feature not only carries the same information as the initial low-level features, but also carries additional contextual cue. We validate the proposed mid-level feature framework on the MSRC dataset, and the segmented results show improvements from both qualitative and quantitative viewpoints compared with other state-of-the-art methods.

Multi-target tracking is an intensively studied area of research and is valuable for a large amount of applications, e.g. video surveillance of pedestrians or vehicles motions for sake of security, or identification of the motion pattern of animals or biological/synthetic particles to infer information about the underlying mechanisms.

We propose a detect-then-track framework to track massive colloids' motion paths in active suspension system. First, a region based level set method is adopted to segment all colloids from long-term videos subject to intensity inhomogeneity. Moreover, the circular Hough transform further refines the segmentation to obtain colloid individually. Second, we propose to recover all colloids' trajectories simultaneously, which is a global optimal problem that can be solved efficiently with optimal algorithms based on min-cost/max flow. We evaluate the proposed framework on a real benchmark with annotations on 9 different videos. Extensive experiments show that the proposed framework outperforms standard methods with large margin.

Résumé

Cette thèse est proposée en deux parties. Une première partie se concentre sur la segmentation d'image. C'est en effet un problème fondamental pour la vision par ordinateur. En particulier, la segmentation non supervisée d'images est un élément important dans de nombreux algorithmes de haut niveau et de systémes d'application. Dans cette thèse, nous proposons trois méthodes qui utilisent la segmentation d'images se basant sur différentes méthodes de graphes qui se révèlent être des outils puissants permettant de résoudre ces problèmes.

Nous proposons dans un premier temps de développer une nouvelle méthode originale de construction de graphe. Nous analysons également différentes méthodes similaires ainsi que l'influence de l'utilisation de divers descripteurs. Le type de graphe proposé, appelé graphe local/global, encode de manière adaptative les informations sur la structure locale et globale de l'image. De plus, nous réalisons un groupement global en utilisant une représentation parcimonieuse des caractéristiques des superpixels sur le dictionnaire de toutes les caractéristiques en résolvant un problème de minimisation ℓ_0. De nombreuses expériences sont menées par la suite sur la base de données <Berkeley Segmentation>, et la méthode proposée est comparée avec des algorithmes classiques de segmentation. Les résultats démontrent que notre méthode peut générer des partitions visuellement significatives, mais aussi que des résultats quantitatifs très compétitifs sont obtenus en comparaison des algorithmes usuels.

Dans un deuxième temps, nous proposons de travailler sur une méthode reposant sur un graphe d'affinité discriminant, qui joue un rôle essentiel dans la segmentation d'image. Un nouveau descripteur, appelé patch pondéré par couleur, est développé pour calculer le poids des arcs du graphe d'affinité. Cette nouvelle fonctionnalité est en mesure d'intégrer simultanément l'information sur la couleur et le voisinage en représentant les pixels avec des patchs de couleur. De plus, nous affectons Ã chaque pixel une pondération à la fois local et globale de manière adaptative afin d'atténuer l'effet trop lisse lié à l'utilisation de patchs. Des expériences approfondies montrent que notre méthode est compétitive par rapport aux autres méthodes standards à partir de plusieurs paramètres d'évaluation.

Finalement, nous proposons une méthode qui combine superpixels, représentation parcimonieuse, et une nouvelle caractéristisation de mi-niveau pour décrire les superpixels. Le nouvelle caractérisation de mi-niveau contient non seulement les mêmes informations que les caractéristiques initiales de bas niveau, mais contient également des informations contextuelles supplémentaires. Nous validons la caractéristisation de mi-niveau proposée sur l'ensemble de données MSRC et les résultats de segmentation montrent des améliorations à la fois qualitatives et quantitatives par rapport aux autres méthodes standards.

Une deuxième partie se concentre sur le suivi d'objets multiples. C'est un domaine de recherche très actif, qui est d'une importance majeure pour un grand nombre d'applications, par exemple la vidéo-surveillance de piétons ou de véhicules pour des raisons de sécurité ou l'identification de motifs de mouvements animaliers

ou de particules synthétiques et biologiques afin de comprendre les mécanismes sous-jacents.

Nous proposons donc une méthode reposant sur la detection puis le suivi de trajectoires de colloïdes massives dans un système de suspension active. Tout d'abord, nous adoptons une méthode de regionalisation par niveau pour segmenter tous les colloïdes de vidéos à long terme dont l'intensité est hétérogène. Par ailleurs, nous utilisons une transformée de Hough circulaire pour affiner la segmentation afin d'identifier les colloïdes individuellement. Ensuite, nous proposons de récupérer les trajectoires de tous les colloïdes simultanément, ce qui est un probléme d'optimisation global qui peut être résolu efficacement avec des algorithmes d'optimisation de type coût minimum / flux maximum. Cette approche est évaluée à partir d'un ensemble de tests reposant sur 9 vidéos différentes de vraies experimentations en physique. Nous montrons ainsi que l'approche proposée surpasse largement les méthodes standards.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Contents

1.1 Computer Vision

Computer vision aims at modeling and imitating human vision using computer software and hardware at different levels. It combines many different branches of knowledge in computer science, mathematics, physiology, and cognitive science. The entire field of computer vision is too broad to be described in details, examples of applications range from industry (e.g. controlling process, automatic inspection), to medicine (e.g. computer-aided diagnosis, tumor detection), biology (e.g. gene expression), or game industry (virtual reality). Although machines are not yet as efficient as the human brain for scene understanding and interpretation, their performances on a few tasks in image denoising, face detection or human pose estimation have reached very good quality. More complex tasks are still intensively studied, as complex image denoising or in painting, image segmentation and classification, object detection and tracking. The reason for such intensive study is that image segmentation and object detection are key steps in a complete computer vision system as illustrated in Fig. 1.1.

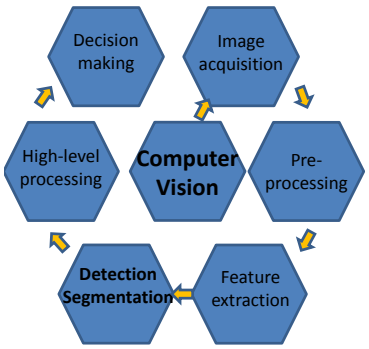


Figure 1.1: Illustration of components consist of a computer vision system.

1.2 Graph Theory

In computer vision, graph theory has been successfully applied to solve many tasks, ranging from low-level tasks (e.g. image segmentation, object tracking, stereo matching, etc.), to high-level tasks (e.g. image classification, object recognition, image parsing, etc.). Beyond this, graphs can be used to model many types of relations and processes in physical, biological, social and information systems, see a few examples in Table 1.1. The popularity of graphs has several reasons: 1) graphs provide discrete and mathematically simple representations that are well adapted to the development of efficient and provably correct methods; 2) they offer data representation flexibility; 3) a lot of methods involving graphs have been developed in other contexts than image processing, yet it is very frequently possible to adapt the methods to an image setting.

Table 1.1: Graph applications

Context	Graph vertices	Graph edges
communication network	telephones, computers	fiber optic cables
transportation network	street intersections, airports	highways, airway routes
hydraulic network	reservoirs, pumping stations	piperline
internet	web pages	hyperlinks
social relationships	people, actors	friendships, movie casts
neural network	neurons	synapses
protein networs	proteins	protein-protein interactions
image	pixels, regions	pairwise relationship

1.3 Image Segmentation

1.3.1 Definition

Image segmentation is a fundamental problem in computer vision. A general definition of image segmentation is to divide an image into meaningful non-overlapping regions, according to some objective criterion, homogeneity in some feature space or

separability in some other one for example. Image segmentation is a core research topic since it is a key element in many computer vision tasks, such as medical imaging for computer-aided diagnosis, cell tracking for biological analysis, motion analysis, see various examples in Fig. 1.2.

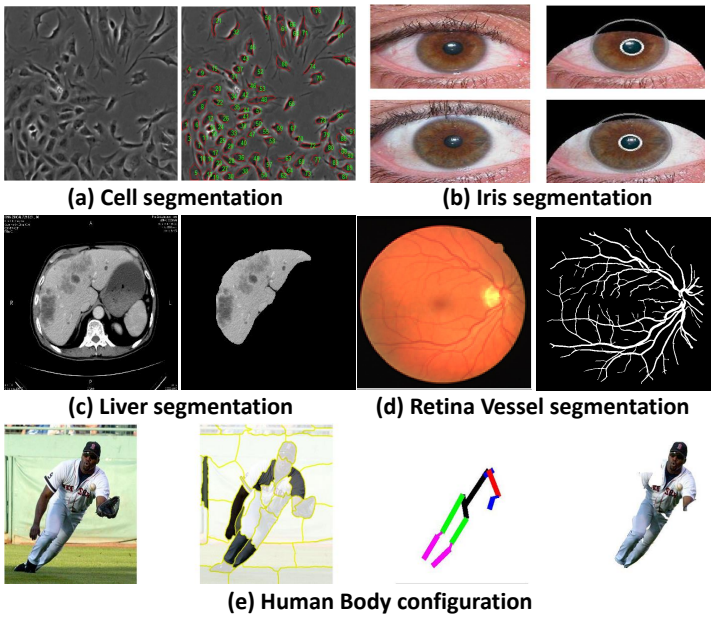


Figure 1.2: Illustration of various image segmentation applications.

1.3.2 The Challenges of Image Segmentation

Image segmentation has been a core research subject in computer vision for many decades. At early stage, image segmentation has been formulated as a bottom-up process. It dates back to Wertheimer Wertheimer [Wertheimer, 1938] who introduced the Gestalt theory for vision perception and studied the factors of perceptual grouping such as proximity, similarity, and good continuity. The low-level principles underlying the Gestalt theory have inspired many approaches to segmentation. Indeed, low-level homogeneous attributes can lead as in Fig. 1.3 to good segmentations which resemble manually-segmented results. Consequently, many works pursued a single optimal segmentation of an image or a few images using one or two low-level or mid-level (e.g. symmetry) cues. However, due to the ambiguity of low and mid-level cues, it is uneasy to get a successful partition. This illustrates of course that image segmentation is an ill-posed problem. It is an easier task for a human observer who performs the segmentation based on different features homogeneity. An example is illustrated in Fig.1.4, where we compare the result produced by a recent efficient bottom-up algorithm [Li et al., 2012] with the corresponding human segmentation. Ideally, we can segment the image as a water region of uniform color, a sky region of uniform smoothness (note that sky color varies across the image), a region of tree leaves of uniform texture, and so forth. However, the automatic segmentation algo-

(a) Color smoothness (b) Texture smoothness

Figure 1.3: Illustration of successful segmentations using low-level perceptual grouping criteria.

rithm is hard to cope with various definitions of uniformity throughout the image, although many methods are designed to combine optimally multiple homogeneous measurements [Cheng et al., 2011a]. It is worth to mention that at early stage, another difficulty of image segmentation is that there is no quantitative benchmarks to evaluate the improvement and to compare with other algorithms and most papers only describe the merits of the output segmentations qualitatively, usually based on results obtained on a few images, as summarized in [Carreira and Sminchisescu, 2010].

Figure 1.4: Illustration of the difficulty of low-level segmentation. SAS is the method presented in Li et al. [2012]

Modern views on the image segmentation have evolved in two aspects: i) one is the role of bottom-up segmentation and ii) the other is taking account mid-level and high-level knowledge to leverage bottom-up segmentation. The underlying force to such change is propelled by : 1) the creation of annotated benchmarks [Arbelaez et al., 2011] [Shotton et al., 2006][Everingham et al.] and new segmentation performance metrics [PRI] [Meila, 2005] [Martin et al., 2001] [Freixenet et al., 2002]; 2) the adoption of machine learning techniques to optimize performance on benchmarks; and 3) relaxing the constraint of working with a single partitioning. An approach became popular by computing several independent segmentations, possibly using different algorithms or varying parameters of the same algorithm [Li et al., 2012][Kim et al., 2010b] [Wang et al., 2013a]. Such new understanding turned image segmentation into a hot topic again, after it got stuck in the nineties [Mundy, 2006].

The first aspect mentioned earlier is that the bottom-up methods should not aim to generate completely correct segments, instead the objective should be to use the results of low-level for mid/high-level process in order to obtain the most appropriate segments in the context of prior world knowledge. A good low-level image segmentation algorithm can significantly reduce the complexity of object segmentation and recognition, which forms the core of high-level vision. For example, some methods [Uijlings et al., 2013] apply low-level segmentation techniques to generate object bounding boxes candidates or use the segmentation result as a kind of feature measuring the objectness of a candidate bounding box [Alexe et al., 2010]. The derived region proposal or bounding boxes can be used in the object detection and localization. In the saliency detection [Cheng et al., 2011b], performing segmentation has become an essential ingredient to obtain high-quality results. In some scene understanding [Li et al., 2009] and semantic segmentation [Carreira et al., 2012a] algorithms, the final results are inferred on over-segmented regions generated by efficient low-level segmentation methods, such as Felzenszwalb and Huttenlocher [Felzenszwalb and Huttenlocher, 2004] or Mean shift [Comaniciu and Meer, 2002].

The second aspect concerns the laws stated in Gestalt theory, which is to say that some mid-level (e.g. symmetry) and high-level vision (past experience and closure) play essential role in vision perceptual process. For example, Shotton et al. [Shotton et al., 2006] incorporate the texture, layout and context using boosting algorithm and the pixel-wise segmentation is achieved with Conditional Random Field (CRF) [Lafferty et al., 2001]. Arbeláez et al. [Arbeláez et al., 2012] proposed region based object detectors which combine top-down poselet detector and global appearance cues.

1.3.3 Graph Based Image Segmentation

In recent years, among the many approaches to image segmentation, graph based methods have become a major trend. In these methods, image segmentation is modeled in terms of partitioning a graph into several sub-graphs such that each of them represents a meaningful object of interest in the image. The very first step is mapping the image elements onto a graph, where the nodes may be pixels, regions, or even user-drawn markers. The graph structure is formed by a set of nodes (also called vertices) and a set of edges that are connections between pairs of nodes. Basically, graph based methods can be categorized into :

- Minimum spanning tree (also called shortest spanning tree) based methods, where the clustering or grouping of image pixels are performed on the minimal spanning tree. The connection of graph vertices satisfies the minimal sum on the defined edge weights, and the partition of a graph is achieved by removing edges to form different sub-graphs. An example of such method is Felzenszwalb and Huttenlocher [Felzenszwalb and Huttenlocher, 2004].

- Graph cut with cost functions. Graph cut is a natural description of image segmentation. Using different cut criteria, the global functions for partitioning the graph will be different. Usually, by optimizing these functions, we can get

the desirable segmentation. Normalized cut [Shi and Malik, 2000] and ratio cut [Wang and Siskind, 2003] are such methods.

- Graph cut on Markov random field models: the goal is to combine the high level interactive information with the regularization of the smoothness in the graph cut function. Under the MAP-MRF framework, the optimization of the function is obtained by the classical min-cut/ max-flow algorithms [Boykov and Funka-Lea, 2006] or its nearly optimal variants, such as multi-label graph cut [Boykov et al., 2001] and interactive graph cut [Rother et al., 2004].

- Shortest path methods, where the object boundary is defined on a set of shortest paths between pairs of graph vertices. That is to say, the problem of finding the best boundary segment is converted into finding the minimum cost path between two vertices. In a weighted graph, the shortest path will connect the two vertices with the minimized sum of edge weights, and the path can be computed for instance with Dijkstra's algorithm [Dijkstra, 1959b]. Shortest path methods require user interactions to guide the segmentation. Therefore, the process is more flexible and can provide friendly feedback.

Graph based approach is gaining popularity primarily due to its ability in reflecting global image properties. It explicitly organizes the image elements into mathematically sound structures, and makes the formulation of the problem more flexible and the computation more efficient, which might require no discretization by virtue of purely combinatorial operators and thus incur no discretization errors.The segmentation problem is solved in a spatially discrete space by the efficient tools from graph theory.

1.4 Multi-target Tracking

1.4.1 Definition

The task of multi-target tracking, also called multiple object tracking, is to follow many moving objects in a dynamic scene. Given a video sequence, multi-target tracking aims to precisely recovering the trajectory of every single, freely moving target from the video. The topic of object tracking is one of the most fundamental tasks in applications of video motion processing, analysis, and data mining, such as human-computer interaction, visual surveillance, and virtual reality. Common application scenarios involve traffic video surveillance for security, sport players tracking to study their motion patterns and parameters during games, cell tracking for research in microbiology, see Fig. 1.5.

1.4.2 The Challenges of Multi-target Tracking

Tracking several objects over time does not only require a correct segmentation of each object at any time, but also a correct identification over time. It is highly challenging and designing an algorithm is generally ad-hoc. It is challenging for various reasons. First of all, visual data is often ambiguous. For example, the

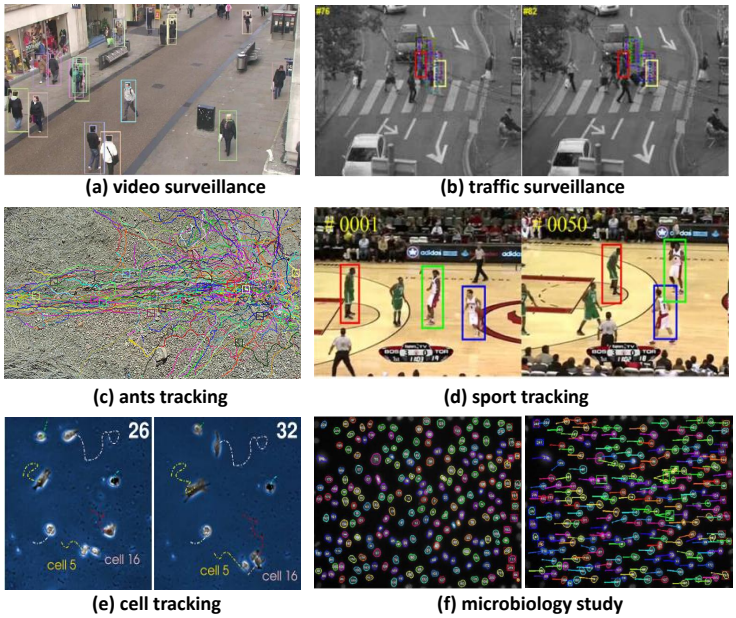


Figure 1.5: Illustration of application examples for multi-target tracking.

objects to be tracked can be missed due to low contrast, noise and occlusion. Second, when multiple objects are close to each other, correspondence ambiguities need to be solved, which leads to more complex problems at every time step. In addition, in realistic applications, the method should take physical constraints into account. This is not only important at the level of individual targets but also regarding interactions between them, which adds to the complexity of the problem.

1.4.3 Brief Literature Review

Object tracking is a well-known problem and a very active research topic in computer vision. Up to now, a substantial body of literature has been proposed dealing with the problem of tracking multiple targets. It is therefore not likely to present a complete review of all works on this topic within the limits of this dissertation. We briefly introduce several very important algorithms in multiple target tracking as well as their extension versions. One typical way to classify multi-target tracking approaches is to distinguish between online methods (e.g. Kalman filters) where the state is estimated at each time step and offline (graph based methods) techniques, which consider an entire temporal sequence at once. In particular, we discuss more on the graph based method proposed for the aim of multi-target tracking in section 1.4.3.2.

1.4.3.1 Online Methods

One of the most popular approaches to this problem is the Kalman filter proposed by Kalman [Kalman, 1960]. In its early version, given a sequence of measurements, the Kalman filter estimates the optimal state of a system in a least squares sense and

under certain assumptions, e.g. linear dependency. To relax the linearity constraint, extensions such as the extended Kalman filter or the unscented Kalman filter [Julier and Uhlmann, 1997] have been proposed. The demand for non-Gaussian models later led to the development of stochastic recursive state estimation techniques called particle filters [Gordon et al., 1993] [Isard and Blake, 1998][Doucet et al., 2000]. Particle filters is used for visual tracking introduced by Isard and Blake to include the posterior probability distribution, e.g. additional density propagation of shape. Later, more complex constraints propagation have been proposed in [Vermaak et al., 2003] and [Okuma et al., 2004]. Note that the above filtering techniques are intrinsically designed to handle single target. But it is easily and more often employed in multiple-target tracking by solving the data association problem iteratively.

Another more frequently used method is the data association. We present a coarse review and we refer readers to [Cox, 1993] and [Blackrnan and House, 1999] for a more thorough discussion. One of the simplest data association techniques is global nearest neighbors [Blackrnan and House, 1999], which keeps the association with the highest probability for all targets and removes all the other ones. An obvious drawback of global nearest neighbors is that all previous information is discarded as soon as the current time step has been processed. To amend this, Reid proposed a more complex data association method, referred as the multiple hypothesis tracker [Reid, 1979]. The idea is to keep all possible association events from the past observations in memory and to choose the best one at each time step, which is an exhaustive search algorithm. Another class of methods is called probabilistic data association originally in [Bar-Shalom and Jaffer, 1972]. The idea is that probabilities for various sources of origin for each detection are accumulated and propagated through time.

1.4.3.2 Offline Methods

- **Markov Chain Monte Carlo Sampling:** Unlike particle filter, Markov Chain Monte Carlo (MCMC) is a more general sampling technique, which works by generating a sequence of variables from a specially crafted Markov Chain. MCMC was first used to solve data association problems by Pasula et al.[Pasula et al., 1999], who showed it to be effective for multi-camera traffic surveillance problems involving hundreds of vehicles. Khan et al. [Khan et al., 2005] introduced a Markov random field motion prior to model the pairwise interaction between targets. The exponential complexity is approached by Markov chain Monte Carlo (MCMC) sampling. Oh et al. [Oh et al., 2004] proposed a general framework to sample the data association hypothesis using a MCMC approach, and converges to the full Bayesian solution if given enough computational resources.

- **Graph-based Methods:** Generally, for graph-based tracking, observations are represented by vertices, and costs are assigned to edges to measure relationships between observations. Solution of the multi-object tracking problem is a combinatorial optimization problem of significant complexity and thus becomes a problem of considerable research interest. A common strategy starts

from a set of short, yet confident tracks, or tracklets, longer trajectories are built based on global information. For example, many early approximation methods proposed greedy bipartite data association on a frame-by-frame basis and can be solved exactly in polynomial time by Hungarian algorithm. [Padfield et al., 2011] explicitly modeled cell behaviors in a graph-theoretic framework as a flow network that can be solved efficiently using the minimum-cost flow algorithm and extended the standard minimum-cost flow algorithm to account for mitosis and merging events through a coupling operation on particular edges. More recent methods proposed to find globally optimal solutions across the entire sequence by creating network flow graphs. Zhang et al. [Zhang et al., 2008b] defined a graph and solved the data association problem by optimizing the cost flow network in order to find the globally optimal trajectories. Some methods are proposed to solve the data association from constructed graph with other algorithms instead of the min-cost/max flow. Yan et al. [Yan et al., 2008] formulated the association problem as an all-pairs shortest path problem in the graph, where each node is a tracklet, and the edge weight between two nodes is defined according to the compatibility of the two tracklets. Brendel et al. [Brendel et al., 2011] formulated the data association as a maximum weight independent set problem. Their algorithm solves for two-frame tracklets independently, and then links these into complete tracks by using a learned distance measure. Zamir et al. [Zamir et al., 2012] incorporated the whole temporal span of the sequence into the data association problem, and solved the data association problem for one object at a time with generalized minimum clique graphs rather than addressing all of them simultaneously. Butt and Collins [Butt and Collins, 2013b] estimated trajectories in a trellis graph, where each node in the network represents a candidate pair of matching observations between consecutive frames and is solved by relaxing these extra constraints using Lagrangian relaxation.

1.5 Contributions

The goal of this dissertation is to contribute to the state of the art in image segmentation and particle tracking using graph-based methods. To this end, four contributions are introduced:

- The first approach is devoted to developing a new graph construction algorithm which encodes adaptively local and global image structure information. Traditional static graphs (e.g. 4-connected graph [Li et al., 2012]) have advantage of good spatial continuity property, but usually fail to detect long range grouping cues. At the same time, a recently proposed graph, called sparse graph (e.g. ℓ_0-graph [Wang et al., 2013a]), can capture long range grouping cues and adaptive neighborhood structure, but fails to guarantee the spatial adjacency property of objects. Therefore, we propose a new graph construction that aims at exploiting the nice properties of both types of graphs. The proposed graph uses without supervision conventional static graph's geometrical adjacency together with sparse graph's properties. As a result, the proposed graph yields

very competitive qualitative and quantitative segmentation results compared to various methods on standard benchmark.

- Our second approach builds on the work [Buades et al., 2005] and [Ji et al., 2012], and develop a new feature descriptor, referred as weighted color patch, which represents a pixel with a weighted patch. To alleviate the over-smooth effect caused by considering each member equally in the patch, we assign different weight to each pixel in the patch. The weighted color patch have two main advantages: i) it can smooth local regions by averaging color information and ii) it can capture texture information by considering context neighboring cue. As a result, the proposed weighted color patch has been proved powerful and discriminating, evaluated on the Prague color texture image benchmark and the Berkeley image segmentation database.

- Our third contribution is to develop a new mid-level feature and apply it in a graph-theoretical framework. We build the dictionary from informative patches centered at interest points detected without any supervision, and each mid-level feature is the sparse coding in the dictionary of the low level feature associated with a superpixel. Consequently, the new mid-level features carry not only the same information as the initial low-level features, but also additional contextual cue. Compared with related works and other benchmark algorithms on the MSRC dataset, the key contribution of this paper is that our new mid-level feature is able to describe better the superpixels.

- Our fourth contribution is to propose a robust framework that jointly segment and track accurately massive colloids in long-term videos. Note that all trajectories are recovered simultaneously from a high-quality trellis graph which builds on all frames. The first contribution is the strategy of two-stage graph construction, first coarse graph then refined graph with additional constraints, that guarantees the final tracking result. The second contribution of this work is that a modified min-cost/max flow algorithm enables recovering simultaneously meaningful trajectories in the graph. Finally, we evaluate the proposed framework on real videos, which have been annotated by 9 different graduate students.

- Our final contribution also lies on the discussion on various graph construction algorithms, extensive experiments to compare graphs and features on public available data set, in order to shed light on how to construct discriminative graphs with suitable features and their combinations. Note that a high-quality graph construction can easily be used in other graphical models for solving other problems involving a graph. Moreover, the tracking algorithm is not limited for tracking massive colloids for physics research, it is feasible for other multi-target tracking problems. Finally, we provided a real benchmark by annotating 9 different videos, which can be used to evaluate and compare new proposed algorithms dealing with multi-target tracking task.

1.6 Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 provides the basic definitions and results in graph theory and presents a systematic survey of graph construction and graph partitioning relevant to this thesis. Chapter 3 reviews the previous work on image segmentation, since the algorithms presented in this dissertation are mainly on this subject.

The main technical contribution presented in the thesis is divided into three parts. Part 1 (Chapter 4) deals with the essential issue of graph construction. We propose a novel graph construction algorithm by encoding adaptively the local and global image structure information. The technical part of this chapter has been published in [Wang et al., 2013a] and is described with details in a submitted paper. Then we also present a thorough comparison with various graph constructions on standard image segmentation dataset. Part II (Chapter 5 and 6) presents two methods to employ perceptually meaningful image properties for image segmentation. In chapter 5, a new feature, called weighted color patch, is proposed to compute the weight of edges in graph construction. The proposed method intends to incorporate both color and neighborhood information by representing pixels with color patches, and is proved powerful on the Prague color texture image benchmark and the Berkeley image segmentation database. This chapter is based on [Wang et al., 2013c]. In Chapter 6, we propose a graph-based unsupervised segmentation approach that combines superpixels, sparse representation, and a new mid-level feature to describe superpixels. These mid-level features not only carry the same information as the initial low-level features, but also carry additional contextual cue. This proposed mid-level feature framework is validated on the MSRC dataset and published in [Wang et al., 2013b]. Part III (Chapter 7) deals with the multi-target tracking problems in active colloids systems. We propose an efficient framework to jointly detect and track each colloid in a long-term video. The proposed modified min-cost/max flow algorithm enables to find all colloids' paths simultaneously and our framework is proved powerful on annotated videos.

In Chapter 8, we summarize the contributions that were developed in this thesis and presents a discussion on the relevance and role of each individual methods as well as an outlook for possible future research direction.

Finally, during the period of the thesis, we also cooperated with Yuxing Tang (PhD candidate) on a different topic, namely object recognition and detection. We present the proposed method in Appendix A.

Chapter 2

Graph based Methods

Contents

2.1 Graph

In graph theory, a graph is a representation of a set of data where some pairs of data elements are connected by links [Wil, Gallier, 2013, Ulrike, 2007]. The data elements are called *vertices, or nodes* and the links are called *edges, or arcs*. In most contexts, a graph is represented as $G = (V, E)$, where V is a set of vertices and E is a set of edges between vertices.

2.1.1 Definitions

Definition 1 *A **directed graph** is a pair $G = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is a set of nodes, and $E \subseteq V \times V$ is a set of ordered pairs of distinct nodes (that is, pairs $(u, v) \in V \times V$ with $u \neq v$), called edges.*

Definition 2 *An**undirected graph** is a pair $G = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is a set of nodes or vertices, and E is a set of two-element subsets of V (that is, subsets $\{u, v\}$, with $u, v \in V$ and $u \neq v$), called edges.*

Since an edge is a set $\{u, v\}$ with $u \neq v$, self-loops are not allowed. Also, for every set of nodes $\{u, v\}$, there is at most one edge between u and v. As in the case of directed graphs, such graphs are sometimes called **simple graphs**.

Definition 3 *Given a directed graph $G = (V, E)$ with $V = \{v_1, \cdots, v_n\}$ and $E = \{e_1, \cdots, e_m\}$, the **incidence matrix** $\tilde{D}(G)$ of G is the $n \times n$ matrix whose entries $\tilde{d}_{ij}$ are given by:*

$$\tilde{d}_{ij} = \begin{cases} -1 & \text{if } e_j = (v_i, v_k) \text{ for some } k \\ +1 & \text{if } e_j = (v_k, v_i) \text{ for some } k \\ 0 & \text{otherwise} \end{cases}$$

If G is undirected then the entries of the incident matrix are given by

$$\tilde{d}_{ij} = \begin{cases} 1 & \text{if } e_j = \{v_i, v_k\} \text{ for some } k \\ 0 & \text{otherwise} \end{cases}$$

Definition 4 *Given a directed or undirected graph $G = (V, E)$, with $V = \{v_1, \ldots, v_n\}$, the **adjacency matrix** $A(G)$ of G is the symmetric $n \times n$ matrix (a_{ij}) such that*

1. *If G is directed, then*

$$a_{ij} = \begin{cases} 1 & \text{if there is some edge } (v_i, v_j) \in E \text{ or some edge } (v_j, v_i) \in E \\ 0 & \text{otherwise} \end{cases}$$

2. *Else if G is undirected, then*

$$a_{ij} = \begin{cases} 1 & \text{if there is some edge } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Definition 5 *A **weighted graph** is a pair $G = (V, W)$, where $V = \{v_1, \ldots, v_n\}$ is a set of nodes, and W is a symmetric matrix called the **weight matrix**, such that $w_{ij} \geq 0 \ \forall i, j \in 1, \ldots, n$, and $w_{ii} = 0$ for $i = 1, \ldots, n$.*

Since $w_{ii} = 0$, these graphs have no self-loops. We can think of the matrix W as a generalized adjacency matrix. The case where $w_{ij} \in [0\ 1]$ is equivalent to the notion of a graph as in Definition 2.

Definition 6 *Given a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$, the **degree matrix** D for G is a $n \times n$ diagonal matrix defined as:*

$$D = \begin{pmatrix} d(v1) & & & \\ & d(v_2) & & \\ & & \ddots & \\ & & & d(v_n) \end{pmatrix} \tag{2.1}$$

where the degree $d(v_i)$ of v_i is the sum of the weights of the edges adjacent to the node v_i, i.e.:

$$d(v_i) = \sum_{j=1}^{n} w_{ij}. \tag{2.2}$$

Note that in the above sum, only nodes v_j such that there is an edge $\{v_i, v_j\}$ have a nonzero contribution. Such nodes are said to be adjacent to v_i.

Given these preliminaries, a series of basic definitions are listed as follows:

1. Given any subset of nodes $A \subseteq V$, the volume $vol(A)$ of A is the sum of the weights of all edges adjacent to nodes in A, defined as:

$$vol(A) = \sum_{v_i \in A} d(v_i) = \sum_{v_i \in A} \sum_{j=1}^{n} w_{ij} \tag{2.3}$$

2. A graph $G = (V', E')$ is called a **subgraph** of $G = (V, E)$ if $V' \subset V$, and $E' = \{e_{ij} \in E | v_i \in V' \text{ and } v_j \in V'\}$

3. A graph is called **bipartite graph** if V can be partitioned into two subsets $V_1 \subset V$ and $V_2 \subset V$, where $V_1 \cap V_2 = \phi$ and $V_1 \cup V_2 = V$, such that $E \subseteq V_1 \times V_2$.

2.1.2 Graph Laplacians

In this section we recall various definitions of graph Laplacians and point out their most important properties. In the following we always assume that G is an undirected, weighted graph with weight matrix W, where $w_{ij} = w_{ji} \geq 0$. Eigenvalues will always be ordered increasingly, respecting multiplicities. By "the first k eigenvectors" we refer to the eigenvectors corresponding to the k smallest eigenvalues

Definition 7 *Given a graph $G = (V, E)$ and its degree matrix D and weighted matrix W, the **unnormalized graph Laplacian** matrix is defined as:*

$$L = D - W. \tag{2.4}$$

Remark 1 *The matrix L satisfies the following properties:*

1. *For every vector $f \in R^n$ we have*

$$f'Lf = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \tag{2.5}$$

2. *L is symmetric and positive semi-definite.*

3. *The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{I}$.*

4. *L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.*

Remark 2 *Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{I}A_1, \ldots, \mathbb{I}A_k$ of those components.*

Definition 8 *Given graph $G = (V, E)$, the **(normalized) graph Laplacians** L_{sym} and L_{rw} of G are defined by:*

$$\begin{aligned} L_{sym} &= D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \\ L_{rw} &= D^{-1} L = I - D^{-1} W. \end{aligned} \tag{2.6}$$

Remark 3 *The normalized Laplacians satisfy the following properties:*

1. *For every vector $f \in R^n$ we have*

$$f' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}})^2. \tag{2.7}$$

2. *λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D^{1/2} u$.*

3. *λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solve the generalized eigen-problem $Lu = \lambda Du$.*

4. *0 is an eigenvalue of L_{rw} with the constant one vector $\mathbb{I}$ as eigenvector. 0 is an eigenvalue of L_{sym} with eigenvector $D^{1/2}\mathbb{I}$.*

5. *L_{sym} and L_{rw} are positive semi-definite and have n non-negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.*

Remark 4 *Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of both L_{rw} and L_{sym} equals the number of connected components $A_1, \ldots, A_k$ in the graph. For L_{rw}, the eigenspace of 0 is spanned by the indicator vectors $\mathbb{I}A_i$ of those components. For L_{sym} , the eigenspace of 0 is spanned by the vectors $D^{1/2}\mathbb{I}A_i$.*

2.2 Affinity Graph

In graph-oriented methods, many tasks such as clustering, dimensionality reduction, or semi-supervised learning, affinity graph (also called similarity graph in context of spectral clustering) plays an essentially important role for the final result. In practice, data is generally not given in form of a graph, but in terms of affinity/similarity values between points [Ulrike, 2007]. For example, in the case of image segmentation, first an affinity graph is constructed to encode the relationship of each data and its neighborhood with certain weight measuring by the weights in the range [0 1], and then a graph partitioning algorithm is applied to this graph.

How to construct a *good* graph describing the relationships between samples has been widely studied in recent years, and it is still an open problem. The quality of graphs is very sensible to the topological structure, the choice of weighting functions and the related parameters:

1. **Topology:** The graph topology depicts the relationship between data points. Whether the graph can accurately determine the local neighborhood of each data point is crucial to infer global structure (meaningful segments) from local information, particularly when the data contain a lot of errors. Formally, given a graph $G = (V, E)$, identified by its vertices $V = \{v_1, \cdots, v_n\}$ and its edges $E = \{v_i, v_j\},\ i \leq n, j \leq n, i \neq j$, the topology is defined by identifying every edge $\{v_i, v_j\}$ within the unit interval $[0, 1]$ and gluing them together at coincident vertices.

2. **Affinity:** To calculate the affinity for an edge connecting two vertices, choosing an appropriate feature and distance function (also called weighted function) are the two key factors to be considered. The basic rule is that the weights should be large for pixels that should be associated and small otherwise.

Studies on the semi-supervised learning [Liu and Chang, 2009] and subspace clustering [Wright et al., 2010], as well as the task of image segmentation [Cour et al., 2005][Li et al., 2012], find that a desirable good graph can generate reasonable results. Basically, a *good* graph [Cheng et al., 2010] [Wright et al., 2010] should have the following properties:

- **High discriminating power**. Pixels from the same object are expected to be assigned large affinities, in contrast with pixels from different objects. Therefore, features selection and their appropriate fusion are critical to capture and compensate the different image properties;

- **Sparsity**. Many works on graph partitioning state that meaningful results derive from a sparse graph [Shi and Malik, 2000] [Cour et al., 2005] because it conveys with low memory cost valuable semantic information of the original high-dimensional data [Wright et al., 2010]. Furthermore, due to storage limitations and the need for efficient solving of eigenvector problems, it is inevitable to build a sparse graph;

- **Adaptivity**. It happens frequently that the data (pixels/regions) is not evenly distributed. Conventional static graphs use fixed size and shape for the neighborhoods that are used for computing affinity weights, which will potentially generate erroneous associations between pixels. In contrast, it is reasonable to ask that different data points should have their corresponding adaptive neighborhood structure.

Up to now, there is a wealthy literature proposing or using various types of graphs for many tasks of machine learning. In this thesis, we classify them into two classes according to graph affinity computation:

1. **Pairwise affinity graph:** The weight is calculated based on pairwise distances (e.g. Euclidean distance) to model local relationships between data points and their neighborhood, that is to say, the method uses the distance between two points to measure their similarity. Typically, topology of neighborhood is determined with parameters. For example, each vertice of ε-ball graph chooses to connect with those points whose pairwise distances are smaller than ε [Ulrike, 2007].

2. **Datum-adaptive affinity graph:** The weight is computed based on reconstruction coefficients, i.e. the distance between any two data points is independent from the other points. The topology of neighborhood is determined depended on all data samples, and each data is supposed to connect to those points which can represent current data linearly. There are many algorithms, e.g. Locally Linear Manifold Clustering [Goh and Vidal, 2007], Sparse Subspace Clustering [Elhamifar and Vidal, 2009], ℓ^1-graph [Cheng et al., 2010][Wright et al., 2010], ℓ^2-graph [Peng et al., 2012], Low Rank Representation [Liu et al., 2010][Liu et al., 2013a].

2.2.1 Topology

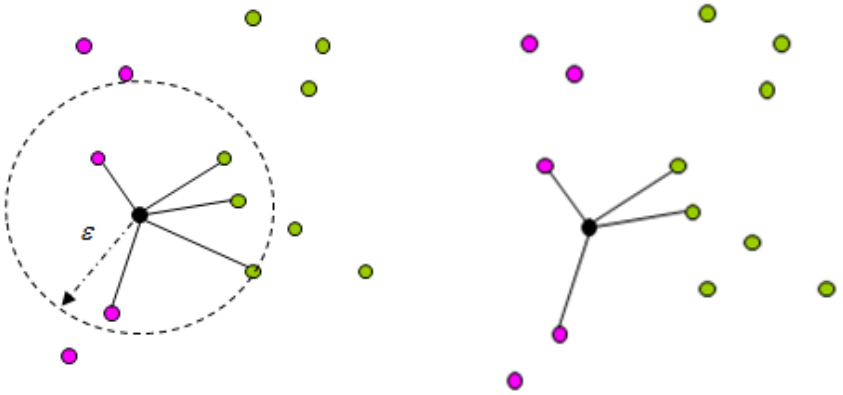
We discuss the graph topology based on the way how the algorithm decides the neighborhood structure, i.e. pairwise graph or datum-adaptive graph. The former, typically means kNN-graph and ε-graph, has been widely applied in different tasks, such as data clustering. Unlike the pairwise affinity graph, in which the edge weights characterize pairwise relations, the edge weights of sparse graph (ℓ^1-graph, ℓ^2-graph) and LRR-graph are determined in a group manner, and the weights related to a certain vertex characterize how the rest samples contribute to the sparse representation of this vertex. The second option has become increasingly popular, especially in high- dimensional data analysis.

2.2.1.1 Pairwise Graph

Most graph-cut approaches for image segmentation build a static graph which only models the local neighborhood relationships between the data points [Ulrike, 2007]. Classical methods (shown in Fig.2.1) for selecting connected vertices are:

- The ε-neighborhood graph (ε-graph), which connects all points whose pairwise distances are smaller than ε. The pairwise similarity is chosen almost constant, making the constructed graph unweighted. However, selecting a single ε for all nodes in the graph might not properly capture the neighborhood structure of the data points;

- The k-nearest neighbor graph (kNN-graph) connects every point to all points that are among its K-nearest neighbors, and the similarity is computed using pairwise distances. The fact that the kNN-graph's neighborhood size is fixed may lead to include noisy edges in the neighborhood of a data and a "good" number of nearest neighbors k may be different for different objects;



(a) ε-graph (b) kNN graph

Figure 2.1: Illustration of two classical pairwise graphs. The dark point is current data, and data in different color form different clusters.

- The fully connected graph connects all points with positive similarity with each other. This construction is only useful if the similarity function itself models local neighborhood relationship. In most cases, Gaussian similarity function $w(x_i, x_j) = e^{(-(x_i - x_j)^2)/(2\sigma^2))}$ is chosen to compute the similarity, where the parameter σ controls the width of the neighborhood. Obviously, a "good" σ would help in pulling intra-class objects together and in pushing interclass objects far away from each other. Therefore the parameter σ is critical in generating a reliable affinity matrix by controlling the neighborhood size and scaling pairwise similarities.

2.2.1.2 ℓ^1-Graph

Wright et al. [Wright et al., 2009] proposed to use sparse representation for face recognition. They demonstrated that the $\boldsymbol{\ell^1}$ linear reconstruction error minimization can naturally lead to a sparse representation for human facial images. Elhamifar and Vidal[Elhamifar and Vidal, 2009] firstly proposed to directly use the sparse representation of vectors lying on a union of subspaces to cluster the data into separate subspaces, which is called sparse subspace clustering. Wright et al. [Wright et al., 2010] extended the sparse representation based on $\boldsymbol{\ell^1}$-minimization to characterize relationships between the data samples, i.e. termed as $\boldsymbol{\ell^1}$-graph, in order to accomplish tasks such as image classification. Cheng et al. [Cheng et al., 2010] proposed a process to build the $\boldsymbol{\ell^1}$-graph and designed a series of new algorithms for various machine learning tasks, e.g. data clustering, subspace learning, and semi-supervised learning. They also concluded the advantages of $\boldsymbol{\ell^1}$-graph: first, $\boldsymbol{\ell^1}$-graph is robust owing to the overall contextual norm formulation and the explicit consideration of data noises. Second, the sparsity of the $\boldsymbol{\ell^1}$-graph is automatically determined instead of manually as in k-NN graph and ε-graph. Finally, the $\boldsymbol{\ell^1}$-graph is datum-adaptive. The number of neighbors selected by $\boldsymbol{\ell^1}$-graph is adaptive to each datum,

which is valuable for applications with unevenly distributed data.

Problem formulation: given a set of data samples $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\} \in \mathbb{R}^{m \times n}$, where n is the data number and m is the feature dimension. Denote the ℓ^1-graph by $G = (V, E, W)$, where V is the set of n vertices, each of which is identified with a sample in $\boldsymbol{X}$, and E is the set of edges connecting a pair of vertices, and $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ is the weighted matrix. The ℓ^1-graph is constructed in an unsupervised manner, with a goal of automatically determining the neighborhood structure as well as the corresponding connection weights for each datum. The process of building the graph is illustrated in Algorithm 1 [Wright et al., 2010].

Algorithm 1: $\boldsymbol{\ell^1}$-graph construction

1) **Input:**the data matrix $\mathbf{X}$.

2) **Sparse coding:** For each datum $\boldsymbol{x_i}$, solve the l^1-norm minimization problem:

$$\min_{\boldsymbol{\alpha^i}} \left\| \boldsymbol{\alpha^i} \right\|_1 \text{ subject to } \quad \boldsymbol{x_i} = \boldsymbol{D^i \alpha^i} \tag{2.8}$$

where $\boldsymbol{D^i} = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_{i-1}}, \boldsymbol{x_{i+1}}, \ldots, \boldsymbol{x_n}, \mathbf{I}] \in \mathbb{R}^{m \times (m+n-1)}$ and $\boldsymbol{\alpha} \in \mathbb{R}^{m+n-1}$.

3) **Weight matrix construction:** for each couple of samples $(\boldsymbol{x_i}, \boldsymbol{x_j})$, the weight is defined as:

$$w_{ij} = \begin{cases} \boldsymbol{\alpha}_j^i & \text{if } i > j \\ \boldsymbol{\alpha}_{j-1}^i & \text{if } i < j. \end{cases} \tag{2.9}$$

2.2.1.3 $\boldsymbol{\ell^2}$-Graph

Peng et al. [Peng et al., 2012] proposed a novel scheme for finding sparse similarity graphs by eliminating the effect of errors from the representation but from the dictionary by developing $\boldsymbol{l^2}$-graph algorithm to corroborate the effectiveness of the scheme. The $\boldsymbol{l^2}$-graph can reveal the latent structure of a data distribution, an ability that is important to a lot of applications. Formally, the $\boldsymbol{l^2}$-graph also calculates the sparse coefficients $\boldsymbol{\alpha^i}$ like other sparse family algorithms by solving :

$$\min_{\boldsymbol{\alpha^i}} \frac{1}{2} \left\| \boldsymbol{x_i} - \boldsymbol{D^i \alpha^i} \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha^i} \right\|_2^2, \tag{2.10}$$

where $\boldsymbol{D^i} = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_{i-1}}, \boldsymbol{x_{i+1}}, \ldots, \boldsymbol{x_n}] \in \mathbb{R}^{m \times n}$, $\boldsymbol{\alpha} \in \mathbb{R}^n$, and $\lambda \geq 0$ is a regularization parameter. For each $\boldsymbol{x_i}$, solving above optimization problem gives

$$\boldsymbol{\alpha^i} = \boldsymbol{P} \left[\boldsymbol{D}^T \boldsymbol{x^i} - \frac{\boldsymbol{e^i}^T \boldsymbol{P} \boldsymbol{D}^T \boldsymbol{x^i}}{\boldsymbol{e^i}^T \boldsymbol{P} \boldsymbol{e_i}} \boldsymbol{e_i} \right] \tag{2.11}$$

where $\boldsymbol{P} = (\boldsymbol{D}^T \boldsymbol{D} + \lambda \boldsymbol{I})^{-1}$, and the union of e_i $(i = 1, \ldots, n)$ is the standard orthogonal basis of $\mathbb{R}^n$, i.e., all entries in $\boldsymbol{e_i}$ are zero, except for the i-th entry, which is one. The process of building the graph is illustrated in Algorithm 2 [Peng et al., 2012]:

Algorithm 2: $\boldsymbol{l^2}$-graph construction

1) **Input:**the data matrix $\mathbf{X}$.

2) **Sparse coding:** For each datum $\boldsymbol{x_i}$, solve the l^2-norm minimization problem in Eq.2.10 via Eq.2.11, obtain the optimal solution $\boldsymbol{\alpha^i}$, and normalize $\boldsymbol{\alpha^i}$ to give a unit $\boldsymbol{l^2}$-norm

3) Eliminate the effects of errors by performing the k-NN or ε-ball method over $\boldsymbol{\alpha^i}$, e.g., $\hat{\boldsymbol{\alpha^i}} = \mathcal{H}_k(\boldsymbol{\alpha^i})$, where $\mathcal{H}_k(\boldsymbol{\alpha^i})$ retains the k largest coefficients of $\boldsymbol{\alpha^i}$ and sets the other entries to zero.

4) **Weight matrix construction:** Construct the weighted matrix $W = [w_{ij}]$ by connecting node i, denoted by $\boldsymbol{x_i}$, with node j, denoted by $\boldsymbol{x_j}$. Assign the connection weight:

$$w_{ij} = |\boldsymbol{\alpha}_j^i| + |\boldsymbol{\alpha}_i^j| \tag{2.12}$$

2.2.1.4 *LRR*-Graph

Like the task of sparse representation, i.e. to recover the subspace structures from the data containing errors, Liu et al. [Liu et al., 2010] proposed a novel method termed *low-rank representation*(LRR), which aims at finding the lowest-rank representation of all data jointly. Formally, let $X = [x_i, x_2, \cdots, x_n]$, the LRR is defined as:

$$\min_{Z,E} = \|Z\|_* + \lambda \|E\|_{2,1} \quad s.t. \quad X = XZ + E \tag{2.13}$$

where $||\cdot||$ denotes the nuclear norm, also known as the trace norm or Ky Fan norm (sum of the singular values), $||E||_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{n}([E]_{ij})^2}$ is the $\boldsymbol{l_{2,1}}$-norm for characterizing noise and the parameter $\lambda > 0$ is used to balance the effects of the two parts.

Due to the discrete nature of the rank function, the above problem was relaxed to a nuclear norm optimization problem, and it can be solved in polynomial time. The optimization problem Eq.2.13 is convex and can be solved with Augmented Lagrange Multiplier (ALM) [Lin et al., 2010], which minimizes the following augmented Lagrange function:

$$\begin{aligned} \mathcal{L} = &\|J\|_* + \lambda \|E\|_{2,1} + tr(Y_1^t(X - XZ - E)) \\ &+tr(Y_2^t(Z - J)) + \frac{\mu}{2}(||X - XZ - E||_F^2 + ||Z - J||_F^2). \end{aligned} \tag{2.14}$$

where $J = Z$.

Minimizing a rank constraint function can address the problems of $\boldsymbol{l^1}$-graph. Yet LRR constructs a dense graph and even a block-diagonal matrix, which is not very desirable for graph-based algorithms. To derive a sparse graph, Zhuang et al.[Zhuang et al., 2012] proposed a novel Non-Negative Low Rank and Sparse graph (NNLRS) which adds a non-negative and sparse constraint on the original LRR model. Global structure of samples is obtained by the low-rank constraint and the locally linear structure is captured by the sparse constraint. The sparsity of the obtained graph is improved and it gets a satisfactory result. The process of building the graph

[Zhuang et al., 2012] is illustrated in Algorithm 3:

Algorithm 3: Nonnegative low rank and sparse graph construction (NNLRS)

1) **Input:** the data matrix X, regularized parameters β and λ, threshold θ.

2) **Low Rank Representation:**

1. Normalize all the samples $\hat{\boldsymbol{x_i}} = \boldsymbol{x_i}/||\boldsymbol{x_i}||_2$ to obtain $\hat{X} = \{\hat{\boldsymbol{x_1}}, \hat{\boldsymbol{x_2}}, \ldots, \hat{\boldsymbol{x_n}}\}$.

2. Solve the following problem:

$$\begin{aligned} \min_{Z,E} &= \|Z\|_* + \beta \|Z\|_1 + \lambda \|E\|_{2,1} \\ s.t. \quad & \hat{X} = \hat{X}Z + E, \quad Z \geq 0 \end{aligned} \tag{2.15}$$

   and obtain the optimal solution (Z^*, E^*).

3. Normalize all column vectors of Z^* by $z_i^* = z_i^*/||z_i^*||_2$, and make small values under given threshold θ zeros by:

$$\hat{z}_{ij}^* = \begin{cases} \hat{z}_{ij}^* & \text{if } \hat{z}_{ij}^* \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

3) **Weight matrix construction:** Construct the weighted matrix $W = [w_{ij}]$ by:

$$W = (\hat{Z} + (\hat{Z})^T)/2. \tag{2.16}$$

2.2.2 Affinity

From visual inspection, the affinity matrix contains information about the correct segmentation. Ren and Malik remark that "pixels are not natural entities; they are merely a consequence of the discrete representation of images" captures some of problems of pixel-based representation [Ren and Malik, 2003].

To overcome these ambiguities, it is necessary to incorporate longer range information

2.2.2.1 Feature Descriptor

In many machine learning task, ranging from data clustering to image classification, feature descriptors have been extensively studied and applied. There is a wealthy literature on designing a powerful feature descriptor which directly leads to large margins on the performance, such as the success of applying the local binary pattern (LBP) for texture image classification [Ojala et al., 2002] and the scale invariant feature transforms (SIFT) [Lowe, 1999] for object recognition and segmentation. In this thesis, we are not focused on developing feature descriptor nor list all existing features, instead, we mainly review those popular features used for image segmentation. In particular, graph based image segmentation, which involves affinity graph

construction. There are hundreds of thousands of feature descriptors in the literature, and there are many good surveys [Tuytelaars and Mikolajczyk, 2008] [Szeliski, 2010] on this topic. In general, based on the way of feature extraction to form feature descriptors for various applications, features can be categorized as local features and global features.

- **Local features:** A local feature is an image pattern which differs from its immediate neighborhood. It is usually associated with a change of an image property or several properties simultaneously, although it is not necessarily localized exactly on this change. Local features can be points, but also edgels or small image patches. Typically, some measurements are taken from a region centered on a local feature and converted into descriptors.
- **Global features:** A global image feature describes an image as a whole, which enables to generalize an entire object with a single vector, such as color histograms (CH).

(a) Camouflage animal (b) Bears

Figure 2.2: What kind of feature descriptor should we use for constructing the affinities.

In the context of graph construction, there are many literature concerning the topic: what is a good feature descriptor to measure the affinity between two data samples, which imposes great influence on the final result. For example, Fig.2.2 present two cases of using different features to measure the affinities, where Fig.2.2(a) prefers the texture feature descriptor, whilst Fig.2.2(b) can use color or contour information.

In graph-oriented image segmentation algorithms, there are generally three ways to use the feature to build the affinity graph:

1. **Pixel.** In this case, each pixel is treated as a vertice in the graph.

   (i) **Color.** Choosing the right color space is also a hot topic in the image segmentation. Although RGB is universal model for video and image display, it is not good for color scene segmentation because of the high correlation among the R,G,B components [Cheng et al., 2001]. As we assume that two pixels belong to the same region based on the color homogeneity, however,

in real natural images, the objects' surface exhibit variance in the highlights, shadows, shadings or texture. Thus, many researchers propose to choose a right color space for ad hoc applications. Shi and Malik [Shi and Malik, 2000] suggested to compute the affinity with HSV value for color image. For example, [Uijlings et al., 2013] proposed using multiple color spaces in order to capture all possible segmentation results. To compare different color spaces, Table 2.2 lists color spaces which can be used to compute the affinity.

(ii) **Contour.** The cue of contour has highly discriminative power to measure the affinity. Most work [Malik et al., 2001][Fowlkes et al., 2003] [Yu and Shi, 2003] used the "intervening contour" [Leung and Malik, 1998] as gradient-based feature, illustrated in Fig.2.3. In more detail, given a pair of pixels, consider the straight-line path connecting them in the image plane. If the pixels lie in different segments, then photometric discontinuity will be found somewhere along the line, otherwise the affinity between the pixels should be large. To use the intervening contour cue, extracting edge information is a precedent yet important step. For example, Canny detector [Canny, 1986] can be applied to detect the step changes in brightness. Most work suggest that the oriented energy approach (also known as the "quadrature energy" at angle 0°) [Knutsson and Granlund, 1983] can be used to detect and localize these changes in a combination of steps, peak and roof profile. Briefly, it is defined as:

$$OE_{0^\circ} = (I * f_1)^2 + (I * f_2)^2 \tag{2.17}$$

where $*$ is the convolution operator. f_1 and f_2 are the quadrature pairs of filters, differ in their spatial phases. The odd-phase filters f_1 are essentially the first-order derivatives, whereas the even-phase filters are the second-order derivatives, both smoothed with Gaussians. OE_{0° has maximum response for horizontal contours. Rotated copies of the two filter kernels are able to pick up composite edge contrast at various orientations.

In particular, it is worth to mention that the joint work of Arbelaez et al. have extensively investigated different features mentioned above and proposed the mPb in their paper [Arbelaez et al., 2011], taking account of combining color, texture etc., which is the state-of-the-art contour detector.

The proposal of multiscale segmentation stems from addressing the segmentation difficulties: (i) camouflage the object by making its boundary edges faint, and (ii) increase clutter by making background edges highly contrasting, particularly those in textured regions. Barbu and Zhu [Barbu and Zhu, 2003] explicitly controls the Markov chain transitions in the space of graph partitions by splitting, merging and re-grouping segmentation graph nodes. Yu [Yu, 2004] constructs a unified graph encoding edge cues at different image scales, and optimizes the average Ncut cost across all graph levels. Cour et al. [Cour et al., 2005] solved this limitation by computing sparse affinity matrices at multiple scales, setting up cross-scale constraints, and deriving a new eigen problem for this constrained multiscale cut.

2. **Patch.** In this case, a patch centered at a pixel is extracted and is treated as a

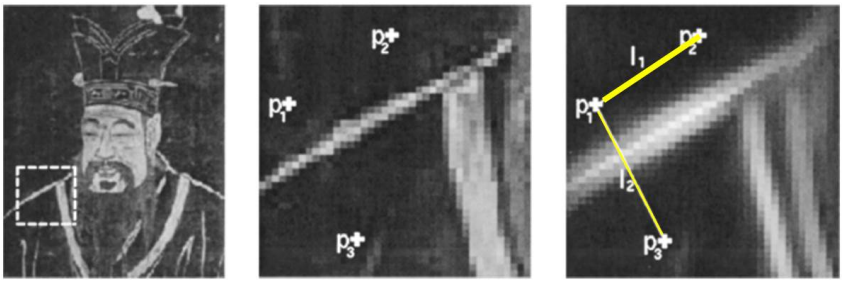


Figure 2.3: Left: the original image. Middle: part of the image marked by the box. The intensity values at pixels p_1, p_2 and p_3 are similar. However, there is a contour in the middle, which suggests that p_1 and p_2 belong to one group while p_3 belongs to another. Just comparing intensity values at these three locations will mistakenly suggest that they belong to the same group. Right: orientation energy. Somewhere along l_2, the orientation energy is strong which correctly proposes that p_1 and p_3 belong to two different partitions, while orientation energy along l_1 is weak throughout, which will support the hypothesis that p_1 and p_2 belong to the same group [Malik et al., 2001]

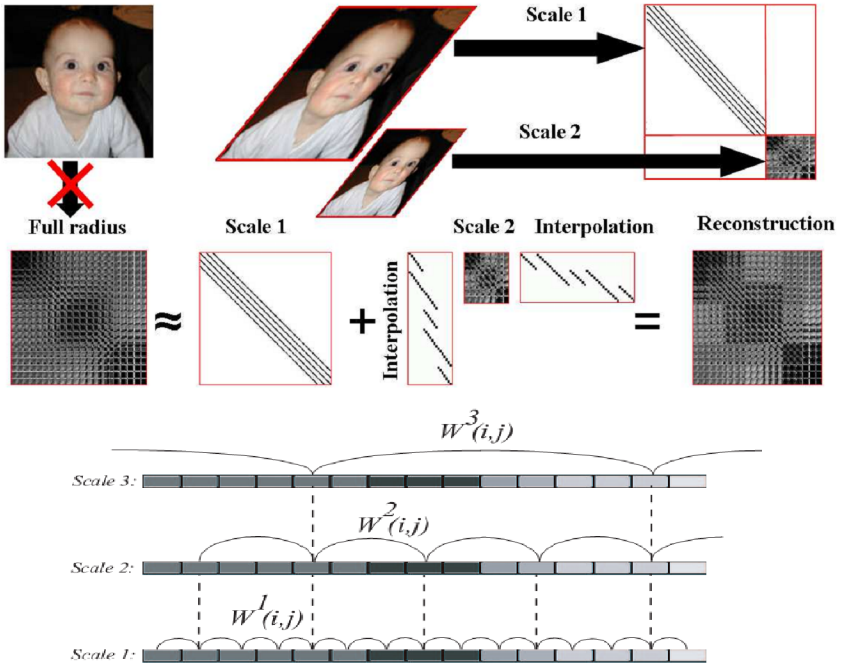


Figure 2.4: Multiscale graph compression

vertex. Since the features are extracted over a local window, the patch based affinity graph incorporates more local appearance in brightness, color and texture. According to [Fowlkes et al., 2003] the color cue is best captured using patches than pixels. [Wang et al., 2013b] proposed to use a bunch of patches centered on the interest points to represent a superpixel with color histogram.

For texture, although patches are not the only useful way to capture texture information, many works choose to study the texture using patches. [Malik et al., 2001] proposed to use windowed *texton* histogram to compute pairwise similarity. Fig.2.5 shows the process of generating the textons. First the input image I is convolved with a bank of filters f_i (shown in the Fig.2.5 (a)), which is clustered into K clusters with K-means (shown in the Fig.2.5 (b)). Each histogram has K bins, one for each texton channel. The value of the kth histogram bin for a pixel i is found by counting how many pixels in texton channel k fall inside the window or patch $P(i)$ (shown in the Fig.2.5 (c)). Thus the histogram represents texton frequencies in a local neighborhood. In

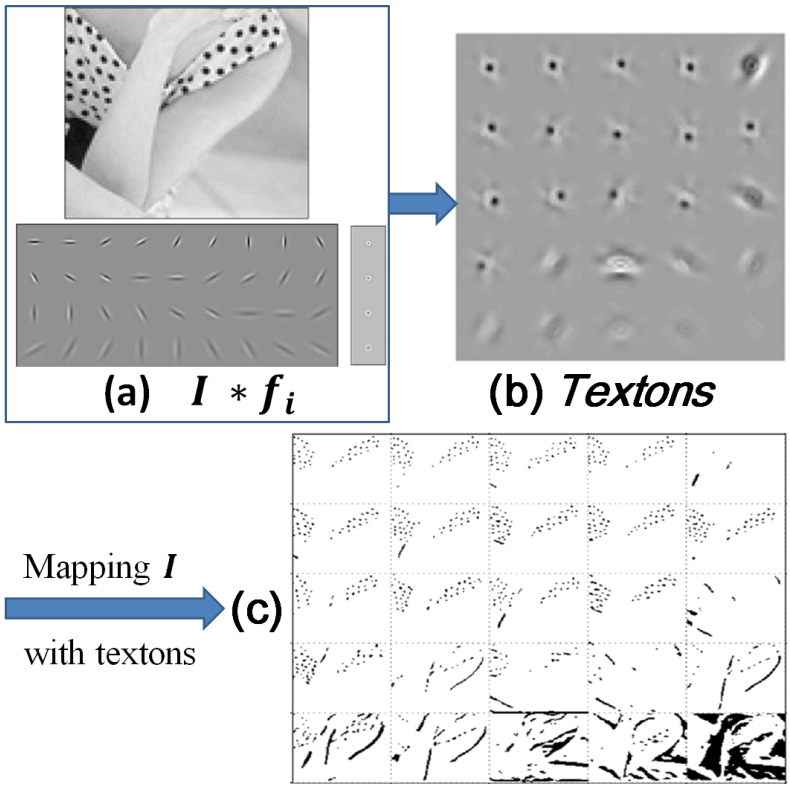


Figure 2.5: (a) Polka-dot image is convolved with a bank of filters. (b) Textons found via K-means with $K = 25$, sorted in decreasing order by norm. (c) Mapping of pixels to the texton channels. The dominant structures captured by the textons are translated versions of the dark spots. [Malik et al., 2001].

[Brunner12 et al., 2010], they extracted a set of features, such as intensity, texture and shape features.

3. **Superpixels.** In recent years, for both low-level and high-level tasks in computer vision, it has been a major trend to use the superpixel, which is a set of homogeneous pixels in certain feature space, as a basic data sample. For example, low-level segmentation [Wang et al., 2008b] [Yang et al., 2008],objectness measurement [Alexe et al., 2012][Uijlings et al., 2013], object detection [Shu

et al., 2013], semantic segmentation [Carreira et al., 2012a]. As pointed in [Yu et al., 2012], operation based on regions or superpixels allows one to investigate and design features much more versatile and powerful. Inspired from successful applications of bag-of-words in generic object classes, they constructed a histogram for each superpixel to quantitatively indicate the proportion of contribution from a specific texton. There are several appealing advantages [Alexe et al., 2010] [Li et al., 2012]:

- They provide additional structure, i.e. the set of possible segmentations is reduced to those aligning well with image boundaries;
- They reduce the computational complexity of segmentation;
- They enforce local smoothness since superpixels generally occupy consecutive image area in which pixels are likely to be grouped together;
- Large elongated superpixels incorporate long-range grouping cues, which has shown to improve segmentation substantially;
- Superpixels generated by different methods with varying parameters can capture diverse and multi-scale visual contents of a natural image.

Basically, in low-level unsupervised grouping algorithms, single-scale superpixel is used to build the similarity graph, which is furthered partitioned. Later, many works take advantage of multi-scale superpixels to improve the performance. Here, we give more details on two frameworks both of which are state-of-the-art. Kim et al. [Kim et al., 2010a] constructed a multi-layer graph with pixels and superpixels, generated by the mean shift (MS) algorithm with three various parameters, as nodes.

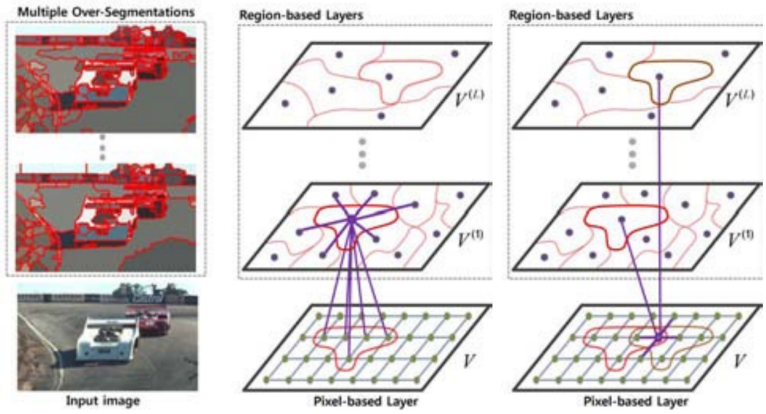


Figure 2.6: Multi-layer graph model [Kim et al., 2010b]. In (a), the graph nodes V^* consist of pixels V and regions $V^{(l)}{}_{l=1,\ldots,L}$, generated by varying the parameters of the mean shift algorithm [Comaniciu and Meer, 2002]. An undirected edge E^* represents the relation between a pair of nodes. (b) and (c) show the examples of edges (violet lines) connected to one region and to one pixel, respectively.

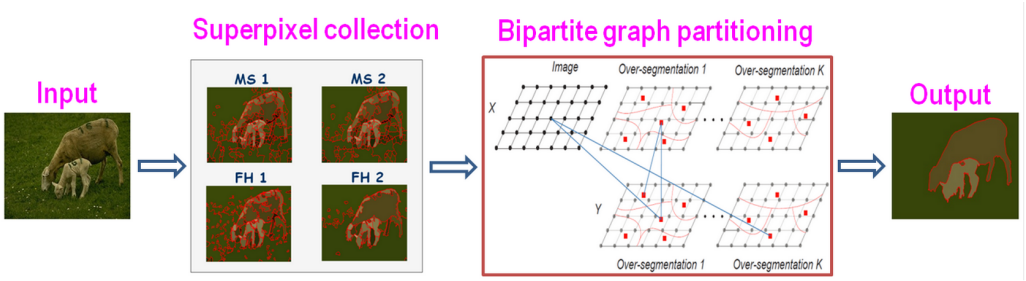


Figure 2.7: The proposed bipartite graph model with K over-segmentations of an image. A black dot denotes a pixel while a red square denotes a superpixel [Li et al., 2012].

$$w_{ij} = \begin{cases} e^{-\theta_g \|g_i - g_j\|} & \text{if } i, j \in V \\ e^{-\theta_g \|\bar{g}_i - \bar{g}_j\|} & \text{if } i, j \in V^{(l)} \\ \gamma & \text{if } i \in V,\ j \in V^{(l)} \end{cases} \tag{2.18}$$

where θ_g is a constant that controls the strength of the weight. g_i denotes the mean color of inner pixels of the region i. $V^(l)$ denotes oversegmentaiton in scale l, and the V denotes the set of pixels in original image domain as shown in Figure 2.7

Another framework is proposed in [Li et al., 2012], which constructs the affinity graph and concatenates 5 to 6 scale of superpixels generated by MS and Felzenszwalb-Huttenlocher (FH) into an unified matrix diagonally, Figure 2.7 shows the overview of the method.

Arose from the application of superpixels, deriving good-quality superpixel has become an increasing attractive topic. Principally, any kind of segmentation method which can generate regions, can be candidate of the method for superpixels. We list the methods frequently used for generating superpixels in Table 2.1, and Neubert and Protzel [Neubert and Protzel, 2012] present a survey for comparing different methods quantitatively.

Table 2.1: Superpixel generation methods

Method	Code
Mean Shift[Comaniciu and Meer, 2002]	http://www.vlfeat.org/
FH [Felzenszwalb and Huttenlocher, 2004]	http://cs.brown.edu/~pff/segment/
Superpixel with normalized cut [Ren and Malik, 2003]	http://www.cs.sfu.ca/~mori/research/superpixels/
Quick Shift[Vedaldi and Soatto, 2008]	http://www.vlfeat.org/
Watersheds[Couprie et al., 2009]	http://www.esiee.fr/~coupriec/code.html
Turbopixels[Levinshtein et al., 2009]	http://www.cs.toronto.edu/~babalex/research.html
ERS[Liu et al., 2011]	https://sites.google.com/site/seanmingyuliu/home/research_segmentation
SLIC [Achanta et al., 2012]	http://www.vlfeat.org/

4. **Combined features.** Up to now, there is no single feature that can describe all types of object properties. Thus it is essential to understand the advantages and disadvantages of different feature descriptors, and extend them or combine them for specific application. Shi and Malik [Shi and Malik, 2000] computed

the affinity of node i and j as the product of a feature similarity term and spatial proximity term:

$$w_{ij} = e^{\frac{-\|\mathbf{F}(i)-\mathbf{F}(j)\|_2^2}{\sigma_I}} * \begin{cases} e^{\frac{-\|\mathbf{X}(i)-\mathbf{X}(j)\|_2^2}{\sigma_X}} & \text{if } \|\mathbf{X}(i)-\mathbf{X}(j)\|_2 < r \\ 0 & \text{otherwise.} \end{cases} \tag{2.19}$$

where $F(i)$ can be any features like intensity, color and $X(i)$ is the position of node i. σ_I and σ_X are the parameters to control the speed of decay. Note that the weight $w_{ij} = 0$ for any pair of nodes that are more than r pixels apart.

Malik et al.[Malik et al., 2001] incorporated the intervening contour cue W_{ij}^{IC} and texture cue W_{ij}^{TX} using the idea which is if either of the cues suggests that i and j should be separated, the composite weight, W_{ij}, should be small, formally defined as:

$$W_{ij} = W_{ij}^{IC} \times W_{ij}^{TX}. \tag{2.20}$$

Cour et al.[Cour et al., 2005] combined the intervening contour cue and intensity cue W_{ij}^{I} with:

$$W_{ij} = \sqrt{W_{ij}^{I} \times W_{ij}^{IC}} + \alpha \times W_{ij}^{I}. \tag{2.21}$$

Cheng et al.[Cheng et al., 2011a] proposed a new solution to fuse multiple types of image features by seeking the sparsity-consistent low-rank affinities from the joint decompositions of multiple feature matrices into pairs of sparse and low-rank matrices. Formally, given a collection of affinity matrices $Z_1, Z_2, \ldots, Z_K$, in order to take account both the advantages of LRR and use of cross-feature information, the unified affinity matrix is obtained by solving the following convex optimization problem:

$$\begin{aligned} \min_{\substack{Z_1,\ldots,Z_K \\ E_1,\ldots,E_K}} & \sum_{i=1}^{K} (\|Z_i\|_* + \lambda \|E_i\|_{2,1}) + \alpha \|Z_i\|_{2,1} \\ s.t. \quad & X_i = X_i Z_i + E_i, \ i = 1, \ldots, K \end{aligned} \tag{2.22}$$

where $\alpha > 0$ is a parameter. Note that minimizing the term $\|Z_i\|_* + \lambda \|E_i\|_{2,1}$ is LRR problem, and $\|Z_i\|_{2,1}$ is a regularization term to ensure the sparsity of resulting matrix. The deriving optimal solution $(Z_1^*, Z_2^*, \ldots, Z_K^*)$, final unified affinity matrix is defined:

$$W_{ij} = \frac{1}{2}\left(\sqrt{\sum_{l=1}^{K} (Z_l)_{ij}^2} + \sqrt{\sum_{l=1}^{K} (Z_l)_{ji}^2}\right) \tag{2.23}$$

5. **Summary.** We list popular features used in graph-oriented algorithm shown in Table 2.2. It can be seen that region based graph construction is more generic and flexible compared with pixel and patch. The disadvantages of using pixel [Fowlkes et al., 2003] is: (1) not "scale invariant"; (2) no explicit control over connectedness; (3) hard to incorporate mid/high-level shape information

such as continuity. and (4) high computational cost to derive solution to the affinity graph.

Concerning the color, each color space has an interesting property, which can efficiently be taken into account in order to make more reliable result. For example, RGB is an additive color system based on trichromatic theory and nonlinear with visual perception. This space color seems to be the optimal one for tracking applications. The HSV is interesting in order to decouple chromatic information from shading effect. The Lab color system approximates human vision, and its component closely matches human perception of lightness. The Luv components provide an Euclidean color space yielding a perceptually uniform spacing of color approximating a Riemannian space. Busin et al. [Busin et al., 2009] proposed a method which automatically selects a specific color space among a set of color spaces in order to preserve their own specific properties.

Concerning the local features, an additional step, called "coding and pooling", to construct a feature vector descriptor. One example is illustrated in the generation of textons with K-means in Fig. 2.5. More detail can be found in [Boureau et al., 2010].

2.2.2.2 Weighted Function

We now discuss distance metrics for computing the w_{ij}. Deciding a good function usually depend on the task at hand. For computing the affinity or similarity, frequently used approaches in the literature, include Euclidean distance and the χ^2 distance (see examples in section 2.2.2). We list several frequent distance metric in Table 2.3. More details can be found in [Schaeffer, 2007][Uijlings et al., 2013]

2.3 Graph Cut Cost Function

Image segmentation can be deemed as cutting edges of a suitable graph to divide data samples into disjoint groups. At the beginning, the graph is partitioned according to a fixed threshold and local properties defined by the Gestalt laws, therefore, global properties of segmentation are hard to guarantee. Then, Wu and Leahy [Wu and Leahy, 1990] proposed the first graph cut with global cost function. From then on, the graph cut based methods have attracted increasing interest on designing different cost functions for image segmentation. In particular, graph cut based methods are extensively applied for image segmentation since the Normalized cut (Ncut) [Shi and Malik, 2000] was proposed, which provides a significant progress over the previous graph cuts methods, both from a theoretical and practical perspective. The normalized cut criterion is further adopted to multi-way partitioning algorithm [Ng et al., 2001, Yu and Shi, 2003].

Give a graph $G = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is a set of vertices corresponding to image elements such as pixels or regions in the Euclidean space. E is a set of edges connecting a pair of vertices. Each edge $(u_i, u_j) \in E$ has a corresponding weight $w(u_i, u_j)$ which measures certain similarity based on the property between

Table 2.2: List of popular features used in graph-oriented algorithms. A +/- means that the feature is partially recommended, + (-) means that the feature can (cannot) be used in the case of pixel, patch or region.

Category	Feature descriptor	Variants	Usage		
			Pixel	Patch	Region
Global	Color	RGB	+	+	+
		Luv	+	+	+
		Lab	+	+	+
		HSV	+	+	+
	LBP		-	-	+
	Shape	Area	-	-	+
		Perimeter	-	-	+
		Centroid	-	+	+
		Roundness	-	+	+
		Compactness	-	+	+
		Eccentricity	-	-	+
		Orientation	-	-	+
		Convex area	-	-	+
		Minor axes length	-	-	+
		Major axes length	-	-	+
	Gestalt properties	Inter-region texton similarity	-	+/-	+
		Intra-region texton similarity	-	+/-	+
		Inter-region brightness similarity	-	+/-	+
		Intra-region brightness similarity	-	+/-	+
		Inter-region contour energy	-	+/-	+
		Intra-region contour energy	-	+/-	+
Local	Interest point	Harris	-	+/-	+
		DoG	-	+/-	+
		LoG	-	+/-	+
		Hessian	-	+/-	+
	SIFT	Gray-SIFT	-	+	+
		Color-SIFT	-	+	+
	HOG		-	+	+
	Bank of filters		-	+	+

the two vertices connected by that edge. The goal of graph cut is to cut or break the edges in E, so as to divide the vertices in V into disjointed groups $V_i \cup V_j = V$ and $V_i \cap V_j = \phi$, $(i, j \in \{1, \cdots, k\}, i \neq j)$.

2.3.1 Minimal Cut

What is a graph cut? It is related to a set of edges by which the graph G will be partitioned into two disjoint sets A and B. The degree of dissimilarity between these

Table 2.3: List of frequent distance metrics for weight calculation. $d_i = (d_i(1), d_i(2), \ldots, d_i(K))$ is an K-dimensional vector in Euclidean space, A and B are two sets of data samples.

Distance	**Definition**
Eulidean distance	$\sum_{k=1}^{K} \sqrt{(d_i(k) - d_j(k))^2}$
Manhattan distance	$\sum_{k=1}^{K} \lvert(d_i(k) - d_j(k))\rvert$
Cosine distance	$\arccos \frac{d_i \cdot d_j}{\sqrt{\sum_{k=1}^{K}(d_i(k)^2)}\sqrt{\sum_{k=1}^{K}(d_j(k)^2)}}$
χ^2 distance	$\frac{1}{2}\sum_{k=1}^{K} \frac{[d_i(k)-d_j(k)]^2}{d_i(k)+d_j(k)}$
Histogram intersection	$\sum_{k=1}^{K} \min(d_i(k), d_j(k))$
Jaccard index	$\frac{\lvert A \cap B\rvert}{\lvert A \cup B\rvert}$

two sets can be computed as the total weight of the edges that have been removed. As a consequence, the segmentation of an image can be interpreted as a graph cut and associated with the following *cut* criterion:

$$cut(A, B) = \sum_{u \in A,\ v \in B} w(u, v) \tag{2.24}$$

Wu and Leahy [Wu and Leahy, 1990] proposed a clustering method based on the minimum cut criterion in Eq.(2.24), namely minimal cut. Solving this minimal cut is well studied problem, for example Gomory and Hu [Gomory and Hu, 1961] in polynomial time. In particular, the author also seek to more general case, that is partitioning a graph into k-subgraphs, such that the maximum cut can be efficiently solved by recursively finding the minimum cuts that bisect the existing segments.

As noticed by Wu and Leahy, the minimum cut criteria has bias on small sets of isolated nodes in the graph, i.e. short boundaries, since the *cut* increases with the number of edges going across the two partitioned parts.

2.3.2 Ratio Regions

Cox et al. [Cox et al., 1996] proposed a cost function called ratio regions to alleviate

minimum cut bias by incorporating both interior region and boundary information. The ratio region criteria minimizes a new cost function based on the ratio of the cost of the perimeter of the segmented region to the benefit assigned to its enclosed interior. Formally, let P be a directed path in graph G that starts and finishes at the same vertices v, denoted by $cost(P)$ representing the length of the boundary. Segment-area cost is denoted by $weight(P)$, the ratio region is defined as:

$$Regioncut(A, B) = \frac{cost(P)}{weight(P)} \tag{2.25}$$

Notice that the cost function favors large regions in the image and the region characteristic of smoothness is measured using the area and perimeters. In particular, the limitation of this criteria is that it can only segment enclosed objects due to its definition. Rao (personal communication) gives a polynomial-time algorithm for finding a cut that minimizes this function with binary search through a space of max-flow problems.

2.3.3 Normalized Cut

Shi and Malik [Shi and Malik, 2000] proposed a new disassociation measure called as Normalized cut (Ncut), to avoid the minimal cut bias, i.e. segmenting small sets of points. Ncut computes the cut cost as a fraction of the total edge connections to all the nodes in the graph, rather than only taking account the value of total edge weight connections between two partitions. Formally, Ncut is defined as:

$$Ncut(A, B) = \frac{cut(A, B)}{vol(A)} + \frac{cut(A, B)}{vol(B)} \tag{2.26}$$

where $vol(\cdot)$ is the total connection from vertices in a set (e.g. A) to all vertices in the graph G. Formally, $vol(A) = \sum_{u \in A v \in V} w(u, v)$, where $w(u, v)$ is weight on edge (u, v) and can be computed with certain affinity function (see section 2.2.2). $cut(A, B)$ is defined in Eq.(2.24).

Finding a cut that minimizes this cost function is NP-complete. Shi and Malik also present a method to compute the optimal partition using spectral graph theory [Ulrike, 2007]. However, Ncut based method (e.g. [Cour et al., 2005] [Yu and Shi, 2003]) tend to break large uniform regions because of the normalization prior. More precisely, the cost criteria favors balanced partitions, at the risk of breaking object boundaries or large uniform region (e.g. sky and grass) into chunks.

2.3.4 Mean Cut

Wang and Siskind [Wang and Siskind, 2001] proposed a cost function called as mean cut with no bias, such as short boundary, large regions, or similar weighted regions. The cost criteria finds cuts which minimize the average edge weight in the cut boundary. Formally, it is defined as:

$$meancut(A, B) = \frac{cut(A, B|w(u, v))}{cut(A, B|\mathbf{1})} \tag{2.27}$$

where $cut(A, B|W(u, v))$ means the cut cost between region A and B given the edge weight $w(u, v)$, and $cut(A, B|\mathbf{1})$ is defined similarly with all edge weights to be 1, i.e. boundary length.

The mean cut criteria can generate both open and closed boundaries and guarantees that partitions are connected as well as does not impose any bias. The author also present polynomial-time global optimization algorithm for this cost function, yet only can be applied in connected planar graphs. To solve the Eq.2.27, there are three reductions: 1) from minimum mean cut to minimum mean simple cycle; 2) from minimum mean simple cycle to negative simple cycle and 3) from negative simple cycle to minimum-weight perfect matching.

2.3.5 Ratio Cut

Wang and Siskind [Wang and Siskind, 2003] generalized the mean cut cost function by removing the restriction that edge weights are 1 and call this new cost function as ration cut. It normalizes the first boundary cost by the second boundary cost, and is formally defined as:

$$Ratiocut(A, B) = \frac{cut1(A, B)}{cut2(A, B)} \tag{2.28}$$

where $cut1(A, B)$ and $cut2(A, B)$ are defined on the graphs of different iterations. The author adopts the same reduction process as mean cut to solve this NP-hard problem.

2.4 Graph Partitioning

Solving most graph cut cost functions yields NP-complete problem. To approximate the optimal solution, there are two broad categories of methods, i.e. local (e.g. the Kernighan-Lin [Kernighan and Lin, 1970] algorithm) and global (e.g. Spectral partitioning). The local method's major drawback is the arbitrary initial partitioning of the vertex set, which can affect the final solution quality. Global methods rely on properties of the entire graph and do not rely on an arbitrary initial partition. More in detail, they can be of three different types, as suggested in [Karypis and Kumar, 1998], listed in following:

- Spectral methods. Many algorithms have been developed that find a reasonably good partition with spectral partitioning, where a partition is derived from the spectrum of the adjacency matrix. Shi and Malik approximate computing the minimum normalized cut criterion with eigen-problem, later, Weiss has shown how the eigenvector problem relate to more standard spectral partitioning methods on graphs. The core idea is to use matrix theory and linear algebra to study properties of the incidence matrix, W, and the Laplacian matrix, $L = D - W$, which provide a great deal of information about graph G. Using the spectrum of graph Laplacian, which is symmetric positive semi-definite matrix, can capture essential cluster structure of a graph, i.e. the eigenvectors of the graph Laplacian or its variants. This can be graphically

illustrated in Fig.2.8. This is a rich area of mathematics and using eigenvectors of the Laplacian for finding partitions of graphs can be traced back to [Cheeger], [Donath and Hoffman, 1973], and [Fiedler, 1975]. For a tutorial introduction to spectral graph theory, Chung[Chung, 1997] presented a good survey on spectral clustering. In mathematics, the spectrum of a symmetric positive semi-definite matrix is the complete real positive eigenvalues $0 \leq \lambda_1 \leq \ldots \leq \lambda_n$, by solving following generalized eigen-problem:

$$L\mathbf{f} = \lambda D\mathbf{f} \tag{2.29}$$

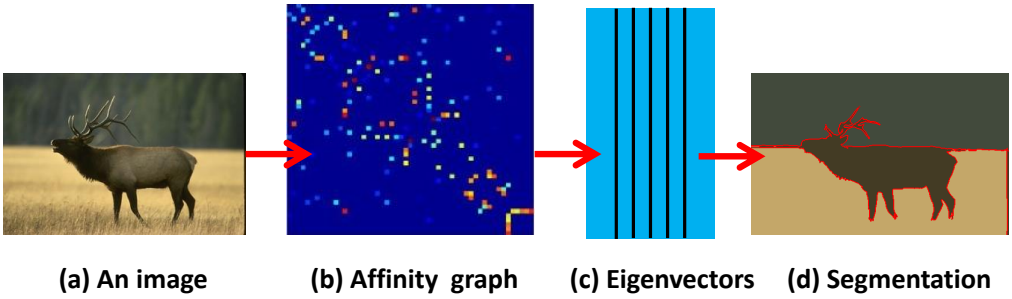
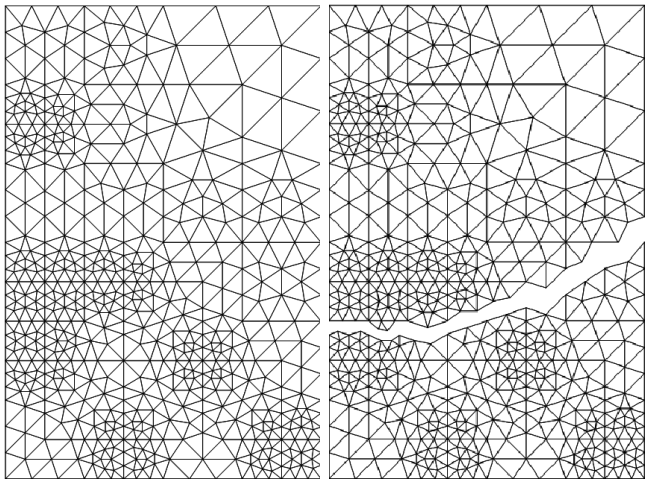


Figure 2.8: Illustration of spectral methods for image segmentation.

However, these methods are very expensive since they require the computation of the eigenvector corresponding to the second smallest eigenvalue (Fiedler vector). Execution time of the spectral methods can be reduced if computation of the Fiedler vector is done by using a multilevel algorithm.

- Geometric partitioning algorithm. These methods use the geometric information of the graph to find a good partition. This algorithm produces partitions that are provably within the bounds that exist for some special classes of graphs (see an example shown in Fig.2.9). Although these methods tend to be fast yet obtain worse partitions than those obtained by spectral methods [Karypis and Kumar, 1998]. Moreover, they are applicable only if coordinates are available for the vertices of the graph.

- Multilevel graph partitioning. These methods reduce the size of the graph (i.e., coarsen the graph) by collapsing vertices and edges, partitions the smaller graph, and then uncoarsens it to construct a partition for the original graph. This process is graphically illustrated in Fig. 2.10. At beginning, researchers designed the multilevel graph partitioning to reduce the computational time at cost of worse partition quality. Later, many methods have been proposed to give both fast execution times and very high quality results. Hendrickson and Leland [Hendrickson and Leland, 1995] construct a sequence of increasingly coarse approximations to a graph, then the smallest graph is partitioned by spectral method, finally the sequence of graphs are projected back to the original graph, periodically improving it with a local refinement algorithm, such as Kernighan-Lin algorithm [Kernighan and Lin, 1970]. In particular,



(a) Input mesh

(b) Partition of the mesh with 42 cut edges

Figure 2.9: Illustration of the geometric partitioning method [Gilbert et al., 1998].

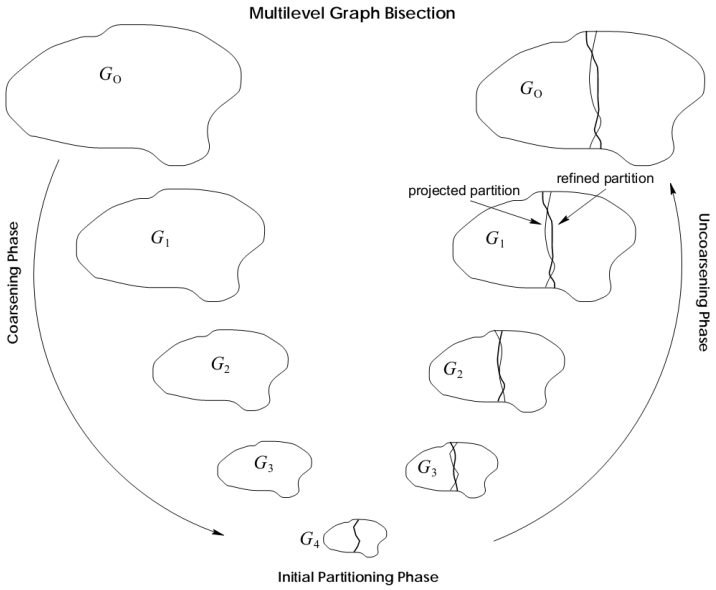


Figure 2.10: The various phases of the multilevel graph bisection. During the coarsening phase, the size of the graph is successively decreased; during the initial partitioning phase, a bisection of the smaller graph is computed; and during the uncoarsening phase, the bisection is successively refined as it is projected to the larger graphs. During the uncoarsening phase the light lines indicate projected partitions, and dark lines indicate partitions that were produced after refinement.

this work showed that multilevel schemes can provide better partitions than spectral methods at lower cost for a variety of finite element problems.

2.5 Summary

Graph cut methods are clearly adapted to the objective of segmentation, i.e. minimizing certain cut cost function makes vertices in different sets (dissimilar vertices) or groups vertices in the same sets (similar vertices). Many different cut criteria have been proposed. In practice, to implement each cost function, finding the solution is a NP-hard problem. Therefore, efficient approximations of the solution need to be studied. Since these methods form different basis for general image segmentation problem, they can be combined with other segmentation techniques for further extension.

Table 1 summarizes different graph cut methods introduced above, including the cost functions, optimization methods, complexity and their properties.

Table 2.4: Comparisons between different graph cut cost functions

Graph cut methods	Objective function	Optimization method	Computational complexity	Bias
Minimal cut [Wu and Leahy, 1990]	$Mincut(A,B)=\sum_{u\in A,v\in B} w(u,v)$	Gomory-Hu's K-way maxflow algorithm	Polynomial time	Short boundaries
Ratio regions [Cox et al., 1996]	$Regioncut(A,B)=\frac{cost(P)}{weight(P)}$	Local rearching for the solution	$O(n\log(n))$	Smooth boundaries
Ncut [Shi and Malik, 2000]	$Ncut(A,B)=\frac{cut(A,B)}{vol(A)}+\frac{cut(A,B)}{vol(B)}$	Generalized eignsystem	$O(mn)$	Similar weight partition
Mean Cut [Wang and Siskind, 2001]	$Meancut(A,B)=\frac{cut(A,B\|w(u,v))}{cut(A,B\|\mathbf{1})}$	minimum-weight perfect matching	polynomial time	No bias
Ratio cut [Wang and Siskind, 2003]	$Ratiocut(A,B)=\frac{cut1(A,B)}{cut2(A,B)}$	Baseline method	$O(n^{7/4})$	No bias

Chapter 3

Literature Review: Image Segmentation

Contents

3.1 Introduction

Image segmentation is a fundamental problem in computer vision. The objective of image segmentation is to segment an image into several non-overlapping regions that are deemed meaningful according to some objective criterion, homogeneity in some feature space or separability in some other one for example. Image segmentation have been a long studied problem. Since the first image segmentation approaches published more than 40 years ago, see for instance [Muerle, 1968], thousands of algorithms have been proposed, and they can be very different using different mathematical models or according to different application goals. Typically, in many literature surveys [Fu and Mui, 1981] [Luccheseyz and Mitray, 2001] [Peng et al., 2013], existing image segmentation methods can be categorized as unsupervised or supervised. Unsupervised methods are also called as low-level or bottom-up methods in the literature. In this thesis, we consider those methods that are without any human interactions nor prior knowledge involving training process or object-class specific information beforehand. In many instances of supervised setting, also called as top-down segmentation or high-level segmentation, there are two versions, namely weakly supervised and fully supervised methods. The former tries to only use the image annotation which describes the visual concepts depicted in the image, instead of manually segmented result for each pixel. Section 3.3 relates to supervised methods in the sense that there is human interaction, e.g. drawing a bounding box

containing objects in the interactive segmentation, and the co-segmentation or object discovery models the segmentation processing provided at least multiple images share a common object. Fully supervised methods involve tasks such as semantic segmentation, image parsing and scene understanding, which need to recognize and label each pixel as a semantic object or event within the image, more detail can be found in Section 3.4.

Note that the literature review is performed from the view-point of general natural image segmentation, specific domain such as medical image processing is not included in the survey.

3.2 Unsupervised Image Segmentation

Unsupervised image segmentation remains a daunting challenge and a hot research topic for computer vision. It is mostly defined as a bottom-up process, employing no high-level knowledge. Low-level knowledge such as coherence of brightness, color, texture, gradient or motion is exploited to design various approaches. Unsupervised image segmentation techniques can be classified into two broad families: (1) region-based, and (2) edge-based approaches. Region-based approaches try to find partitions of the image pixels into sets corresponding to coherent image properties such as brightness, color and texture. Edge-based approaches usually start with a first stage of edge detection, followed by a linking process that seeks to exploit curvilinear continuity.

I am not going to survey the review from the above point of view, instead, I would roughly summarize the literature under the whole domain development. As we know, there are countless papers published around the topic of image segmentation. In following review, those segmentation methods or features, which pursue optimal performance on single or a few images, will be classified to the early stage of image segmentation. While those approaches, which combine different algorithms or different parameters of the same algorithm and validated on large image annotated benchmarks with segmentation metrics, will be deemed as modern stage of image segmentation. A broad family of approaches to segmentation make use of low-level features such as brightness, color, or texture, they are rough be introduced as follows.

- **Histogram Thresholding.** Histogram thresholding is one of the widely used techniques for gray-level image segmentation. It assumes that images are composed of regions with different gray level ranges, the histogram of an image can be separated into a number of peaks (modes), each corresponding to one region, and there exists a threshold value corresponding to valley between the two adjacent peaks. Color images can also be thresholded. One approach is to perform threshold for each of the RGB components of the image and then combine them with an AND operation, see more information in [Cheng et al., 2001]. Automatic thresholding techniques can be roughly categorized as global thresholding and local thresholding. The Otsu's method [Otsu, 1975] and Kittler and Illingworth's method [Kittler and Illingworth, 1986] are the most popular methods for image thresholding. Other interesting methods

are, as summarized in [Sezgin et al., 2004]: histogram shaped-based method, clustering-based method [Tizhoosh, 2005], entropy-based methods [Abutaleb and Eloteifi, 1988], spatial methods [Wang et al., 2008c] and local methods. It is worth to mention that early methods for image thresholding were basically devised to separate classes that are unimodal and lies in the assumption that class data are Gaussian. Recently, researchers have used other distribution types to provide better image thresholding methods by modeling histogram classes using, for instance, Poisson, generalized Gaussian, skew-normal and Rayleigh distributions. Another parallel trend is using mixture methods for segmentation, where data are clustered to classes determined by the components of a learned mixture model. For instance, Boulmerka et al. [Boulmerka et al., 2014] propose a thresholding method by modeling non-Gaussian and multi-modal class-conditional distributions using mixtures of generalized Gaussian distributions.

(a) original image (b) thresholded result

Figure 3.1: Illustration of thresholding result with the Otsu method [Otsu, 1975].

- **Contour detector.** The problems of contour detection and segmentation are related, but not identical. In general, contour detectors offer no guarantee that they will produce closed contours and hence do not necessarily provide a partition of the image into regions. But, one can always recover closed contours from regions in the form of their boundaries. Early edge detection methods, such as the Robert edge detector [Roberts, 1963], the Canny edge detector [Canny, 1986] are based on the abrupt changes in image intensity. Later, more complex techniques are obtained by considering the response of the image to a family of filters of different scales and orientations, such as Gaussian smoothing at multiple scales [Witkin, 1987], and the Oriented Energy approach [Freeman and Adelson, 1991]. In modern local methods, one develops a powerful contour detector by taking into account color and texture information and make use of cue combination [Jitendra et al., 2001]. Many researcher still concern the issue, that is objects may appear at large scales of the image. Ren[Ren, 2008] combines strengths from both large-scale detection (robust but poor localization) and small-scale detection (detail-preserving but sensitive to clutter) multiple scales of the local operators developed by [Ar-

belaez et al., 2011]. As pointed in [Ren and Bo, 2012], there are something in common among modern contour detection methods: they are built on top of a set of gradient features measuring local contrast of oriented discs, using chisquare distances of histograms of color and textons. For example, the state-of-the-art contour detector, global probability boundary (gPb) [Arbelaez et al., 2011], whose output $E(x, y, e)$ predicts the probability of an image boundary at location (x, y) and orientation. They also present the method to build hierarchical regions by exploiting the information from this contour signal using a sequence of two transformations, the Oriented Watershed Transform (OWT) [Arbelaez, 2006] and Ultrametric Contour Map (UCM) [Arbelaez, 2006].

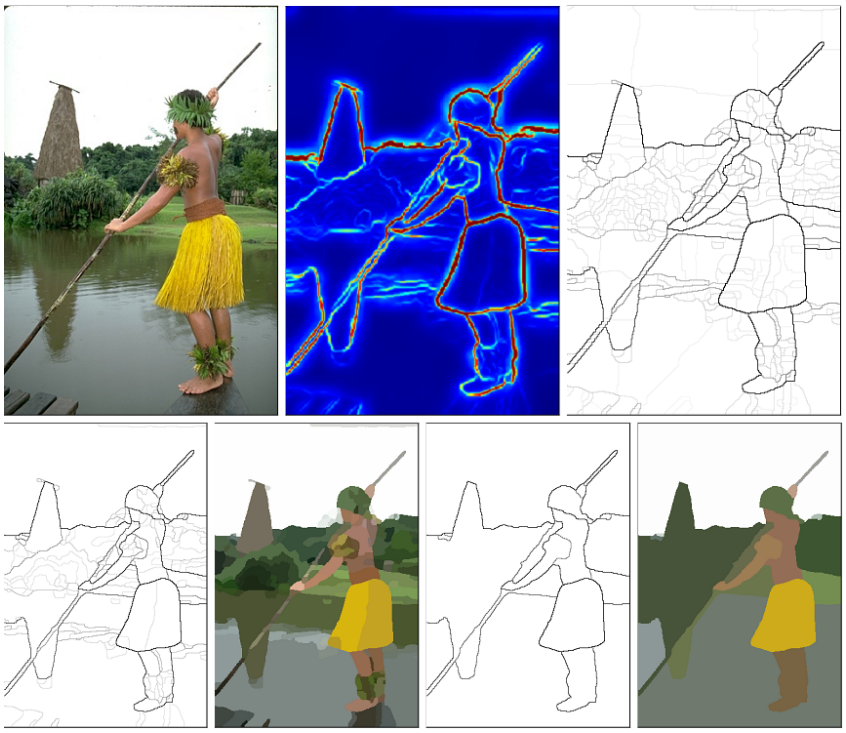


Figure 3.2: Hierarchical segmentation from contours [Arbelaez et al., 2011] . Top Left:Original image. Top Middle: Maximal response of contour detector gPb over orientations.Top Right: Weighted contours resulting from the Oriented Watershed Transform - Ultrametric Contour Map algorithm using gPb as input. Bottom: Contours and corresponding segmentations obtained by thresholding the UCM at levels 0.1(left), and 0.5 (right), with segments represented by their mean color.

- **Clustering.** Cluster analysis is one of the most fundamental modes of understanding and learning. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). It has broad application in many contexts and appealed by researchers in many disciplines. Thousands of clustering algorithms have been proposed, which makes it extremely difficult to review all the published approaches. We refer readers to the review by Jain and Dubes[Jain and Dubes, 1988], Jain et al.

[Jain et al., 1999] and Jain[Jain, 2010] for more information.

Clustering broadly divided into two groups: hierarchical and partitional (hierarchical methods produce a nested series of partitions, while partitional methods produce only one). Most hierarchical clustering algorithms are variants

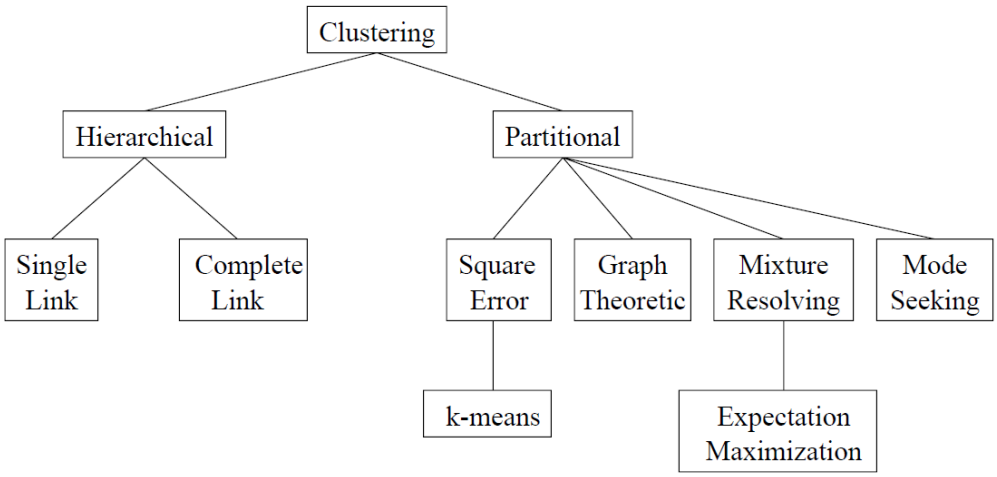


Figure 3.3: A taxonomy of clustering algorithm [Jain et al., 1999]

of the single-link [Sneath et al., 1973], complete-link [King, 1967]. The difference between the single-link and complete-link algorithm is the distance between two clusters, which are measured by the minimum and maximum of the distances between all pairs of patterns drawn from the two clusters, respectively. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria. Compared to hierarchical clustering algorithms, partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure.

The most popular and the simplest clustering algorithm is K-means. The basic idea of the K-means method is to cluster a set of points in some metric space into K clusters by iteratively improving K cluster centres and grouping each point to the cluster with the closest centre (called hard assignment); the centres are chosen to minimize the sum-of-squares of the intra-cluster distances. Such iterative algorithms for clustering provide a partial clustering for the data already seen from an unknown data stream to be clustered. In order to cluster a large database, incremental clustering is useful for clustering data sets that undergo frequent modification, such as addition, removal or editing of the data elements. It is noteworthy that in iterative clustering, the order in which the data are processed may significantly affect the resulting clusters. To avoid these problems, commonly the existing partial clustering is constantly optimized with respect to some carefully selected global measure as new data are processed, reassigning also the old data as necessary. For instance, fuzzy c-means [Bezdek, 1981], is an extension of K-means where

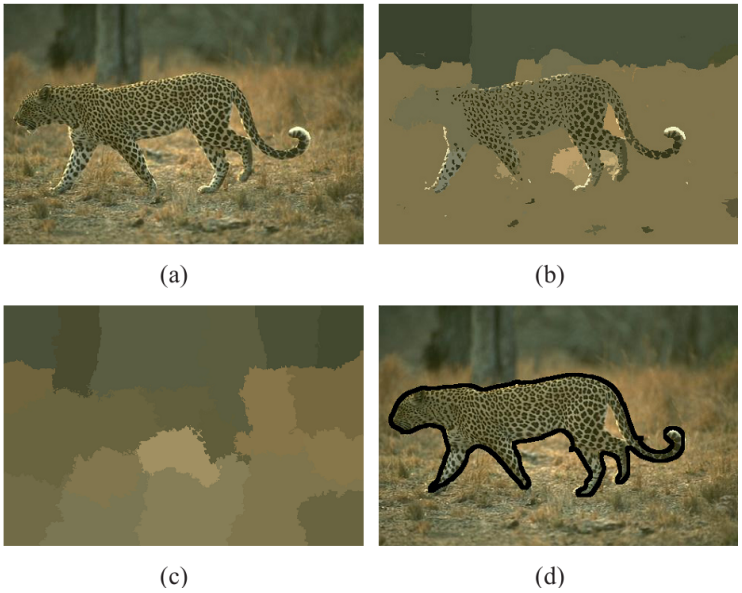
each data point can be a member of multiple clusters with a membership value (soft assignment).

Compared to the traditional algorithm such as K-means or single linkage, spectral clustering has many fundamental advantages. Results obtained by spectral clustering often outperform the traditional approaches, spectral clustering is very simple to implement and can be solved efficiently by standard linear algebra methods. Spectral clustering is typically based on computing the eigenvectors corresponding to the second-smallest eigenvalue of the normalized Laplacian or some eigenvector of some other matrix representing the graph structure.

As a single cluster can be well modeled by a Gaussian distribution, it is straightforward to assume that each probabilistic distribution is Gaussian, so known as the mixture of Gaussian model (GMM). Then the problem of segmenting the data is converted to a model estimation problem. The EM algorithm [Dempster et al., 1977] is used to infer the parameters in mixture models. Recently established minimum description length (MDL) [Ma et al., 2007], explicitly cast segmentation problem as density estimation, which entails estimating the mixture of all the models and then assigning each data point to the model with the highest likelihood. But when the data is high-dimensional, the feature space is usually sparse, making it difficult to distinguish high-density regions from low-density regions. Subspace space clustering (SSC) [Elhamifar and Vidal, 2009] overcome this limitation by finding clusters embedded in low-dimensional subspaces of the given high-dimensional data. SSC uses the sparsest representation produced by ℓ_1-minimization to define the affinity matrix of an undirected graph. Then subspace segmentation is performed by spectral clustering algorithms such as the Normalized cuts [Shi and Malik, 2000].

- **Mode-seeking.** Mode seeking provides a versatile tool for feature space analysis by finding local density maxima (or modes) in the feature space. In mode seeking clustering, data belonging to the same cluster fall within the same density attraction basin where the attraction force points to the direction that mostly increases the estimated density. Mean shift (MS) [Comaniciu and Meer, 2002] is regarded as one of the most canonical mode seeking algorithms with numerous real applications in computer vision. Given a collection of data samples distributed according to an unknown distribution on a Euclidean space, the mean shift is designed to iteratively locate the underlying modes together with the points that belong to the cluster associated with each mode. The success of the mean shift algorithm inspired many researchers to develop different variants of the standard version. For instance, there is a great deal of works that focus on improving the mean shift in: 1) speed [Paris and Durand, 2007]; 2) accuracy via adaptive bandwidths [Georgescu et al., 2003] and asymmetric kernels [Yilmaz, 2007]; 3) manifold. Subbarao and Meer [Subbarao and Meer, 2006] extend the Euclidean MS formulation to two particular analytic manifolds, Grassmann manifolds and Lie groups, which is actually the nonlinear MS. Inspired by the nonlinear MS, Medoid shift [Sheikh et al., 2007] and the

quick shift [Vedaldi and Soatto, 2008] algorithms are designed to cluster data on non-Euclidean spaces and employed for image segmentation and categorization. Cetingul and Vidal [Cetingul and Vidal, 2009] generalize mean shift to non-linear manifolds and intrinsically model curved mean shift space. Yu et al. [Yu et al., 2012] modify mode seeking method called as convex shift to group superpixels using bag of features. Fig.3.4 present the perceptual comparison of several popular mode seeking methods, i.e. mean shift, quick shift as well as the convex shift.



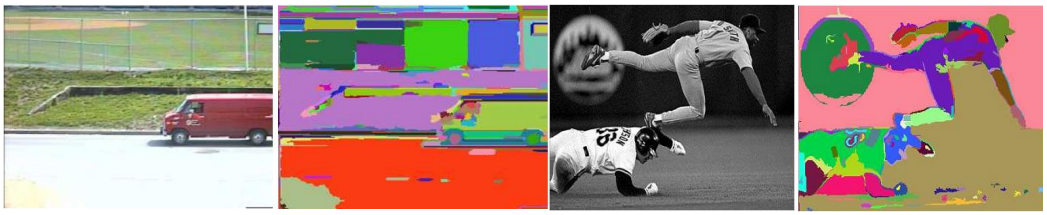
(a) (b) (c) (d)

Figure 3.4: Comparison of segmentation result: (a) The original image. (b)-(c) Segmentation generated by mean shift [Comaniciu and Meer, 2002], quick shift [Vedaldi and Soatto, 2008] and convex shift [Yu et al., 2012].

- **Graph based methods.** In Chapter 2, we focused on reviewing unsupervised graph based methods with cost functions, which are designed to minimize the similarity between pixels that are then split with graph partitioning method. In this Chapter, we briefly review other graph based methods in unsupervised manner. More specifically, we will introduce the minimal spanning tree based methods, since other graph based methods such as graph cut on Markov random field models and the shortest path based methods incorporate high-level knowledge or user's guidance to segment semantic objects in images. We will discuss them in the supervised segmentation setting.

  The minimal spanning tree (MST) is a spanning tree with the smallest weights among all spanning trees. The MST is essentially related to graph clustering, where the MST can be used to compute the weights between two vertices. A typical MST method, such as the Prim's algorithm [Prim, 1957], is constructed by iteratively adding the frontier edge of the smallest edge-weight. For instance, Felzenszwalb and Huttenlocher' s (FH) graph-based method [Felzenszwalb and Huttenlocher, 2004] is a very efficient algorithm and recently it has

been frequently applied in generating superpixels [Li et al., 2012][Wang et al., 2013a] or as starting step in other high-level machine learning tasks [van de Sande et al., 2011]. The algorithm considers both the difference across two regions and the difference inside a region. It merges regions greedily according to these differences and returns a gross segmentation. Fig. 3.5 illustrates the segmentation results obtained with this algorithm.



(a) Street scene (b) segmentation with FH (c) Baseball scene (d) segmentation with FH

Figure 3.5: Illustration of the segmentation result by MST based algorithm, FH [Felzenszwalb and Huttenlocher, 2004].

- **Partial differential equation-based methods.** Those methods are more recent, can be very efficient and have broad applications in image processing, such as image interpolation, denoising and segmentation, etc. In this review, we briefly introduce classical PDE methods for segmentation. The basic idea is to evolve a curve, subject to constraints from a given image, in order to detect objects in that image. For instance, starting with a curve around the object to be detected, the curve moves toward its interior normal and has to stop on the boundary of the object. The early methods in segmentation are parametric methods, such as the famous snake [Kass et al., 1988] and its variants. Later, the level set method for capturing dynamic interfaces and shapes was introduced in Osher and Sethian [Osher and Sethian, 1988] and adapted for segmentation in Caselles [Caselles, 1995] and Malladi et al. [Malladi et al., 1995]. The basic idea of the level set method is to represent a contour as the zero level set of a higher dimensional function and formulate the motion of the contour as the evolution of the level set function. The active contour models in level set can also be classified as edge-based methods, e.g. geodesic active contour (GAC) [Caselles et al., 1997] and region-based method, e.g. the Mumford-Shah model [Mumford and Shah, 1989] and its piecewise constant version addressed numerically with the Chan-Vese (CV) model [Chan and Vese, 2001]. In particular, the CV model has gained great success due to its simplicity and efficiency. Based on CV model, many works have been devised to generalize it, e.g. extending the CV model to multi-channel images [Chan et al., 2000] and multiphase model [Vese and Chan, 2002]. More recently, some works improved the active contour model to segment images with intensity inhomogeneities [Li et al., 2008][Zhao et al., 2012]. Another direction is incorporating prior shape [Cootes et al., 1995] [Chan and Zhu, 2005] into the energy function to make it robust to occlusion, severe pollution etc. As discussed in many works, edge-based, region-based or prior shape based

methods have advantages and weaknesses. State-of-the-art models, e.g. [Ali and Madabhushi, 2012], try to combine all their merits in a unified energy function to obtain more reasonable segmentations.

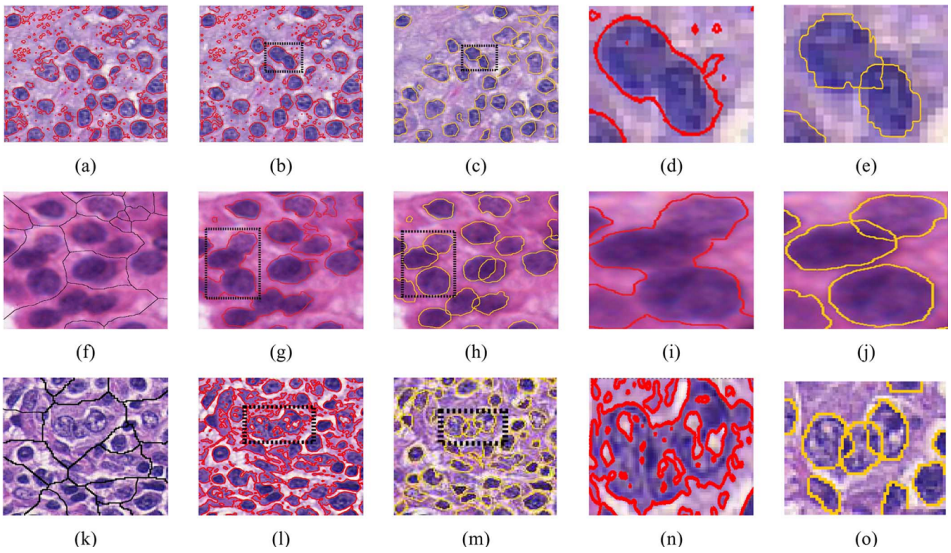


Figure 3.6: (a), (f), (k) Watershed initialization of nuclei and lymphocytes on prostate and breast cancer histopathology with corresponding segmentation results obtained via GAC [Caselles et al., 1997] (b), (g), (l); Method in [Ali and Madabhushi, 2012] (c), (h), (m); magnified region (d), (i), (n) from (b), (g), (l); magnified region (e), (j), (o) from (c), (h) and (m)

3.3 Foregound/Background Segmentation

Foreground/background segmentation has been seen as an important and even necessary precursor for object recognition when supervisory information is available in the form of labeled training data (full image or, in interactive settings, smaller groups of pixels).

3.3.1 Interactive Segmentation

Interactive segmentation is an important problem in computer vision where discrete optimization techniques have had a significant impact. Generally the user assigns some pixels to the foreground and background regions manually and these constrain an energy function, which is optimized using a global minimization algorithm. The goal is to allow the user to quickly segment foreground objects from the rest of the image. This is done via iterative process, where the user has the opportunity to correct or improve the current segmentation as the algorithm progresses.

There have been proposed a wealthy literature on this topic. One of them is the shortest path based methods. They are to finding the shortest path between two vertices in a weighted graph, i.e. it will connect the two nodes with the minimized

sum of edge weights. In image segmentation context, the problem of finding the minimum cost path between the two vertices corresponds to finding the best boundary segment. Several algorithms can be used to solve it, one of which is Dijkstra's algorithm. For instance, Intelligent scissors [Mortensen and Barrett, 1995] find the object contour via shortest paths in a graph near the boundary of the target clicked by user. Given an initial pixel s by user, the shortest paths can derive an optimal curve from s to any pixel within the image. So with user clicking the mouse around the target, the contour of the target can be computed with a shortest paths in a graph.

In these approaches, graph-cuts (also known as s/t cut) methods have been quite popular for foreground/background segmentation [Boykov and Funka-Lea, 2006, Boykov and Kolmogorov, 2004], for that they can obtain optimal solution for defined energy minimization problems that involve region and boundary properties. These methods employ appearance models for the foreground and background which are estimated through user interactions. Graph-cut method demands user input seeds in the foreground region and background region. For example, the lazy snapping [Li et al., 2004] require loosely foreground position seed points and editing the boundary to modify results. [Bagon et al., 2008] require a user click a point inside the object of interest, and use EM to estimate a sophisticated self-similarity energy. GrabCut [Rother et al., 2004] demands less human interaction, where a simple rectangular seed around the object of interest is initialized. It iteratively uses graph-cuts in the model of foreground/background. Chen et al. [Chen et al., 2012] proposed adaptive figure-ground classification algorithm with a prior bounding box defined by user. The image is oversegmented with mean shift [Comaniciu and Meer, 2002], then the background and foreground regions are gradually refined and multiple segmentations according to several evaluation scores are selected. The final segmentation is determined with a voting or weighted combination scheme. Liu et al. [Liu et al., 2009] propose "Paint Selection" which gives user instant feedback when they drag the mouse.

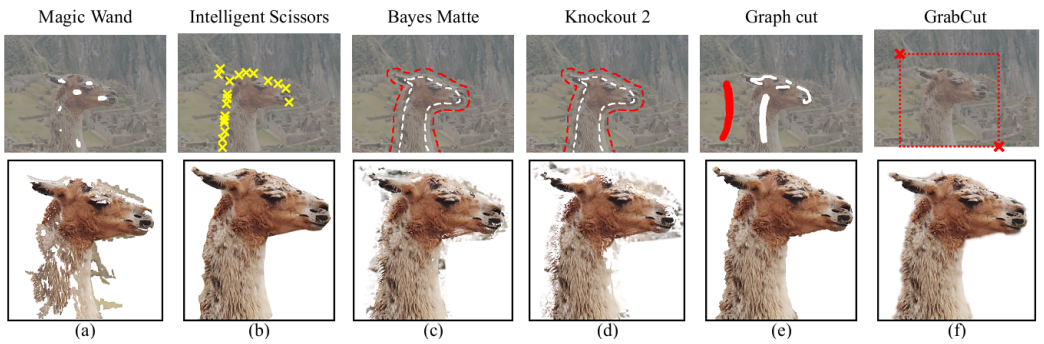


Figure 3.7: Illustration of comparison of interactive segmentation method. The top row shows the user interaction required to complete the segmentation or matting process: white brush/lasso (foreground), red brush/lasso (background), yellow crosses (boundary). The bottom row illustrates the resulting segmentation.

3.3.2 Class Segmentation

Given a set of training examples from *an* object class, class segmentation pursue to be able to automatically learn a class specific prior knowledge to categorize and segment images which contains the specific object class. The class segmentation differs from unsupervised image segmentations and interactive segmentation in that it not only divides the image into a set of coherent regions, but also assigns a class labels to each region.

Many methods have been proposed on such topic to use top-down knowledge to leverage the segmentation performance. Learning a prior class-specific shape model have been intensively studied for the task of object recognition. Borenstein and Ullman 2002 used prior shape characteristics of objects within a given class to guide the segmentation process, where the segmentation result is obtained by fitting the fragments to the image, which is analog to jigsaw-puzzle. Leibe and Schiele 2006 defined a implicit shape model using codebook which groups and encodes specific object class's local appearance represented with patches centered around interest points over all training images. Alexe et al. 2010 learned the object class's shape structure with a reference coordinate frame common across images. Such reference frame is determined in every image with a salient object detector.

3.3.3 Cosegmentation/Object Discovery

The term "cosegmentation" (also known as object discovery) is first coined in [Rother et al., 2006], and have attracted increasing interest in computer vision. The early aim of cosegmentation is to simultaneously segment the common parts within a pair of images by proposing a generative model, which encodes constraints (such as spatial coherency) in the formulation of MRF. As shown in Figure 3.8, cosegmentation is superior to classical foreground/background methods (GrabCut [Rother et al., 2004]) which do not model joint foreground. However, numerous methods have been proposed, they can only handle a pair of images with the same object at a time [Rother et al., 2006], and/or need user interaction [Batra et al., 2010].

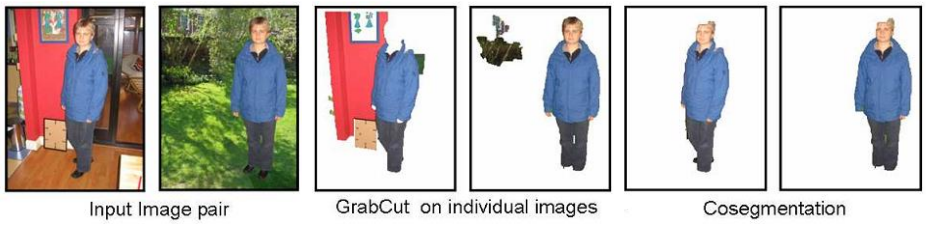


Figure 3.8: Illustration of cosegmentation [Rother et al., 2006]: given a pair of images, the task is to segment the common part in both images. Compared to GrabCut [Rother et al., 2004], which segment the foreground individually, cosegmentation outperforms traditional foreground/background method.

There has been increasing interest on the developing cosegmentation by taking into account: (1) multiple object classes and 2) more images. Their target is to

jointly segment K different objects classes from multiple images, each of which contains unknown subset of K object classes. Vicente et al. 2011 introduced object-like into cosegmentation by generating a pool of region proposals [Carreira and Sminchisescu, 2010] followed by learning similarity measure using Random Forest regressor, which evidently boost the performance on standard benchmark.

It is worth to mention that although many cosegmentation and object discovery methods claim as unsupervised, to clarify the ambiguities, we categorize them as weakly/fully supervised, in the sense that the assumption of multiple images supposed to have recurring visual categories provides a weak form of supervision. But the strong assumption that the object is present in all of the images is relaxed. Further improvements are made to handle images which might not contain the common object shown in Figure 3.9. These methods aim to handle multiple object classes. In particular, Kim and Xing 2012 explicitly handled this challenge by foreground segmentation(with/without user input label) and region assignment. Joulin et al. 2012 implicitly handled multiple object classes with non-convex energy function which combines spectral- and discriminative-clustering. Ma and Latecki 2013 solved this problem based on graph transduction semi-supervised learning. Rubinstein et al. 2013 further extended the cosegmentation to discover object in internet images, and can handle noisy images which do not contain common object of interest, with constructing a large-scale graphical model.

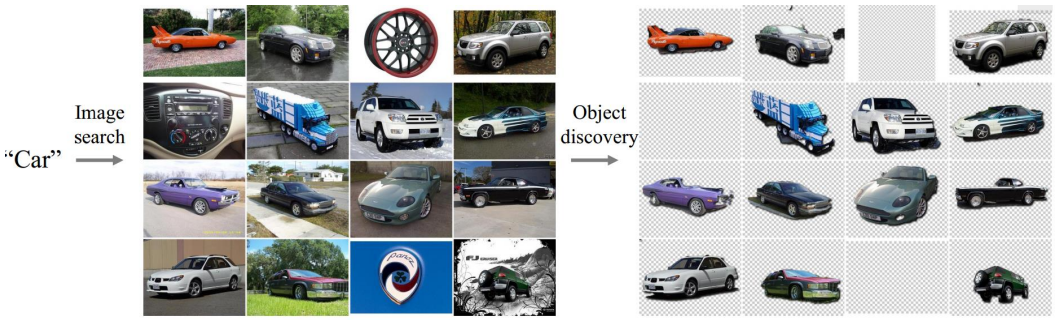


Figure 3.9: Illustration of cosegmentation [Rubinstein et al., 2013] applied in images obtained from Internet search. Note that no objects are discovered for noisy images.

3.4 Semantic Segmentation

Semantic segmentation (also known as object segmentation and image parsing), has attracted wide and intensive interest in computer vision. The task of semantic segmentation involves recognition and localization, that is to assign a class label to each pixel. This has high practical value in many applications, such as image editing, object retrieval and intelligent image coding.

In recent years, numerous semantic segmentation methods have been proposed. They can be generally categorized as bottom-up and top-down. Increasing studies (see [Li et al., 2002]) suggest that humans can perform high level scene and object categorization tasks as fast as low level texture discrimination and other so-called

pre-attentive vision tasks, i.e. humans can detect both low and high level visual patterns at early stages in visual processing. This evidence makes two paradigms to pursue to semantic segmentation.

The bottom-up methods generally consist of four steps: (1) extract low level image features (e.g., SIFT) over pixel or regions, 2) perform feature coding (e.g., K-means and sparse coding) and pooling (e.g., max-pooling) to construct mid-level feature vector descriptor (e.g., bag-of-words), and 4) train a discriminative classifier (e.g., SVM). For example, Carreira et al. 2012b proposed a method where figure-ground (regions) are generated by solving constrained parametric min-cut (CPMC) [Carreira and Sminchisescu, 2010] problems with various choices of the parameter. The semantic hypotheses are then ranked and classified by making use of support vector regression (SVR) based on their "objectness". Another typical example using this paradigm (shown in Fig.3.10), Zou et al.[Zou et al., 2012] first generate multilevel regions for an input image with *globalPb*-UCM [Arbelaez et al., 2011], extract color-SIFT [van de Sande et al., 2010] and derivatives of Gaussians on the regions, generate bag-of-words with K-means, then train a region based classifiers with support vector machine with multiple kernel learning [Varma and Ray, 2007]. The semantic labeling is derived by considering SVM scores, region sizes and common sense. Many works in literature focus on proposing discriminative

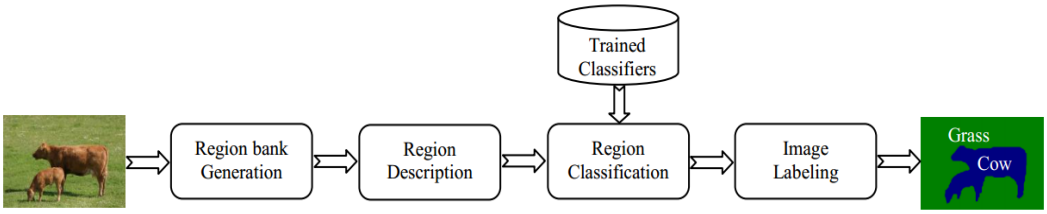


Figure 3.10: Illustration of bottom-up semantic segmentation framework in paper Zou et al. [2012]

feature descriptors for semantic segmentation. Shotton et al. [Shotton et al., 2006] proposed bag of semantic texton forest, which is computed over local rectangular regions for semantic segmentation. Carreira et al.[Carreira et al., 2012a] proposed *second order pooling* to encode the second order statistics of local descriptors inside a region. Combining this pooling technique with CPMC, leading to be the winner of segmentation competition on PASCAL VOC [Everingham et al.].

Top-down approaches generally exploit the acquired class-specific prior knowledge with low-level grouping cues. Most of these approaches use Conditional Random Fields (CRFs) over regions which in standard form have two terms: data term which considers appearance information of each region and a smoothness term which encourages similar neighboring regions to have the same labels. Ladicky et al. 2009 proposed a multilevel hierarchical conditional random field (CRF) model to incorporate information from different scales, which is combined with top-down detectors and global occurrence information. Arbeláez et al. 2012 proposed region based object detectors which combine top-down poselet detector and global appearance cues.

It is worth to mention that semantic segmentation can also be categorized as

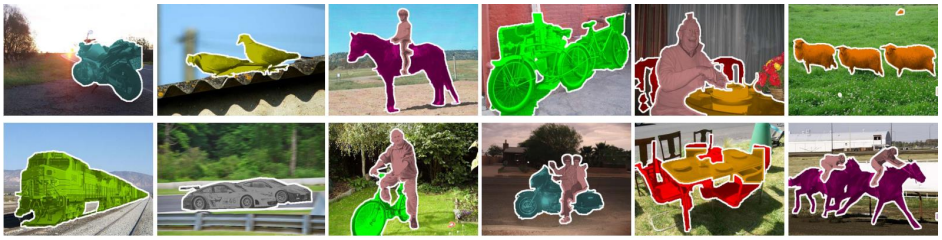


Figure 3.11: Illustration of semantic segmentation on PASCAL VOC 2012 test dateset by the method in [Xia et al.]. Different colors correspond to different object classes and the boundaries are colored in white.

three different types, ones that estimate labels pixel by pixel ([Ladicky et al., 2009, Shotton et al., 2006]), ones that combine features over regions([Carreira et al., 2012b, Zou et al., 2012]), and ones that exploit the information of the whole image ([Gonfaus et al., 2010]).

3.5 Image Segmentation: Dataset

Berkeley Segmentation Dataset (BSD). The public benchmark [Arbelaez et al., 2011] has two versions. One of them called as BSD300, which includes 300 images and its corresponding ground truth data (each image has at least 4 human annotations), is divided into train set which contains 200 images and test set including 100 images. The other is the BSDS500, an extended version of the BSDS300 that includes 200 fresh test images. Each image size is 481×321. Table.3.1 presents some examples from the BSD. Note that each image has multiple ground truths annotated by different human observers.

The **Microsoft Research Cambridge (MSRC) dataset** [Shotton et al., 2006] was first introduced in the context of supervised class segmentation. Soon, this dataset has been widely used to evaluate scene labeling including both image segmentation and multi-class object recognition. The MSRC dataset contains two versions. The MSRC-v0 has 21-classes (3,457 images), and the MSRC-v2 has 21-classes (591 images). The MSRC-v2 has been used as the standard benchmark for evaluating image segmentation [Wang et al., 2013b] as well as co-segmentation and object discovery [Rubinstein et al., 2013], since it has the largest number of categories, and provides clean ground-truth labeling for all objects. Table.3.2 presents some examples from the MSRC-v2, where each color corresponds to a class.

3.6 Image Segmentation: Evaluation

Since tens of thousands of image segmentation algorithms have been proposed, it is essential to compare them using different parameters, in order to understand them on a solid experimental ground. This domain has attracted great interest, and many different evaluation metrics have been developed. Zhang et al. [Zhang et al., 2008a] presented a good survey on this topic. Typically, an evaluation metric allows to

Table 3.1: Berkeley Segmentation Dataset

Image	Human Subjects					Boundary
	#1	#2	#3	#4	#5	

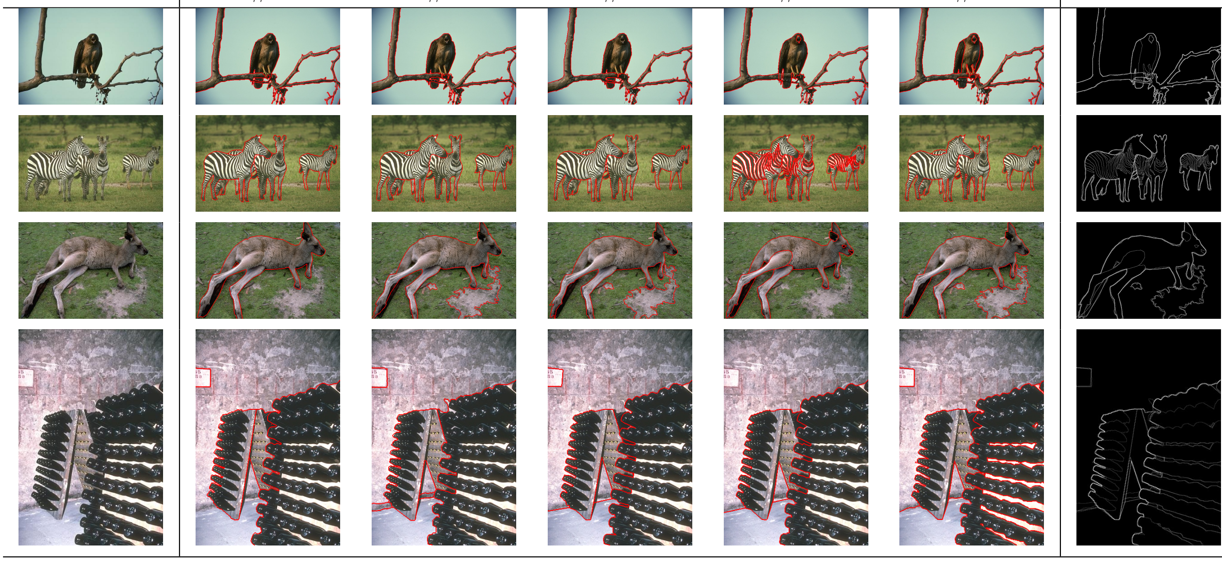


Table 3.2: MSRC-v2 object category image database

grass, tree, sky, flower, water, mount

building, road, sign, aeroplane, sky

cow, sheep,bird, dog, grass, sky...

chair, book, bike, cat

compare the results between a machine algorithm and human observer. The degree of similarity between the human and machine segmented images determines the quality of the segmented image.

From a different perspective, many evaluation algorithms have been proposed. Generally, researchers tend to evaluate the proposed algorithm with multiple different evaluation metrics to fully present its performance. For example, many methods [Rao et al., 2009] [Yining and Manjunath, 2001b] [Li et al., 2012] [Wang et al., 2013b] [Wang et al., 2013a] use the PRI, VoI, GCE and BDE as standard combination to

compare with other standard benchmark algorithms. According to them, a segmentation result is good, at least quantitatively, whenever the comparison with ground truth yields a high value for PRI and small values for the other three indicators. We list several popular evaluation metrics used in image segmentation.

1. The **Probabilistic Rand Index (PRI)** measures the fraction of pixel pairs whose labels are consistent between the segmentation result and the ground truth. In practice, PRI can be computed in a simple form. Let S_{ground} and S_{test} be two clusterings of the same image with different number of clusters, and let n_{ij} be the number of points in the ith cluster of S_{ground} and the jth cluster of S_{test}. N is the total number of pixels of the image. The similarity between the two clusterings is:

$$PR(S_{ground}, S_{test}) = \Big\{ \binom{N}{2} - 1/2\{\sum_i (\sum_j n_{ij})^2 + \sum_j (\sum_i n_{ij})^2 - \sum\sum n_{ij}^2\}\Big\} \Big/ \binom{N}{2} \tag{3.1}$$

   From (3.2), the value of PRI, which measures the similarity of two clusters, ranges from 0 (when there is no intersection at all between S_{ground} and S_{test}) to 1 when the two clusterings are actually the same.

2. The **Volume of Information (VoI)** [Meila, 2005] computes the amount of information loss/gain between the compared images, and can therefore measure the extent to which one image can explain the other, with lower values representing greater similarity. Formally, it is defined as:

$$VI(S_{ground}, S_{test}) = H(S_{ground}) + H(S_{test}) - 2I(S_{ground}, S_{test}) \tag{3.2}$$

   where H and I represent respectively the entropies of and the mutual information between the two clusterings, see [Meila, 2005] for more details.

3. The **Global Consistency Error (GCE)** [Martin et al., 2001]computes the degree to which two segmentations are mutually consistent. Let $R(S_{ground}, p_i) \,\Delta\, R(S_{test}, p_i)$ denote the symmetric difference between $R(S_{ground}, p_i)$ (the subregion of S_{ground} containing the pixel p_i) and $R(S_{test}, p_i)$ (the subregion of S_{test} containing the pixel p_i). Let $|\cdot|$ denote set cardinality. The non symmetric local consistency error is defined as:

$$E(S_{ground}, S_{test}, p_i) = \frac{|R(S_{ground}, p_i) \Delta R(S_{test}, p_i)|}{|R(S_{ground}, p_i)|} \tag{3.3}$$

and the global consistency error is obtained by symmetrization and averaging:

$$GCE(S_{ground}, S_{test}) = \frac{1}{N} \min\{\sum_i E(S_{ground}, S_{test}, p_i), \sum_i E(S_{test}, S_{ground}, p_i)\} \tag{3.4}$$

GCE is valued in $[0, 1]$, where the null value indicates of course that both segmentations are equivalent.

4. The **Boundary Displacement Error (BDE)** [Freixenet et al., 2002] measures the average displacement error of boundary pixels between two segmentation results. More precisely, it defines the error of one boundary pixel as the distance between the pixel and its closest boundary pixel in the other image. Denoting

$$d(p_i, B_2) = \min_{p \in B_2} \|p_i - p\| \tag{3.5}$$

the distance of a boundary point $p_i \in B_1$ to the boundary set B_2, and N_1, N_2 the total number of points in the boundary sets B_1 and B_2, BDE is defined as:

$$BDE(B_1, B_2) = \frac{\sum_i^{N_1} d(p_i, B_2)/N_1 + \sum_i^{N_2} d(p_i, B_1)/N_2}{2} \tag{3.6}$$

A value of BDE close to zero is a good indication that both segmentations are similar.

5. The **Segmentation Covering** [Arbelaez et al., 2009] has been used for the evaluation of the pixel-wise classification task in recognition. The overlap between two regions R and R' is defined as:

$$O(R, R') = \frac{|R \cap R'|}{|R \cup R'|} \tag{3.7}$$

It measures the region-wise covering of the ground truths by a machine segmentation.

6. The **Precision** and **Recall** criteria [Arbelaez et al., 2009] and their weighted harmonic mean **F-measure** evaluate the accuracy of the segmentation algorithms by computing the percentage of matched boundary pixels between the segmented result and the ground-truth image.

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{3.8}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{3.9}$$

$$\mathrm{F} = \frac{2TP}{2TP + FP + FN} \tag{3.10}$$

where TP is short for True Positives, FP means False Positive, and FN represents False Negative.

Chapter 4

A Global/Local Affinity Graph for Image Segmentation

Contents

4.1 Introduction

Image segmentation aims to partition an image into meaningful regions and is a fundamental step for many computer vision tasks, *e.g.*, object recognition [Lee and Grauman, 2010], scene interpretation [Kumar and Koller, 2010], or content-based image retrieval [Belongie et al., 1998]. It proves to be extremely challenging due to the huge diversity and ambiguity of visual grouping patterns in natural scene images, in particular in presence of faint object boundaries and cluttered background (see Fig.4.1(a)). When no restrictive prior is imposed, segmenting an image is an inherently ill-posed task which requires incorporating prior knowledge into the algorithm and keeps attracting many researcher's attention. .

In this work, we are interested in graph-oriented methods which turn the problem of segmenting an image into a problem of partitioning a graph. They prove to be very versatile while providing the ability to encode perceptual grouping laws which play a major role in human visual perception [Felzenszwalb and Huttenlocher, 2004,

Shi and Malik, 2000, Wertheimer, 1938]. However, it is also well known that the quality of the final segmentation result strongly depends on the way the initial graph is built from the input image. Building a graph requires defining its nodes and the relationships between them, *i.e.,* the edges and their weights. However, for graph-based image segmentation, constructing a reliable graph with such requirements faces the following challenges:

1. Numerous feature descriptors have been proposed to differentiate data points. It is actually uneasy to choose the most appropriate descriptor since the relative performances of the various features vary depending on the type of data and are not well known in general;

2. Both local geometrical adjacency and global grouping cues are critical to obtain reliable segmentation results, while their adaptive combination still has to be understood in graph-construction algorithms.

4.1.1 Motivation and contribution

In this work, we propose to construct a sparse and discriminative graph over superpixels to implement not only some obvious perceptual grouping laws, *e.g.*, proximity, similarity, but also enable some others, less straightforward, *e.g.*, continuity, to enter into action for the purpose of perceptual image segmentation. Based on empirical observations, we first postulate a gravitation law over superpixels for their perceptual grouping. Specifically, as can be seen in Fig.4.1 (b)-(f), in dividing broadly superpixels into small, medium and large sized sets, colored in yellow, green and blue in Fig.4.1, respectively, the postulated gravitation law states that:

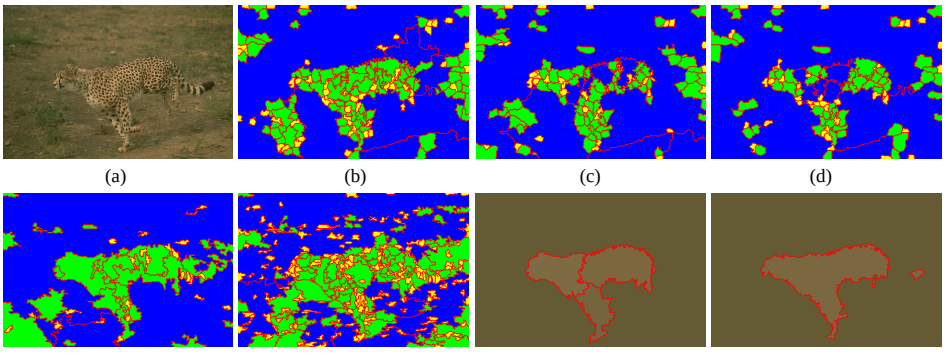


Figure 4.1: Illustration of the gravitation law in perceptual grouping: (a) leopard running on the ground, (b)-(d) are superpixels of 3 different scales by Mean Shift (MS) by oversegmenting (a) using 3 parameter settings, and (e)-(f) superpixels of 2 other different scales by Felzenszwalb-Huttenlocher (FH). Superpixels are divided into small, medium and large sized sets colored in yellow, green and blue, respectively. (g) and (h) are segmented result by SAS and the proposed GL-graph with a number of segments $k = 4$ and $k = 2$ respectively.

1. Small sized superpixels are tiny regions which tend to be perceptually attracted by nearby medium or large sized superpixels while large sized superpixels are

wide regions, *e.g.*, ground regions in blue in Fig.4.1(b), which could span as large as more than half of an image. They are structuring visual patterns that already convey long range information and tend to strongly attract their direct medium and small sized superpixels in perceptual grouping;

2. Medium sized superpixels express long range visual grouping patterns, *e.g.*, skin spots of the leopard in green in Fig.4.1.(b), which need to be captured to further enable propagation of local grouping cues across long range connections;

As a result, we propose to construct an adjacent-graph over small and large sized superpixels to encode the proximity, and adopt our previously proposed sparse ℓ_0 graph over medium sized ones to capture continuity and promote sparsity. As the proposed graph can capture both local and global relationships among data points, we call it a *Global/Local Graph*, or *GL*-graph in short. Furthermore, to enable propagation of grouping cues among superpixels of different scales, we also introduce a bipartite graph which expresses relationships between pixels and superpixels.

Another important perceptual grouping law is similarity of data points within an object which can be characterized by three major perceptual visual features, namely color, texture and shape. According to a few works in psychophysics of human vision [Toni P.Saarela, 2012][M.Peterson and B.Gibson., 1993], these features jointly contribute to perceptual grouping but with different emphases. However, human vision of a scene is perceptual, making use of perceptual laws [Wertheimer, 1938], *e.g.*, similarity, proximity, continuity, *etc.* Given the unsupervised context while dealing with a huge diversity of natural scene images, *e.g.*, indoor and outdoor scenes, landscapes, cityscapes, plants, animals, people, objects, *etc.*, the challenge here is thus to construct a reliable graph fulfilling the aforementioned requirements while encoding some prior knowledge, *e.g.*, perceptual laws. In this work, we implement this paradigm and evaluate the aforementioned three visual features in our *GL* graphs, through mlab and color histogram, Color LBP and SIFT-based codebooks for color, texture and shape, respectively, both individually and their weighted combinations for their effectiveness in unsupervised image segmentation.

The contributions of the proposed approach are threefold:

1. A sparse global/local graph over superpixels of different scales is proposed to capture both short and long range grouping cues of an image, thereby enabling perceptual grouping laws, *e.g.*, proximity, similarity, continuity, to enter into action through a suitable graph cut algorithm. This is achieved in over-segmenting the input image into superpixels at different scales, postulating and implementing a gravitation law which makes use of small and large sized superpixels to encode local smoothness, *e.g.*, proximity, while medium sized superpixels to capture sparse long range grouping cues, *e.g.*, continuity, through ℓ_0 sparsity. A bipartite graph is also introduced to further enable propagation of grouping cues across superpixels of different scales.

2. Using *GL*-graph, we also evaluate three major visual grouping features, namely color, texture and shape, for their discriminating power in perceptual image

segmentation, as well as simple weighted fusion schemes which implement findings from psychophysics which suggest combining color, texture and shape cues with different emphases for perceptual grouping. These evaluations are not only conducted on the proposed *GL*-graph but also on a number of state of the art graph construction methods to shed light on how constructing discriminative graphs with suitable features and their combinations.

3. Extensive experiments are carried out on the Berkeley Segmentation Database (BSD) using 4 different criteria, namely PRI, VoI, GCE and BDE. The experimental results show the effectiveness of the proposed approach, which generate perceptually meaningful partitions and display very competitive objective results in comparison with a number of state of the art algorithms.

Paper Organization. The rest of this paper is organized as follows: in Section II we discuss the basic principles underlying standard graph construction methods in the literature. Section III presents the proposed *GL*-graph in detail and introduces the graph cut method for general image segmentation tasks. In Section IV we carry out extensive experiments on different graphs with different features, and compare the proposed graph with existing graphs as well as other state-of-the-art segmentation methods both visually and quantitatively. Finally, the conclusion is drawn in Section V.

4.2 Proposed Global Local Affinity Graph based on Superpixel and Sparse Representation

In this chapter, we propose a new efficient affinity graph and an unsupervised image segmentation method. An overview of the scheme is shown in Fig.4.2. We start by

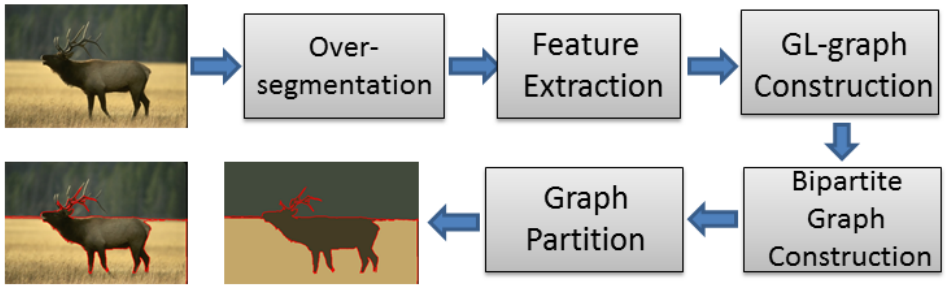


Figure 4.2: The framework of our proposed graph-cut approach for image segmentation

over-segmenting the image and refer to the segments as "superpixels". Then, several feature vectors are extracted from the superpixels by using different types of feature descriptors: color, texture, gradients, localization. The key point is to know how to construct a graph based on these features of different size and type. We propose, for each type of feature descriptor, to calculate the connection information between superpixels by adaptively choosing local and nonlocal neighbors, and by defining the affinities between them using a sparse representation error. The multi-feature

affinity graph is the combination of these graphs. We repeat this graph construction scheme several times for different scales of over-segmentations and concatenate the resulting multi-feature affinity graph into a new multi-feature multi-level affinity graph. Finally, unlike usual unsupervised approaches like normalized cut (Ncut) [Shi and Malik, 2000], the image segmentation problem is solved by computing the partition of the bipartite graph obtained with the unified affinity graph and the association between pixels and superpixels.

4.2.1 Multi-scale Superpixels Generation and Representation

As pointed in [Li et al., 2012], superpixels generated by different methods with varying parameters can capture various and multiscale visual patterns of a natural scene image. By superpixel, we mean here a connected maximal region in a segmented image. As shown in Fig. 4.3, an input image is oversegmented into superpixels of different scales, *e.g.*, 5 scales in the figure, using one or several state of the art segmentation methods, *e.g.* the Mean Shift algorithm (MS) [Comaniciu and Meer, 2002] and the Felzenszwalb-Huttenlocher (FH) graph-based method [Felzenszwalb and Huttenlocher, 2004] in this work. Fig.4.3 shows 5 oversegmentations at 5 different scales using the same parameters as the method referred to as Segmentation by Aggregating Superpixels (SAS) [Li et al., 2012] in the sequel. Then, to obtain a discriminative affinity graph, we compute for each superpixel various visual features. While any kind of region-based feature could be used, we evaluate the discriminating power of three perceptual visual features, namely color, texture and shape, which play a major role in human vision-based segmentation [Toni P.Saarela, 2012][M.Peterson and B.Gibson., 1993]. Specifically, in this work, color feature is characterized using mean value in the L*a*b space ($mLab$) and Color Histogram (CH) in the RGB space, texture through Local Binary Pattern (LBP) while shape cues using SIFT based bag-of-visual-words (BoW) [Lowe, 1999][Cheng et al., 2011a] as shown in Fig. 4.3. Unlike RGB, Lab color space is designed to approximate human vision and its L component closely matches human perception of lightness. Local Binary Patterns (LBP) are reputed to encode micro-texture and robust to monotone light changes.

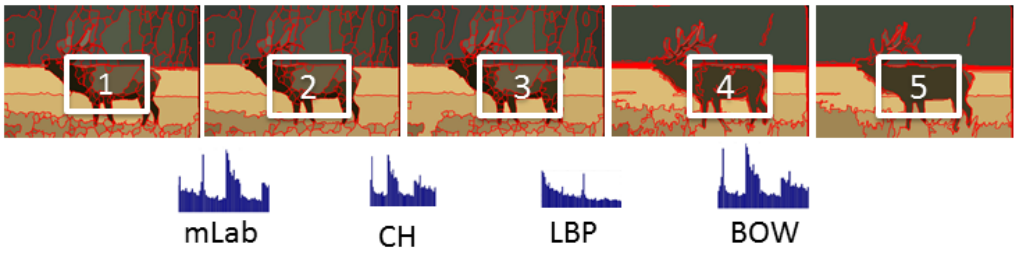


Figure 4.3: Multi-scale superpixels generation and representation with multiple features: each superpixel can be described by feature vectors, as color (mean value in L*a*b, $mlab$, color histogram in RGB (CH), texture (Local Binary Pattern, LBP), and gradient appearance cue (Bag-of-Words) with SIFT.

4.2.2 Global/local Affinity Graph Construction

We postulate the gravitation law from empirical observations on superpixels and broadly divide them into *small*, *medium* and *large* sized sets for their perceptual grouping. *Adjacency*-graph is used for both small and large sized superpixels with respect to their spatial neighbors to capture local smoothness while ℓ_0-graph is applied to medium sized pixels. The final result is a sparse Global/Local graph, namely GL-graph, as illustrated in Fig.(4.5), which implement proximity, long-range continuity and similarity in the same framework.

Specifically, given an input image I, and a collection of superpixels $S_l = \{s_1, s_2, ..., s_N\}$ at a given scale l, a GL-graph is built in a given feature space, *e.g.*, mLab, using the superpixels as graph nodes. Superpixels are divided adaptively into three disjoints sets: *small*, *medium* and *large* sized ones. The *small* sized superpixels can be directly defined using the minimum area parameter involved in the oversegmentation algorithms used for the computation of the superpixels. To decide the *large* sized superpixels, we first sort all the superpixels areas in an ascending order, then we compute the cumulative sum $\mathcal{C}(s_l)$ of the reordered areas. Fig.4.4 illustrates the graph of this cumulative sum for superpixels of 5 different scales. Calculating the second derivative of each curve, we identify its maximal value, and the corresponding area is chosen as threshold value (see the corresponding blue mark on the cumulative graphs in Fig.4.4). This simple procedure seems rather robust in our experiments. Indeed, they depict almost the same performance when the threshold for deciding the large sized superpixels varies in a range close to the inflection identified by the aforementioned procedure through second derivative.

Building a ℓ_0-graph for medium-sized superpixels. Our previously proposed ℓ_0-graph in [Wang et al., 2013a] is applied to *medium*-sized superpixels in order to capture long range grouping cues. More precisely, in context of image segmentation problem, the basic principle is to approximate every data point, *i.e.*, superpixel in a given feature space, as a linear combination of other superpixels of the same image, which are considered as neighbors, and their pairwise similarities or affinities are computed from the corresponding representation error. Formally, such an approximation can be written as:

$$y_i = \mathbf{Y}c_i, \ c_{ii} = 0 \tag{4.1}$$

where $c_i \in \mathbb{R}^n$ is the sparse representation of the data point $y_i \in \mathbb{R}^m$ over the dictionary $\mathbf{Y}$ which is a matrix representation of data points. The constraint $c_{ii} = 0$ prevents the self-representation of y_i.using Eq.(4.1) to approximate every *medium*-sized superpixel from other *medium*-sized ones in a given feature space, *e.g.*, *mLab*.

However, Eq.(4.1) is generally underdetermined and can have an infinite number of solutions whereas we seek to build a sparse image graph in line with the requirements. It turns out that the sparsest solution of Eq.(4.1) measured in the sense of ℓ_0-norm is unique and conveys the most meaningful information of a signal [Elad, 2010].

Formally, this sparsest solution can be written as the following minimization

problem:

$$\min \ ||c_i||_0 \ \ s.t. \ \ y_i = \mathbf{Y}c_i, \ c_{ii} = 0 \tag{4.2}$$

where $\|\cdot\|_0$ denotes the ℓ_0 norm, which counts the number of nonzero values in a vector. c_{ii} denotes the ith elements in the coefficient vector c_i.

However, the problem of finding the sparsest solution of linear equations is NP-hard. Nevertheless, there are many sparse approximation methods, the most two common ones being the ℓ_1-norm approximation and the orthogonal matching pursuit (OMP).

The ℓ_1-norm can be used to approximate the ℓ_0-norm:

$$\min \ ||c_i||_1 \ \ s.t. \ \ y_i = \mathbf{Y}c_i, \ c_{ii} = 0 \tag{4.3}$$

under the condition if the solution sought is sparse enough [Breen, 2009, Wright et al., 2009]. However, within our context of image segmentation using superpixels, such a condition is not necessarily satisfied, given the fact that the number of superpixels given by an oversegmentation is quite limited, *e.g.*, a few hundreds, and even less for *medium* sized superpixels. Furthermore, because of the huge diversity of natural scene images, the dictionary,*i.e.*,the data point representation matrix $\mathbf{Y} = [y_i, ..., y_N] \in \mathbb{R}^{d \times N}$ in Eq.(4.1), could be very unbalanced, for instance with far much more sky superpixels than others, thus missing to be overcomplete for some visual patterns. As a result, we keep to solve Eq.(4.2) using the ℓ_0-norm but make use of orthogonal matching pursuit (OMP) to seek an approximation of the sparsest solution. Experimental results discussed later on in section 4.3 are in line with our analysis and provide further support in favor of our choice of the ℓ_0-sparsity.

OMP is a simple and fast greedy method for approximately solving the ℓ_0-norm sparse formulation through the following optimization problem:

$$\tilde{c}_i = \text{argmin}_{c_i} \left\{ \|y_i - \mathbf{Y}c_i\|_2^2, \ \|c_i\|_0 \leq L, \ c_{ii} = 0 \right\} \tag{4.4}$$

where the parameter L controls the sparsity of the representation. The OMP takes linear time $O(NL)$ with the N representing total number of entries in the dictionary $\mathbf{Y}$, and the L be the maximal number of coefficients for each input data atom y_i.

Once achieved a sparse representation for each data point whose nonzero elements are expected to indicate superpixels from a same object, these superpixels will be considered as graph neighbors of the given data point. The next step of the algorithm is to define the similarity matrix W using the sparse reconstruction error:

$$r_{ij} = \|y_i - c_{ij}y_j\|_2^2. \tag{4.5}$$

c_{ij} denotes the jth elements in the coefficient vector c_i.

The similarity coefficient w_{ij} between superpixels s_i, s_j is defined as

$$w_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 - (r_{ij} + r_{ji})/2 & \text{if } i \neq j. \end{cases} \tag{4.6}$$

Building an adjacency-graph for small and large sized superpixels with

respect to their neighbors. As for the superpixels in the *small-* and *large-*sized sets, every superpixel is connected to all its adjacent superpixels, denoted as *adjacency*-graph. Traditionally, the pairwise similarities are computed with the Gaussian kernel function which is influenced greatly by the choice of the standard deviation σ [Li et al., 2012, Shi and Malik, 2000]. In our combining scheme, it is hard to decide adaptively the value of σ in order to maintain the same order of magnitude with ℓ_0-graph. Therefore, we adopt the same principle as for ℓ_0-graph to compute the similarities: given a superpixel s_i associated with its corresponding feature vector x_i and the matrix-representation $\mathcal{D}$ of all its adjacent neighbors, we try to represent x_i as a linear combination of elements in $\mathcal{D}$. In practice, we solve the following optimization problem:

$$\tilde{c_i} = \text{argmin}_{c_i} ||x_i - \mathbf{D}c_i||_2 \tag{4.7}$$

Once a minimizer $\tilde{c}_i$ has been computed, the similarities between a superpixel and its graph neighbors are computed as in Eq. (4.5) and (4.6).

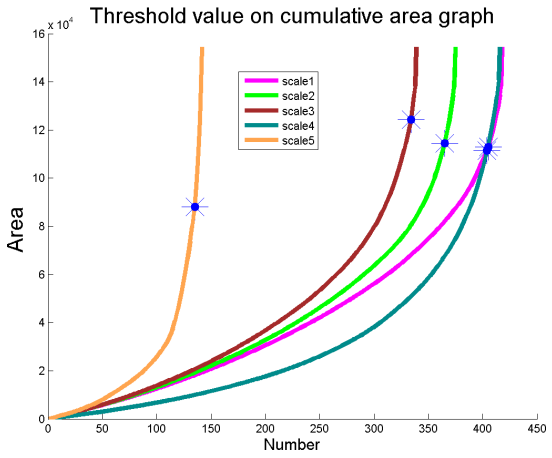


Figure 4.4: Illustration of the adaptive threshold selection of large regions.

4.2.3 Fusing GL-graphs of different visual features and different scales

In summary, for each scale of oversegmented superpixels $S_l = \{s_1, ..., s_N\}$, and its associated feature matrix $[x_1, ..., x_N]$, we construct a GL-graph $\mathcal{G}_l$. In this work, as explained in subsection 4.2.1, we aim to evaluate the effectiveness of three major perceptual visual features, namely color, texture and shape, for their discriminating power, and therefore generate for each of them f_k a similarity matrix W^{f_k}. Furthermore, following [M.Peterson and B.Gibson., 1993, Toni P.Saarela, 2012] which suggest combining color, texture and shape cues with different emphases. Given with m different types of features, we implement a simple weighted sum as in Eq.(4.8) to

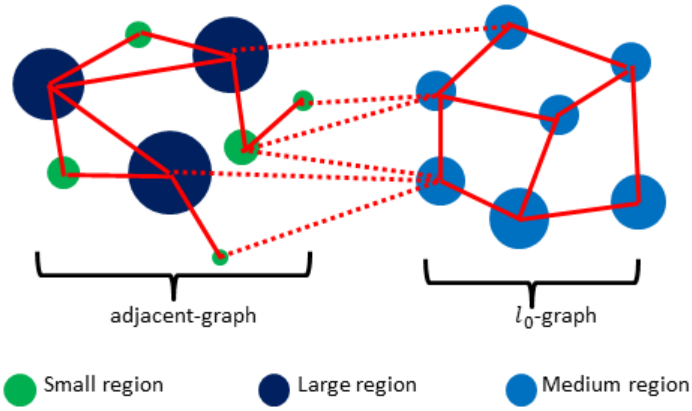


Figure 4.5: Illustration of the *GL*-graph's structure: for each over-segmentations, all the superpixels are divided into three sets: *small* (the green dots), *medium* (the blue dots) and *large* (the ink blue dots) according to their area. Over *small* and *large* sets, all data points will connect to their adjacent neighbors, while over *medium* set, each data point will search its neighbors all over the set. Note that bold red lines represent undirected edges connecting data points within sets, while the dashed red lines describe the edges connecting data points between two different sets.

fuse these similarities into a single affinity matrix.

$$w_{ij} = \sum_{k=1}^{m} (\beta^{f_k} w_{ij}^{f_k}) \tag{4.8}$$

where β^{f_k} is a weight assigned to feature f_k, which controls this feature's importance, and $w_{ij}^{f_k}$ denotes the similarity of superpixels s_i and s_j with feature f_k. For comparison, a baseline fusion scheme as defined in Eq.(4.9) is also used.

$$w_{ij} = \sqrt{\sum_{k=1}^{m} (w_{ij}^{f_k})^2} \tag{4.9}$$

To fuse all scales of superpixels, we plug each scale affinity matrix W_l corresponding to its *GL*-graph $\mathcal{G}_l$ into a block diagonal multiscale affinity matrix W_{ss} like [Cour et al., 2005] as follows:

$$W_{ss} = \begin{pmatrix} W_1 & & 0 \\ & \ddots & \\ 0 & & W_l \end{pmatrix} \tag{4.10}$$

Note that this multiscale affinity matrix of superpixels gathers all the informative intra-scale similarities for grouping. Furthermore, in packing them diagonally, we are ready also to enable propagation of long-range grouping cues across scales, which

is achieved by constructing and diagonalizing a pixel-superpixel graph, or bipartite graph, as introduced in the next subsection.

4.2.4 Bipartite Graph Construction and Partition

To map the relationships between pixels and superpixels and enable propagation of grouping cues across superpixels of different scales, we build a bipartite graph which consists of two parts describing the pixel-superpixel and superpixel-superpixel relationships, respectively. Fig. 4.6 illustrates the structure of such a bipartite graph which encodes the information between pixels and superpixels in blue lines, and the information between superpixels in yellow ones. In particular, taking into account the demand of sparsity for a good-quality graph, pixels are only connected to the superpixels to which they belong. More precisely, let $\mathcal{G}_{\mathcal{B}} = \{\mathcal{U}, \mathcal{V}, B\}$ denote the bipartite graph, where $\mathcal{U} = I \cup S$, $\mathcal{V} = S$, I is the set of pixels and S the set of superpixels. $B = \begin{bmatrix} W_{IS} \\ W_{SS} \end{bmatrix}$, with $W_{IS} = (b_{ij})_{|I|\times|V|}$, and $b_{ij} = \gamma$, if pixel i belongs to superpixel j (in our experiments, we set $\gamma = 10^{-3}$), $b_{ij} = 0$ otherwise. W_{SS} is the affinity graph between superpixels computed in section 3.2. Note that the resultant bipartite graph is highly sparse[1] because of its unbalanced nature. Furthermore, superpixels sharing a large number of pixels are likely to be grouped together, thanks to connections between pixels and their superpixels containing them, thus enabling propagation of grouping cues across scales.

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a similarity matrix W and a number of segments k, various techniques can be applied to group the data points into k different clusters, as cuts [Shi and Malik, 2000], maximum-flow techniques [Goldberg and Tarjan, 1988] and spectral clustering algorithms [Chung, 1997, Ulrike, 2007]. Among these methods, spectral clustering algorithms have been proven successful in many applications, and in particular image segmentation [Shi and Malik, 2000]. They have been in recent years a major trend to achieve clusters from a sparse graph, mainly using representations as linear combinations of eigenvectors of the Laplacian matrix. Basically, spectral clustering consists of partitioning the graph using eigenspaces associated with the following generalized eigen problem [Shi and Malik, 2000]:

$$L\mathbf{f} = \lambda D\mathbf{f}, \tag{4.11}$$

where $L = D - W$ denotes the graph Laplacian, and $D = diag(W\mathbf{1})$ with $\mathbf{1}$ a vector with all components equal to 1. The Lanczos method [Golub and Van Loan, 1996] and the partial SVD [Xiaofeng et al., 2001] can be applied to solve the above

[1] In the bipartite graph, a pixel is connected to only l superpixels for l over-segmentations of an image, we used $l = 5$ *or* 6 for our experiments . For instance, an image named 2092, it is oversegmented into 5 scales with 123, 121, 105, 209 and 53 superpixels, respectively, thus resulting in 611 superpixels in total. The pixel-superpixel graph's size is 154401×611. The total number of nonzero elements in this graph is 154401×5. The percentage of nonzero elements can be viewed as a measurement of sparsity, *i.e.*, $\frac{5}{611} = 0.0081$ Additionally, the unbalanced structure of the constructed bipartite graph $\mathcal{G}_{\mathcal{B}} = \{\mathcal{U}, \mathcal{V}, B\}$ makes the graph further sparser. The final bipartite graph has the size $|\mathcal{U}| = |\mathcal{V}| + |I| = (154401 + 611) > |I| = 481 \times 321 = 154401$.

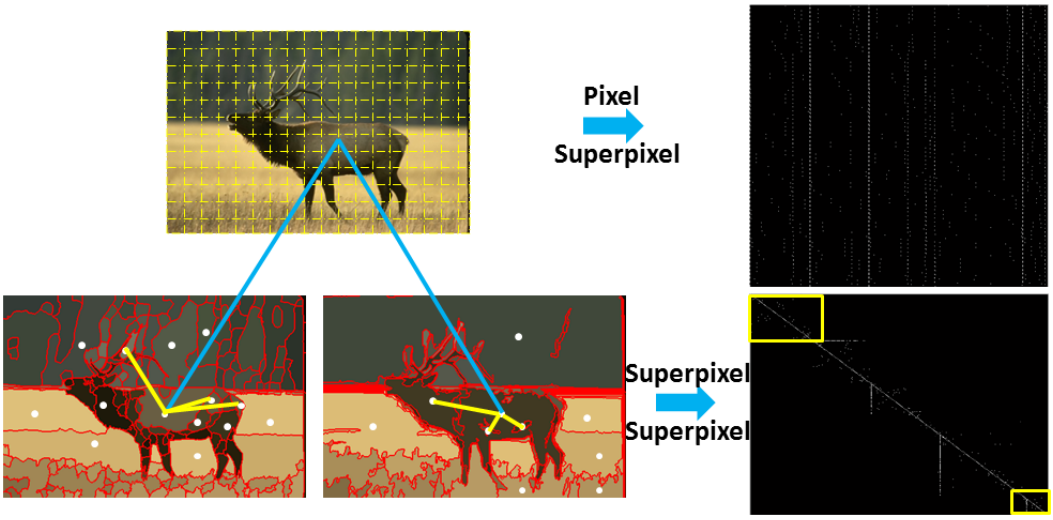


Figure 4.6: Illustration of the construction of an unbalanced bipartite graph over multi-scale over-segmentations: a yellow dot denotes a pixel, and a white dot denotes a superpixel. The blue lines show that each pixel is only connected to its corresponding superpixel in each scale of over-segmentations which is represented as a pixel-superpixel affinity matrix (upper block matrix), while the yellow lines show undirected edges representing the relationships between two superpixels, represented by a superpixel-superpixel affinity matrix (lower block matrix).

eigen problem.

However finding eigenvalues of large matrices is in general computationally demanding, for example, given the bipartite graph $\mathcal{G}_B$, it takes $O(k(|\mathcal{U}|+|\mathcal{V}|)^{3/2})$ [Li et al., 2012] for the Lanczos method and the partial SVD. Note that the bipartite graph in our case is unbalanced, i.e. $|\mathcal{U}| = |\mathcal{V}|+|I|$, and $|I| >> |\mathcal{V}|$ in general, which gives $|\mathcal{U}| >> |\mathcal{V}|$. We use the Transfer Cuts method [Li et al., 2012] which has been proposed to solve efficiently the unbalanced bipartite graph partitioning problem. Interestingly, Transfer Cuts solve a problem which has similar form as (4.11), but holds on a much smaller graph over superpixels only

$$L_{\mathcal{V}}\mathbf{f} = \lambda D_V \mathbf{f}, \tag{4.12}$$

where $L_{\mathcal{V}} = D_{\mathcal{V}} - W_{\mathcal{V}}$, $D_{\mathcal{V}} = diag(B^\top \mathbf{1})$, and $W_{\mathcal{V}} = B^\top D_{\mathcal{U}}^{-1} B$, $D_{\mathcal{U}} = diag(B\mathbf{1})$. Note that solving (14) takes linear time $O(k|\mathcal{V}|^{2/3})$ with a small constant.

4.3 Experiment and Analysis

All experiments are carried out on the Berkeley Segmentation Database (BSD) [Arbelaez et al., 2011], which includes 300 images and the corresponding ground truth data (each image has at least 4 human annotations). It is divided into a training set which contains 200 images and a test set including 100 images. Each

image's size is 481×321. Four standard measurements are used for quantitative evaluation: the Probabilistic Rand Index (PRI) [Unnikrishnan et al., 2007], the Variation of Information (VoI) [Meila, 2005], the Global Consistency Error (GCE) [Martin et al., 2001], and the Boundary Displacement Error (BDE) [Freixenet et al., 2002].

4.3.1 Experiments Setup

Using the framework depicted in Fig.4.2, the proposed GL-graph is first evaluated through a single visual feature in comparison with several state of the art graphs. It is then further evaluated when fusing visual features as proposed by psychophysicists and several global graphs to capture different grouping cues. This means that only the GL-graph construction is evaluated and compared while keeping the same all the other steps, *e.g.*, over-segmentation, feature extraction, bipartite graph construction and graph partition using spectral clustering. Please refer to section 4.2 for further details.

The state of the art graphs studied are the *adjacent*-graph as in SAS [2][Li et al., 2012] and four popular global graphs, namely KNN-graph [3] [Grady, 2004], ℓ_1-graph[4] [Elhamifar and Vidal, 2013], LRR-graph (Low Rank Representation)[5] [Liu et al., 2013a], and ℓ_0-graph [Wang et al., 2013a].

For each method, we tuned the parameters to achieve the best performance:

1. *adjacent*-graph: the standard deviation of the Gaussian kernel function is defined as $\sigma = 20$;

2. KNN-graph: we adopt Euclidean distance as the similarity metric, and use Gaussian kernel function to compute the weights of edges, with $\sigma = 20$ as [Li et al., 2012]. Various numbers of neighbors are tested;

3. ℓ_1-graph: we construct the graph following the method in [Elhamifar and Vidal, 2013]. Since the affinity matrix is asymmetric, we replace it with $\tilde{W} = (|W| + |W|^T)/2$;

4. ℓ_0-graph: we derive the graph and symmetrize it following [Wang et al., 2013a]. Parameters are also the same;

5. LRR-graph: we construct the LRR-graph and symmetrize it as for the ℓ_1-graph following [Liu et al., 2013a]. We set the balance parameter $\lambda = 0.18$;

6. GL-graph: For our GL-graph, the threshold value for defining small regions is empirically set to 300 pixels and the threshold for large regions is decided adaptively following the discussion in Section 3.2. Performances with various L are presented.

[2]http://www.ee.columbia.edu/ln/dvmm/SuperPixelSeg/
[3]http://cns.bu.edu/ lgrady/software.html
[4]http://www.cis.jhu.edu/ ehsan/
[5]https://sites.google.com/site/guangcanliu/

As explained in subsection 4.2.1, following the findings of psychophysicists, we evaluate three major perceptual visual features, namely color using the mean value of a superpixel in L*a*b denoted as $mLab \in \mathbb{R}^3$ or RGB color histogram denoted as $CH \in \mathbb{R}^{256}$, texture through Uniform Color Local Binary Pattern[6] [Zhu et al., 2010] denoted as $CLBP^{u2} \in \mathbb{R}^{177}$, and shape cues using the Bag-of-Visual-Words (BoW). In the experiments, we compute the scale invariant feature transform (SIFT)[7][Lowe, 1999] at each pixel and then perform the vector quantization by fast K-means[8] to construct the visual vocabulary. The number of clustering centers is 100, 150, 200 and 300, denoted as $BoW100$, $BoW150$, $BoW200$, $BoW300$, respectively.

4.3.2 Experimental results using single visual feature

Graph's performances are closely related to neighborhood's topology and to features' choice. This experiment aims to compare the quality of the proposed GL-graph with 5 other state of art graph constructions and highlights the discriminating power of each visual feature. Table 4.1 tabulates the performance of the 6 tested graph construction methods over each visual feature. First we present the average score on each number of segments over the BSD test set I and II (Fig. 4.7 and 4.8). In detail, for each feature, we use the four evaluation metrics (PRI, VoI, GCE, and BDE) to compare different graphs: *adjacent*-graph, KNN-graph, ℓ_1-graph, LRR-graph, ℓ_0-graph, and GL-graph. Together with the feature performance, we can make the following observations:

1. The *adjacent* graph is more sensitive than global graphs to the number of segments k and to feature selection. In particular, its performance varies greatly with the feature $mLab$. However, the additional use of BoW makes it less sensitive to k. In addition, on average, $mLab$ performs better as k increases. As a quantitative example, when k=2, the evaluation scores on the BSD test set for $mLab$ are: PRI=0.6097, VoI=2.0647, GCE=0.1211, BDE=41.2471; and using $BoW100$ one gets PRI=0.7705, VoI=2.3657, GCE= 0.2047, BDE=16.0532. The reason may lie on the feature's dimension and on the use of global information. Indeed, $BoW100$ is encoded from SIFT with K-means into a 100-dimensional histogram. Generally the sensitivity to k decreases as the dimension increases;

2. The performances of global graphs are essentially stable with respect to k and such stability is invariant to feature selection. This property is mainly due to the fact that global graphs choose each node's neighbors by searching globally, which enables the constructed graph to capture long-range grouping cue. It is worth mentioning that such property of global graph makes it promising for practical applications in object recognition, image annotation, etc. Note that the family of sparsity-based graphs (e.g. ℓ_0-graph) has better performance than rank minimization graph (LRR-graph) or ℓ_1-graph. The reason is that in a ℓ_0-graph, each node has very few neighbors, which makes the graph much

[6] http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab
[7] http://www.vlfeat.org/
[8] https://gforge.inria.fr/frs/?group_id=2151

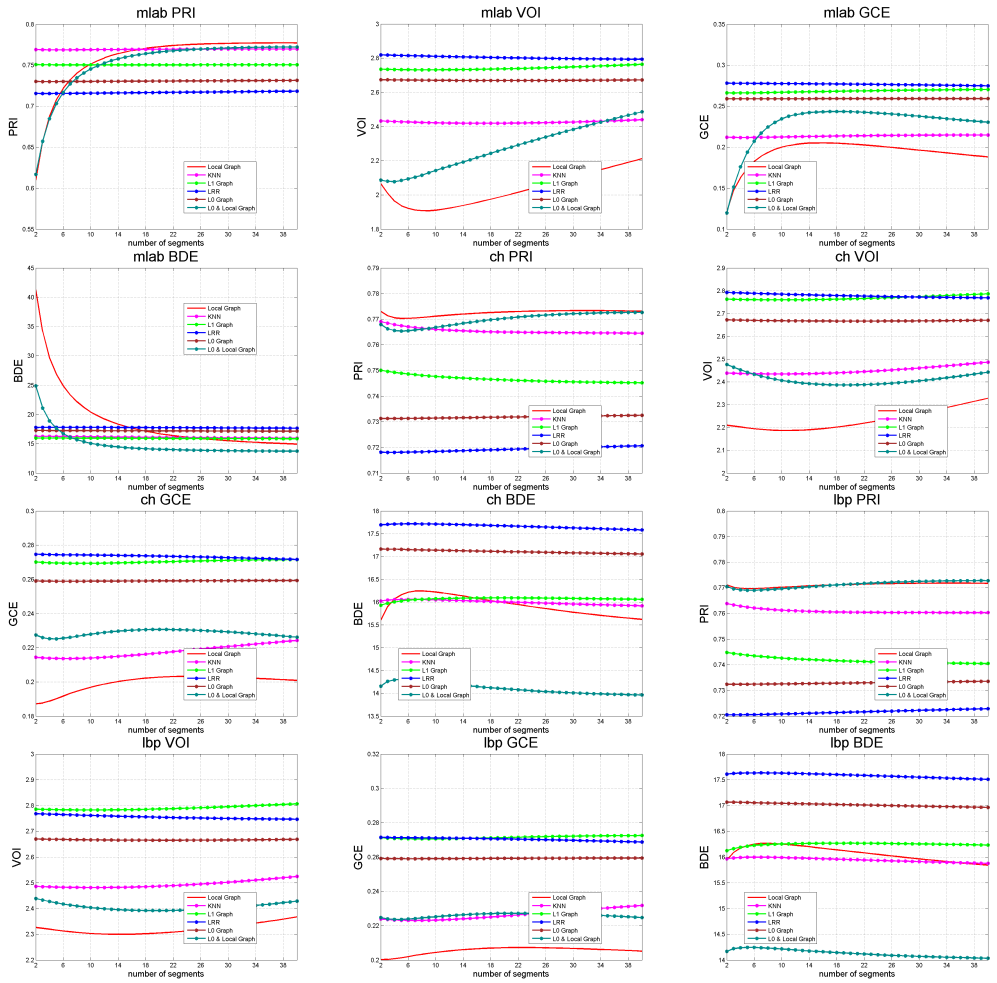


Figure 4.7: The performance comparison of the GL-graph with other graphs on different features ($mlab$, CH and LBP) on Berkeley Segmentation Database I: for each feature, graphs are compared by the average score over each number of segments from 2 to 40 by the four metrics PRI, VoI, GCE and BDE simultaneously.

sparser compared with ℓ_1-graph and LRR-graph, see Table 4.1 where scores with various values of the parameter L are reported;

3. The proposed GL-graph combines local graph and ℓ_0-graph's nice properties. It achieves the best performances in Table 4.1 in comparison with the *adjacency*-graph and the ℓ_0-graph. As shown in Table 4.2, it is however somewhat sensitive to the parameter L.

As for the evaluation of the respective performances of features, it is essentially an open problem. In what follows, we shed light on the performance of seven features on different graphs by comparing the *best* results according to the maximization of the evaluation measurement PRI shown in Table 4.1, from which we can deduce that:

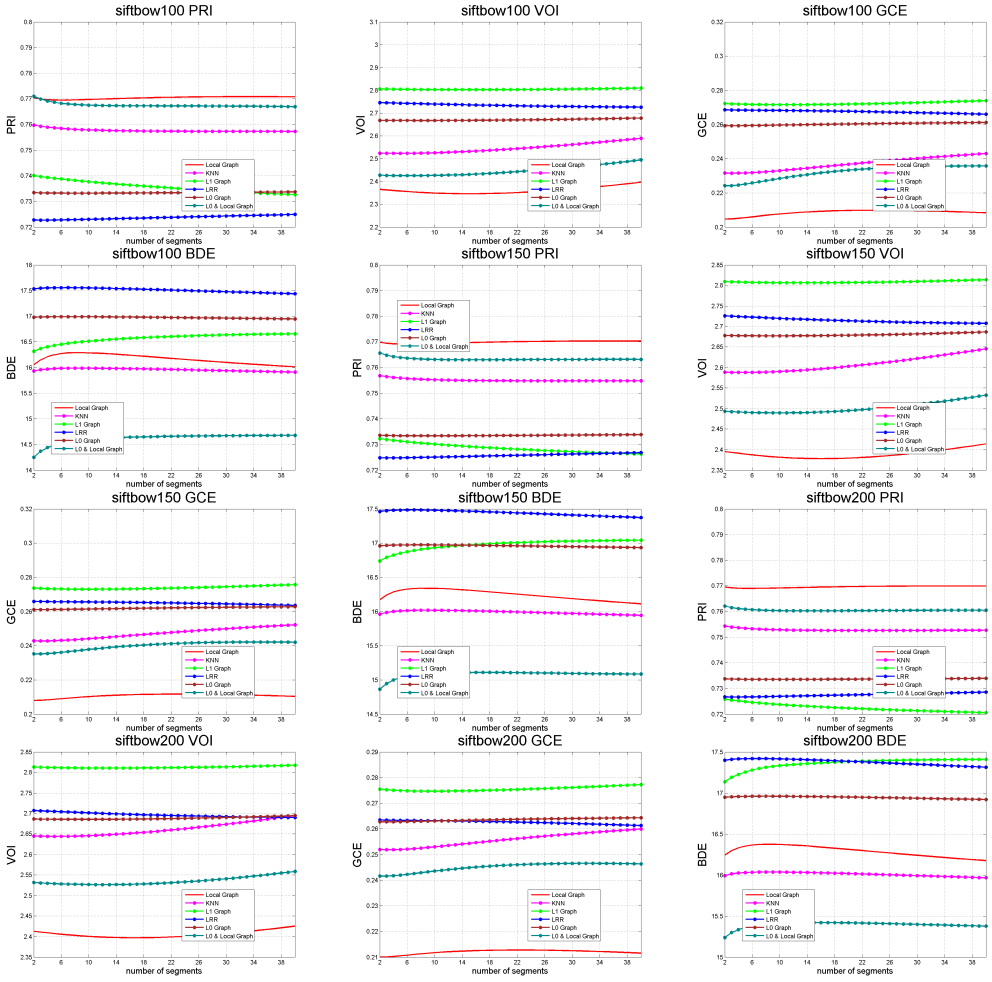


Figure 4.8: The performance comparison of the *GL*-graph with other graphs on different features on Berkeley Segmentation Database II: for each feature, graphs are compared by the average score over each number of segments from 2 to 40 by the four metrics PRI, VoI, GCE and BDE simultaneously.

1. color is a faithful cue for almost all graphs except the LRR-graph and GL-graph on which LBP has equivalent performance with color. More specially, choosing appropriate color space can boost the performance, such as $mLab$ for KNN-graph and ℓ_1-graph;

2. CH and LBP perform almost the same for all kinds of graph;

3. BoW's performance does not vary greatly with respect to the number of centers. Note that LRR-graph's performance is invariant to the feature selection according to Table 4.1.

Remark that these findings are in perfect accordance with those of psychophysicists on human vision-based segmentation which suggest that appearance grouping cues, *i.e.*, color and texture, outweight shape-based ones [M.Peterson and B.Gibson., 1993] while human vision makes joint use of color and texture for image segmentation but with asymmetric role in favor of color [Toni P.Saarela, 2012]. These findings will be fully explored in the fusion scheme as explained in subsections 4.3.3.

Table 4.1: Performance validation for the proposed GL-graph and other types of graphs, using various features, on the Berkeley Segmentation Database test set. Four metrics are used: PRI, VoI, GCE and BDE. For each graph, the best performance over features is highlighted. Note that the best performance result is computed by maximizing PRI of each image over all its evaluation results ranging from 2 to 40.

adjacent-graph	PRI↑	VoI↓	GCE↓	BDE↓	KNN-graph	PRI↑	VoI↓	GCE↓	BDE↓
mlab	**0.8264**	**1.7537**	**0.1935**	**12.7985**	**mlab**	**0.8290**	**2.0732**	**0.2316**	**12.1872**
CH	0.8133	1.9811	0.2204	13.9598	CH	0.8016	2.7882	0.3229	14.4206
$CLBP^{u2}$	0.8133	1.9811	0.2204	13.9598	$CLBP^{u2}$	0.8016	2.7882	0.3229	14.4206
BoW100	0.8106	1.9983	0.2301	14.7859	BoW100	0.7862	3.2387	0.3440	16.1826
BoW150	0.8112	2.0210	0.2302	14.9699	BoW150	0.7891	3.1858	0.3385	16.2013
BoW200	0.8104	2.0179	0.2286	14.7858	BoW200	0.7899	3.2239	0.3402	15.3621
BoW300	0.8113	1.9954	0.2285	14.9503	BoW300	0.7871	3.2063	0.3383	16.1223
ℓ_1-graph	PRI↑	VoI↓	GCE↓	BDE↓	LRR-graph	PRI↑	VoI↓	GCE↓	BDE↓
mlab	**0.8036**	**2.9053**	**0.3079**	**12.7745**	mlab	0.8155	1.8788	0.2071	13.7015
CH	0.7710	2.8919	0.3012	13.5910	**CH**	**0.8153**	**1.8794**	**0.2068**	**13.6949**
$CLBP^{u2}$	0.7710	2.8919	0.3012	13.5910	$CLBP^{u2}$	**0.8153**	**1.8794**	**0.2068**	**13.6949**
BoW100	0.6963	2.9473	0.3691	19.1577	BoW100	0.8148	1.8809	0.2072	13.6680
BoW150	0.7009	2.9678	0.3695	19.6824	BoW150	0.8146	1.8864	0.2084	13.7504
BoW200	0.7046	2.9428	0.3702	24.5510	BoW200	0.8140	1.8838	0.2083	13.7732
BoW300	0.7096	2.8871	0.3495	23.2067	BoW300	0.8147	1.8901	0.2078	13.6894
ℓ_0-graph	PRI↑	VoI↓	GCE↓	BDE↓	GL-graph	PRI↑	VoI↓	GCE↓	BDE↓
mlab	0.8141	2.2969	0.2470	12.2632	mlab	0.8230	2.0848	0.2260	11.7124
CH	**0.8185**	**2.2426**	**0.2622**	**12.8445**	**CH**	**0.8266**	**1.9585**	**0.2204**	**12.0042**
$CLBP^{u2}$	0.8152	1.8793	0.2068	13.6948	$CLBP^{u2}$	**0.8266**	**1.9584**	**0.2204**	**12.0043**
BoW100	0.7896	2.7465	0.3057	15.7107	BoW100	0.7970	2.4072	0.2545	15.2672
BoW150	0.7878	2.7624	0.2994	15.5692	BoW150	0.7959	2.4067	0.2542	14.7353
BoW200	0.7859	2.7847	0.3050	15.2595	BoW200	0.7991	2.3744	0.2521	15.1711
BoW300	0.7872	2.7346	0.2968	15.1443	BoW300	0.7997	2.3612	0.2502	15.4163

Obtaining visually meaningful results requires inevitably the careful tuning of the number of segments k. We show in Fig. 4.9 the different performances of the graphs for various values of k and the following observations can be made: 1) the

Table 4.2: Quantitative scores for different values of the parameter L for the GL-graph over the Berkeley Segmentation Database test set.

Sparsity (CH)	PRI↑	VoI↓	GCE↓	BDE↓
L=2	0.8213	2.1111	0.2453	13.2554
L=3	0.8185	2.2426	0.2622	12.8445
L=4	0.8195	2.2958	0.2645	12.4510
L=5	0.8177	2.3079	0.2622	12.2648
L=6	0.8185	2.3086	0.2631	12.8950
L=7	0.8190	2.2913	0.2624	12.8970
L=8	0.8185	2.3253	0.2667	12.1963

adjacent-graph considers only the local structure of image, which leads to wrong segmentations (see the results segmented in first row for each image) when the objects cover a large part of the image. 2) the ℓ_1-graph tends to oversegment the image (see third rows for every example in Fig. 4.9), due to its high sensitivity to noise and outliers, which is a convenient skill for face recognition [Wright et al., 2009], but not for image segmentation; 3) unlike the graph based on sparse minimization, which finds the sparse representation of every point, the LRR-graph finds a global lowest rank representation, therefore further enforces the global structure over the data points. However, as pointed in [Zhuang et al., 2012], LRR-graph often produces a dense graph which fails to meet the demand of sparsity for a desirable graph.

4.3.3 Results on fusing different graphs and visual features

The experimental results shown in subsection 4.3.2 in perfect accordance with the findings of psychophysicists on human vision-based segmentation [M.Peterson and B.Gibson., 1993] strengthen the simple weighted sum fusion scheme as defined in Eq.(4.8) in subsection 4.2.3 [Toni P.Saarela, 2012] which enables combining color, texture and shape cues with different emphases. Specifically, following both the findings of psychophysics and the experimental results shown in subsection 4.3.2, we empirically implement several fusion schemes, namely fusion schemes combining color and texture features as well as those combining color, texture and shape at the same time. When color and texture cues are jointly used, more weight is given to color-based affinities than those of texture-based one; When all the three visual features are used at the same time, shape receives less weight in comparison with color and texture. As a baseline, we also implement the baseline fusion scheme as defined in Eq.(4.9) which gives an equal weight to each kind of visual grouping cues. As can be seen from the Table 4.3, very competitive results are achieved by the proposed GL-graph when fusing color, texture and shape with different emphases.

We showed in the previous section that that different graphs capture different affinities between superpixels. Given a visual feature, *e.g.*, color $mLab$, fusion can also be carried out at the graph level in combining the proposed GL-graph with other ones, *e.g.*, ℓ_1, KNN, LRR, and in averaging their affinities. Specifically, the segmentation framework as defined in section 4.2 is kept the same, the fusion

Table 4.3: Quantitative performance of the proposed method (GL-graph) with simple weighted sum fusion scheme.

Methods	PRI↑	VoI↓	GCE↓	BDE↓
$\sqrt{(LBP^2 + mlab^2 + SIFT^2)}$	0.8332	1.8890	0.1998	10.7904
$\sqrt{(LBP^2 + CH^2 + SIFT^2)}$	0.8355	1.8716	0.2048	10.9985
(0.4LBP+0.6mlab)	0.8355	1.8965	0.1765	10.9157
(0.4LBP+0.6CH)	0.8363	**1.6776**	**0.1727**	11.0456
(0.4LBP+0.4mlab+0.2SIFT)	0.8368	1.8347	**0.1706**	10.8552
(0.4LBP+0.4CH+0.2SIFT)	0.8381	1.8753	0.1741	**10.6787**
(0.3LBP+0.5mlab+0.2SIFT)	**0.8384**	1.8012	0.1934	**10.6633**
(0.3LBP+0.5CH+0.2SIFT)	**0.8383**	**1.7927**	0.1958	11.4088

only takes place at the graph level of medium sized superpixels. Table 4.4 reports the experimental results of such graph level fusion schemes. As can be seen from Table 4.4, when the input image is simply segmented into two clusters ($k = 2$), the baseline, *i.e.*, the proposed GL-graph, outperforms all three combinations. However, when the number of clusters is increased, *i.e.*, $k = 10, 30, 40$, all three combinations outperform the baseline GL-graph. These results suggest that, when the number of clusters is increased, new connections are brought in by other global graphs, *i.e.*, ℓ_1, KNN, LRR, definitively contribute to improve the segmentation result. Furthermore, both KNN and LRR graphs prove to bring more complementary information with respect to the GL-graph than the ℓ_1 graph.

4.3.4 Comparison with state-of-the-art algorithms

Our work follows a similar, yet not identical, strategy as the SAS algorithm [Li et al., 2012], i.e., building a bipartite graph over multiple superpixels and pixels, then using Tcuts for image segmentation. The main difference between both methods is the affinity graph construction. In SAS, adjacent neighborhoods of superpixels are used, and the pairwise superpixel similarity is computed by the Gaussian weighted Euclidean distance in the color feature space. In our method, we build a GL-graph combining classical spatial homogeneity of objects and long range clustering based on sparse representation over multiple types of features and multi-scale superpixels, making the constructed graph having the characteristics of a long range neighborhood topology, yet with sparsity and high discriminative power. Fig.4.10 shows various segmentation results obtained with either the SAS method (second image of each experiment), or with our algorithm (third image). Notice that the results of SAS are the best results reported by the authors, and require a careful tuning of the number of segments k (e.g. for *starfish*, *owl* and *leopard*, $k = 11, 4, 5$ respectively). For our method that takes into account the global information, a desirable result can be usually achieved with either k=2, 3, or 4 (e.g. for *starfish*, *owl* and *leopard*, $k = 2$). Especially, compared with SAS, our method achieves a correct segmentation even in the difficult cases where: 1) The detected object is highly textured (this is for instance the case of *starfish*, *moray eel*, *leopard*, and *owl*), and the background may be highly unstructured. In the particularly difficult case of the *owl* image,

Table 4.4: Quantitative comparison of different combinations of two global graphs, associating with *adjacent*-graph over the Berkeley Segmentation Database.

Combinations (mlab)	PRI↑	VoI↓	GCE↓	BDE↓
$k = 2$				
baseline: GL-graph	0.6205	2.0445	0.1240	25.0000
$adjacency + \ell_0 + KNN$	0.5646	2.0936	0.0960	43.7168
$adjacency + \ell_0 + \ell_1$	0.5276	2.1655	0.1001	47.4737
$adjacency + \ell_0 + LRR$	0.5732	2.1191	0.1138	43.4317
$k = 10$				
baseline: GL-graph	0.7456	2.1730	0.2381	15.0301
$adjacency + \ell_0 + KNN$	0.7851	1.9744	0.2290	14.5649
$adjacency + \ell_0 + \ell_1$	0.7518	2.0892	0.2404	16.8827
$adjacency + \ell_0 + LRR$	0.7892	1.9773	0.2306	14.7932
$k = 30$				
baseline: GL-graph	0.7703	2.3802	0.2350	13.5401
$adjacency + \ell_0 + KNN$	0.7968	2.3235	0.1988	12.8590
$adjacency + \ell_0 + \ell_1$	0.7900	2.3705	0.2166	13.3426
$adjacency + \ell_0 + LRR$	0.7964	2.3166	0.1904	12.8149
$k = 40$				
baseline: GL-graph	0.7752	2.5688	0.2301	13.5003
$adjacency + \ell_0 + KNN$	0.7957	2.4623	0.1845	12.8569
$adjacency + \ell_0 + \ell_1$	0.7911	2.4922	0.2005	13.2135
$adjacency + \ell_0 + LRR$	0.7951	2.4603	0.1743	12.8511

our method segments it correctly while the segmentation provided by SAS is not meaningful; 2) The object and its surrounding are quite similar in color or texture (*river otter*, *leopard* and *bird*). For example, the SAS algorithm oversegments the river otter and the leopard into several parts, while our method yields a correct segmentation. 3) Objects of the same type appear in a large, possibly disconnected, region of the image, as for instance the bottles or the mountain. SAS is not competitive with our method for long-range grouping, hence it tends to split the object into different parts (e.g. the bottles into 4 parts and the mountain into 4 parts). On the contrary, our proposed method can derive the right partition.

We also report quantitative comparison with SAS and other standard benchmarks: Ncut [Shi and Malik, 2000], JSEG [Yining and Manjunath, 2001b], Multiscale Ncut (MNcut) [Cour et al., 2005], Normalized Tree Partitioning (NTP) [Wang et al., 2008a], Saliency Driven Total Variation (SDTV) [Donoser et al., 2009], Texture and Boundary Encoding-based Segmentation (TBES) [Rao et al., 2009], Ultrasound Contour Map (UCM) [Arbelaez et al., 2011], Learning Full Pairwise Affinity (LFPA) [Kim et al., 2010a], SAS [Li et al., 2012], Context-sensitive [Bai et al., 2010], Co-transduction [Bai et al., 2012], Tensor Product Graph (TPG) [Yang et al., 2013], and Fusion with TPG [Zhou et al., 2012]. The results are shown in Table 4.5, where we highlight in bold the best two results for each qualitative criterion.

Most of the average scores of the benchmark methods are collected from [Li

Table 4.5: Quantitative comparison of the proposed method (*GL*-graph) with state-of-the-art methods over the Berkeley Segmentation Database.

Methods	PRI↑	VoI↓	GCE↓	BDE↓
NCut [Shi and Malik, 2000]	0.7242	2.9061	0.2232	17.15
JSEG [Yining and Manjunath, 2001b]	0.7756	2.3217	0.1989	14.40
MNCut [Cour et al., 2005]	0.7559	2.4701	0.1925	15.10
NTP [Wang et al., 2008a]	0.7521	2.4954	0.2373	16.30
TBES [Rao et al., 2009]	0.8000	1.7600	N/A	N/A
UCM [Arbelaez et al., 2011]	0.8100	**1.6800**	N/A	N/A
SDTV [Donoser et al., 2009]	0.7758	1.8165	0.1768	16.24
LFPA [Kim et al., 2010a]	0.8146	1.8545	0.1809	12.21
Context-sensitive (mlab) [Bai et al., 2010]	0.7937	3.9174	0.4165	**9.9046**
Cotransduction $(mlab + LBP)$ [Bai et al., 2012]	0.8083	2.3644	0.2681	14.1972
TPG [Yang et al., 2013]	0.8227	1.7696	N/A	N/A
FusionTP [Zhou et al., 2012]	0.7771	3.3089	0.3654	13.2428
SAS [Li et al., 2012]	0.8319	1.6849	0.1779	11.29
ℓ_0-graph [Wang et al., 2013a]	0.8355	1.9935	0.2297	11.1955
GL-graph	**0.8384**	1.8012	0.1934	**10.6633**

et al., 2012], [Kim et al., 2010a] and [Yang et al., 2013], with exception of [Bai et al., 2010, 2012, Zhou et al., 2012], the graphs proposed in which are for task of shape retrieval and visual tracking. Nevertheless, we compare with their graph construction methods, by only replacing the *GL*-graph in our segmentation framework, while keeping other settings such as multi-scale superpixels and bipartite graph structure the same, for the sake of fairness. From Table 4.5, we can observe that only with one feature mlab, the context-sensitive graph [Bai et al., 2010] has very promising performance. It is worth to mention that the graph construction techniques proposed in [Bai et al., 2010, 2012, Yang et al., 2013, Zhou et al., 2012], are with very high computational cost and even hardly acceptable for the bottom-up segmentation, which is usually pre-process for high-level computer vision task, e.g. object recognition and detection. We can see that our method ranks first for PRI, VoI, GCE, and BDE, in particular the gain is significant for PRI and BDE.

Additionally, to demonstrate the advantage of our algorithm in practical applications, we present visual segmentation results of our method with k =2. As can be seen in Fig. 4.11, our method tends to first segment the most salient objects in the image even in the following cases where: 1) the detected object is tiny (see the *aeroplane*, the *boat*); 2) multiple objects are needed to segment in the same image (as in both middle rows); 3) the color of background and object are quite similar (see the last row).

4.3.5 Algorithm time complexity

The framework of the proposed algorithm as depicted in Fig.4.2 includes steps of oversegmentation, feature extraction, GL-graph and bipartite graph construction and graph partitioning. They are all coded as Matlab routines. The time complexity of OMP for GL-graph construction is analyzed in section 4.2.2. The time complexity of graph partitioning using Transfer cut is analyzed in section 4.2.4. Using a standard computer (Intel Core (TM) 2.3GHz CPU with 16G memory) to segment an image from BSD, *e.g.*, *"2092.jpg"*, generating multi-scale superpixels with MS and FH takes 4.68 seconds, extracting all the visual features listed in Section 4.2.1 takes 1872.23 seconds, building the bipartite graph with a single feature requires 2.12 seconds, of which the superpixel graph constructed by the proposed GL-graph only lasts 0.12 seconds for a graph with size 123×123, 0.11 seconds for 121×121, 0.13 seconds for 105×105, 0.39 seconds for 209×209, and 0.03 seconds for 53×53; cutting the bipartite graph into 11 clusters, the computational time is 0.87 seconds.

4.4 Conclusion

In this chapter, we introduced a sparse global/local graph which encodes in a sparse way the perceptual grouping laws, e.g., proximity, similarity, and continuity. Unlike classical methods, our GL-graph is able to encode adaptively both local and global homogeneity of an object via fusing two types of graphs: the *adjacent*-graph and the sparse ℓ_0-minimization based graph built separately on three different classes of superpixels, i.e. enforcing proximity and similarity over small and large sized superpixels and encoding long range similarity on medium sized ones. Moreover, the discriminative power of the GL-graph is further enhanced by fusing several different features over multi-scale superpixels. The derived GL-graph is plugged into an efficient graph-cut method for unsupervised image segmentation. Extensive validations on the BSD data set show that our method yields very competitive qualitative and quantitative segmentation results compared to state-of-the-art methods.

As future extension of our work, it could be interesting to be able to learn an optimal fusion scheme that combines color, texture and shape cues using training data. That would be an interesting step toward semi-supervised image segmentation.

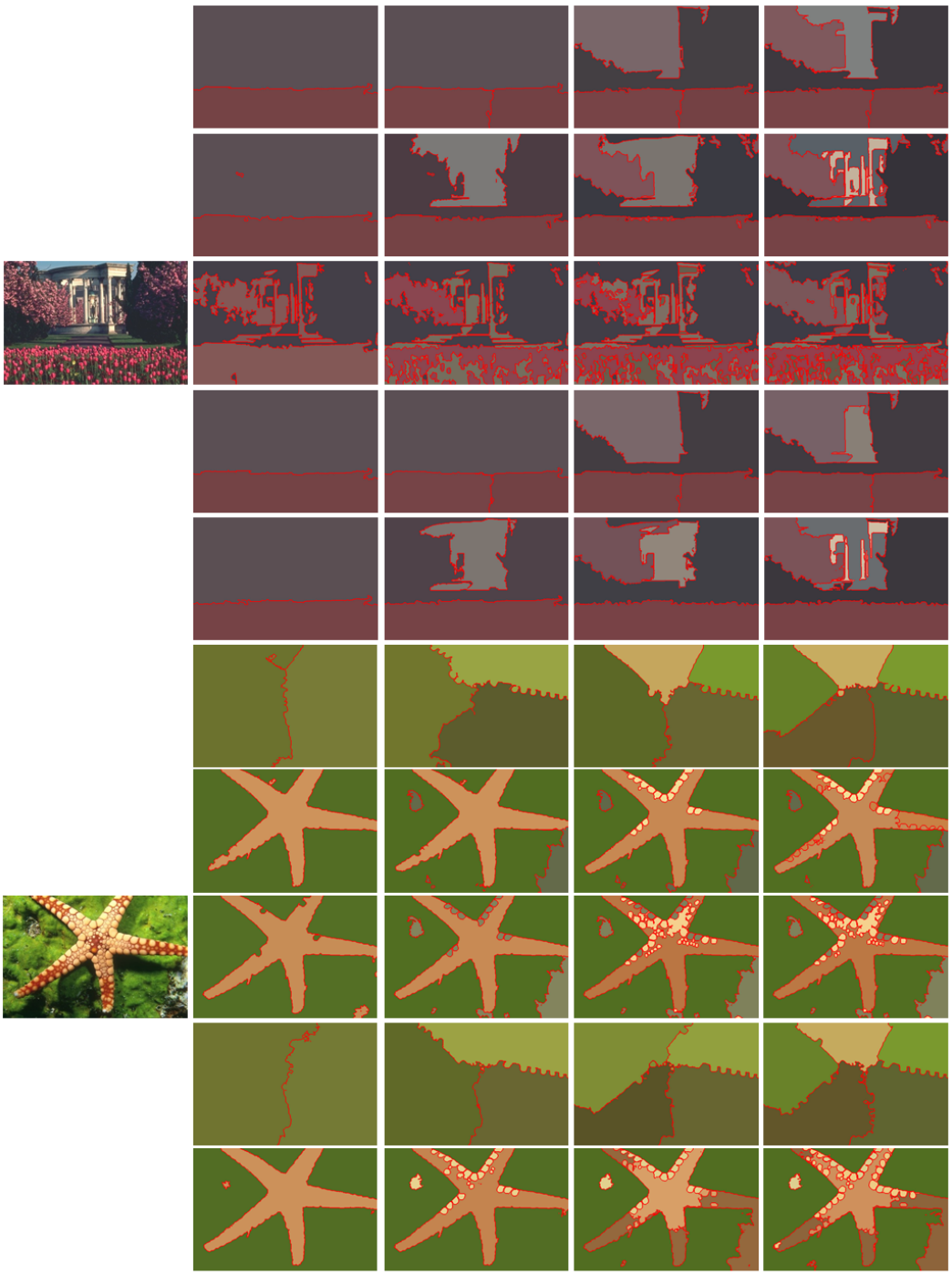


Figure 4.9: Visual comparison of the results obtained with the GL-graph and with other graphs. Each line from top to bottom corresponds to the segmentation result obtained with the following graphs: *adjacency*, kNN, ℓ_1, LRR, and the proposed GL-graph. From left to right, the results for various choices of $k = 2, 3, 4, 5$. . Note that the result is segmented using the most appropriate feature for each kind of graph according to Table I, e.g. we use the feature *mlab* for the *adjacency*-graph and the KNN-graph.

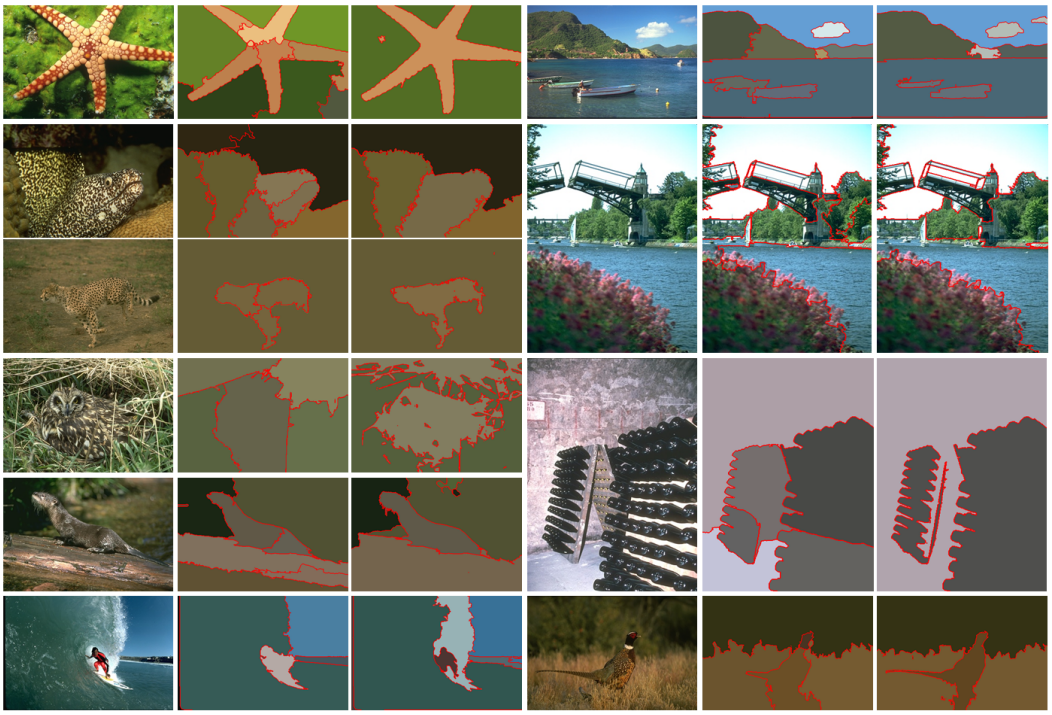


Figure 4.10: Visual comparison with SAS. For each experiment, the second image shows the results of SAS, and the third image is obtained with our method. Our results require significantly less tuning for k and are visually better in general, in particular often more accurate.

Figure 4.11: Visual segmentation examples by the proposed method: all images are segmented into 2 regions (k=2). Note that the salient objects or parts can be segmented accurately, such as the plane, boat, flower with insects, elephants, hill. Even multiple objects with large inner color variation can be segmented correctly, as the cactus flowers or the men in water.

Chapter 5

Graph-based Image Segmentation Using Weighted Color Patch

Contents

5.1 Introduction

In recent years, the graph-based methods have been proven successful and widely applied to image segmentation, mainly because they have an efficient tool to solve the optimization problem of segmentation [P.Bo et al., 2013] and can naturally incorporate different type of features in the affinity graph. In particular, the graph-based methods first construct an affinity graph from a given image, and then partition the resulting graph into different clusters with certain cut criteria [Hagen and Kahng, 1992] [Shi and Malik, 2000]. Thus, constructing a discriminative affinity graph plays an essential role in such methods. For a desirable partition result, the pixels should be similar to each other in intra-clusters while different from each other in inter-clusters. The similarity between two pixels can be measured by the distance of different features such as color, boundary, texture, etc. Feature is an important factor since its properties directly influence the discriminative power of the resulting affinity graph.

5.1.1 Motivation and our proposed method

In this chapter, we propose a new feature descriptor based on the weighted color patch to construct a more discriminative affinity graph. The idea of representing a pixel with a patch has been proven successful in non-local image denoising [Buades et al., 2005]. However, it produces the over-smooth effect due to considering each member equally in the patch. Therefore, it is necessary to assign different weight to each pixel in the patch. J. Zexuan et al. [Ji et al., 2012] investigated this idea in their work on fuzzy c-means clustering, but they only considered gray intensities to compute the similarity of two pixels. For image segmentation, it is insufficient to use only gray intensities, while color is also a very discriminative and efficient feature for identifying different objects, especially in natural images. Therefore, our proposed method intends to incorporate both color and neighborhood information. There are two main advantages: i) it can smooth local regions by averaging color information and ii) it can capture texture information by considering context neighboring cue. Furthermore, in order to incorporate spatial information, we also propose to assign a global weight to each pixel in an image according to different proportion of the object and background, so that the contrast between them is enhanced and a more discriminative affinity graph is constructed.

The rest of the chapter is organized as follows: we introduce the proposed weighted color patch (WCP) method elaborately in section 2, where local and global weights are presented in section 2.1 and 2.2 respectively, and we introduce the affinity graph construction based on WCP in section 2.3; in section 3, we present extensive experiments on the Prague texture image benchmark [Haindl and Mikes, 2008] and the Berkeley image segmentation database [Arbelaez et al., 2011], and report the quantitative results with associated multiple evaluation metrics; the conclusions are drawn in section 4.

5.1.2 Related work

In the literature, numerous works have been proposed to design powerful features for image segmentation. Color has been proven powerful in many works for image segmentation. Early works [Shi and Malik, 2000][Felzenszwalb and Huttenlocher, 2004] on graph cut based technique only consider the color information with one pixel. Recent results in [Fowlkes et al., 2003] suggest that the color cue is best captured using patches to task of image segmentation, for it is well known that color patches are a stable cue. Furthermore, using sliding window (patch) for object detection has proven a huge success, e.g., Deformable Part-based Models (DPM) [Felzenszwalb et al., 2010] and its variants [Girshick et al., 2011] [Azizpour and Laptev, 2012]. Developing new feature descriptor based on patch attracts intensive interest of researchers. One evident advantage of patch is that a set of intensity, texture and shape features can be extracted and computed for each patch. For instance, Malik et al.[Malik et al., 2001] constructed textons based on clustering of filter response over patches. [Brunner12 et al., 2010], they extract intensity, texture and shape descriptors from patches and combine them linearly in the graph-cut formulation. Another advantage is that using patch makes derived descriptor robust against noise. More importantly, the use of a patch can also incorporate the context

information, which has been treated as a key factor. [Torralba, 2009] pointed out that human performance in scene categorization remains high no matter whether low or multi-mega pixel images are used. Moreover, they demonstrated that very small-size patch 32×32 color pixels provides sufficient information to recognize the object category. Later, in [Lee and Grauman, 2012], the authors proposed that context information of known object is a helpful hint for discovering unknown objects in an image. In [Bai et al., 2010], the computation of graph similarities for shape retrieval is context-sensitive by considering neighbors' influence.

5.2 Proposed method

5.2.1 Local weights computation

As introduced in the introduction, using patches directly will cause the over-smooth effect mainly due to considering each member in the patch equally. Therefore, it is necessary to assign different weights to different pixels. In this paper, we adopt the method described in [Ji et al., 2012] to compute the local weights adaptively.

Let an image represented by $I = \{g_1, ..., g_x\}$ with g_x as pixel intensity, and a patch vector denoted as $P_k = (g_k, N_k)$, where N_k is the neighborhood around the central pixel g_k with the size $w \times w$. For each pixel g_r in the patch, its mean-square deviation σ_r is defined as follows:

$$\sigma_r = \Big[\frac{\sum_{n \in N_k \setminus \{r\}} (g_r - g_n)^2}{n_k - 1}\Big]^{1/2} \tag{5.1}$$

The computed mean-square deviation σ_r is then applied in the following exponential kernel function:

$$\xi_r = exp\Big[-\Big(\sigma_r - \frac{\sum_{r \in N_k} \sigma_r}{n_k}\Big)\Big] \tag{5.2}$$

Finally, the local weight of pixel g_r is obtained by normalizing the value of ξ_r:

$$\omega_r = \frac{\xi_r}{\sum_{r \in N_k} \xi_r} \tag{5.3}$$

Since the applied kernel function decays very fast, those pixels whose mean-square deviation is far away from the average value will have a relatively small weights. An illustration of how to calculate the local weights is shown in Fig.5.1, and we take a patch from a natural image to depict the effectiveness of the local weights. We can observe that the patch is extracted from an inhomogeneous boundary region, thus relative to the central pixel, those pixels lying on the other side of the boundary are assigned with smaller weights in order to decrease their impact to the patch.

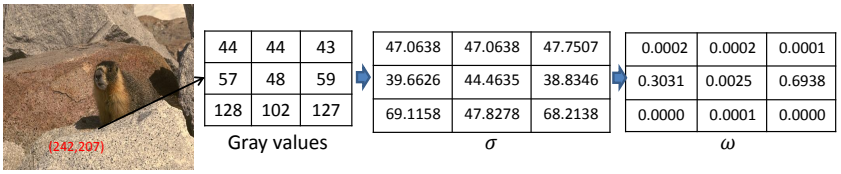


Figure 5.1: An illustration of the local weights calculation of a patch extracted from the boundary region in a natural image (the first column shows the gray values, the second column is the mean-square deviation of each pixel, and the last column shows the weights assigned to each pixel).

5.2.2 Global weights assignment

In addition to the local weights, which only reflect the structure of a local patch, we also propose to assign a global weight to each pixel in an image according to different proportion of the object and background, since we observed that they should have different contribution to the affinity graph construction because of different structure of the whole image content. More precisely, the proposed global weights are obtained by calculating a normalized histogram of the image based on the pixel values.

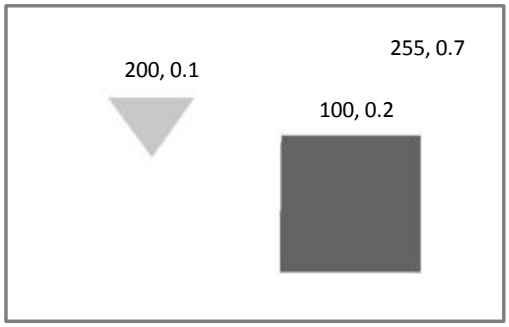


Figure 5.2: An illustration of the effectiveness of the global weights in a synthetic image.

Fig.5.2 presents an example to show the effectiveness of using the global weights. Suppose that the intensities of the background, the triangle object and the rectangle object are 255, 200, 100 respectively, then their calculated global weights will be 0.7, 0.1 and 0.2 respectively. Without the global weights, the distances between the background and the objects are 55 and 155 respectively, while both distances become 158.5 when considering the global weights. Thus we can see that i) the distances between the background and the objects are increased; ii) both objects have the same distance to the background, which makes them easier to be segmented simultaneously.

5.2.3 Affinity graph construction

Given an image I, it can be represented as a graph $G = (V, E)$, with V being the set of vertices and E being the set of edges connecting two vertices. We apply the

proposed WCP method to compute the weight of edges in the graph. In order to incorporate color information, the affinity graph is first computed in each channel of the RGB color space, formally defined as follows:

$$W(x_i, x_j) = e^{-(\|\sum_{i=1}^{N} P_{wi}^c - \sum_{j=1}^{N} P_{wj}^c\|^2/\sigma)}, ||x_i - x_j||_2 < r \tag{5.4}$$

where W is the affinity graph, and $W(i, j)$ defines the edge weight of two vertices i and j in the graph. According to the derived weights, we discard those pixels in the patch whose weights are smaller than a threshold value which is set to $1/(n_k) \times 1/N$ with n_k the size of the local patch, and N the total number of pixels in the image. x_i represents the spatial coordinates of pixel i, and r is the graph radius.

$$P_{wi} = (g_r,\ r \in N_k,\quad if\quad \omega_r \times \xi_r >= (1/(n_k) \times 1/N)) \tag{5.5}$$

with ξ_r represents the global weight assigned to pixel g_r. σ in Eq.(4) is a positive constants to control the decaying speed of gaussian kernel function. c represents each channel of the RGB color space.

The final affinity graph is obtained by averaging the results from all the channels.

5.2.4 Graph partitioning

Given the affinity graph W, we apply the normalized cut (NCut) algorithm to partition the graph into k groups by solving the following generalized eigen-vector problem:

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{5.6}$$

where $L = D - W$ is the Laplacian matrix, $D = diag(W\mathbf{1})$ is the diagonal degree matrix. The bottom k eigenvectors are computed either by k-means [Ulrike, 2007] or discretization method [Shi and Malik, 2000].

5.3 Experimental Evaluation

In this section, we evaluate the proposed WCP method for image segmentation on two popular databases: the Prague color texture benchmark [Haindl and Mikes, 2008] and the Berkeley image segmentation database (BSD) [Arbelaez et al., 2011]. For simplicity, we fix the parameters for all the following experiments as: $\sigma = 10$, $r = 10$ in Eq.(4) and the patch size is 7×7.

5.3.1 Results on Prague texture benchmark

The Prague texture benchmark datasets are computer generated 512×512 random mosaics filled with randomly selected textures. This benchmark provides a bunch

of criteria for evaluation (see Table 1), and we list them in Appendix B which describes them in detailed manner. The proposed method is compared with the other unsupervised benchmark algorithms, including: EDISON [Christoudias et al., 2002], JSEG [Yining and Manjunath, 2001a], SWA [Sharon et al., 2001], and GL-graph in Chapter 4. Fig. 5.3 presents seven selected 512 × 512 experimental benchmark mosaics and Table 1 gives their corresponding numerical scores w.r.t. different indicators. It can be observed that EDISON and JSEG tend to oversegment images while SWA and our method have better trade-off between over-/under-segmentation. From the results presented in Table 1, we can see that no single algorithm can outperform all the others on all the measurements. Note that our method GL-graph presented in Chapter 4 ranks the first place on 10 indicators (highlighted in bold), the proposed method WCP and JSED has two and SWA has four best results. In particular, although EDISON also has 8 best performances, its other performances such as OS, O and C lagged far behind ours, which makes our method the best overall algorithm except GL-graph regarding to all associated indicators.

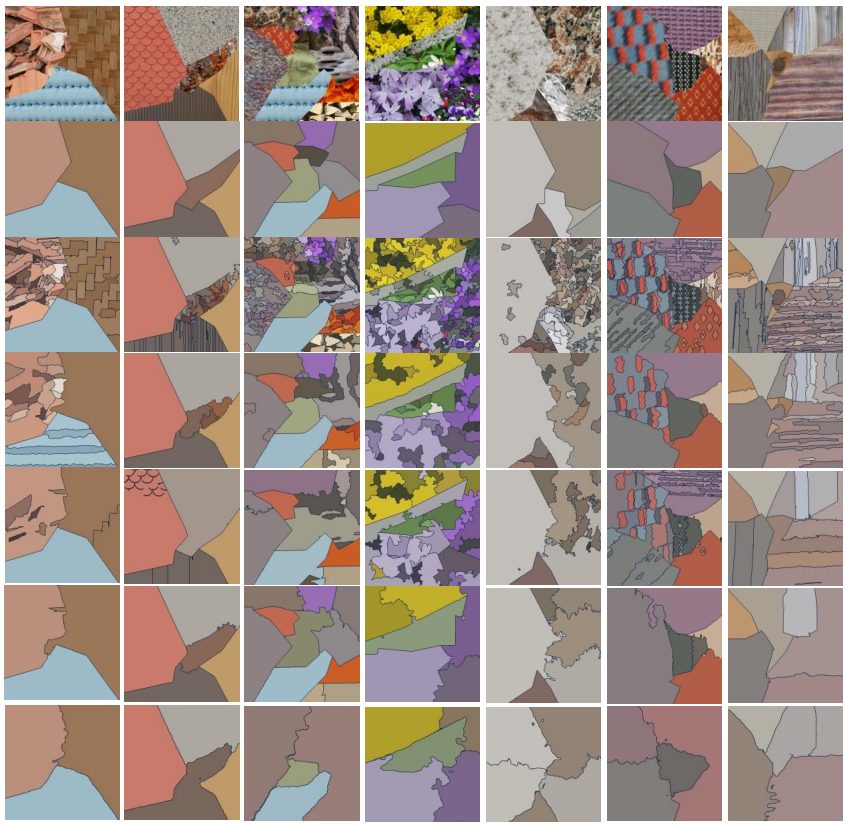


Figure 5.3: Visual comparison of our results with other methods on the Prague benchmark (examples presented in row-wise, from up to down, are respectively the original images, ground truth maps, EDISON, JSEG, SWA, GL-graph presented in Chapter 4 and our results WCP).

Table 5.1: Quantitative comparison of our results with other methods on the Prague benchmark with multiple measurements.

Metrics	region-based					consistency measure		clustering			-
Methods	CS↑	OS↓	US↓	ME↓	NE↓	GCE↓	LCE↓	dM↓	dD↓	dVI↓	-
EDSION	12.68	86.91	**0.00**	**2.48**	**4.68**	**3.55**	**3.44**	35.37	16.84	25.65	-
JSEG	27.47	38.62	5.04	35.00	35.50	18.45	11.64	23.38	15.19	17.37	-
SWA	27.06	50.21	4.53	25.76	27.50	17.27	11.49	24.20	**13.68**	17.16	-
GL-graph	**41.42**	15.04	12.48	27.64	26.92	20.48	11.25	**11.22**	17.13	14.40	-
WCP	30.92	**4.12**	26.67	37.40	35.72	20.28	14.82	22.27	16.83	**13.25**	-
Metrics	pixel-wise										
Methods	O ↓	C↓	CA↑	CO↑	CC↑	I.↓	II.↓	EA↑	MS↑	RM↓	CI↑
EDSION	73.17	100.00	31.19	31.55	**98.09**	68.45	**0.24**	41.29	31.13	**3.21**	50.29
JSEG	37.94	92.77	55.29	61.81	87.70	38.19	3.66	66.74	55.14	4.96	70.27
SWA	33.01	85.19	54.84	60.67	88.17	39.33	2.11	66.94	53.71	6.11	70.32
GL-graph	**17.80**	**15.13**	**66.53**	**75.75**	82.19	**24.25**	4.17	**76.10**	**63.63**	6.72	**77.48**
WCP	41.32	28.70	53.55	67.49	63.39	32.51	6.60	62.69	51.23	9.34	64.00

5.3.2 Results on Berkeley image database

The Berkeley image database contains 300 images and their corresponding ground truth (each image has at least 4 human annotations). In our experiments, we test the proposed method on all the 300 images, since the algorithm has no parameter to be trained. The number of segments k is set from $[3, 5, 7, 10, 12, 15, 18, 20, 23, 25, 28, 30, 31, 32, 35, 40]$. The final results are evaluated according to 4 associated measurements, including: Probabilistic Rand Index (PRI) [Unnikrishnan et al., 2007], Variation of Information (VoI) [Meila, 2005], Global Consistency Error (GCE) [Martin et al., 2001], and Boundary Displacement Error (BDE) [Freixenet et al., 2002]. The popular NCut, GBIS [Felzenszwalb and Huttenlocher, 2004] and Normalized Tree Partitioning (NTP) [Wang et al., 2008b] are applied for the purpose of comparison, and their parameters are the same as [Kim et al., 2010a], which manually tuned the number of segments for each image.

The quantitative results are presented in Table 2, with the best results highlighted in bold for each measurement. It is obvious to see that the proposed WCP method ranks the first place with respect to VoI and BDE compared with the other methods. Fig.5.4 presents some visual comparisons of our results with the other methods, and we can see that NCut tends to split homogenous large region into separate regions and GBIS has thick edges, while our proposed method can obtain more meaningful region with accurate boundary. We also present some examples segmented by our proposed method in Fig.5.5. It can be observed that our method can well segment the texture images (the penguin, the leopard, web girl), and it has high discriminative power to detect objects from different backgrounds.

5.4 Conclusion

In this paper, we propose a new method based on the weighted color patch to construct the affinity graph for image segmentation. The proposed method is invariant to uneven light conditions and noise benefitting from the usage of image patches.

Figure 5.4: Visual comparison of our results with other methods on the Berkeley database (examples presented in column-wise, from left to right, are respectively the original images, NCut, GBIS and our results).

Furthermore, we assign a local weight to each member in the patch to overcome the over-smooth effect, and also calculate a global weight for each pixel in the image to enhance the contrast between the background and the objects. The proposed method is evaluated by extensive experiments on two popular segmentation databases, and is quantitatively compared with some other standard algorithms. The results show that our method is powerful and competitive, and can be further applied on other clustering problems.

Table 5.2: Quantitative comparison of our results with other methods on the Berkeley database with multiple measurements: the results of our method are obtained over the best tuned parameter for each image.

Methods	PRI↑	VoI↓	GCE↓	BDE↓
NCut	0.7242	2.9061	0.2232	17.15
GBIS	0.7139	3.3949	**0.1746**	16.67
NTP	**0.7521**	2.4954	0.2373	16.30
WCP	0.7496	**2.4399**	0.2392	**15.7416**

Figure 5.5: Some examples segmented by our method on the Berkeley database.

Chapter 6

Sparse Coding and Mid-Level Superpixel-Feature for ℓ_0-Graph Based Unsupervised Image Segmentation

Contents

6.1 Introduction

Most unsupervised image segmentation methods, which are frequently used for high-level vision tasks like object recognition and image annotation, involve essentially low level features such as color, boundary or texture. In particular, various method using graphs and spectral clustering have been proposed in recent years, however it remains challenging for those methods to provide desirable visually semantic partitions.

Generally, for those methods, building a faithful graph is critical to the final quality. The graph nodes can be pixels or regions, and the graph affinity matrix encodes the similarity between either low level features or top down features associated with the nodes. Low level features capture object basic properties and they can be obtained with various descriptors or operators, such as color histograms, histogram of oriented gradients (HOG), scale invariant feature transform (SIFT), local binary patterns (LBP), etc. Despite progresses in the design of more informative low-level

features, performances remain limited. Top down features usually convey semantic or prior knowledge about the segmented regions or objects. Many works treat the output of trained classifiers and object detectors [Li et al., 2010], or semantic segmentation algorithm [Fu and Qiu, 2011] as top down information to guide the low level unsupervised segmentation. However, all these top-down semantic methods require non-trivial amounts of human-labeled training data, which is unrealistic in practical situation.

6.1.1 Motivation and the proposed method

In this chapter, we focus on mid-level features based on sparse coding, as in [Zou et al., 2012] where first a dictionary is built by learning or human labeling, then the coefficients of the sparse representation in this dictionary are used to define mid-level features for classification or grouping. In contrast to [Zou et al., 2012], we build the dictionary from informative patches centered at interest points detected without any supervision, and each mid-level feature is the sparse coding in the dictionary of the low level feature associated with a superpixel. This way, the contextual information, which has been proved an efficient cue to discriminate two objects or images [Lee and Grauman, 2012], is added to the original low-level features to improve the robustness of the similarity coefficient between two superpixels in the graph construction, whose quality plays a critical role to the segmentation result.

More precisely, the whole segmentation model starts by extracting interest points from the image, associating with them a set of low-level features whose collection forms a dictionary, and over-segmenting the input image into multi-layer superpixels. Then, each superpixel is associated with a sparse representation of its low level feature in the previously built dictionary. This proposed feature inherits of the original descriptors' property and covers also adaptive contextual information. Compared with related works and other benchmark algorithms on the MSRC dataset [Shotton et al., 2006], the key contribution of this paper is that our new mid-level feature is able to describe better the superpixels. The similarities between superpixels are then computed based on ℓ_0 graph construction in the spirit of [Wang et al., 2013a] (where only low-level features were used). Finally, the constructed graph is plugged into a robust unsupervised segmentation framework introduced in [Li et al., 2012]. The proposed method can segment visually semantic regions, and can be used in many high-level computer vision tasks.

The organization of the chapter is as follows: in Section 2 we introduce the proposed mid-level features based on the sparse coding and the segmentation framework, and in Section 3 we present and comment a few segmentation results on the MSRC dataset. We conclude in Section 4.

6.1.2 Related work

In recent years, features popularized in other domains, e.g. image classification and sliding-window detection, has been renewed interest in segmentation together with recognition based on bottom-up segments. In this problem, the typical processing pipeline is: 1) extracting local feature (e.g., SIFT and HOG), 2) encoding the local

features in an image descriptor (e.g., histogram of the quantized local feature), and 3) representing resultant descriptor to a classifier (e.g, support vector machine). The role of feature encoding is the core component, which produces a global description of an image or a region to summarize the local features inside the region [Carreira et al., 2012a][Huang et al., 2014]. Note that in the thesis, we denote the feature descriptor derived from 1) and 3) as mid-level feature, also known as bag-of-visual features in other literatures. Successful applications of discriminative power of local features also have motivated their introduction for bottom-up image segmentation directly. For instance, [Yu et al., 2012] proposed bag of textons, namely, filter responses encoded by soft clustering technique for image segmentation.

There are bunch of feature coding methods proposed in literature. As suggest by the survey [Huang et al., 2014], we can group the existing techniques into two major categories, *global* and *local coding* according to the motivations presented in their original papers. *Global coding* focuses on the global description of all features rather than each individual feature. More specifically, there are two major methods:

- Voting-based methods describe the distribution of features with a histogram, also referred as codebook in literature, which carries the occurrence information of codewords. Such a histogram can be constructed by hard quantization or soft quantization. For instance, in [Csurka et al., 2004], a bag of key points called as the bag-of words for word-document or bag of features for image classification, was proposed as mid-level feature, which corresponds to a histogram of the number of occurrences of particular image patterns in a given image. To avoid drawbacks of the codebook produced by K-means or radius based clustering, [van Gemert et al., 2008] proposed an uncertainty modeling method to form the codebook.

- Fisher coding-based methods estimate the distribution of features or codebook with the Gaussian mixture models (GMM), consisting of the weights, the means, and the covariance matrix of multiple Gaussian distributions, each of which reflects one pattern of features. Fisher Kernel introduced by Jaakkola et al.[Jaakkola et al., 1999] and applied by Perronnin and Dance[Perronnin and Dance, 2007] to image classification.

Yet *local coding* is proposed to describe each individual feature. More precisely,

- Reconstruction-based methods use a small part of codewords to describe each feature via solving a least-square-based optimization problem with constraints on codewords, i.e., a feature can be represented with a small error. There is a huge body of literature existing on this topic. Their difference mainly lies on constraint term. For instance, sparse coding is with a constraint term: [Yang et al., 2009]with constraint term $\sum_{i=1}^{M} |v(i)|$; LLC [Wang et al., 2010] is subject to constraint: $\sum_{i=1}^{M}(v(i)exp(||x - b_i||_2/\sigma))$. Besides this, there are many other reconstruction-based coding methods, e.g., Laplacian sparse coding [Gao et al., 2010], mixture sparse coding [Yang et al., 2010], nonnegative sparse coding [Zhang et al., 2011], hierarchical sparse coding [Yu et al., 2011]. All of them extend sparse coding by substituting the constraint term.

- Local tangent-based methods, e.g., [Yu and Zhang, 2010] assume that all features constitute a smooth manifold where codewords are located. They derive an exact description for each feature through approximating the Lipschitz smooth manifold. In this way, features are not independent but closely related, expressed by a Lipschitz smooth function.

- Saliency-based methods e.g.,[Huang et al., 2011] encode each feature by the saliency degree, which is calculated using the ratio or the difference of the distances from a feature to its nearby codewords. The core idea is that a strong response on a codeword indicates relative proximity, which means that this codeword, compared with all other codewords, is much closer to a feature belonging to this codeword.

Finally, we summarize the characteristics of the different methods mentioned above. *Global coding* pursues to model the probability density distribution of features, and therefore, it is not easy to be influenced by a small number of unusual features. In particular, Fisher coding uses GMM for probability density estimation, which is more robust than the histogram-based manner. *Local coding* aims to describe each individual feature and, thus, is sensitive to unusual features, however, as the codebook size increases, it can describe more patterns of features. Thus, it has better adaptiveness compared with *global coding*.

6.2 Superpixels, mid-level features, and sparse representation

Our approach consists of three steps: 1) interest points extraction, low-level features computation, and dictionary building; 2) over-segmentation of the original image, extraction of superpixels (defined as the over-segmented regions), computation of a low-level feature for each superpixel, and sparse representation in the dictionary of step 1; 3) graph construction and partitioning.

6.2.1 Low-Level Features Detection and Extraction

We use low-level features extraction to build a meaningful dictionary to represent a given image. First, we extract a set of key points from the image. The meaningfulness of the low-level dictionary is highly dependent on the choice of the key points. If they capture the main structural information of the input image, then the derived dictionary will be highly meaningful. In practice, we have tested various approaches, see Fig. 6.1: either the interest points are randomly or densely sampled, or they are obtained using a feature descriptor, e.g., the Harris detector, the Difference of Gaussians (DoG), or the Hessian detector. The respective performances are discussed in Section 3.

Once interest points have been extracted, we consider the local image patches around them, from which low-level features can be computed (we use in this paper RGB color histograms for its strong discriminative skill, but other features as LBP histogram or SIFT may be used). Finally, our low-level dictionary is defined as the collection of all these low-level features, see Fig. 6.2.

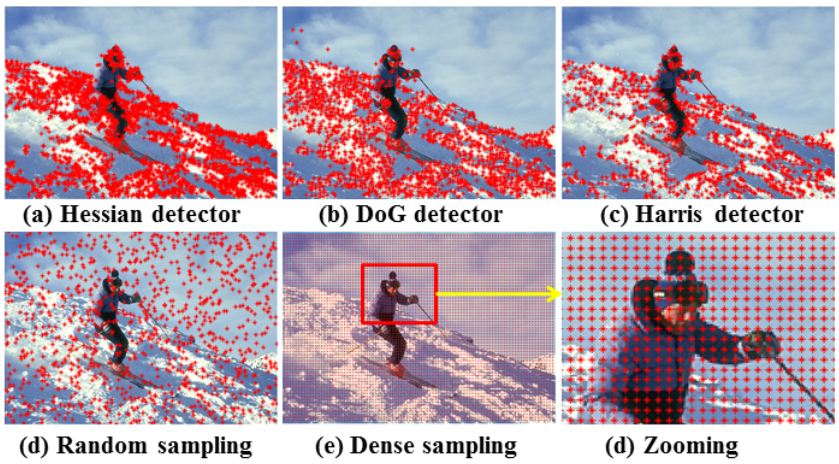


Figure 6.1: Illustration of different types of interest points.

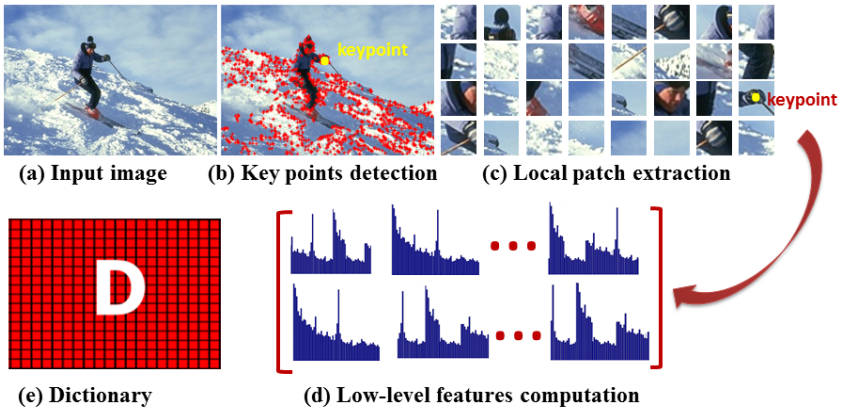


Figure 6.2: Illustration of low-level features computation.

6.2.2 Mid-Level Features Extraction over Superpixels

We call superpixel a region of an over-segmentation of the original image. In practice, we compute several over-segmentations, and we associate with each superpixel a low-level feature (in our experiments, we used RGB color histograms for its strong discriminative skill). Then we define the mid-level feature associated with a superpixel as the sparse representation of its low-level feature in the dictionary built previously, see Fig. 6.3 for an illustration of the whole process. More precisely, given a superpixel, suppose $x \in \mathbb{R}^m$ is the low-level feature associated with it, and let $D = [d_1 \cdots d_n] \in \mathbb{R}^{m \times n}$ be the low-level dictionary built in section 6.2.1. The sparse representation of x in D is obtained by solving the following optimization problem:

$$\min_{\alpha} ||x - D\alpha||_2^2 \quad s.t. \quad ||\alpha||_0 \leq L, \tag{6.1}$$

where $\alpha \in \mathbb{R}^n$, and $\|\alpha\|_0 := \|\alpha\|_{\ell_0}$ is the number of its non-zero coefficients. Suppose $\hat{\alpha}$ is a solution of the problem and $\Lambda_{\hat{\alpha}} = \{j | \hat{\alpha}(j) \neq 0\}$ is the index set of non-zero

coefficients of $\hat{\alpha}$, then the mid-level feature associated with the low-level feature x is defined as

$$\hat{x} = D\hat{\alpha} = \sum_{j \in \Lambda_{\hat{\alpha}}} d_j \hat{\alpha}(j). \tag{6.2}$$

Therefore, the mid-level feature $\hat{x}$ is a linear combination of several low-level features, thus not only carries the same information as the original low-level features, but also carries additional contextual cue.

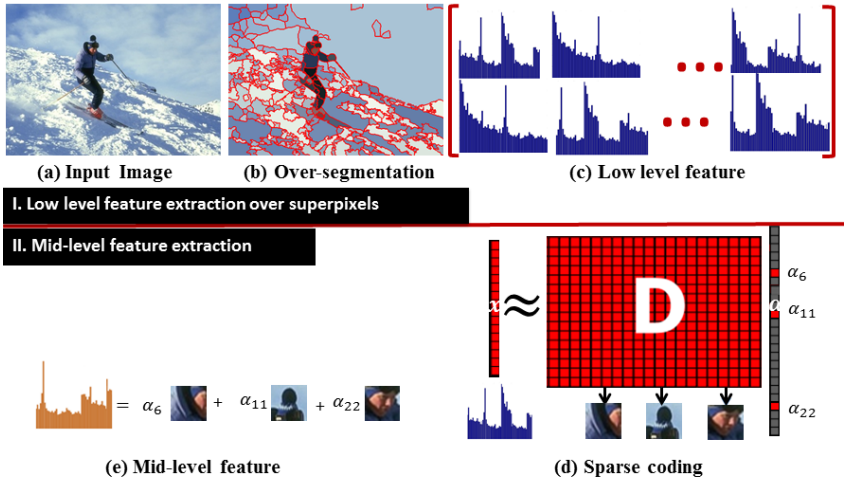


Figure 6.3: Illustration of mid-level features computation.

6.2.3 Graph Construction and Partitioning

Once mid-level features have been computed, we build the graph that will be plugged into a spectral clustering algorithm to perform image segmentation. This is done as follows: For each scale of over-segmentation (i.e. for each instance of over-segmentation), we construct a graph whose nodes are the superpixels at that scale, and whose graph edges and weights are computed using ℓ_0-sparse representation. More precisely, we consider as dictionary the mid-level features associated with the superpixels. Then, as in Equation (6.2), each mid-level feature $\hat{x}_i$ can be represented as a sparse linear combination $\hat{x}_i = \sum_j \alpha_j^i \hat{x}_j$ of the other mid-level features. The similarity coefficient of any pair $\hat{x}_i, \hat{x}_j$ of superpixels is defined as $w_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 - (r_{ij} + r_{ji})/2 & \text{if } i \neq j. \end{cases}$ where r_{ij} is the sparse representation error of $\hat{x}_i$ and $\hat{x}_j$, i.e. $r_{ij} = \|\hat{x}_i - \alpha_j^i \hat{x}_j\|_2^2$.

We collect all ℓ_0 affinity matrices obtained from all over-segmented images, and we concatenate them diagonally into a unique matrix denoted as W_{SS}, together with the pixel-superpixels affinity matrix W_{IS}. Then we consider the bipartite graph associated with the matrix $B = \begin{bmatrix} W_{IS} \\ W_{SS} \end{bmatrix}$ and the Transfer Cut algorithm [Li et al., 2012] is applied to partition the bipartite graph into K clusters by solving the

following generalized eigenvalue problem over superpixels only $L_V\mathbf{f} = \lambda D_V\mathbf{f}$, where $L_V = D_V - W_V$, $D_V = diag(B^\top\mathbf{1})$, and $W_V = B^\top D_U^{-1}B$, $D_U = diag(B\mathbf{1})$, see [Li et al., 2012] for more details.

6.3 Experimental Results

6.3.1 Database and Parameter Settings

We evaluate our approach on the Microsoft Research Cambridge (MSRC) database, which contains 591 images from 23 object classes, and we use for the evaluation the accurate ground-truth segmentations of [Malisiewicz and Efros, 2007]. To quantitatively evaluate the performance, we apply four popular measurements : 1) Probabilistic Rand Index (PRI) [PRI]; 2) Variation of Information (VOI) [Meila, 2005]; 3) Global Consistency Error (GCE) [Martin et al., 2001]; and 4) Boundary Displacement Error (BDE) [Freixenet et al., 2002]. A segmentation result is better if PRI is higher and the other three ones are lower. For low-level features extraction, we only use the color feature in RGB space, and the feature dimension is reduced from 256×3 to 64 by PCA. For mid-level dictionary building via sparse coding, we use the Orthogonal Matching Pursuit (OMP) algorithm [OMP] to solve Eqn. 6.1 and set the sparsity number $L = 4$ according to the experimental results.

On the step of graph construction and partitioning, we proceed as in our previous work [Wang et al., 2013a], i.e. we derive from the original image 5 or 6 oversegmented images (this number of scales being experimentally satisfactory) obtained by the Mean Shift (MS) method [Comaniciu and Meer, 2002] and by the FH method [Felzenszwalb and Huttenlocher, 2004]. More precisely, we derive three images by the MS method using the sets of parameters (hs, hr, M)= $\{(7, 7, 100), (7, 9, 100)$, and $(7, 11, 100)\}$, respectively, where hs and hr are bandwidth parameters in the spatial and range domains, and M is the minimum size of each segment. Either two of three oversegmented images are provided by the FH method using as parameters (σ, c, M) either $\{(0.5, 100, 50), (0.8, 200, 100)\}$, or $\{(0.8, 150, 50), (0.8, 200, 100), (0.8, 300, 100)\}$. To build the ℓ_0 graph, the sparsity number $L = 3$ is used for all the experiments, see [Wang et al., 2013a] for more details. We organize our experimental results as follows: first, we compare the

Table 6.1: Comparison of different feature detectors on the whole MSRC database (red color indicates the best result).

Detector	PRI↑	VoI↓	GCE↓	BDE↓
Harris detector	0.8195	1.4214	0.1694	9.4530
Hessian detector	0.8177	1.4366	0.1691	9.9951
DoG detector	0.8226	1.3900	0.1670	9.3955
Random sampling	0.8069	1.5578	0.1781	10.1746
Dense sampling	0.8280	1.3452	0.1633	9.4403

performances of the five different kinds of low-level feature detectors introduced in section 6.2.1; then, we list the quantitative results of our proposed method on differ-

ent subsets of MSRC database and compare it with several state-of-the-art methods; finally, we show some visual examples of our method.

Table 6.2: Performances of our method on MSRC and comparison with state-of-the-art methods.

Metric	**PRI↑**		**VoI↓**		**GCE↓**		**BDE↓**	
Object class	**baseline**	**new**	**baseline**	**new**	**baseline**	**new**	**baseline**	**new**
1. grass, cow	0.8889	0.8978	0.7927	0.8417	0.1006	0.1059	4.8316	4.9181
2. tree, grass, sky	0.7865	0.7963	1.2569	1.3664	0.1727	0.1990	18.6141	13.6065
3. building, sky	0.8429	0.8697	1.2660	1.3768	0.1670	0.1755	8.0268	8.3904
4. aeroplane, grass, sky	0.9083	0.9202	1.3133	1.2662	0.1463	0.1649	4.1802	4.3369
5. cow, grass, mount	0.9038	0.8647	0.5641	0.7804	0.0752	0.0889	4.2286	4.8817
6. face, body	0.7176	0.7277	2.2429	2.3892	0.2601	0.2669	16.1357	15.2383
7. car, building	0.7423	0.7624	2.2676	2.1879	0.2044	0.2546	12.3907	12.3268
8. bike, building	0.7037	0.7196	2.0662	2.1575	0.2729	0.2854	10.7725	10.9580
9. sheep, grass	0.8837	0.8867	0.7287	0.7166	0.0853	0.0874	4.7323	4.9983
10. flower	0.8712	0.8766	0.6368	0.7172	0.0836	0.0927	6.8501	5.7331
11. sign	0.8581	0.8839	0.7668	0.7591	0.0929	0.0940	6.4911	6.3972
12. bird, sky, grass, water	0.8820	0.8932	0.6977	0.7215	0.0963	0.0831	5.6918	5.9985
13. book	0.6714	0.6613	1.7574	1.9669	0.1596	0.1633	18.9275	17.7393
14. chair	0.7395	0.7806	1.3144	1.6839	0.1862	0.1807	11.7096	7.7027
15. cat	0.7532	0.7483	1.3479	1.2819	0.1272	0.1240	12.0134	11.8589
16. dog	0.8030	0.8029	1.2856	1.2436	0.1394	0.1613	9.7475	9.5381
17. road, building	0.8439	0.8610	1.6346	1.7412	0.2002	0.2025	9.0031	8.4299
18. water, boat	0.8548	0.8424	1.0310	1.0947	0.0935	0.1088	9.1329	12.4533
19. body, face	0.8376	0.8275	1.6961	1.9347	0.1931	0.2124	7.4399	8.8790
20. water, boat, sky, mount	0.8884	0.9154	1.1942	1.0002	0.1602	0.1279	6.3682	5.6792
Average performance								
Method	**PRI↑**		**VoI↓**		**GCE↓**		**BDE↓**	
Our new method	0.8269		1.3614		0.1590		9.0032	
Baseline [Wang et al., 2013a]	0.8190		1.2930		0.1508		9.3644	
NCut [Shi and Malik, 2000]	0.8052		1.2516		-		-	
LRR(CH)[Liu et al., 2013a]	0.7912		1.3002		-		-	
MS[Comaniciu and Meer, 2002]	0.7307		1.7472		-		-	

6.3.2 Experimental Results

As mentioned in section 6.2.1, the property of the low-level dictionary is highly dependent on the selection of the key points. Therefore, we compared the Harris detector, Difference of Gaussian (DoG), Hessian detector, random sampling, and the dense sampling (see Fig. 6.1). The results are shown in Tab. 6.1, from which we can deduce that dense sampling is the most efficient way to extract interest points. The main reason is that dense sampling can capture almost all information of the image and is well-suited for sparse coding that requires an over-complete dictionary.

We compare in Table 6.2 the performances of our method on the MSRC database and the performances of the method we proposed in [Wang et al., 2013a] (limiting to RGB histogram as superpixel feature, and calling *baseline* this reference algorithm). Obviously, our new method can achieve excellent performances on segmenting object classes such as *cow*, *building*, *sheep*, *flower*, *sign*, *bird*, *road*, and *boat*, but is less efficient for *tree*, *face*, *cat*, *dog*, *bike*, etc. The visual results are also shown in Fig.6.4.

The reasons for the difference performances are various: **1)** objects like *face*, *cat*, and *dog* usually have complex backgrounds mainly associated with indoor scene which makes the evaluation unfair for the machine algorithms since the ground-truth does not label the indoor objects. On the other side, in the case of objects without complex backgrounds, our method can segment them correctly even if the object itself presents obvious color variations like on *cow*, *building* and *flower*; **2)** objects like *face* or *bike* can be subject to strong illumination changes which prevent the machine algorithms from grouping object correctly if only color is used as low level descriptor. Results should be improved if other descriptors as LBP were used, and this is the purpose of future work. **3)** the quality of segmentation can also be influenced greatly by the way superpixels are extracted.

We compare the performances of our approach with other state-of-the-arts algorithms in Tab. 6.2. We used the scores given in [Liu et al., 2013a], observing that GCE and BDE were not reported. Our method ranks first according to PRI and BDE, which makes it one of the most competitive algorithms.

6.4 Conclusion

We introduced a new unsupervised image segmentation method based on ℓ_0-graph, superpixels, mid-level features, and sparse coding. An nice property of the mid-level feature we propose is that it can capture adaptive contextual information and carries as well the original low level feature information. Quantitative comparison with the state-of-art methods, as well as visual results, indicate that our new algorithm is a competitive image segmentation method.

Figure 6.4: Examples of segmented results on the MRSC dataset (for each experiment, we show the segmentation result, and the segmentation superimposed with the original image).

Chapter 7

Active Colloids Segmentation and Tracking

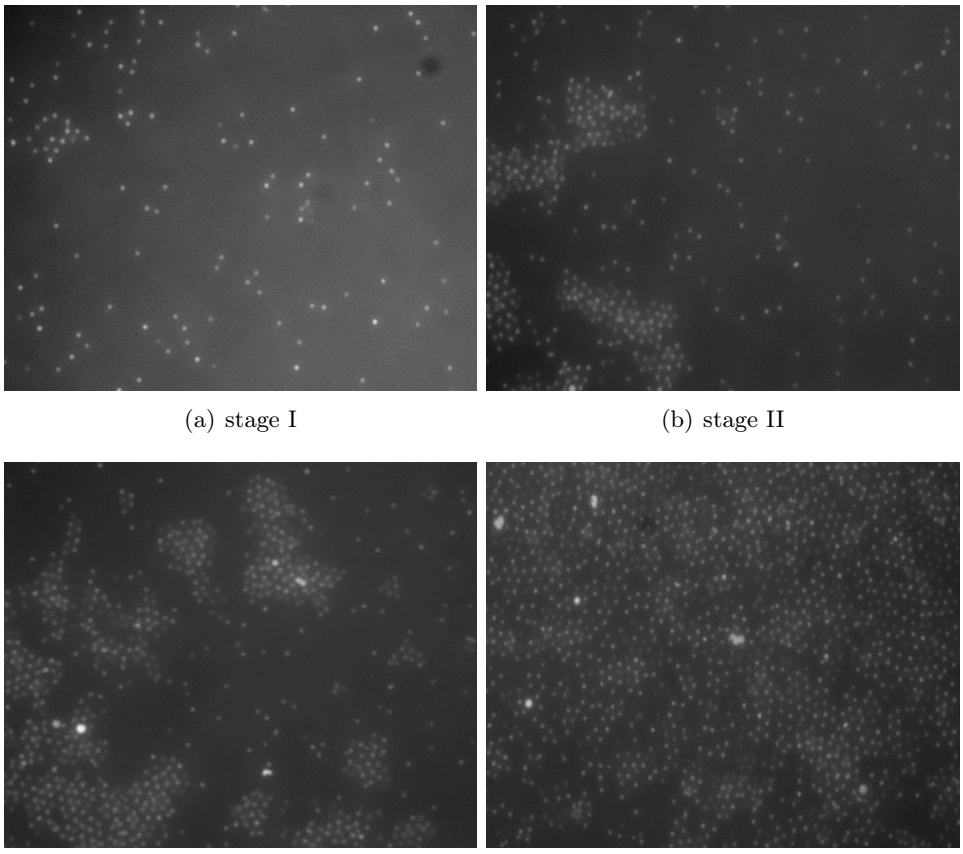
Contents

7.1 Introduction

Designing an efficient and robust tracking algorithm has been an active research topic in computer vision community over the past decades. The basic task of video tracking is to detect the position or shape of objects and to link correctly trajectories of moving single/multiple targets in consecutive frames. In practice, multiple objects tracking (MOT) is a canonical method for wider range of applications ranging from human surveillance [Benfold and Reid, 2011, Butt and Collins, 2013b, Milan et al., 2013], vehicles tracking [Betke et al., 2000, Rubio et al., 2012], to biological analysis (e.g. cell tracking [Chenouard et al., 2013]), or behavior pattern study [Veeraraghavan et al., 2008].

7.1.1 Context

In this thesis, we aim to develop a reliable system to track tens of thousands of active colloids in certain environment. Here, active colloids are tiny-scaled particles which can propel themselves by consuming energy at individual level. These spherical colloidal particles (see Fig. 7.1) can be sedimenting, platinum-coated gold particles [Buttinoni et al., 2013] or an embedded hematite cube [Palacci et al., 2013]. The



(a) stage I (b) stage II

(c) stage III (d) stage IV

Figure 7.1: Illustration of dynamical-clustering of self-propelled colloidal particles at different stages: (a) illustrates temporal evolution at low densities; (b) and (c) show the cluster grow stably, and (d) presents the final stage of the system.

modeling of dense active suspensions of artificial self-propelled colloids has aroused recently the interest of physicists [Buttinoni et al., 2013, Theurkauff et al., 2012], with the general purpose of understanding the mechanisms of self-aggregation and the formation of clusters. Very recent studies found that active systems exhibit a wide variety of collective behaviors, structures and patterns as illustrated in Fig. 7.1 where typical situations at low and high density are presented. Fig.7.1 (a) shows the stage of temporal evolution of a small cluster. During the period, the aggregation is dynamical, i.e., particles join and leave the cluster frequently. Fig.7.1(b) and (c) show that the cluster size increases approximately linearly, as observed by Buttinoni

et al.. Fig.7.1 (d) illustrates that at higher densities, clusters grow until the system consists of a few big clusters according to the observations of Buttinoni et al.. Moreover, such mechanisms can be observed in many living systems, such as cells, sheep flocks, birds, fishes swarms, etc. However, the collective patterns hide many different mechanisms in the individual motion, e.g. move and interaction. Therefore, there is a need for identifying the underlying models in order *"to disentangle the universal from the specific behavior of those complex phases"* [Theurkauff et al., 2012]. A natural approach is to derive the models from the experimental observation of colloidal suspensions, and more precisely from the identification of the individual trajectories of colloids. However, labeling manually massive colloidal suspensions in long video sequences to obtain accurate trajectories leads to a tremendous amount of work. It is therefore necessary to design a reliable, automatic, and systematic algorithm, which can process long video sequences with good resolution.

Although several publications on colloidal suspensions [Crocker and Grier, 1996, Royall et al., 2003] have provided standard procedures to identify independent paths, they still differ from our problem in the context of active systems in some respects, for example, irregular motion and frequent interactions among the colloids. Nevertheless, the task of tracking active colloids shares several common difficulties with the particle tracking in biological study [Chenouard et al., 2013, 2014, Padfield et al., 2011], where a "particle" may be anything, such as a single molecule, or cells, genes etc. Difficulties include similar appearance, large number of objects, frequent events like entering, exiting, splitting, merging etc..

Yet in the context of active systems, we face with new challenges which are still open issues. First, due to that colloids are all similar, the only useful information is spatial location. Second, colloids can have very complicated motion [Buttinoni et al., 2013]. For example, a colloid may follow ballistic motion at short times, while at longer times, it transits to a diffusive regime. Moreover, unlike the application in human or cell tracking, the motion of a colloid can change frequently because of collisions with other colloids. Thus, it is difficult to infer an accurate pre-defined model to predict colloid's location, especially in a long video sequence. Furthermore, as will be explained below, in the experiments that we use, colloids essentially move in a 2-dimensional plane, yet some 3d behavior may rarely occur which results in that some colloids may exhibit low intensity or disappear, and then appear again in a short period of time. Consequently, only trajectories fragments may be observed for some colloids.

In addition, in contrast with living cells tracking where merging usually happens among few cells and small clutter moves at small speed, in our case, as pointed in [Buttinoni et al., 2013], large clusters can form and move. It is therefore essential, in order to identify accurately the motion law, to be able to distinguish colloids individually.

7.1.2 Motivation and contribution

Our goal is to design a faithful detection and tracking algorithm to answer physicist's needs, i.e. detecting precisely the positions and the independent trajectories of many self-propelled colloids. Our work is based on video captures of a two-dimensional

dense active suspension of artificial self-propelled colloids [Buttinoni et al., 2013], see the experimental details in Section 7.4. By two-dimensional, it is meant that the colloids being heavy, they settle at the bottom of the observation cell and only 2D motions are observed. In particular, the colloids being observed from below, there is essentially no occlusion phenomenon. We propose a framework to jointly segment, localize and track each colloid. The difficulty of the detection task follows from the severe intensity inhomogeneities in each frame, the high number of colloids, and the poor temporal resolution. In other experiments where the intensity is more homogeneous and the temporal resolution is high, a much simpler algorithm can be used (for instance adaptive threshold technique [Otsu, 1975], morphological top hat [Serra, 1982]). The method that we propose in this paper can handle much more complicated situations, e.g. it is also applicable to dense assemblies of passive colloids, such as colloidal glasses and gels.

To sum up, our contribution is as follows:

- To obtain high-quality segmentation, we propose to combine the level set method and circular Hough transform in the same framework, to handle the severe intensity inhomogeneities and highly cluttered colloids.
- We build a graph over all frames instead of over a very few, and further refine by three additional configurations. We insist on the fact that no assumption is made on colloids motion, except that the positions of a colloid in two consecutive frames are close. In particular, we do not assume any smoothness of the trajectories and we do not favor a priori linear motions.
- We propose to recover all trajectories from the trellis graph simultaneously. More precisely, we model the problem by finding the maximum flow with minimum cost over the whole graph. We propose an additional procedure called tag-then-delete together with the successive shortest path (SSP) algorithm to efficiently solve the min-cost/max flow optimization.
- Unlike other methods, validated on simulated image data, we choose to evaluate quantitatively on a real benchmark, the active colloids suspension, provided to us by physicists. These videos have been annotated by different graduate students majored in computer vision. As far as we know, there is very limited previous work using real dataset with human observations labeling the ground truth, for the obvious reasons that it is a tedious task to track all the particles in long video sequences. Instead, most methods first evaluate on a synthetic benchmark [Chenouard et al., 2014, Ruusuvuori et al., 2008, Smal et al., 2010a] with other standard algorithms, then provide performances on real data, e.g. cells in fluorescence image sequences.

Organization. The remainder of this paper is organized as follows. In Section 2, we present a survey on previous literature on multiple object tracking. In Section 3, we introduce our proposed detection algorithm and trellis graph construction for modeling the generic dynamics in the active systems, and present in detail the Min-cost/Max flow algorithm and our optimized solution to this optimization problem as well. In Section 4, our real-data benchmark is introduced and extensive experiments are presented to validate our method. Finally, we conclude in Section 5.

7.2 Related Work

Advances in imaging techniques and applications in some specific domains have encouraged greatly the research on automatic tracking methods capable of establishing accurate trajectories. In this paper, we concentrate on the tracking-by-detection paradigm to solve the task of multiple object tracking. Existing proposed methods in literature can generally be divided into two components: (i) particles detection, where targets are detected from each video frame, and (ii) multiple objects tracking, where detected objects or particles are connected across frames by a suitable tracker algorithm. For each of the above two steps, there are a huge body of literature over the years. Nevertheless, several good-quality surveys with relevant to the object tracking [Chenouard et al., 2014, Meijering et al., 2012, Yilmaz et al., 2006] are available to provide practical yet insightful summarization these developments. We briefly review these techniques concerning multiple particles detection and tracking.

7.2.1 Particles Detection

Generally, most proposed particle detection frameworks can be split into three steps and some of them can be optional or implement in a different way, according to [Chenouard et al., 2014, Smal et al., 2010a,b]: (i) noise reduction, e.g. Guassian smoothing (ii) signal enhancement, e.g. wavelet multiscale product [Olivo-Marin, 2002], and (iii) signal thresholding. For example, in [Basset et al., 2014], they exploited Laplacian of Gaussian at several scales to detect the minima of LoG values, which is then thresholded to derive segmentation result. (LoG) of the images at several scale In [Chenouard et al., 2014], they performed an objective comparison of different methods with an open competition, and found that approaches to particles detection algorithm varied greatly, ranging from simple threshold [Winter et al., 2011], or local-maxima finding [Coraluppi and Carthel, 2011] and morphological processing [Anoraganingrum, 1999], to linear and nonlinear model fitting [Liang et al., 2010], and centroid estimation [Sbalzarini and Koumoutsakos, 2005]. Most of detection algorithms have incorporated two or more of these mentioned techniques, and inevitably they shared many techniques in common.

7.2.2 Multiple Object Tracking

Multiple object tracking has been intensively studied by researchers, and can be divided into two basic classes [Poore and Gadaleta, 2006]: (i) the probabilistic based methods. (e.g. Kalman filters [Blackman, 1986], particle filters [Kitagawa, 1987], multiple hypothesis tracking[Blackman, 2004], probabilistic multiple hypothesis tracking [Gelgon et al., 2005], inference on Bayesian network [Nillius et al., 2006], joint probabilistic density association filters [Fortmann et al., 1983]), Monte Carlo Markov Chains methods [Oh et al., 2004]; (ii) non-probabilistic methods (e.g. nearest neighborhood, minimum-cost [Butt and Collins, 2013a][Wu et al., 2011][Padfield et al., 2011], shortest path [Berclaz et al., 2011, Jiang et al., 2013], minimal paths [Bonneau et al., 2005], Hungarian algorithm [Kuhn, 1955]).

Most proposed algorithms are locally greedy, that is run a low-level tracker in a small window (two frames or three) to obtain tracklets, and then link the par-

tial track fragments with method like network-flow [Zhang et al., 2008b], linear-programming [Jiang et al., 2007], matching algorithm [Kuhn, 1955], Bayesian network [Nillius et al., 2006], or solving a set cover problem [Wu et al., 2011]. These locally greedy trackers are applied widely for the sake of low computational cost, nonetheless such trackers tend to have identity swap errors, which are hard to be corrected when future information is included. Another drawback is that such strategy can easily miss the global optimal solution. It is also fair to mention that Multiple Hypothesis Tracking and Monte Carlo Markov Chains methods search the largest nonconflicting trajectories of all objects satisfying the expected motion behavior, but cannot guarantee a globally optimal solution in sub-exponential time.

7.3 Proposed Framework for Colloids Detection and Tracking

7.3.1 Accurate Active Colloids Detection

In the paradigm of track-before-detect, accurate detection method is essential for the quality of tracking. In general, the difficulty of detecting particles yields corrupted sets of detections: some particles are missed by the detection procedure and some artifacts are wrongly considered as being some spots of interest. In our context of the active colloids detection, due to low dose concentration, all images have low contrast-to-noise. Moreover, the images are subject to uneven illumination, i.e. the flat-field from the lamp causes the interior to look brighter than the edges of the image (e.g. see Fig. 7.2(a)). In addition, objects can be highly cluttered, which can cause severe ambiguities in the tracking stage.

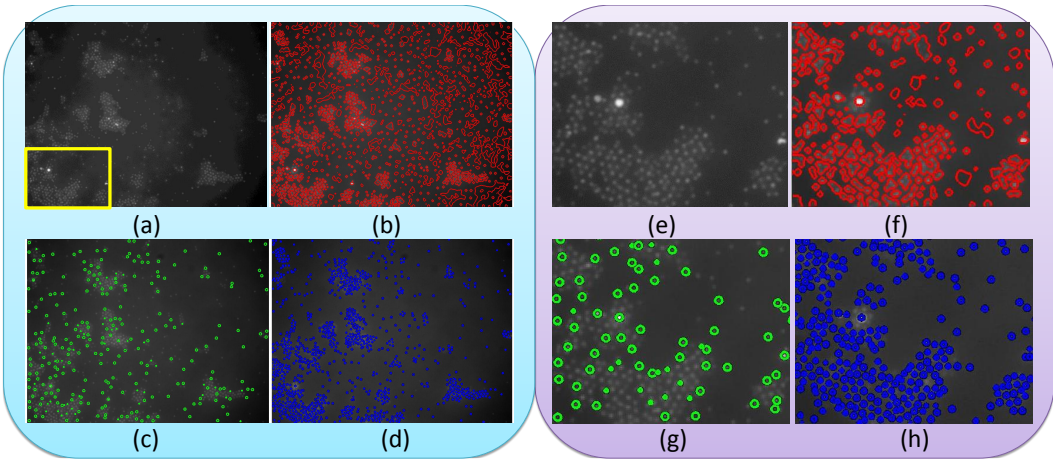


Figure 7.2: Illustration of active colloid detection. In light-blue block: (a) a video frame with highly dense colloids; (b) result segmented only by level set [Li et al., 2008]; (c) result detected with Gaussian smoothing and circular Hough transform; and (d) result obtained by the combination of level set and circular Hough transform. In light-purple block: (e) zoom in version of the yellow rectangle shown in (a); and (f-h) are the zoom-in results corresponding to (b-d) respectively.

We propose an accurate method to detect meaningful objects based on the modified Circular Hough Transform (CHT) [Atherton and Kerbyson, 1999] and the level set method proposed in [Li et al., 2008]. It is known that all objects in the image are round-shape spots, although some colloids may have deformations due to twinkle or uneven illumination. The Circular Hough Transform has been proved efficient in detecting circle targets thanks to its nice properties such as the robustness to noise, invariance to slight occlusion and illumination.

However, when dealing with real-world images, the circular Hough transform can miss potentially targets altered by intensity inhomogeneities. In this paper, we adopt an efficient level set variational model [Li et al., 2008] to overcome the difficulties arisen from such inhomogeneities, thanks to the region-scalable fitting (RSF) energy which quantifies how well, given a contour C in the image domain Ω, the image intensities in the outer and inner domains with respect to C are well approximated locally by two functions f_1 and f_2:

$$\begin{aligned}\varepsilon^{RSF}(C, f_1, f_2) = \int_{\Omega} (\lambda_1 \int_{\text{inside}(C)} \mathbf{K}_\sigma(\mathbf{x}-\mathbf{y})|I(\mathbf{y}) - f_1(\mathbf{x})|^2 d\mathbf{y} \\ +\lambda_2 \int_{\text{outside}(C)} \mathbf{K}_\sigma(\mathbf{x}-\mathbf{y})|I(\mathbf{y}) - f_2(\mathbf{x})|^2 d\mathbf{y}) d\mathbf{x} + \nu|C|\end{aligned} \tag{7.1}$$

where Ω is the image domain, $\lambda_1, \lambda_2, \nu$ are positive constants, K_σ is a Gaussian kernel whose standard deviation σ tunes the locality of the approximation, and the length $|C|$ of C is a regularization parameter that avoids spurious contours when the energy is minimized. The fitting energy ε^{RSF} is able to segment objects even in severe illumination condition, due to that, in contrast to global threshold in classical level set based active contour models (e.g. the Chan-Vese model [Chan and Vese, 1999]), the approximating functions f_1 and f_2 are not necessarily constant within the outer and the inner domains denoted as $outside(C)$ and $inside(C)$ respectively. This allows a good robustness to light changes.

As shown in [Li et al., 2008], the energy admits a level-set formulation, so that the associated minimizing flow can better handle the topological changes. More precisely, a contour $C \subset \Omega$ is represented as the zero level set of a function $\phi : \Omega \to \mathbb{R}$, which is called level set function (a classical example of such function is the signed distance function to C). We solve the level set minimization problem with standard gradient descent method in Algorithm 1, and we refer the reader to [Li et al., 2008] for more details. Convex formulations of $\varepsilon^{RSF}(C, f_1, f_2)$ in the spirit of [Chan et al., 2006, Pock et al., 2008] could be used, and would decrease the computational time, but they would not be equivalent to Algorithm 1 which involves an additional regularization term that is well suited for the recovery of objects with smooth boundaries.

The proposed detection method can segment individual tiny objects in dense populations in the presence of noise and intensity inhomogeneities. Fig.7.2 shows our motivation and the result of active colloids detection. It is obvious to observe that Fig.7.2 (a) is a difficult image for the task of segmenting individual tiny colloids from poor illumination condition and large clusters. As we can see, Fig.7.2 (b) (c) illustrate that RSF and CHT fail to detect objects when used individually. The

Algorithm 4: Active colloids detection

1: **for** $i = 1 : N$ **do**
2: Input frame I_i
3: Segment frame I_i with the RSF level set method, i.e.:
4: Initialize the level set ϕ (using for instance the signed distance function to the discontinuity set of the segmentation), the number of iterations $numIter$, and the parameters $\lambda_1, \lambda_2, \nu, \mu, \sigma$;
5: **for** $k = 1 : numIter$ **do**
6: Compute the approximate Heaviside function and its derivative:

$$H_\varepsilon(\phi) = \frac{1}{2}[1 + \frac{2}{\pi}\operatorname{atan}(\frac{\phi}{\varepsilon})], \tag{7.2}$$

$$\delta(\phi) = \frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2 + \phi^2}; \tag{7.3}$$

7: Compute the fitting intensities:

$$f_1 = \frac{\mathbf{K}_\sigma * [H_\varepsilon(\phi) I_i]}{\mathbf{K}_\sigma * [H_\varepsilon(\phi)]}, \tag{7.4}$$

$$f_2 = \frac{\mathbf{K}_\sigma * I_i - \mathbf{K}_\sigma * [H_\varepsilon(\phi) I_i]}{\mathbf{K}_\sigma * \mathbf{1} - \mathbf{K}_\sigma * [H_\varepsilon(\phi)]} \tag{7.5}$$

8: Update ϕ:

$$\begin{aligned}\frac{\partial \phi}{\partial t} = &-\delta_\varepsilon(\phi)(\lambda_1 e_1 - \lambda_2 e_2) + \nu \delta_\varepsilon(\phi) \operatorname{div}(\frac{\nabla \phi}{|\nabla \phi|}) \\ &+\mu \left(\Delta \phi - \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right)\end{aligned} \tag{7.6}$$

where, $e_k(x) = \int_\Omega \mathbf{K}_\sigma(y - x)|I_i(x) - f_k(y)|^2 \mathrm{dy}$, $k = 1, 2$.
9: Threshold the level set ϕ with Otsu's method [Otsu, 1975], get coarse object interest regions I_{roi}.
10: Find circles with CHT on I_{roi}: $[centers\ radii] = CHT(I_{roi},\ [minR\ maxR])$, where $[minR\ maxR]$ is the range of possible radii;
11: Generate binary image by creating round regions centered at *centers* and get each colloid's center.

reason lies in that the RSF model is unable to *separate* clustered colloids and the CHT misses some ambiguous targets. Fig.7.2 (d) demonstrates that combining both methods is much more efficient.

7.3.2 Active Colloids Tracking

Given detected colloids, the next task is to infer meaningful trajectories throughout the video. Unlike common methods, which first implement partial matching between two or three frames, then link all fragments to form meaningful paths, we build the graph using all colloids within all frames, and then solve the multi-frame data

association problem.First, the output of the detection step consists in all colloids' centers associated with frame indexes, and this information is used to construct an initial graph. Second, we refine the initial graph in order to correct local "errors" (see Fig. 7.3), which may cause ambiguity during later data association. Finally, the min-cost/max flow algorithm is adopted to solve the problem of finding meaningful paths over the trellis graph, and the solution is guaranteed to be a global optimum with respect to the current graph.

7.3.2.1 Directed Acyclic Graph Construction

A directed graph $G = (V, E)$ is constructed to model the image frames over the time, where V is the nodes set representing all colloids centers, and E is the edge set of pairwise nodes representing the relationship of these nodes. The quality of the constructed graph is very important to the final tracking result. We have two stages for the graph construction: 1) initial graph construction; 2) initial graph refinement.

Initial Graph Construction: In collision-free situations, a single colloid exhibits Brownian-like motion. In practice however, the population density yields frequent collisions, which induce abrupt changes of colloids' motion velocity and direction. It implies that we cannot impose any motion smoothness, in particular because the purpose of the study is to identify the underlying motion model, not to impose it a priori. Therefore, we construct the initial graph using only a simple spatial rule, i.e. each node links to its neighbors within a search range d_{max}, and the edge weight coincides with the normalized distance between both neighbors.

Mathematically, for a node $\mathbf{v} = (x', y')$ at frame f_{k-1} finds its neighbor $\mathbf{u} = (x, y)$ in the next frame f_k, the cost ω_{uv} of linking them is defined as:

$$\omega_{uv} = \frac{\|\mathbf{u} - \mathbf{v}\|_2}{d_{max}} \tag{7.7}$$

where d_{max} is a parameter which controls the search range.

Initial graph refinement: Given the initial weighted graph, we now want to recover meaningful paths in it. However, the graph's construction rules being uniform, a colloid in a given frame may be connected with none, one or several colloids in the previous and next frames. Heuristically, we identified three configurations that may require a modification of the graph which, in practice, improves the final result. These configurations are illustrated in Fig. 7.3:

Configuration I (PI)

Fig.7.3 (a) shows a case of missing connection. The track numbered 80 in pink is terminated at frame f_k, due to that its corresponding node at time $t = k - 1$ has no connected node at time $t = k$. This can be easily fixed by simply increasing the search range until a neighborhood is found (shown in dash line).

Configuration II (PII)

Fig.7.3 (b) illustrates the case where one node (numbered 154 in the frame f_{k-1}) connects to two neighbors (numbered 154 and 361 in frame f_k), which are *far* from each other. We observed that the global result is improved if the

link with the most distant node is removed from the graph. Let us emphasize that in case a node has more than three neighbors, only the most distant one is removed.

Configuration III (PIII)

Fig.7.3 (c) presents another conflicting situation where two nodes (numbered 88 and 344 in frame f_{k-1}) are connected to the same neighbor in f_k (labeled as 88 in the figure). If only one of the node has more than two neighbors then its connection with the common neighbor is removed. If both nodes have more than two neighbors, the longest connection with the common neighbor is removed.

The graph refinement corresponding to fixing these configurations is synthesized in Algorithm 2. It is worth mentioning that the refinement removes at most one connection at each node, and that of course there remain afterward, in particular in zones with clusters, nodes with more than two neighbors. Experimentally, we observed that there is no need to iterate the refinement until convergence to a stable graph. There is actually no significant improvement of the final optimal flow after two iterations of this intermediate refinement step.

7.3.2.2 Full Trajectories via Min-cost/max Flow

In the context of tracking, our goal is converted to finding the largest set of independent trajectories, which can be expressed as a maximum flow problem. While there are many maximal solutions to the same flow network, the fact that colloid's motion shows natural spatial continuity between consecutive frames yields naturally to defining the optimal tracking result as the solution of a minimum cost / maximum flow problem (min-cost/max flow).

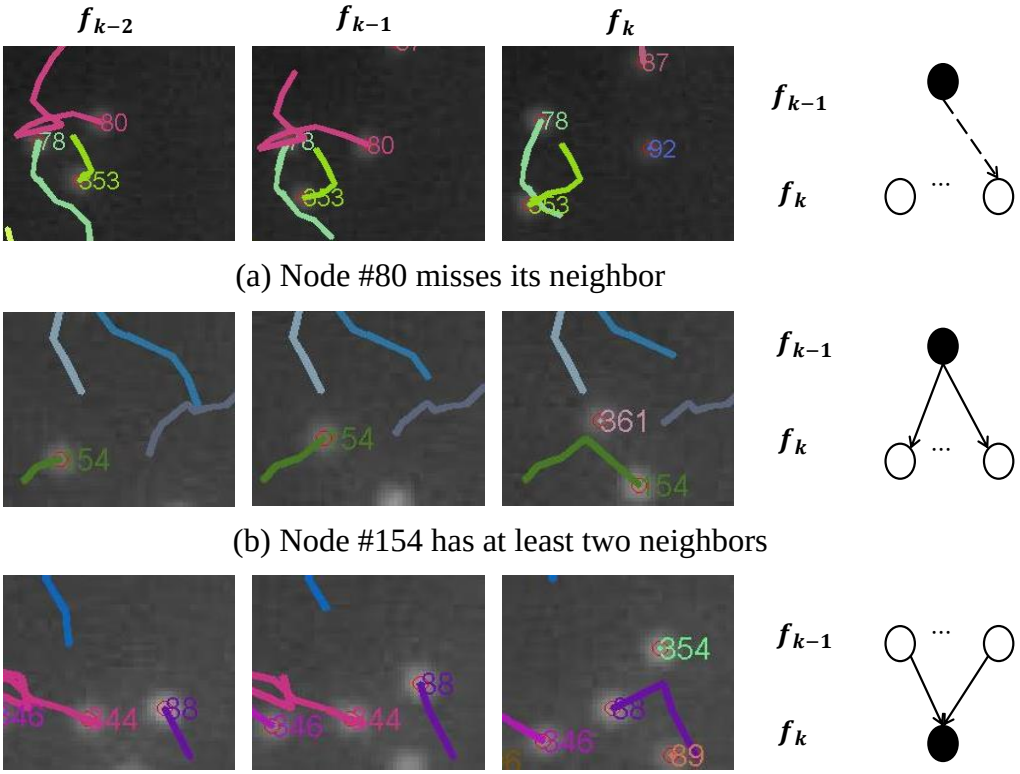
Mathematical formulation: we consider a directed graph $G = (V, E)$ with source $s \in V$ and sink $t \in V$, where each edge $e_{uv} \in E$ has unit capacity. We will denote as f_{uv} a volume of flow passing from node u to node v along e_{uv}. If e_{uv} is positively oriented then $0 \leq f_{uv} \leq 1$, otherwise $-1 \leq f_{uv} \leq 0$. The cost per unit flow on e_{uv} is denoted as ω_{uv} – we already assumed that $\omega_{uv} = \|u - v\|_2 / d_{\max}$ whenever $u, v \neq s, t$, and of course edges from s or to t have no cost. We shall work only with circulation flows, i.e. the inflow and outflow volumes are equal at each node except s and t.

Sending a volume m of a circulation flow from s to t, and looking for the flow repartition along the graph with minimal cost is equivalent to solving the following problem [Ahuja et al., 1993]:

$$\text{argmin}_f \sum_{(u,v)\in E} \omega_{uv} f_{uv} \tag{7.8}$$

subject to the following properties

(i) $|f_{uv}| \leq 1$;
(ii) $f_{uv} = -f_{vu}$;
(iii) $\sum_{w\in V} f_{uw} = 0 \quad \forall u \neq s, t$;



(a) Node #80 misses its neighbor

(b) Node #154 has at least two neighbors

(c) Node #88 shares a neighbor with another node

Figure 7.3: Illustration of local confliction in the initial graph. **Left**: temporal trajectories in different color among three consecutive frames recovered from the initial graph with min-cut/max flow algorithm. **Right**: the graph connection between two frames f_{k-1} and f_k. **Note** that dot in black represent the current node and its corresponding neighbors in next frame is dot in white. The dash line means the nodes should be considered connected and the $\times$ in red means the connection should be removed.

Algorithm 5: Initial graph refinement

The initial directed graph $G = (V, E)$, where each node's position is denoted as $p = (p_i, p_j)$, and the search radius d_{max},
Configuration I: node $p^{current}$ has no neighbors in $B(p^{current}, d_{max})$
repeat
  $d_{max} \leftarrow 1.5 \times d_{max}$
until $p^{current}$ *has neighbor in* $B(p^{current}, d_{max})$ **or** $B(p^{current}, d_{max}) \cap ImageBorder \neq \emptyset$;
Configuration II: node $p^{current}$ connects with conflicting nodes p_1, p_2 on next frame (i.e. $\|p_1 - p_2\|_2 \geq d_{max}$).
if $p^{current} \notin ImageBorder$ **then**
  if p_1 *or* $p_2 \in ImageBorder$ **then**
    remove the node near image border;
  else
    remove the connection with largest cost.
  end
else
  remove the connection with largest cost.
end
Configuration III: nodes $p_1^{current}$ and $p_2^{current}$ connect with the same node $p^{neighbor}$ on next frame.
if $\exists! p_k^{current}$ *with more than* 2 *neighbors* **then**
  remove the corresponding connection with $p^{neighbor}$
else
  remove the connection with $p^{neighbor}$ which has largest cost
end
Refined directed graph G^*

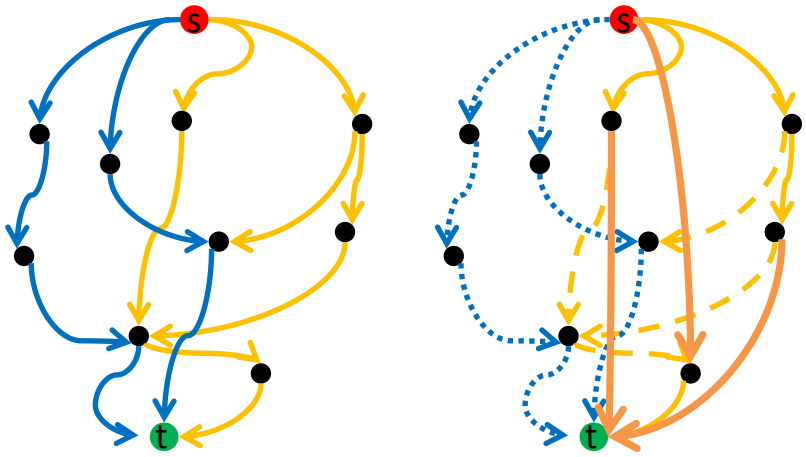
$$(iv)\ \sum_{w \in V} f_{sw} = \sum_{w \in V} f_{wt} = m.$$

Ensuring that no two trajectories share an object, each colloid can just only link with one colloid between consecutive frames respectively. Thus, we assume that each edge contains unit capacity $c(u, v) = 1$ and there are no negative cost edges $\omega(u, v) \geqslant 0$. Many algorithms exist to find the solution based on such optimality criteria, such as cycle canceling, linear programming, push-relabel method [Zhang et al., 2008b], We refer the reader to [Ahuja et al., 1993] for a proof of the optimality criteria and more details of these algorithms.

Global optimal solution: Many algorithms exist to find the solution of this minimum cost maximal flow problem, such as cycle canceling, linear programming, or the push-relabel method [Zhang et al., 2008b]. We refer the reader to [Ahuja et al., 1993] for a proof of the optimality criteria and more details on these algorithms.

A popular method is the successive shortest path (SSP) algorithm[pir, Ahuja et al., 1993]. However, one pass of SSP is insufficient to find all tracks, because graph construction from frames cannot guarantee to connect every head node of potential track to the source, with the same situation for rear node to the sink.

Thus many potential trajectories are actually hidden in the initial graph. Instead of performing SSP once, we apply the method iteratively followed by a tag-then-delete procedure. In each iteration, the SSP first finds a set of optimal trajectories (shown in blue lines in Fig. 7.4(a)) in the current graph. The tag-then-delete procedure then tags and deletes the nodes as well as their corresponding edges within the flow (dots in gray color and dash lines in Fig. 7.4(b)). The rest of nodes with 0 in-degree will be connected to the source and the ones with 0 out-degree will be linked to the sink, which is shown in Fig. 7.4(b), where bold edges in orange are the new edges connected with the dummy nodes, i.e. source and sink in red and green respectively. The newly constructed graph is fed into the next iteration. As the number of tracks is unknown, the iterative algorithm will stop when all nodes in the initial graph have been deleted, which means every potential colloids' path has been found.



(a) Input directed graph (b) Updated graph during SSP

Figure 7.4: Illustration of our tracking scheme by iteratively finding the min-cost path with SSP and tag-then-delete procedure. In (a), the bold blue edges are part of an optimal path found by the SSP among all edges (edges in blue and yellow); in (b), the dash edges in blue and some edges in yellow are tagged and deleted, some new edges in orange are added to connect the node directly with dummy nodes

In summary, our tracking pseudo-code is presented in Algorithm 3. We first define the residual graph $G_r(x)$, which has the same nodes as the original graph G, but has reversed edges with capacities $c_{uv}{}' = c_{uv} - f_{uv}$, and with the negative of their original cost. The computational complexity is $O(|P|(|E| + |V|\log|V|))$. In each iteration, we use Dijkstra's algorithm to find the shortest path in $O(|E|+|V|\log|V|)$, and we find in total $|P|$ paths.

Algorithm 6: Finding min-cost paths with SSP on dynamic graph

Input: a directed graph $G = (V, E)$.
While $\sim isempty(V)$ except the s and t **do**
  Step I: initialize graph G, and compute its residual graph G^*;
  Step II: find min-cost paths
    while $K < |E|$ **do**
      (i) $\forall u \in V$, computer the shortest distance $d(u)$ from s, by $Dijkstra's\ algorithm$ with cost $\omega_{uv}^{(K)}$;
      (ii) update each edge cost $\omega_{uv}^{(K+1)}$, $\forall e_{uv} \in E$ by

$$\omega_{uv}^{(K+1)} = \omega_{uv}^{(K)} + d(u) - d(v) \tag{7.9}$$

      (iii) **if** $\exists p^K$ (p^K is a min-cost path) **then** push flow from s to t;
      else
      break;;
      end
      $K = K + 1$
  end
  Step III: do depth-first-search to find every path from s to t;
  Step IV: tag nodes V_l in path set $P = \{p^K\}$ and delete the nodes and their corresponding edges E_l;
  Step V: construct a new graph G with nodes $V = V \setminus V_l$ and $E = E \setminus E_l$;
end
Output: All tracks P

7.4 Experiments and Discussion

7.4.1 Benchmark

Data acquisition: the active particles in all videos are homemade spherical gold colloids of radius $\alpha \simeq 1.1 \pm 0.1$ µm, half covered with platinum. In the presence of hydrogen peroxide, the particles self-propel consuming H_2O_2 under a self-phoretic motion (a combination of diffusiophoresis and self-electrophoresis) [Palacci et al., 2010]. The active system is observed with an inverted optical microscope and a Hammatsu Orca-ER camera. Each video contains 500 frames (taken at 1 fps), and each frame's size is 1024×1344 pixels. We already mentioned in the introduction that a better acquisition system can provide a better temporal resolution and images with less noise and less intensity inhomogeneities. The purpose of this paper is to propose a general algorithm which works also in more difficult situations.

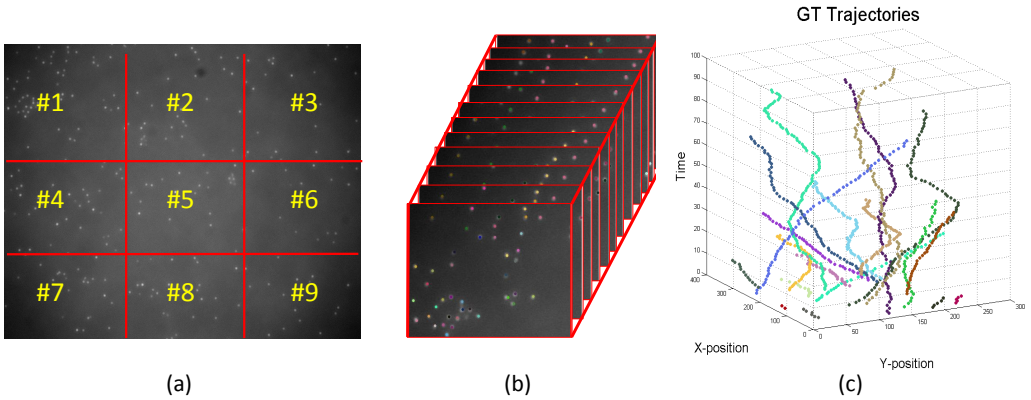


Figure 7.5: Illustration of ground-truth. (a) 9 subvideos (out of 16) extracted from the original large scale video; (b) Stack of frames in subvideo #1 annotated by human observers; (c) 3D visualization of a few trajectories tracked by observers in subvideo #1.

Ground-truth generation: the ground-truth is obtained from the manual tracking of particles in subvideos. More precisely, we divide the first large-scale video into 16 small subvideos of size 256×336 pixels per frame. The ground-truth is generated as follows:

- each subvideo is labeled by various observers in order to take into account that tracking results may differ between various observers.
- each time, only one colloid is tracked across all 500 frames, as the ability of human vision system to track multiple objects is limited [Pylyshyn and Storm, 1988].
- each observer (graduate-level student majored in computer science) is not instructed nor has prior knowledge by watching the video before tracking.

We illustrate the process in Fig.7.5, which presents the ground-truth generation and the sub-videos' annotation result in 3D. More precisely, 9 subvideos are randomly chosen out of 16 extracted from the original sequence. The parameters of each subvideo are provided in Table 1. In average, each subvideo has a cumulative number of 9587 colloids (adding all colloids in all frames, i.e. without any identification of identical colloids across frames) and 268 trajectories. In total, we have labeled 2408 meaningful trajectories extracted from a cumulative number of 86287 colloids recovered from 9 different videos by 9 different observers.

Table 7.1: Parameters of ground-truth

Labeled videos	Total colloids	Mean colloids per frame	Total trajectories	Mean/min/max trajectory length
# 1	12443	25	352	35/1/283
# 2	14479	29	358	40/1/361
# 3	11571	23	318	36/1/212
# 4	7974	16	236	34/1/145
# 5	9802	20	248	40/1/500
# 6	8420	17	253	33/1/173
# 7	8561	17	258	33/1/159
# 8	6437	13	198	32/1/105
# 9	6600	13	187	35/1/146
Av.	9587	19	268	36/1/232
Total	86287	-	2408	-

7.4.2 Evaluation Metrics

In this paper, following common practice (e.g. [Benfold and Reid, 2011][Berclaz et al., 2011]), our results are evaluated using the standard CLEAR MOT metrics [Keni and Rainer, 2008], listed as follows:

- The multiple object tracking Precision (MOTP): the ratio of the number of track switches over the number of objects present in all frames,

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{7.10}$$

  where c_t is the number of matches found for time t. For each of theses matches, d_t^i is the distance between the object o^i and its corresponding hypothesis.

- The Mismatch ratio (MM): the ratio of the number of track switches over the number of objects present in all frames,

$$\overline{mme} = \frac{\sum_t mme_t}{\sum_t g_t} \tag{7.11}$$

where mme_t is the number of mismatch errors for time t. g_t be the number of objects present at time t.

- Multiple Object Tracking Accuracy (MOTA): the final score to summarize tracking,

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{7.12}$$

m_t, fp_t, mme_t represent respectively the number of misses, of false positives, and of mismatches respectively for time t.

MOTP evaluates the alignment of tracks with the ground truth, and MOTA produces a score based on the amount of false positives, missed detections and identity switches. Note that during constructing object-hypothesis, the distance between object o_i and hypothesis h_j should not exceed the threshold, 15 pixels. In other words, the maximal observed difference between computed and hand-marked cell positions was 15 pixels, i.e. less than the search range.

7.4.3 Detection and Analysis

In this part, we compare the proposed method, i.e. the combination of a level set method with the circular Hough transform, with four other standard baselines which are frequently used techniques for particle detection. As summarized in [Chenouard et al., 2014, Smal et al., 2010b], we choose to use the adaptive threshold method [Otsu, 1975], the local minimum/maximum detection, and the wavelet based method used in [Padfield et al., 2011] as well as the particle detection method proposed in [Sbalzarini and Koumoutsakos, 2005]. Let us briefly introduce them.

In the experiments, we use the Otsu's method to decide the optimal threshold for the image after performing the background subtraction. We model the background by simply averaging over all video frames. Before we perform the local minimum/maximum algorithm, we also subtract background and denoise the result with Gaussian kernel. In [Padfield et al., 2011], they choose to use wavelets based method to decompose the image into both the spatial and frequency domain. Such decomposition enables the images to be directly denoised in the wavelet coefficient space and the final segmentation can be obtained from the correlation stack for coefficients greater than zero and needs no post-processing. The segmentation parameters for the lower scales and upper scales are tuned according to the rules introduced in [Padfield et al., 2011]. Note that the code can be found online at `http://www.dirkpadfield.com/papers`. In [Sbalzarini and Koumoutsakos, 2005], their proposed particle detection algorithm mainly consists of four steps: image restoration, estimation and refinement of the particle position and non-particle discrimination. The parameters relevant for detection are: (1) radius: approximate radius of the particles, $radius = 4$ pixels; (2) cutoff: the score cut-off for the non-particle discrimination, $cutoff = 0$; and (3) percentile: the percentile which determines which bright pixels are considered as particles, $percentile = 0.7$. Note that we turn the parameter to get reasonable performance with the software online available at `http://mosaic.mpi-cbg.de/?q=downloads/imageJ`.

The quantitative results of detection are presented in Table 7.2, which shows that our method gain more accuracy in terms of MOTP criterion compared with other standard baselines. The visual comparison results are shown in the Fig.7.6 and Fig.7.7. The adaptive threshold technique and wavelet based method [Padfield et al., 2011] can obtain reasonable good segmentation (see in Fig.7.6 (d)) where there aren't cluttered colloids, otherwise they fail to successfully separate the colloids individually (shown in Fig.7.6 (c) and (g)). The local minimum/maximum finding technique, is prone to be sensitive to noise of the image. Therefore, we set up a threshold in order to cut off non-particle detection (shown in Fig.7.6 (e) and (f)). The detection generated by method in [Sbalzarini and Koumoutsakos, 2005] shows its weakness in detecting objects in low-resolution background, which leads to miss the detections shown in the Fig. 7.7(c) and (d).

Table 7.2: Quantitative evaluation of different detection methods measured by MOTP.

Method	Otsu	local maximum	Padfield et al.	Sbalzarini and Koumoutsakos	ours
# 1	**0.8975**	0.8950	0.8793	0.8929	0.8969
# 2	0.8920	0.8977	0.8741	0.89	**0.8996**
# 3	0.9032	0.9006	0.8829	0.8871	**0.9058**
# 4	**0.8661**	0.8635	0.8542	0.8506	0.8646
# 5	0.8627	0.8637	0.8356	0.8545	**0.8661**
# 6	0.8434	**0.8444**	0.8315	0.8327	0.8427
# 7	0.9064	0.9031	0.8886	0.8907	**0.9085**
# 8	**0.8654**	0.8642	0.8526	0.8563	0.8645
# 9	0.8218	0.8248	0.8135	0.8153	**0.8254**
AV	0.8731	0.8730	0.8569	0.8633	**0.8749**

7.4.4 Tracking and Analysis

7.4.4.1 Graph Refinement Validation

First we evaluate the influence of the three refinement operations separately as well as their different combinations. In Table 7.3, each type of proposed refinement is proved useful quantitatively, compared to the case without any refinement operation, the performance goes from 94.03% to 94.13%, 94.17%, and 94.23% respectively. Moreover, combining two or all of them further leverages accuracy to 94.45%, which means that the three types of refinements are complementary. We also investigate in Table 7.4 the influence of different orders of refinement operations, and it follows from the results that the order does not affect significantly the final performance. The reason lies in the fact that we perform the three configurations iteratively.

7.4.4.2 Comparison with standard baselines

To compare systematically, we report results on nine annotated video sequences with five standard baselines: the Kalman's filter [Blackman, 1986], Dijkstra's short-

Table 7.3: Quantitative evaluation of the efficiency of the three types of refinement, **PI**, **PII** and **PIII**. $\oplus$ and $\ominus$ mean with/without operator.

Refinement			Metric
PI	**PII**	**PIII**	MOTA% ↑
$\ominus$	$\ominus$	$\ominus$	94.03%
$\oplus$	$\ominus$	$\ominus$	94.13%
$\ominus$	$\oplus$	$\ominus$	94.17%
$\ominus$	$\ominus$	$\oplus$	94.23%
$\oplus$	$\oplus$	$\oplus$	94.45%

Table 7.4: Quantitative evaluation of different combinations of the three types of refinement, **PI**, **PII** and **PIII**. Numbers 1,2 3 mean 1st, 2nd,3th order.

Refinement			Metric
PI	**PII**	**PIII**	MOTA%↑
1	2	3	94.45%
1	3	2	94.45%
2	1	3	94.45%
2	3	1	94.45%
3	1	2	94.45%
3	2	1	94.45%

est path algorithm [Dijkstra, 1959a, Jiang et al., 2013], the Nearest neighbor, the Hungarian [Kuhn, 1955] and the [Sbalzarini and Koumoutsakos, 2005]. The first baseline algorithm is the Kalman's filter [Blackman, 1986], which recursively estimates the state of a process by minimizing the mean of the squared error. The Kalman filter model assumes the true state at time t is evolved from the state at $(t-1)$ according to:

$$\mathbf{x_t} = \mathbf{A_t}\mathbf{x_{t-1}} + \mathbf{B_t}\mathbf{u_t} + \mathbf{w_t} \tag{7.13}$$

and at time t an observation (or measurement) $\mathbf{z_t}$ of the true state $\mathbf{x_t}$ is made according to:

$$\mathbf{z_t} = \mathbf{H_t}\mathbf{x_t} + \mathbf{v_t} \tag{7.14}$$

where, $\mathbf{A_t}$ is the state transition model, $\mathbf{B_t}$ is the control-input model and $\mathbf{w_t}$ is the process noise, which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $\mathbf{Q}$, $\mathbf{w_t} \sim N(0, \mathbf{Q})$, and $\mathbf{H_t}$ is the observation model, $\mathbf{v_t}$ is the observation noise, $\mathbf{v_t} \sim N(0, \mathbf{R})$. In the experiment, we set $\mathbf{A_t} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $\mathbf{B_t} = \begin{bmatrix} dt^2/2 \\ (dt^2/2) \\ dt \\ dt \end{bmatrix}$, $\mathbf{Q} = \begin{bmatrix} dt^4/4 & 0 & dt^3/2 & 0 \\ 0 & dt^4/4 & 0 & dt^3/2 \\ dt^3/2 & 0 & dt^2 & 0 \\ 0 & dt^3/2 & 0 & dt^2 \end{bmatrix}$,

$\mathbf{H_t} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, and $\mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$.

The second baseline algorithm is the algorithm proposed in [Jiang et al., 2013], which first constructs a directed graph associating all frames and then iteratively uses Dijkstra's shortest path algorithm until a specified number of iterations is reached or no more optimal paths can be extracted from the graph. In the experiments, to conduct fair comparison, the maximum displacement of a colloid moving between consecutive frames is fixed with the same of ours, naming $d_{max} = 25$ pixels. All the nine videos use the same detection results proposed in this paper, the only difference is the tracking algorithm.

The third and fourth algorithms are the nearest neighbor algorithm and the Hungarian algorithm [Kuhn, 1955]. Both of them recover the trajectories of colloids by first doing a frame-to-frame linking by virtue of either nearest neighbor or the Hungarian algorithm, and then do a second iteration which is to link the partial trajectory with its subsequent trajectory. Their code is available online at `http://fr.mathworks.com/matlabcentral/fileexchange/34040-simple-tracker`. In addition, during the second iteration, they also consider gap-closing, which is a link spanning between multiple frames to restore the track. Note that, the Hungarian algorithm provides guarantee that the sum of the pair distance (measured by Euclidean distance) is minimized over all colloids between two frames. The complexity of this algorithm is in $O(n^3)$, which can be prohibitive for problems with a huge number of objects in each frame. In this case, it is prone to use the nearest neighbor, which runs in $O(n^2)$, although it only achieves a local optimum for a pair of colloids.

The fifth baseline proposed in [Sbalzarini and Koumoutsakos, 2005] find the trajectories also within two or more frames, where the optimal set of associations is solved by minimizing a cost functional. In practice, their implementation is based on a particle matching algorithm [Dalziel, 1992] using a graph theory technique. In experiments, the maximum displacement is set 25 pixels also, the number of subsequent frames that are considered to find the optimal association is two frames.

The quantitative comparison results are reported in the Table 7.5. Our proposed tracking scheme has the best performance over all videos with respect to the MOTA score, and at the same time, has the smallest mismatch measured by MM, compared with the other standard baselines. In average, our method gains large margins compared with the Kalman, the Dijkastra shortest path used in the [Jiang et al., 2013], and the [Sbalzarini and Koumoutsakos, 2005] by both measurements of MM and MOTA. The performances of the nearest neighbor and the Hungarian are quite close to ours, their average accuracies measured by the MOTA are 92.93% and 92.99% respectively, which is close to our average accuracy 93.19%.

In addition, we also compare visually the results obtained by different algorithms to show the performance in a particular context where a small cluster is moving slowly and meanwhile encounters with other individual colloids. As shown from Fig.7.8 to Fig.7.14, we show only a few trajectories, but each illustration is finished at the end frame of trajectories, for example, in Fig.7.8, all trajectories are finished at the 126th-frame. Besides, we also show all the trajectories in the three-dimensional illustration located at the last of each figure. As observed in Fig.7.9, the Kalman's filter yields identity switches errors with long-distance colloids (note the large gap

Table 7.5: Quantitative comparison between five standard tracking methods and our algorithm on 9 small video samples.

Method	Kalman		Jiang et al.		*Nearest Neighbor*	
Evaluation	MM% ↓	MOTA%↑	MM%↓	MOTA%↑	MM%↓	MOTA%↑
# 1	1.20%	92.04%	3.75%	90.61%	0.35%	94.35%
# 2	1.57%	82.95%	4.50%	90.46%	0.43%	95.20%
# 3	1.48%	82.90%	3.22%	82.80%	1.56%	91.18%
# 4	0.74%	93.53%	3.50%	92.53%	0.51%	96.25%
# 5	1.15%	88.59%	2.68%	86.40%	0.48%	90.77%
# 6	0.78%	91.60%	2.41%	87.00%	0.90%	94.03%
# 7	0.81%	91.78%	3.38%	86.72%	1.30%	93.47%
# 8	0.79%	89.90%	2.81%	85.29%	1.52%	91.28%
# 9	2.64%	80.35%	2.65%	76.62%	2.34%	89.89%
Average	1.24%	88.18%	2.82%	86.49%	1.04%	92.93%
Method	*Hungarian*[Kuhn, 1955]		Sbalzarini and Koumoutsakos		**Ours**	
# 1	0.33%	94.36%	1.38%	85.24%	0.26%	**94.45%**
# 2	0.32%	**95.31%**	1.20%	86.12%	0.37%	95.26%
# 3	1.37%	91.38%	0.91%	88.98%	0.70%	**92.05%**
# 4	0.44%	**96.32%**	1.76%	82.22%	0.30%	94.46%
# 5	0.48%	90.77%	0.66%	87.87%	0.43%	**90.83%**
# 6	0.80%	94.13%	0.83%	89.19%	0.39%	**94.54%**
# 7	1.31%	93.49%	0.89%	86.11%	0.62%	**94.17%**
# 8	1.43%	91.37%	1.23%	73.62%	0.47%	**92.34%**
# 9	3.40%	89.84%	2.00%	75.68%	1.65%	**90.59%**
Average	1.09%	92.99%	1.20%	83.89%	**0.58%**	**93.19%**

in the purple-colored path). A relatively high number of fragments and identity switches can be observed from the method proposed by [Jiang et al., 2013] shown in Fig.7.10 at the same time, leaving colloids tend to link with entering colloids, which leads to very long-range paths (e.g. the red, blue and green ones). These one-pass greedy algorithms do not work well when there is target interaction in a scene. In contrast, the nearest neighbor, the Hungarian algorithm [Kuhn, 1955], and our modified min-cost/max flow algorithm can recover the correct trajectories even in a dense scene as shown in Fig.7.11, Fig.7.12 and Fig.7.14 respectively. One reason for our tracking method is the use of unit capacity constraint for each edge and the global optimization algorithm solved by the modified SSP.

Finally, we present visual results of colloids in highly dense context shown in Fig.7.15. We can observe that our proposed method can detect colloids individually correctly and track them without identity switch even in such highly dense and cluttered video.

Note that the ground-truth videos used in the experiments, as well as the tracking results obtained with either our algorithm or the methods presented in [Blackman, 1986, Jiang et al., 2013], are available in a frame-by-frame format at `http://math.univ-lyon1.fr/~masnou/colloids`.Together with these ground truth videos, we provide the segmentation and tracking result for a video with a middle dense population of colloids.

7.4.5 Code and computational time

In the previous part, extensive experiments were presented to evaluate the efficiency of our method. The segmentation step is implemented as a Matlab routine, and the tracking step combines C++ and matlab implementations. The code is ran on a standard computer (Intel Xeon 3.3GHz CPU with 16G memory). For each subvideo (frame size 256×336, 500 frames), the computational time was approximately 28mins for the segmentation, 2s for the graph construction, and 0.5s for the iterated min-cost/max-flow algorithm. Improving the computational burden for segmentation is the purpose of future research.

7.5 Conclusion

In this chapter, an efficient and reliable framework has been proposed to detect and track individual colloids in a long-term video sequence. First, all colloids are detected individually by combining a region-based level set method with the circular Hough transform. Then, all meaningful trajectories are recovered from a trellis graph (which iterative refinements) using an iterative optimization algorithm. The min-cost/max flow algorithm guarantees the global optimum at each iteration. The combination of a tag-then-delete method and the successive shortest paths algorithm enables to find all colloids' paths simultaneously with higher accuracy, compared with five state-of-the-art methods, according to MOTA and MM criteria on annotated real videos.

Acknowledgment

The authors would like to thank Bingxue Zhang, Doming Chen, Huaxiong Ding, Huiliang Jin, Ying Lv, Yuxing Tang, Yinghang Tang, Cheng Wang, and Wuming Zhang for contributing their precious time to annotate each colloid in the video sequences presented in the paper. In particular, many thanks to Xiaoyan Jiang, for her fruitful suggestions on our work, and for having provided the results for 9 video sequences using the algorithm proposed in [Jiang et al., 2013].

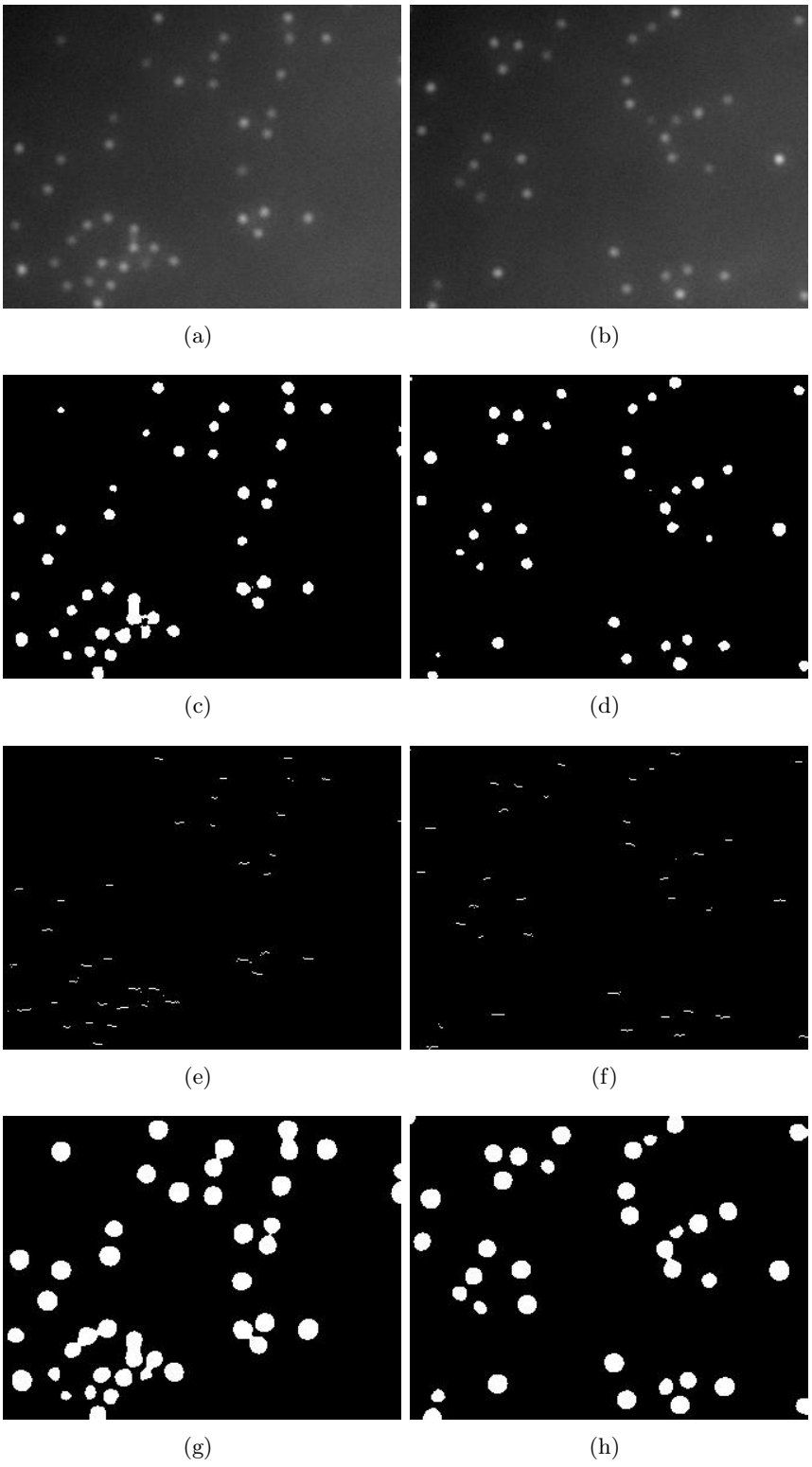


Figure 7.6: Visual comparison of different particle detection methods. (a, b) show two original video frames. (c, d), (e, f), (g, h) are the result obtained by the Otsu's threshold [Otsu, 1975], local maximum detection method, and wavelet based method [Padfield et al., 2011] respectively.

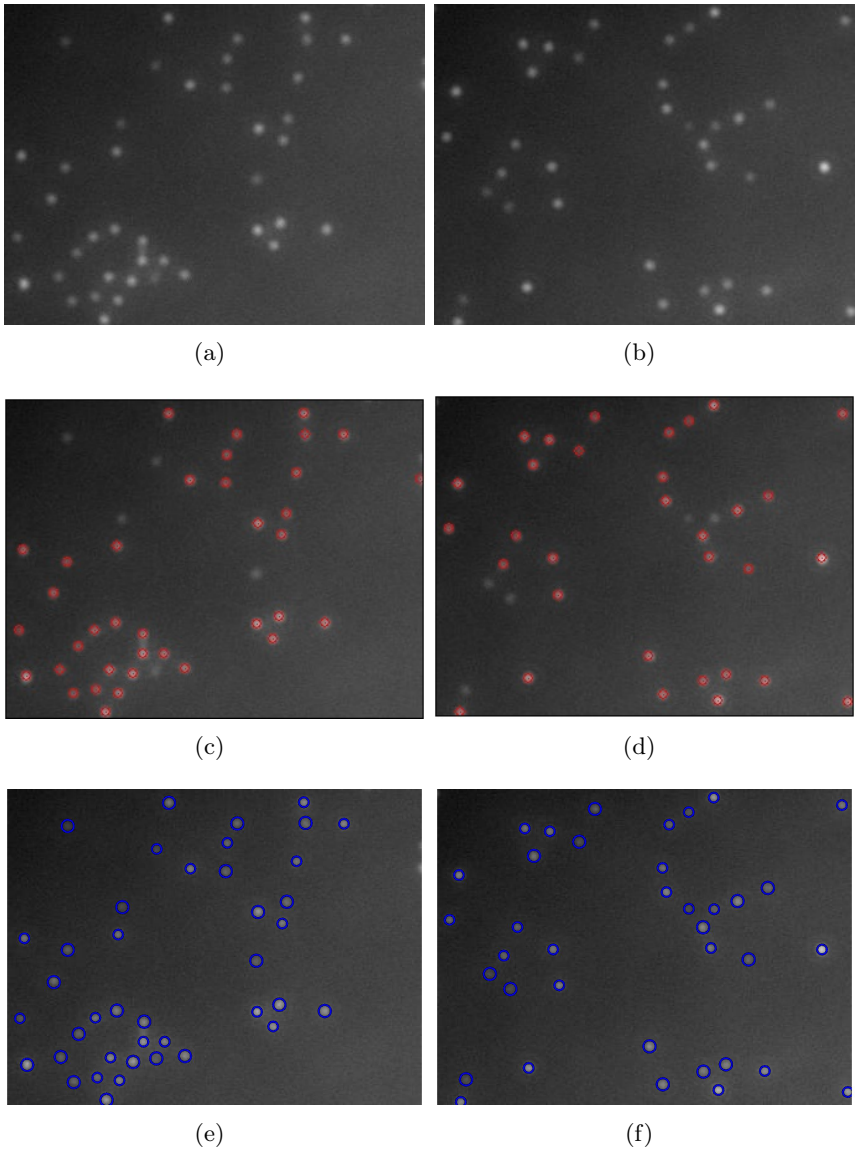


Figure 7.7: Visual comparison of different particle detection methods. (a, b) show two original video frames. (c, d), (e, f) are the result obtained by the method proposed in [Sbalzarini and Koumoutsakos, 2005] and our method respectively.

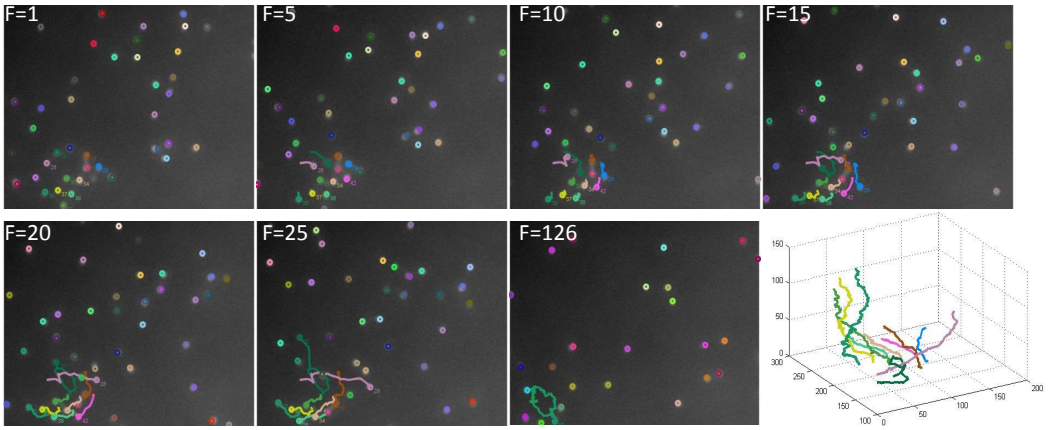


Figure 7.8: Illustration of cluttered colloid's tracking: ground truth.

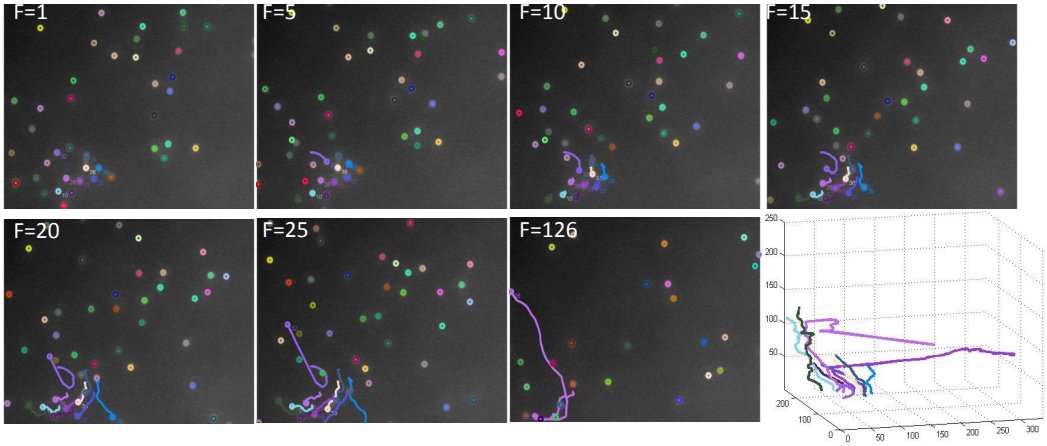


Figure 7.9: Illustration of cluttered colloid's tracking: Kalman filter [Blackman, 1986].

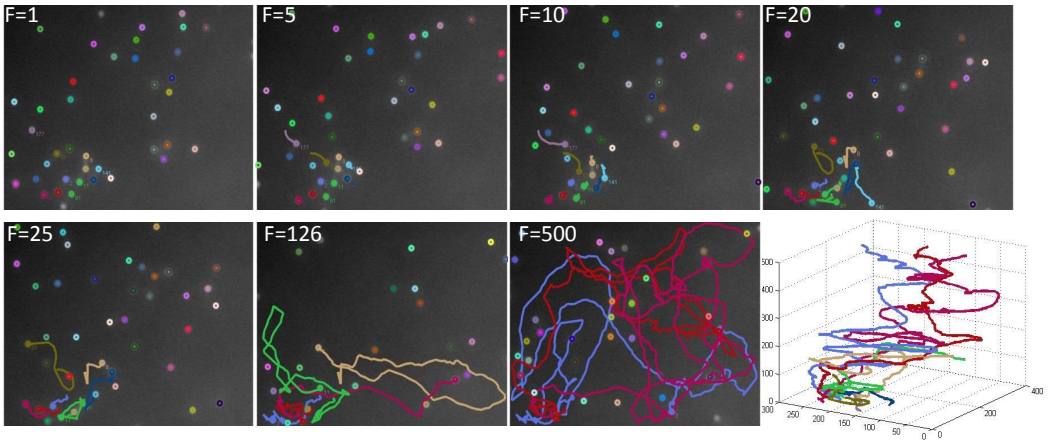


Figure 7.10: Illustration of cluttered colloid's tracking: Dijkstra shortest path used in [Jiang et al., 2013].

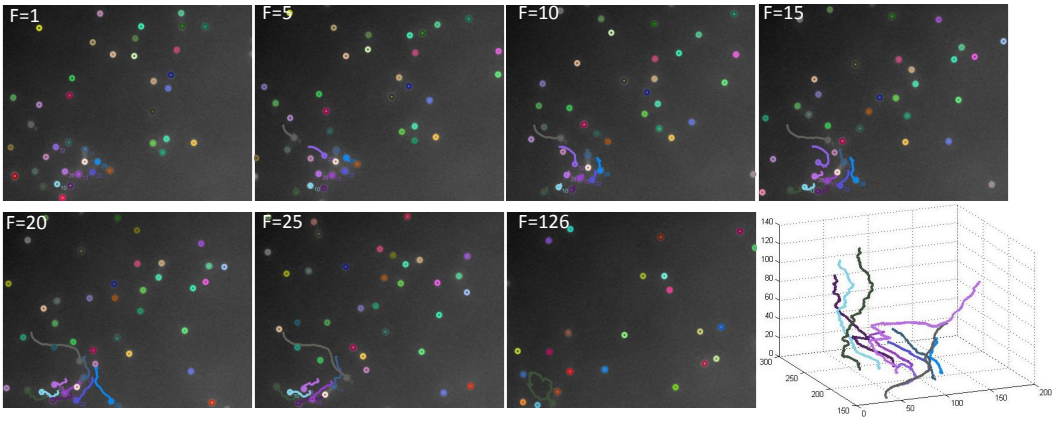


Figure 7.11: Illustration of cluttered colloid's tracking: Nearest neighbor linking.

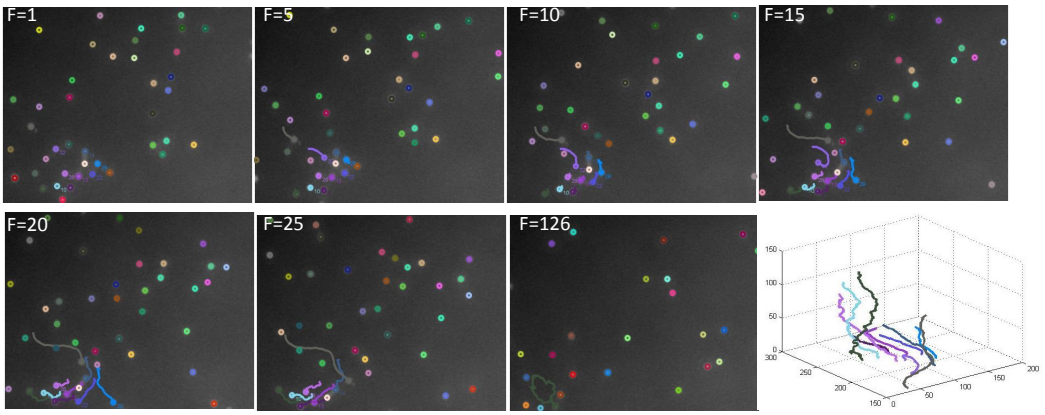


Figure 7.12: Illustration of cluttered colloid's tracking: Hungarian algorithm [Kuhn, 1955].

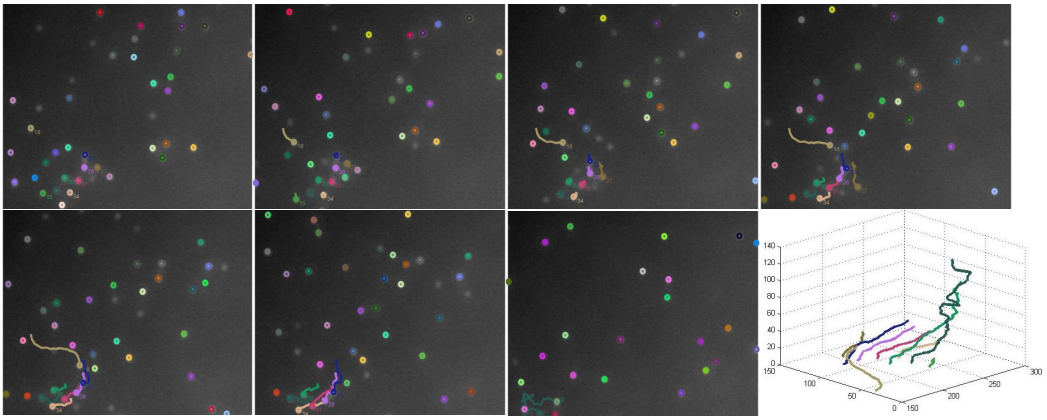


Figure 7.13: Illustration of cluttered colloid's tracking: track algorithm proposed in [Sbalzarini and Koumoutsakos, 2005]

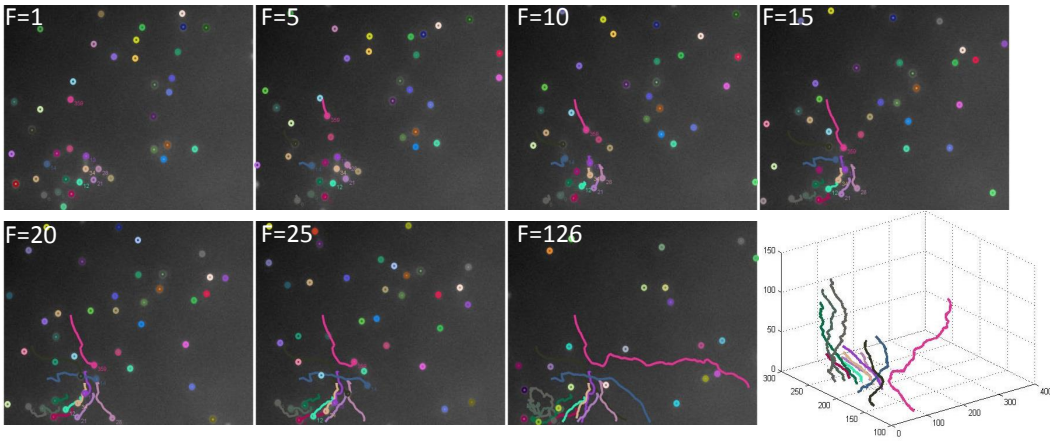
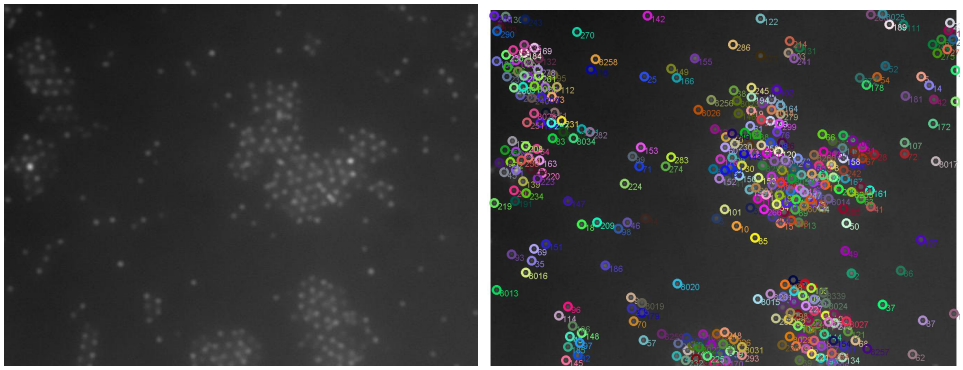
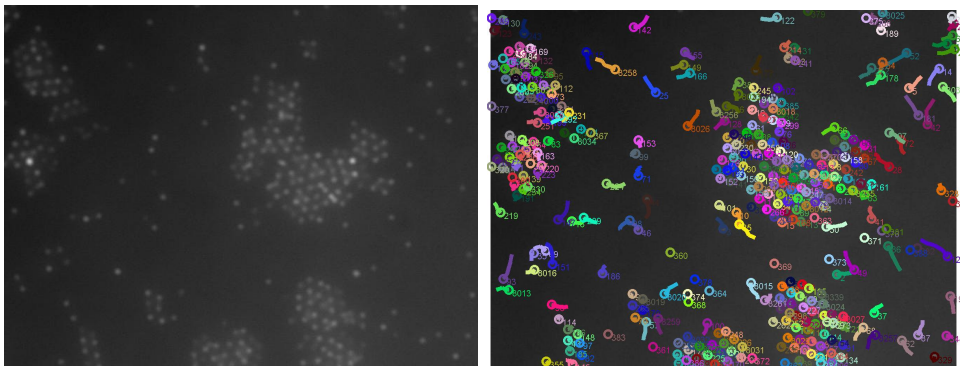


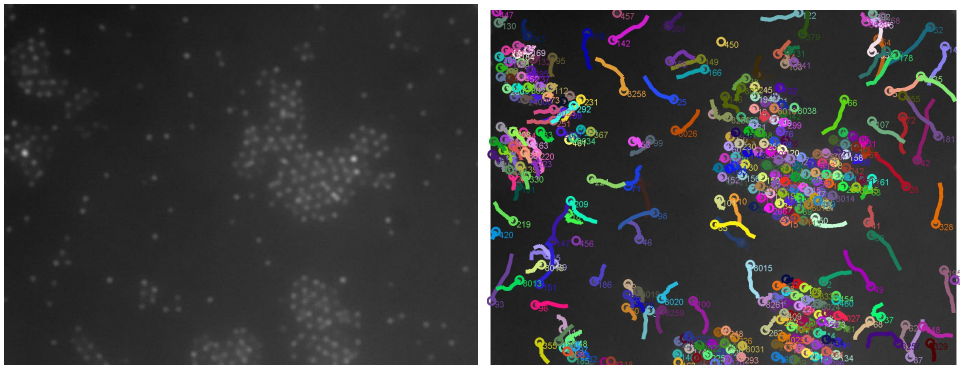
Figure 7.14: Illustration of cluttered colloid's tracking: our proposed method.



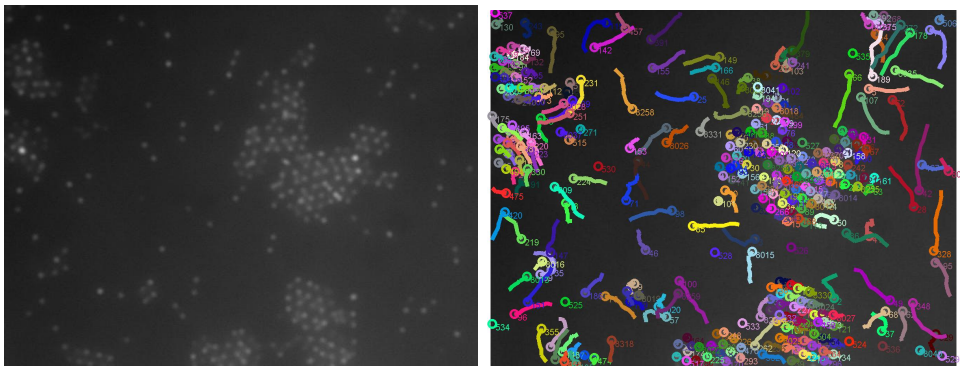
(a) Frame1 (b) Frame1



(c) Frame5 (d) Frame5



(e) Frame10 (f) Frame10



(g) Frame15 (h) Frame15

Figure 7.15: Illustration of results of colloids detection and tracking in highly dense video.

Chapter 8

Conclusions and Future Work

Contents

This thesis mainly concentrates on the application of graph theory and develops algorithms based on it to solve two classical computer vision problems: image segmentation and multi-target tracking. In chapter 1, we give a general introduction of the contexts, motivations, objectives, and contributions. In chapter 2, we introduce the basic definitions in graph-theoretical methods and review graph construction algorithms and the graph partitioning methods. In chapter 3, we review the previous work on image segmentation, since the algorithms presented in this dissertation are mainly on this subject. Our main research works presented in the thesis can be divided into three parts. Part 1 (Chapter 4) deals with the graph construction. To combine adaptively the local and global image structure information, we propose a novel graph construction algorithm taking account of nice properties, e.g. spatial consistency, sparsity, long-range grouping cues. Part II (Chapter 5 and 6) focus on the topic of developing powerful and discriminative grouping cues. In chapter 5, to include color and neighborhood structure as well as avoid the over-smooth effect, we propose a new feature descriptor weighted color patch, which is further used to compute the weight of edges in graph construction. In chapter 6, we propose a graph-based unsupervised segmentation approach that combines superpixels, sparse representation, and a new mid-level feature to describe superpixels. Part III (Chapter 7) deals with the multi-target tracking problems in active colloids systems. We propose an efficient and robust framework to jointly detect and track each colloid in a long-term video sequences.

8.1 Contributions

The contributions in this thesis are as follows.

8.1.1 A Global/Local Affinity Graph for Image Segmentation

To construct a high-quality graph in the graph-cut based image segmentation methods, we propose a novel sparse global/local affinity graph over superpixels of an input image to capture both short and long range grouping cues, thereby enabling perceptual grouping laws, *e.g.*, proximity, similarity, continuity, to enter in action through a suitable graph cut algorithm. Moreover, we also evaluate three major visual features, namely color, texture and shape,for their effectiveness in perceptual segmentation and propose a simple graph fusion scheme to implement some recent findings from psychophysics which suggest combining these visual features with different emphases for perceptual grouping. Specifically, an input image is first oversegmented into superpixels at different scales. We postulate a gravitation law based on empirical observations and divide superpixels adaptively into small, medium and large sized sets. Global grouping is achieved using medium sized superpixels through a sparse representation of superpixels' features by solving a ℓ_0-minimization problem, thereby enabling continuity or propagation of local smoothness over long range connections. Small and large sized superpixels are then used to achieve local smoothness through an adjacent graph in a given feature space, thus implementing perceptual laws, *e.g.*, similarity and proximity. Finally, a bipartite graph is also introduced to enable propagation of grouping cues between superpixels of different scales. Extensive experiments are carried out on the Berkeley Segmentation Database in comparison with several state of the art graph constructions. The results show the effectiveness of the proposed approach which outperforms state of the art graphs using 4 different objective criteria, namely PRI, VoI, GCE and BDE.

8.1.2 Graph-based Image Segmentation Using Weighted Color Patch

To construct a discriminative affinity graph in graph-based image segmentation, we propose a new method based on the weighted color patch to compute the weight of edges in an affinity graph. The proposed method intends to incorporate both color and neighborhood information by representing pixels with color patches. Furthermore, we assign both local and global weights adaptively for each pixel in a patch in order to alleviate the over-smooth effect of using patches. The normalized cut algorithm is then applied on the resulting affinity graph to find partitions. We evaluate the proposed method on the Prague color texture image benchmark and the Berkeley image segmentation database. The extensive experiments show that our method is competitive compared to the other standard methods using multiple evaluation metrics.

8.1.3 Sparse Coding and Mid-Level Superpixel-Feature for ℓ_0-Graph Based Unsupervised Image Segmentation

We propose a graph-based unsupervised segmentation approach that combines superpixels, sparse representation, and a new mid-level feature to describe superpixels. We first extract a set of interest points either by sampling or using a local feature detector, and we compute a set of low-level features associated with the patches centered at the interest points. A low-level dictionary is defined as the collection of all these low-level features. We call superpixel a region of an oversegmented image obtained from the input image, and we compute the low-level features associated with it. Then we compute for each superpixel a mid-level feature defined as the sparse coding of its low-level features in the aforementioned dictionary. These mid-level features not only carry the same information as the initial low-level features, but also carry additional contextual cue. We use the superpixels at several segmentation scales, their associated mid-level features, and the sparse representation coefficients to build graphs at several scales. Merging these graphs leads to a bipartite graph that can be partitioned using the Transfer Cut algorithm. We validate the proposed mid-level feature framework on the MSRC dataset, and the segmented results show improvements from both qualitative and quantitative viewpoints compared with other state-of-the-art methods.

8.1.4 Active Colloids Tracking: Recover Trajectories Globally via Min-cost/max Flow

To track massive colloids' trajectories independently for the study of the active suspension system, we propose a new detect-then-track method. First, a region based level set method is adopted to segment all colloids from long-term video sequences. Moreover, the circular hough transform further refines the segmentation to obtain colloid individually. Second, we propose to recover all the colloids' trajectories simultaneously, which is a global optimal problem and can be solved efficiently with the min-cost/max flow. A high-quality graph construction strategy guarantees the final tracking result. We first construct an initial graph using simple yet effective measurement such as colloid displacement, and further resolve local conflicts introduced by the unambiguity of initial graph with additional constraints. A modification of min-cost/max flow algorithm combined with iterative tag-then-delete is proposed to recover all colloids' trajectories simultaneously. Finally, we evaluate the proposed framework on a real benchmark with annotations. Extensive experiments demonstrate that the proposed framework outperforms standard state-of-the-art methods with large margin measured by CLEAR MOT.

8.2 Perspectives for Future Work

8.2.1 Image Segmentation

Image segmentation has been a fundamentally studied topic and continues to attract intensive interest in computer vision. After a short period time, when part of the recognition community lost confidence in bottom-up segmentation for its ill-posed

property, it becomes a major trend to apply the results of bottom-up segmentation as a first step in many high-level vision tasks. These methods provide high-quality and category-independent object candidates, which can then be described with richer representations and used as input to more sophisticated learning methods. Recently, this paradigm has dominated the PASCAL segmentation challenge [Carreira and Sminchisescu, 2010] [Carreira et al., 2012a] [Arbeláez et al., 2012], leveraged object detection [Alexe et al., 2010] [Uijlings et al., 2013] [Girshick et al., 2013][Wang et al., 2013d][Manen et al., 2013] and demonstrated competitive in large-scale classification [Uijlings et al., 2013].

Graph based methods build the basis of many state-of-the-art methods in image and video segmentation. These methods gain popularity for well-sound mathematical structure and efficient discrete optimization techniques as well as the flexible data representation ability. As consequence, methods using graph have been a dominant paradigm to generate state-of-the-art performance on popular benchmark dataset. We present possible extensions and future directions.

8.2.1.1 Multiple Cues Combination

For general-purpose segmentation, it is insufficient to use a single feature to describe all type of image perceptual properties. At the same time, various feature descriptors are proposed in literature possessing different advantages in color, gradient or texture etc. Consequently, optimizing multiple grouping cues in the same framework is an essentially interesting area in bottom-up segmentation. For graph oriented methods, the affinity graph plays essentially important role for the quality of result by graph partitioning technique. A good combination of different features can improve the accuracy due to its ability of capturing key perceptual grouping property.

It will be a very valuable research if one can design a general and fair comparison framework to comprehensively understand the performance of various feature descriptors based on graph weight computation. As a preliminary extension, based on the idea presented in [Wang et al., 2013a], we investigate various feature descriptors including color, texture, and other appearance cues. Several works have been proposed toward this direction also. For instance, in the global probability boundary (gPb) [Arbelaez et al., 2011] algorithm, brightness, color and texture gradients at three fixed disk sizes are first computed. These local contour cues are globalized using spectral graph-partitioning, resulting in the gPb contour detector as state-of-the-art in BSDS. Cheng et al.[Cheng et al., 2011a] proposed a new solution to fuse multiple types of image features by seeking the sparsity-consistent low-rank affinities from the joint decompositions of multiple feature matrices into pairs of sparse and low-rank matrices.

8.2.1.2 Multi-scale Segmentations Combination

Multi-scale Segmentation have been frequently considered in order to design a robust and general framework. Combining scales from coarse to fine is a powerful processing strategy in computer vision, as a single hierarchy is not enough to get a sufficiently diverse set of regions but by using several ones can capture different meaningful object candidates. One example is the winner [Uijlings et al., 2013]

in the localization task of the ImageNet Large-Scale Visual Recognition Challenge 2011, it use several over-segmentations computed in several color spaces and obtain high recall which is a critical index for the object detection task. [Arbeláez et al., 2012] considered hierarchical segmentations at three different scales and combining pairs and triplets of adjacent regions from the two coarser scales to produce object region. [Arbeláez et al., 2014] proposed a hierarchical segmenter that leverages multiscale information and produces accurate object candidates by efficiently exploring the combinatorial space of our multiscale regions. For graph oriented methods, graph partitioning technique provides a natural globalization of the combination of multi-scale affinity graph, e.g. [Cour et al., 2005] [Kim et al., 2010b]. It is worth to mention that varying parameters of candidate oversegmentation methods can also obtain multi-scale superpixels, which are further used as mid-level grouping unites either in graph cut framework [Li et al., 2012] or conditional random field framework [Liu et al., 2013b], both being mathematical model to integrate the spatial coherency and across-scale consistency of multi-scale superpixels.

8.2.1.3 Multiple Algorithms Combination

Recently, the paradigm which treats bottom-up segmentation as input to more sophisticated task, e.g. object detection, has dominated the PASCAL challenges on segmentation and object detection [Carreira et al., 2012a][Carreira and Sminchisescu, 2012][Fidler et al., 2013] and it has been also proved very competitive in large-scale classification [Uijlings et al., 2013]. Motivated by potential fail of only one segmentation method, which may often lead to degrade performance substantially, several works have pointed out *"some segments in some of the segmentations appear to provide good spatial support objects"* [Malisiewicz and Efros, 2007]. For example, [Hoiem et al., 2005] [Russell et al., 2006] used multiple segmentations from an image, and treated them as hypotheses for object support rather than a single partitioning of the image for recognition and detection. [Malisiewicz and Efros, 2007] took one of the first steps towards combinatorial grouping, by running multiple segmenters with different parameters and merging up to three adjacent regions. As a preliminary extension, we proposed in [Wang et al., 2013b][Wang et al., 2013a] to construct an unified affinity graph by concatenating 5 to 6 scale of superpixels generated by MS [Comaniciu and Meer, 2002] and FH [Felzenszwalb and Huttenlocher, 2004] into an unified matrix diagonally.

8.2.2 Multi-target Tracking

Multi-target tracking is one of the most challenging problems in computer vision. Although much progress has been made in the last several years, it is still far from solved. Tracking-by-detection paradigm is frequently used to this task, these methods tend to be more robust as they can access all observations simultaneously. Many algorithms are proposed to solve the tracking problem in specific domain, and therefore algorithms can be very varied according to the track target. Nevertheless, we can point out some further improvement based on our observation in our preliminary work in Chapter 7. First of all, as its name tells, the tracking result surely

benefits from further advances in object detection which is largely an independent research area and is really ad-hoc. Some assumption can be learned from colloid' s properties and motion mechanism throughout the whole video sequences. For example, the prior knowledge round-shaped target can be further considered in the stage of segmenting (e.g. imposing as shape prior in the active contour model's energy function [Chan and Zhu, 2005]), and the smooth change in terms of mean gray level of colloids can indicate the motion towards the third direction, which can be a useful hint to detect the target even in severe inhomogeneous condition. Another improvement can be made in the efficiency of level set formation. For example, the convex formulations of $\varepsilon^{RSF}(C, f_1, f_2)$ in Chapter 7, in the spirit of [Chan et al., 2006] [Pock et al., 2008] could be used, and would decrease the computational time. Second, recovering trajectories from video sequences is in fact a combinatorial optimization problem of significant complexity. There exists many proposals to solve this problem, e.g. Hungarian algorithm [Kuhn, 1955] and k-shortest paths [Berclaz et al., 2011]. More recent methods [Zhang et al., 2008b] [Berclaz et al., 2011] have attempted to find globally optimal solutions across the entire sequence by creating network flow graphs, which can be solved optimally and efficiently by min-cost/max flow algorithms. In Chapter7, we also treat the tracking task as the min-cost/max flow problem. A further improvement can be made by considering integrating higher-order track smoothness constraints such as constant velocity. Although additional constrain to the original binary flow variables in the network graph leads to a problem, which cannot be solved by min-cost flow any more, [Butt and Collins, 2013b] proposed an iterative solution method that relaxes these extra constraints using Lagrangian relaxation, resulting in a series of problems that are solvable by min-cost flow.

Appendix A

Fusing Generic Objectness and Deformable Part-based Models for Weakly Supervised Object Detection

A.1 Introduction

Object detection/localization in images is one of the most widely studied problems in computer vision. For most of the existing methods, a fully supervised learning (FSL) approach is adopted [Dalal and Triggs, 2005, Felzenszwalb et al., 2010], where positive training images are manually annotated with bounding boxes encompassing the objects of interest. However, manual annotation for large-scale image database is extremely laborious and unreliable [Siva and Xiang, 2011b]. As a result, in contrast to the traditional FSL, there has been a great interest in weakly supervised learning (WSL) for object detection [Deselaers et al., 2010][Pandey and Lazebnik, 2011] [Crandall and Huttenlocher, 2006][Nguyen et al., 2009] [Siva and Xiang, 2011a][Deselaers et al., 2012][Siva et al., 2012], where the exact object locations in positive training examples are not provided, given only the binary labels indicating the presence or absence of the objects.

Deformable Part-based Models (DPM) [Felzenszwalb et al., 2010] and its variants [Azizpour and Laptev, 2012, Girshick et al., 2011], are the leading techniques to object detection with full supervision on the challenging PASCAL VOC datasets [Everingham et al., 2010]. The DPM represent an object with a coarse root filter that approximately covers an entire object and several higher resolution part filters that cover smaller parts of the object. In the standard (fully supervised) DPM framework, the positive ground-truth object bounding boxes are treated as the initial root filters, and it is allowed to move around in its small neighborhood to maximize the filter score. The locations of parts are treated as latent information as the annotations for parts are not available. Megha *et al.* [Pandey and Lazebnik, 2011] modify the fully supervised DPM to a weakly supervised one, without object-level annotations, by treating the location of root filter and part filters full latent, and learning structural object detectors based on the entire image (root filter location is initialized randomly based on a window which has at least 40% overlap with the positive training image, and its aspect ratio is initialized roughly to the average of the aspect ratios of positive training examples). However, the specific size and location of the initial root filter, as well as their aspect ratio are indicated to have a significant impact on the final localization result [Dalal and Triggs, 2005,

Felzenszwalb et al., 2010, Pandey and Lazebnik, 2011]. And to our best knowledge, methods for initializing the root filter as well as the definition of the aspect ratio of the objects in weakly supervised DPM, have not been well studied in [Pandey and Lazebnik, 2011]. To take advantage of the outstanding object detection performance

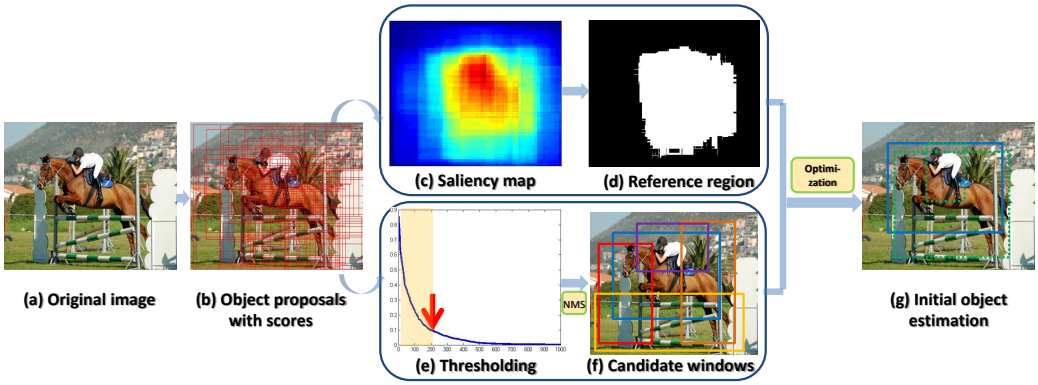


Figure A.1: Illustration of our proposed method to extract the initial object estimation: for an input image (a), 1000 object proposals (b) are sampled with corresponding scores to their probability to have object inside via the objectness measurement. (c) is the saliency map derived from (b), and (d) is the reference region obtained by thresholding (c). A finer set of candidate windows (f) are selected on the sorted proposals (e) by NMS. The blue window in (g) is our initial object estimation obtained by optimizing the overlap between (d) and (f).

of fully supervised DPM, in this paper, we propose a model enhancing the weakly supervised DPM by emphasizing the importance of location and size of the initial class-specific root filter. To be precise, we explore the objectness approach [Alexe et al., 2012], which generates class-independent object proposals with corresponding scores to their probabilities of being object windows, and adaptively extract a reliable window from the derived object proposals for each image as the initial root filter estimate for training DPM detector. Finally, a flexible enlarging-and-shrinking post-processing procedure is proposed to modify the predicted output of DPM detector, which can effectively generate more accurate bounding boxes by better conserving foreground and cropping out plain background regions. Experimental results on the challenging PASCAL VOC 2007 database demonstrate that our proposed framework is effective for initialization of root filter, and shows competitive final localization performance with the other weakly supervised object detection methods[Pandey and Lazebnik, 2011, Siva et al., 2012].

The rest of the chapter is organized as follows: we present our method to extract reliable initial root filter for weakly supervised DPM and our technique to post-process the bounding box in Section A.2, and in Section A.3 we present our experimental results and the comparison with other methods on PASCAL VOC 2007 datasets. In Section A.4, we conclude our work.

A.2 Our Approach

In this section, we present our approach for improving the performance of DPM for weakly supervised object detection. In particular, we explore objectness measurement [Alexe et al., 2012], which has been widely applied for various purposes in computer vision, to generates category-independent object proposals with corresponding scores to their likelihood of being object bounding boxes, and adaptively extract a faithful window from the derived object proposals for each image as the initial root filter size and position for DPM detector. We then briefly describe the training and detecting procedures with DPM. Finally we propose our new post-processing method to further modify the predicted object bounding box obtained by DPM detector, so as to cover the object more precisely.

A.2.1 Initialization of object bounding box estimation

Figure A.2: Examples of bounding box enlarging and shrinking. Boxes before and after post-processing are shown in red and yellow, respectively.

Given an input image I (shown in Fig.A.1(a)), we first compute a set of N windows $\mathcal{W} = \{w_1, \cdots, w_k, \cdots, w_N\}$ with corresponding Bayesian posterior probabilities, denoted as $\mathcal{S} = \{s_1, \cdots, s_k, \cdots, s_N\}$ (shown in Fig.A.1 (b)) using the objectness approach [Ali and Madabhushi, 2012]. We set $N = 1000$, which ensures covering most objects even in very difficult images [Ali and Madabhushi, 2012]. Based on the fact that the objectness is designed to capture all possible objects within an image, we assume it has the reliability for providing at least *one* good candidate window w^* which covers the object of interest. However, the window with the highest objectness score $\max(\mathcal{S})$ is not always an effective choice [Shi et al., 2012], which usually encompasses other noisy objects, or locates poorly on object target.

To extract a reliable window from the pool of 1000 windows, we design a recursive selective scheme shown in Fig.A.1 (c)-(g). Inspired by the success of visual saliency applied in object recognition, we compute the reference region $\mathcal{T}$ (shown in Fig.A.1 (d)) by thresholding the saliency map $\mathcal{M}$(shown in Fig.A.1 (c)). The value of saliency map $\mathcal{M}$ at pixel $I(i,j)$ is obtained by summing up the objectness scores of the windows that cover this pixel:

$$\mathcal{M}(i,j) = \sum_{k=1}^{1000} \mathcal{M}_k(i,j) \tag{A.1}$$

where,

$$\mathcal{M}_k(i,j) = \begin{cases} s_k, & \text{if } I(i,j) \in w_k, \forall w_k \in \mathcal{W}, \\ 0, & \text{otherwise.} \end{cases} \tag{A.2}$$

Meanwhile, we also adaptively select windows with high score as candidates, according to the histogram of 1000 sorted windows (shown in Fig.A.1(e)). To avoid near duplicate candidate windows, we further perform non-maximum suppression (NMS) to get a finer set of candidates. Contrary to the common practice, which starts the suppression procedure from highest scoring windows, we randomly choose one, for the reason that the highest scoring window is not necessarily the best. Fig.A.1 (f) illustrates the derived smaller set of n confident candidates $\hat{\mathcal{W}} = \{\hat{w}_1, \cdots, \hat{w}_i, \cdots, \hat{w}_n\}$, and their corresponding score denoted as $\hat{\mathcal{S}} = \{\hat{s}_1, \cdots, \hat{s}_i, \cdots, \hat{s}_n\}$.

Given the reference region $\mathcal{T}$ which implies the most salient region within an image, and confident candidate windows, the overlap between them provides valuable information to find the location of target object. The final estimate of the initial object bounding box w^* (Fig.A.1(g)) is determined by optimizing the following function:

$$w^* = \underset{\hat{w}_i \in \hat{\mathcal{W}}, \hat{s}_i \in \hat{\mathcal{S}}}{\arg\max} \quad \gamma \hat{s}_i + (1-\gamma)\frac{area(\mathcal{T} \cap \hat{w}_i)}{area(\mathcal{T} \cup \hat{w}_i)}, \quad i \in [1, n] \tag{A.3}$$

where γ is a parameter used to control the influence of the objectness score s_i. In practice, we set $\gamma = 0.2$.

A.2.2 Detection with deformable part-based models

We start training the DPM detectors with the derived bounding boxes from Section A.2.1, which are treated as our positive training windows. Similarly to [Felzenszwalb et al., 2010], each root filter hypothesis in a positive training image is initialized with the corresponding derived bounding box (ground-truth bounding box is used in [Felzenszwalb et al., 2010]), and it is allowed to move around in a small neighborhood to maximize the filter score to compensate for imprecise bounding box estimation from Section A.2.1. We refer the reader to [Felzenszwalb et al., 2010] for more details concerning the DPM training and detection procedures. As in [Pandey and Lazebnik, 2011], we represent an image by a multiscale HOG feature pyramid [Dalal and Triggs, 2005] of 16 levels. For our DPM model, we use only a single component, since the multiple components are used for detecting objects with different views. We set the number of parts in DPM as 8 in all our experiments. And for negative training examples, we use random negatives from other object classes.

A.2.3 Bounding box post-processing

In many cases, the bounding boxes generated by DPM detectors are too large (resp. small) when detecting very small (resp. large) objects due to the restrictions of the size of the root filter and the scale of the feature pyramid. To improve the localization and to obtain a more precise estimate of the bounding box aspect ratio, we post-process each bounding box by enlarging or shrinking it to cover the object as much as possible. This is done using an improved version of the method proposed in [Y. Ke and Jing, 2006] which measures the amount of area that the edge energy occupies. In brief, we first augment the original bounding box to 120% of the original width and height (*i.e.* 144% in total area), and calculate the absolute values of the

gradients over the augmented bounding box and set the values which are less than 10% of the maximum to 0. To easily calculate the edge spatial distribution, then we resize the gradient magnitude image size to 100×100 and normalize the image sum to 1. Finally, we expand the bounding box in 4 directions from the centroid and stop until it contains 98% of the total gradient magnitude (edge energy) in the augmented box. This post-processing technique is not only able to crop out plain background regions, but also can expand to cover the foreground regions which are not encompassed by the original box. However, the cropping method in [Pandey and Lazebnik, 2011] is probably to fail with the latter. Fig. A.2 shows a few examples of our bounding box post-processing results. It is also worth noticing that this post-processing technique works efficiently for the objects with a unique or plain background, but has limited help for those with cluttered or textured background.

A.3 Experimental Evaluation

Table A.1: Average detection results (in %) compared with state-of-the-art competitors on the two variations of the PASCAL VOC 2007 datasets.

no post-processing	*VOC07-6×2*			
	Initialization	Refinement 1	Refinement 2	Refinement
[Pandey and Lazebnik, 2011]	37.22	51.63	56.99	59.32
ours	**38.72**	55.85	**59.82**	-
	VOC07-14×2			
[Pandey and Lazebnik, 2011]	19.88	25.11	27.69	**28.98**
ours	21.73	27.46	28.95	-
with post-processing	*VOC07-6×2*			
[Pandey and Lazebnik, 2011]	44.62	53.11	59.31	61.02
ours-[Pandey and Lazebnik, 2011]	47.85	56.78	63.31	-
ours-ES	**48.59**	58.02	**63.91**	-
	VOC07-14×2			
[Pandey and Lazebnik, 2011]	23.00	26.38	29.39	30.31
ours-[Pandey and Lazebnik, 2011]	24.20	28.21	**32.87**	-
ours-ES	**25.12**	28.94	32.82	-

Dataset: Following the protocol of previous works [Deselaers et al., 2010, Pandey and Lazebnik, 2011, Siva et al., 2012], we evaluate the performance of our proposed weak supervision framework on two subsets from the training and validation set (*trainval*) of the PASCAL VOC 2007 dataset (*VOC07*)[Everingham et al., 2010]: *VOC07-6×2* and *VOC07-14*. The *VOC07-6×2* subset contains 6 classes with *Left* and *Right* views (aspects) of each class, resulting in a total of 12 separating classes. The *VOC07-14* subset (same with *PASCAL07-all* defined in [Pandey and Lazebnik, 2011]) consists of 42 class/view combinations covering 14 classes and 5 views. Similar to [Pandey and Lazebnik, 2011], we remove all the images annotated as *difficult* or *truncated* in both training and evaluation steps.

Appendix A. Fusing Generic Objectness and Deformable Part-based Models for Weakly Supervised Object Detection

Evaluation criteria: To make fair comparisons, we only choose the detection window with highest score per image, although our method can detect multiple instances appeared in the image using sliding window approach. We also report both results for initial and refined localization as [Pandey and Lazebnik, 2011, Siva et al., 2012]. A refined localization is obtained by an iteratively trained DPM detector for one/several iteration(s) to refine the initial detection using the previous annotations as ground truth. Performance is evaluated with the percentage of training images in which an object is correctly covered by the window, if the strict PASCAL-overlap criterion is satisfied (intersection-over-union > 0.5).

Experimental evaluation: As Table A.1 shows, our method outperforms [Deselaers et al., 2010] and our baseline approach [Pandey and Lazebnik, 2011] on both datasets. Our average performance of initial detection before cropping boxes on the *VOC07-6×2* and *VOC07-14* subsets is 38.74% and 21.73% respectively, versus 37.22% and 19.98% for [Pandey and Lazebnik, 2011]. These improvements are due to the initial object estimate of our method described in Section A.2.1, which gives a better initialisation of the root filter of DPM detectors. We can also observe that both the cropping post-processing method from [Pandey and Lazebnik, 2011] (*i.e.* ours-[Pandey and Lazebnik, 2011] in Table A.1) and our enlarging-or-shrinking (*i.e.* ours-ES) post-processing method steadily improve the average localization accuracy. In particular, our ES cropping method is superior to that of [Pandey and Lazebnik, 2011], as our cropped bounding box is not only able to shrink to crop out the background regions, but also capable of enlarging to cover the whole foreground object resulted by incomplete coverage of the original window. An example is shown in the last row of Fig. A.3, where the target object (motorbike) is only partially localized by the initial detector (shown in red rectangles in the middle and right images) for both [Pandey and Lazebnik, 2011] and our method. However, in the final detection (shown in yellow), our method is able to enlarge the bounding box to nearly include the whole object, while [Pandey and Lazebnik, 2011] tends to crop out both foreground and background regions. The middle rows in Table A.1 indicate that localization accuracy can benefit from the refinement process. It is worth mentioning that with a better initialisation, our models converge to a steady level of performance after one less round of costly re-training (*i.e.* 2 iterations) than [Pandey and Lazebnik, 2011], and achieve slightly better results in the mean time. The detailed comparisons of our method with state-of-the-art methods on the *VOC07-6×2* dataset are listed in Table A.2. The results show that our method outperforms [Pandey and Lazebnik, 2011] for most of the categories. Especially, our method achieves the state-of-the-art results in some classes where the target object possesses the most salient regions in that category (*e.g.* *aeroplane, bus, horse*). Interestingly, even without refinement process, the accuracy for our method with certain category (*e.g.* *aeroplane left*) is superior to the competitors with the time-consuming refinement procedure. Fig. A.3 visually compares some of our results with those of [Pandey and Lazebnik, 2011].

Appendix A. Fusing Generic Objectness and Deformable Part-based Models for Weakly Supervised Object Detection

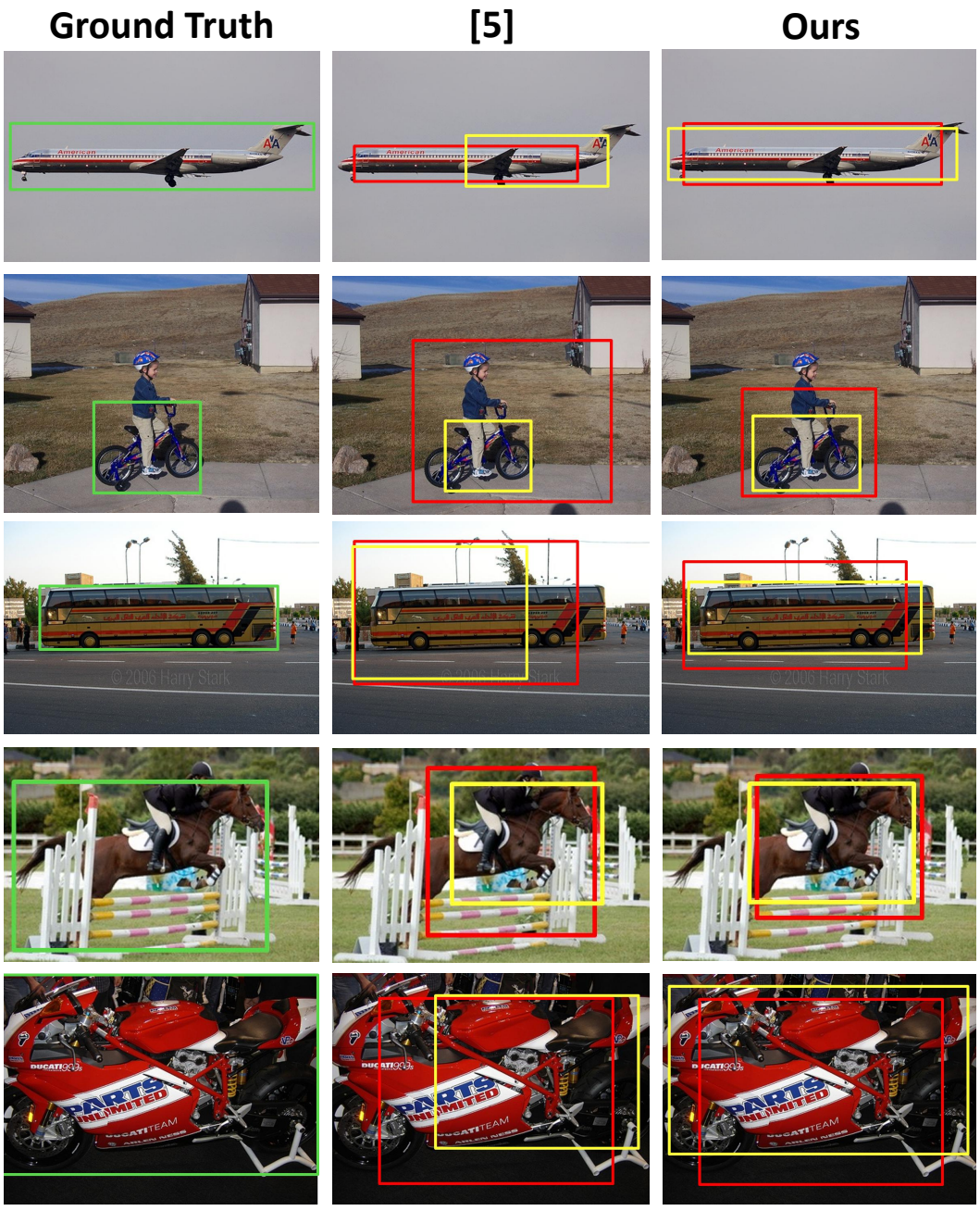


Figure A.3: Examples of detection results. The left column: ground-truth bounding boxes in green rectangles. The middle and right columns are detection results with [Pandey and Lazebnik, 2011] and our method, respectively. Initial detections are shown in red and detections refined by detectors are shown in yellow. Both results are with individual post-processing approach.

Appendix A. Fusing Generic Objectness and Deformable Part-based Models for Weakly Supervised Object Detection

Table A.2: Class-level localization accuracy (in %) for the *VOC07-6×2* dataset for our method vs. [Deselaers et al., 2010, Pandey and Lazebnik, 2011, Siva et al., 2012].

	Initialization		
	ours	[Pandey and Lazebnik, 2011]	[Siva et al., 2012]
aero left	**65.1**	55.8	39.1
aero right	**64.1**	61.5	50.0
bike left	31.3	**31.3**	28.4
bike right	42.0	**44.0**	30.6
boat left	9.1	4.6	**15.1**
boat right	9.3	9.3	**20.7**
bus left	23.8	23.8	**31.0**
bus right	**65.2**	52.2	35.1
horse left	**64.6**	60.4	48.5
horse right	**73.9**	67.4	45.2
mbike left	**64.1**	48.7	46.3
mbike right	70.6	**76.5**	55.3
average	**48.6**	44.6	37.1
	Refined by detector		
	ours	[Pandey and Lazebnik, 2011]	[Deselaers et al., 2010]
aero left	**69.7**	65.1	58.0
aero right	**84.6**	82.1	59.0
bike left	85.4	**87.5**	46.0
bike right	54.0	**68.0**	40.0
boat left	**13.6**	2.3	9.0
boat right	14.0	7.0	**16.0**
bus left	**42.9**	28.6	38.0
bus right	69.6	47.8	**74.0**
horse left	**87.5**	83.3	58.0
horse right	76.1	**80.4**	52.0
mbike left	87.2	**92.3**	67.0
mbike right	82.4	**88.2**	76.0
average	**63.9**	61.1	50.0

A.4 Conclusion

In this paper, we proposed a model enhancing the weakly supervised learning by emphasizing the importance of location and size of the initial class-specific root filter of deformable part model (DPM). We follow the general setup of [Pandey and Lazebnik, 2011] and introduce several substantial improvements to the weakly supervised DPM. The main contributions included new approaches based on objectness approach in generating the initial candidate window estimates. Furthermore we designed a flexible enlarging-and-shrinking post-processing procedure to modify the output bounding boxes of DPM, which can effectively further improve the final accuracy. Experimental results on the challenging PASCAL VOC 2007 database demonstrate that our proposed framework is efficient and competitive with state-of-the-art methods.

Appendix B

The Criteria Set on Prague Texture Benchmark

The segmentation benchmark criteria is divided into four subsets:

- Region-Based Criteria [Hoover et al., 1996].
- Pixel-Wise Weighted Average Criteria.
- Consistency Error Criteria [Martin et al., 2001]
- Clustering Comparison Criteria [Meila, 2005]

Note that we have introduced the GCE in Chapter 2.

Symbols ↑ and ↓ denote required increase / decrease of the corresponding criterion.

B.1 Region-Based Criteria

The region-based criteria mutually compare the machine segmented regions R_i $i = 1, ..., M$ with the correct ground truth regions $\hat{R}_j$ $j = 1, ..., N$ where $|R|$ is the corresponding set cardinality. The regions overlap acceptance is controlled by the threshold $0.5 < T \leq 1]$) $k = 0.75$. Single region-based criteria are defined as follows:

↑ **CS** (correct detection): $[R_m, \hat{R}_n]$ iff

(i) $|R_m \cap \hat{R}_n| \geq k|R_m|$

(ii) $|R_m \cap \hat{R}_n \geq k|\hat{R}_n|$

↓ **OS** (over-segmentation): $[R_{m1}, ..., R_{mx}; \hat{R}_n]$, $2 \leq x \leq M$ iff

(i) $\forall i \in < [1\ x], |R_{mi} \cap \hat{R}_n| \geq k|R_{mi}|$

(ii) $\sum_{i=1}^{x} |R_{mi} \cap \hat{R}_n| \geq k|\hat{R}_n|$

↓ **US** (under-segmentation): $[R_m; \hat{R}_{n1}, ..., \hat{R}_{nx}]$, $2 \leq x \leq N$ iff

(i) $\sum_{i=1}^{x} |R_m \cap \hat{R}_{ni}| \geq k|R_m|$

(ii) $\forall i \in < [1\ x], |R_m \cap \hat{R}_{ni}| \geq k|R_{ni}|$

↓ **ME** (missed): $[R_n]$ iff

(i) $\hat{R}_n \notin$ correct detection

(ii) $\hat{R}_n \notin$ over-segmentation

(iii) $\hat{R}_n \notin$ under-segmentation

$\downarrow$ **NE** (noise): $[R_m]$ iff

(i) $R_m \notin$ correct detection

(ii) $R_m \notin$ over-segmentation

(iii) $R_m \notin$ under-segmentation

B.2 Pixel-Wise Weighted Average Criteria

Let us denote

$$n_{i,\bullet} = \sum_{j=1}^{N} n_{i,j}, \tag{B.1}$$

$$n_{\bullet,i} = \sum_{j=1}^{M} n_{j,i}, \tag{B.2}$$

where N, M are the correct number of classes and the interpreted number of classes (or regions), respectively. $K = \arg\max\{M; N\}$, n is the number of pixels in the test set, $n_{i,j}$ is the number of pixels interpreted as the i-th class but belonging into the j-th class. The error matrix ($\{n_{i,j}\}$) extended into $K \times K$ is obtained by padding missing entries with zeros. $\hat{i}$ is either i for supervised tests or mapping of the i-th class ground truth into an interpretation segment based on the Munkres algorithm (for unsupervised test). The following pixel-wise criteria were implemented:

$\downarrow$ **O** (omission error - the overall ratio of wrongly interpreted pixels):

$$O = median\{\frac{O_i}{n_{\bullet,i}}\}_{i=1}^{N} = median\{\frac{n_{\bullet,i} - n_{\hat{i},i}}{n_{\bullet,i}}\}_{i=1}^{N} \tag{B.3}$$

where O_i is the $i-th$ class omission error.

$\downarrow$ **C** (commission error - the overall ratio of wrongly assigned pixels):

$$C = median\{\frac{C_i}{n_{\hat{i},\bullet}}\}_{\hat{i}=1}^{M} = median\{\frac{n_{\hat{i},\bullet} - n_{\hat{i},i}}{n_{\hat{i},\bullet}}\}_{\hat{i}=1}^{M} \tag{B.4}$$

where C_i is the i-th class commission error.

$\uparrow$ **CA** (the weighted average class accuracy):

$$CA = \frac{1}{n}\sum_{i=1}^{K} \frac{n_{\hat{i},i} n_{\bullet,i}}{n_{\bullet,i} + n_{\hat{i},\bullet} - n_{\hat{i},i}} \tag{B.5}$$

↑ **CO** (recall, the weighted average correct assignment):

$$CO = \frac{1}{n}\sum_{i=1}^{K} n_{\bullet,i} CO_i = \frac{1}{n}\sum_{i=1}^{K} n_{\hat{i},i} \tag{B.6}$$

↑ **CC** (precision, object accuracy, overall accuracy):

$$CC = \frac{1}{n}\sum_{i=1}^{K} n_{\bullet,i} CC_i = \frac{1}{n}\sum_{i=1}^{K} \frac{n_{\hat{i},i} n_{\bullet,i}}{n_{\hat{i},\bullet}} \tag{B.7}$$

↓ **I** (type I error, the weighted probability of wrong assignment of classes pixels):

$$I = \frac{1}{n}\sum_{i=1}^{K} (n_{\bullet,i} - n_{\hat{i},i}) = 1 - CO \tag{B.8}$$

↓ **II** (type II error, the weighted probability of commission error):

$$II = \frac{1}{n}\sum_{i=1}^{K} \frac{n_{\hat{i},\bullet} n_{\bullet,i} - n_{\hat{i},i} n_{\bullet,i}}{n - n_{\bullet,i}} \tag{B.9}$$

↑ **EA** (mean class accuracy estimate):

$$EA = \frac{1}{n}\sum_{i=1}^{K} \frac{2 n_{\hat{i},i} n_{\bullet,i}}{n_{\bullet,i} + n_{\hat{i},\bullet}} \tag{B.10}$$

↑ **MS** (mapping score - emphasizes the error of not recognizing the test data):

$$MS = \frac{1}{n}\sum_{i=1}^{K} (1.5 n_{\hat{i},i} - 0.5 n_{\hat{i},\bullet}) \tag{B.11}$$

↓ **RM** (root mean square proportion estimation error):

$$RM = \sqrt{\frac{1}{K}\sum_{i=1}^{K} (\frac{n_{\hat{i},\bullet} - n_{\bullet,i}}{n})^2} \geq 0 \tag{B.12}$$

indicates unbalance between the omission O_i and commission C_i errors, respectively.

↑ **CI** (comparison index - includes both these types of errors):

$$CI = \frac{1}{n}\sum_{i=1}^{K} n_{\hat{i},i} \sqrt{\frac{n_{\bullet,i}}{n_{\hat{i},\bullet}}} = \frac{1}{n}\sum_{i=1}^{K} n_{\bullet,i} \sqrt{CC_i CO_i} \tag{B.13}$$

where CC_i, CO_i are the object precision and recall. CI reaches its maximum either for the ideal segmentation or for equal commission and omission errors for every region (class).

The F measure curve (see Region-Based Criteria)

$$F = \frac{1}{n}\sum_{i=1}^{K} n_{\bullet,i}\frac{CC_i CO_i}{\gamma CO_i + (1-\gamma CC_i)} \tag{B.14}$$

B.3 Clustering Comparison Criteria

A clustering $\mathcal{S}$ is a partition of a data set D into sets $R_1, R_2, ..., R_M$ called clusters such that $R_k \cap R_l = \emptyset$ and $\cup_{k=1}^{M} R_k = S$

Let the number of data points in D and in cluster R_k be n and n_k respectively. We have, of course, $n = \sum_{k=1}^{M} n_k$.

The number of points in the intersection of clusters R_k of $\mathcal{S}$ and $R'_{k'}$ of $\mathcal{S}'$ is denoted $n_{kk'}$

$$n_{kk'} = |R_k \cap R'_{k'}| \tag{B.15}$$

Note that the metric d_{VI} has been presented denoted as VoI in Chapter 2.

$\downarrow d_M$(Mirkin metric):

$$d_M(\mathcal{S},\mathcal{S}') = \frac{d'_M(\mathcal{S},\mathcal{S}')}{n^2} \tag{B.16}$$

$$d'_M(\mathcal{S},\mathcal{S}') = \sum_k n_k^2 + \sum_{k'} n'^2_{k'} - 2\sum_k\sum_{k'} n_{kk'}^2 \tag{B.17}$$

$\downarrow d_D$(Van Dongen metric):

$$d_D(\mathcal{S},\mathcal{S}') = \frac{d'_D(\mathcal{S},\mathcal{S}')}{2n} \tag{B.18}$$

$$d'_D(\mathcal{S},\mathcal{S}') = 2n - \sum_k \max_{k'} n_{kk'} - \sum_{k'} \max_k n_{kk'} \tag{B.19}$$

Appendix C

Publications

During my PhD studies, I have published one journal paper and five papers in international conferences. A second journal paper is under review.

Published International Conference Papers:

1. **Xiaofang Wang**, Huibin Li, Simon Masnou, Liming Chen: Sparse Coding and Mid-Level Superpixel-Feature for ℓ_0-Graph Based Unsupervised Image Segmentation. Computer Analysis of Images and Patterns, pp. 160–168, York, UK, 2013.

2. **Xiaofang Wang**, Huibin Li, Charles-Edmond Bichot, Simon Masnou, Liming Chen: A graph-cut approach to image segmentation using an affinity graph based on ℓ_0- sparse representation of features. Image Processing (ICIP), 20th IEEE International Conference on, pp. 4019-4023, Melbourne, Australia, 2013.

3. **Xiaofang Wang**, Huibin Li, Bichot, C.-E., Simon Masnou, Liming Chen: Graph-based image segmentation using weighted color patch. Image Processing (ICIP), 20th IEEE International Conference on, pp. 4064-4068, Melbourne, Australia, 2013.

4. Dongming Chen, Mohsen Ardabilian, **Xiaofang Wang**, Liming Chen: An improved non-local cost aggregation method for stereo matching based on color and boundary cue. Multimedia and Expo (ICME), 2013 IEEE International Conference on, pp. 1-6, California, US, 2013.

5. Yuxing Tang, **Xiaofang Wang**, Emmanuel Dellandrea, Simon Masnou, Liming Chen: Fusing Generic Objectness and Deformable Part-based Models for Weakly Supervised Object Detection. Image Processing (ICIP), 21th IEEE International Conference on, Pairs, France, 2014.(**Equal contribution**)

Accepted Journal Paper:

1. **Xiaofang Wang**, Yuxing Tang, Simon Masnou, Liming Chen: A Global/Local Affinity Graph for Image Segmentation. IEEE Transactions on Image Processing (TIP), to appear, 2015.

Submitted Journal Paper:

1. **Xiaofang Wang**, Boyang Gao, Simon Masnou, Liming Chen: Active Colloids Segmentation and Tracking. Pattern Recognition, submitted, 2015.

Bibliography

97

4, 97

13

112

AS Abutaleb and A Eloteifi. Automatic thresholding of gray-level pictures using 2-d entropy. In *31st Annual Technical Symposium*, pages 29–35. International Society for Optics and Photonics, 1988. 41

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Su¨sstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274 –2282, nov. 2012. 28

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-617549-X. 110, 112

Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Classcut for unsupervised class segmentation. In *Computer Vision–ECCV 2010*, pages 380–393. Springer, 2010. 5, 27, 49, 134

Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2189–2202, 2012. 26, 138, 139

Sahirzeeshan Ali and Anant Madabhushi. An integrated region-, boundary-, shape-based active contour for multiple object overlap resolution in histological imagery. *Medical Imaging, IEEE Transactions on*, 31(7):1448–1460, 2012. xiv, 47, 139

D. Anoraganingrum. Cell segmentation with median filter and mathematical morphology operation. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 1043–1046, 1999. doi: 10.1109/ICIAP.1999.797734. 105

P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011. xiv, 4, 24, 41, 42, 51, 52, 67, 75, 76, 82, 85, 134

Pablo Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 182–182. IEEE, 2006. 42

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. From contours to regions: An empirical evaluation. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 2294–2301. IEEE, 2009. 55

Pablo Arbeláez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir Bourdev, and Jitendra Malik. Semantic segmentation using regions and parts. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3378–3385. IEEE, 2012. 5, 51, 134, 135

Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014. 135

Tim J Atherton and Darren J Kerbyson. Size invariant circle detection. *Image and Vision computing*, 17(11):795–803, 1999. 107

H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *Computer Vision-ECCV 2012*, 2012. 82, 137

Shai Bagon, Oren Boiman, and Michal Irani. What is a good image segment? a unified approach to segment extraction. In *Computer Vision–ECCV 2008*, pages 30–44. Springer, 2008. 48

Xiang Bai, Xingwei Yang, Longin Jan Latecki, Wenyu Liu, and Zhuowen Tu. Learning context-sensitive shape similarity by graph transduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):861–874, 2010. 75, 76, 83

Xiang Bai, Bo Wang, Cong Yao, Wenyu Liu, and Zhuowen Tu. Co-transduction for shape retrieval. *Image Processing, IEEE Transactions on*, 21(5):2747–2757, 2012. 75, 76

Y Bar-Shalom and AG Jaffer. Adaptive nonlinear filtering for tracking with measurements of uncertain origin. In *Decision and Control, 1972 and 11th Symposium on Adaptive Processes. Proceedings of the 1972 IEEE Conference on*, volume 11, pages 243–247. IEEE, 1972. 8

Adrian Barbu and Songchun Zhu. Graph partition by swendsen-wang cuts. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 320–327. IEEE, 2003. 24

Antoine Basset, Jérôme Boulanger, Patrick Bouthemy, Charles Kervrann, and Jean Salamero. Slt-log: A vesicle segmentation method with automatic scale selection and local thresholding applied to tirf microscopy. In *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, pages 533–536. IEEE, 2014. 105

Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3169–3176. IEEE, 2010. 49

S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using em and its application to content-based image retrieval. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 675–682, 1998. 57

B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3457–3464, 2011. 101, 116

Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1806–1819, 2011. 105, 116, 136

Margrit Betke, Esin Haritaoglu, and Larry S Davis. Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications*, 12(2):69–83, 2000. 101

James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981. 43

Samuel S Blackman. Multiple-target tracking with radar applications. *Dedham, MA, Artech House, Inc., 1986, 463 p.*, 1, 1986. xvii, 105, 118, 119, 122, 126

S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19(1):5–18, 2004. 105

Samuel Blackrnan and Artech House. Design and analysis of modern tracking systems. *Boston, MA: Artech House*, 1999. 8

Stéphane Bonneau, Maxime Dahan, and Laurent D Cohen. Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *Image Processing, IEEE Transactions on*, 14(9):1384–1395, 2005. 105

Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *Computer Visio ECCV 2002*, pages 109–122. Springer, 2002. 49

Aïssa Boulmerka, Mohand Saïd Allili, and Samy Ait-Aoudia. A generalized multiclass histogram thresholding approach based on mixture modelling. *Pattern Recognition*, 47(3):1330–1348, 2014. 41

Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 111–118, 2010. 30

Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006. 6, 48

Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004. 48

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001. 6

Philip Breen. Algorithms for sparse approximation. *Project, School of Mathematics University of Edinburgh*, 2009. 63

William Brendel, Mohamed Amer, and Sinisa Todorovic. Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280. IEEE, 2011. 9

Gerd Brunner12, Deepak R Chittajallu, Uday Kurkure, and Ioannis A Kakadiaris. Patch-cuts: A graph-based image segmentation method using patch features and spatial relations. In *British Machine Vision Conference (BMVC)*, pages 1–11, 2010. 26, 82

A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, pages 60–65, 2005. 10, 82

Laurent Busin, Jiambo Shi, Nicolas Vandenbroucke, and Ludovic Macaire. Color space selection for color image segmentation by spectral clustering. In *Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on*, pages 262–267. IEEE, 2009. 30

A.A. Butt and R.T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1846–1853, 2013a. 105

Asad A Butt and Robert T Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1846–1853. IEEE, 2013b. 9, 101, 136

Ivo Buttinoni, Julian Bialké, Felix Kümmel, Hartmut Löwen, Clemens Bechinger, and Thomas Speck. Dynamical clustering and phase separation in suspensions of self-propelled colloidal particles. *Phys. Rev. Lett.*, 110:238301, Jun 2013. 102, 103, 104

John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986. 24, 41

Joao Carreira and Cristian Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010. 4, 50, 51, 134

Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1312–1328, 2012. 135

Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *Computer Vision–ECCV 2012*, pages 430–443. Springer, 2012a. 5, 27, 51, 93, 134, 135

João Carreira, Fuxin Li, and Cristian Sminchisescu. Object recognition by sequential figure-ground ranking. *International journal of computer vision*, 98(3):243–262, 2012b. 51, 52

Vicent Caselles. Geometric models for active contours. In *Image Processing, 1995. Proceedings., International Conference on*, volume 3, pages 9–12. IEEE, 1995. 46

Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997. xiv, 46, 47

Hasan Ertan Cetingul and René Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1896–1902. IEEE, 2009. 45

Tony Chan and Luminita Vese. An active contour model without edges. In *Scale-Space Theories in Computer Vision*, pages 141–151. Springer, 1999. 107

Tony Chan and Wei Zhu. Level set based shape prior segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, volume 2, pages 1164–1170. IEEE, 2005. 46, 136

Tony F Chan and Luminita A Vese. Active contours without edges. *Image processing, IEEE transactions on*, 10(2):266–277, 2001. 46

Tony F Chan, B Yezrielev Sandberg, and Luminita A Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11(2):130–141, 2000. 46

Tony F. Chan, Selim Esedoglu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal of Applied Mathematics*, 66(5):1632–1648, 2006. 107, 136

J. Cheeger. A lower bound for the smallest eigenvalue of the laplacian. *Problems in analysis*. 35

Yisong Chen, Antoni B Chan, and Guoping Wang. Adaptive figure-ground classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 654–661. IEEE, 2012. 48

B Cheng, J Yang, S Yan, Y Fu, and TS Huang. Learning with l1-graph for image analysis. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 19(4):858, 2010. 17, 18, 19

Bin Cheng, Guangcan Liu, Jingdong Wang, Zhongyang Huang, and Shuicheng Yan. Multi-task low-rank affinity pursuit for image segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2439–2446. IEEE, 2011a. 4, 29, 61, 134

H.D. Cheng, X.H. Jiang, Y. Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259 – 2281, 2001. 23, 40

Ming-Ming Cheng, Guo-Xin Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 409–416, 2011b. 5

Nicolas Chenouard, Isabelle Bloch, and Jean-Christophe Olivo-Marin. Multiple hypothesis tracking for cluttered biological image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013. 101, 103

Nicolas Chenouard, Ihor Smal, Fabrice De Chaumont, Martin Maška, Ivo F Sbalzarini, Yuanhao Gong, Janick Cardinale, Craig Carthel, Stefano Coraluppi, Mark Winter, et al. Objective comparison of particle tracking methods. *Nature methods*, 2014. 103, 104, 105, 117

C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *ICPR*, pages 150–155, 2002. 86

F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. 35, 66

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002. xiii, xiv, 5, 27, 28, 44, 45, 48, 61, 97, 98, 135

Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995. 46

Stefano Coraluppi and Craig Carthel. Multi-stage multiple-hypothesis tracking. *J. Adv. Inf. Fusion*, 6(1):57–67, 2011. 105

Camille Couprie, Leo Grady, Laurent Najman, and Hugues Talbot. Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 731–738. IEEE, 2009. 28

T. Cour, F. Bénézit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, pages 1124–1131, 2005. 17, 24, 29, 33, 65, 75, 76, 135

Ingemar J Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993. 8

Ingemar J Cox, Satish B Rao, and Yu Zhong. "ratio regions": A technique for image segmentation. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 2, pages 557–564. IEEE, 1996. 32, 37

D. Crandall and D. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *Computer Vision-ECCV 2006*, 2006. 137

John C Crocker and David G Grier. Methods of digital video microscopy for colloidal studies. *Journal of colloid and interface science*, 179(1):298–310, 1996. 103

Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2, 2004. 93

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, 2005. 137, 140

Stuart B Dalziel. Decay of rotating turbulence: some particle tracking experiments. *Applied scientific research*, 49(3):217–244, 1992. 120

Arthur P Dempster, Nan M Laird, Donald B Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39 (1):1–38, 1977. 44

T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *Computer Vision-ECCV 2012*, 2010. xii, 137, 141, 142, 144

T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *International Journal of Computer Vision*, 100(3):275–293, 2012. 137

Edsger. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959a. 119

Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959b. 6

William E Donath and Alan J Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973. 35

M. Donoser, M. Urschler, M. H., and H. Bischof. Saliency driven total variation segmentation. In *Computer Vision (ICCV), 2009 IEEE International Conferenc on*, pages 817–824, 2009. 75, 76

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000. 8

Michael Elad. *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*. Springer, 2010. ISBN 978-1-4419-7010-7. doi: 10.1007/978-1-4419-7011-4. URL http://dx.doi.org/10.1007/978-1-4419-7011-4. 62

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2765–2781, November 2013. 68

Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 2790–2797. IEEE, 2009. 18, 19, 44

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 4, 51

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2), 2010. 137, 141

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. xiv, 5, 28, 45, 46, 57, 61, 82, 87, 97, 135

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. 82, 137, 138, 140

Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3294–3301. IEEE, 2013. 135

Miroslav Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4): 619–633, 1975. 35

Thomas E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173–184, 1983. 105

Charless Fowlkes, David Martin, and Jitendra Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In *Computer Vision and Pattern Recognition (CVPR), 2003 IEEE Conference on*, volume 2, pages II–54. IEEE, 2003. 24, 25, 29, 82

William T. Freeman and Edward H Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991. 41

Jordi Freixenet, Xavier Muñoz, David Raba, Joan Martí, and Xavier Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *Computer Vision-ECCV 2002*, pages 408–422, 2002. 4, 55, 68, 87, 97

Hao Fu and Guoping Qiu. Integrating low-level and semantic features for object consistent segmentation. In *Int. Conf. on Image and Graphics (ICIG),*, pages 39–44, 2011. 92

King-Sun Fu and JK Mui. A survey on image segmentation. *Pattern recognition*, 13(1):3–16, 1981. 39

Jean Gallier. Notes on elementary spectral graph theory. applications to graph clustering using normalized cuts. *arXiv preprint arXiv:1311.2492*, 2013. 13

Shenghua Gao, Ivor Waihung Tsang, Liang-Tien Chia, and Peilin Zhao. Local features are not lonely–laplacian sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3555–3561. IEEE, 2010. 93

Marc Gelgon, Patrick Bouthemy, and Jean-Pierre Le Cadre. Recovery of the trajectories of multiple moving objects in an image sequence with a pmht approach. *Image and Vision Computing*, 23(1):19–31, 2005. 105

Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 456–463. IEEE, 2003. 44

John R Gilbert, Gary L Miller, and Shang-Hua Teng. Geometric mesh partitioning: Implementation and experiments. *SIAM Journal on Scientific Computing*, 19(6): 2091–2110, 1998. xiii, 36

R. Girshick, P. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *NIPS*, 2011. 82, 137

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 134

Alvina Goh and René Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1–6. IEEE, 2007. 18

Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum flow problem. *J. ACM*, 35:921–940, October 1988. 66

Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. 66

Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *Journal of the Society for Industrial & Applied Mathematics*, 9(4):551–570, 1961. 32

Josep M Gonfaus, Xavier Boix, Joost Van de Weijer, Andrew D Bagdanov, Joan Serrat, and Jordi Gonzalez. Harmony potentials for joint classification and segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3280–3287. IEEE, 2010. 52

Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993. 8

Leo John Grady. *Space-variant computer vision: a graph-theoretic approach*. PhD thesis, Boston, MA, USA, 2004. 68

L. W. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 11(9): 1074–1085, 1992. 81

M. Haindl and S. Mikes. Texture segmentation benchmark. In *ICPR*, pages 1–4, 2008. 82, 85

Bruce Hendrickson and Robert W Leland. A multi-level algorithm for partitioning graphs. *SC*, 95:28, 1995. 35

Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *Computer Vision (ICCV), 2011 IEEE International Conferenc on*, volume 1, pages 654–661. IEEE, 2005. 135

Adam Hoover, Gillian Jean-Baptiste, Xiaoyi Jiang, Patrick J Flynn, Horst Bunke, Dmitry B Goldgof, Kevin Bowyer, David W Eggert, Andrew Fitzgibbon, and Robert B Fisher. An experimental comparison of range image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18 (7):673–689, 1996. 147

Yongzhen Huang, Kaiqi Huang, Yinan Yu, and Tieniu Tan. Salient coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1753–1760. IEEE, 2011. 94

Yongzhen Huang, Zifeng Wu, Liang Wang, and Tieniu Tan. Feature coding in image classification: A comprehensive study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(3):493–506, March 2014. 93

Michael Isard and Andrew Blake. Condensation¡ªconditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998. 8

Tommi Jaakkola, David Haussler, et al. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999. 93

Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. 43

Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988. 42

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999. xiv, 42, 43

Z. Ji, Y. Xia, Q. Chen, Q. Sun, D. Xia, and D. D. Feng. Fuzzy c-means clustering with weighted image patch for image segmentation. *Appl. Soft Comput.*, 12(6): 1659–1667, 2012. 10, 82, 83

Hao Jiang, S. Fels, and J.J. Little. A linear programming approach for multiple object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8, 2007. 106

Xiaoyan Jiang, Daniel Haase, Marco Körner, Wolfgang Bothe, and Joachim Denzler. Accurate 3d multi-marker tracking in x-ray cardiac sequences using a two-stage graph modeling approach. In *CAIP (2)*, pages 117–125, 2013. xvii, 105, 119, 120, 121, 122, 123, 126

M. Jitendra, B. Serge, L. K. Thomas, and S. Jianbo. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001. 41

Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 542–549. IEEE, 2012. 50

Simon J Julier and Jeffrey K Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3, pages 3–2. Orlando, FL, 1997. 8

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 7

George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998. 34, 35

Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 46

Bernardin Keni and Stiefelhagen Rainer. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008. 116

Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970. 34, 35

Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1805–1819, 2005. 8

Gunhee Kim and Eric P. Xing. On multiple foreground cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012. 50

T. H. Kim, K. M. Lee, and S. U. Lee. Learning full pairwise affinities for spectral segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2101–2108, 2010a. 27, 75, 76, 87

T. H. Kim, K. M. Lee, and S. U. Lee. Learning full pairwise affinities for spectral segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2101–2108, 2010b. xiii, 4, 27, 135

Benjamin King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101, 1967. 43

Genshiro Kitagawa. Non-gaussian state-space modeling of nonstationary time series. *Journal of the American statistical association*, 82(400):1032–1041, 1987. 105

Josef Kittler and John Illingworth. Minimum error thresholding. *Pattern recognition*, 19(1):41–47, 1986. 40

H. Knutsson and G. H. Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management - CAPAIDM*, Pasadena, October 1983. URL `http://www.imt.liu.se/Publications/kg83a.pdf`. 24

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. xvii, 105, 106, 119, 120, 121, 122, 127, 136

M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *CVPR*, pages 3217–3224, 2010. 57

Lubor Ladicky, Christopher Russell, Pushmeet Kohli, and Philip HS Torr. Associative hierarchical crfs for object class image segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 739–746. IEEE, 2009. 51, 52

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. 5

Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *CVPR*, pages 1–8, 2010. 57

Yong Jae Lee and Kristen Grauman. Object-graphs for context-aware visual category discovery. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):346–358, 2012. 83, 92

Bastian Leibe and Bernt Schiele. *Interleaving object categorization and segmentation*. Springer, 2006. 49

Thomas Leung and Jitendra Malik. Contour continuity in region based image segmentation. In *Computer Vision—ECCV'98*, pages 544–559. Springer, 1998. 24

Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12): 2290–2297, 2009. ISSN 0162-8828. 28

Chunming Li, Chiu-Yen Kao, John C Gore, and Zhaohua Ding. Minimization of region-scalable fitting energy for image segmentation. *Image Processing, IEEE Transactions on*, 17(10):1940–1949, 2008. xvii, 46, 106, 107

Fei Fei Li, Rufin VanRullen, Christof Koch, and Pietro Perona. Rapid natural scene categorization in the near absence of attention. *Proceedings of the National Academy of Sciences*, 99(14):9596–9601, 2002. 50

Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Computer Vision and Pattern Recognition, 2009.*, pages 2036–2043. IEEE, 2009. 5

Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. *Advances in Neural Information Processing Systems*, 2010. 92

Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 303–308. ACM, 2004. 48

Z. Li, X. Wu, and S. Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 789–796, 2012. xiii, 3, 4, 9, 17, 27, 28, 46, 53, 61, 64, 67, 68, 74, 75, 76, 92, 96, 97, 135

Liang Liang, Hongying Shen, Pietro De Camilli, and James S Duncan. Tracking clathrin coated pits with a multiple hypothesis based method. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, pages 315–322. Springer, 2010. 105

Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010. 21

Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 663–670, 2010. 18, 21

Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):171–184, January 2013a. 18, 68, 98, 99

Han Liu, Yanyun Qu, Yang Wu, and Hanzi Wang. Class-specified segmentation with multi-scale superpixels. In *Computer Vision-ACCV 2012 Workshops*, pages 158–169. Springer, 2013b. 135

Jiangyu Liu, Jian Sun, and Heung-Yeung Shum. Paint selection. In *ACM Transactions on Graphics (ToG)*, volume 28, page 69. ACM, 2009. 48

M.Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2097 –2104, 2011. 28

Wei Liu and Shih-Fu Chang. Robust multi-class transductive learning with graphs. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 381–388. IEEE, 2009. 17

D. G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision (ICCV), 1999 IEEE International Conferenc on*, pages 1150–1157, 1999. 22, 61, 69

L Luccheseyz and SK Mitray. Color image segmentation: A state-of-the-art survey. *Image Processing, Vision, and Pattern Recognition*, 67(2):207–221, 2001. 39

Tianyang Ma and Longin Jan Latecki. Graph transduction learning with connectivity constraints with application to multiple foreground cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1955–1962. IEEE, 2013. 50

Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(9):1546–1562, 2007. 44

Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1): 7–27, 2001. xiii, 24, 25, 26, 29, 82

Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *British Machine Vision Conference (BMVC)*, 2007. 97, 135

Ravi Malladi, James A Sethian, and Baba C Vemuri. Shape modeling with front propagation: A level set approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(2):158–175, 1995. 46

Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim's algorithm. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2536–2543. IEEE, 2013. 134

D. R. Martin, C. Fowlkes, D. Tal, , and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision (ICCV), 2011 IEEE International Conferenc on*, pages 416–425, 2001. 4, 54, 68, 87, 97, 147

Erik Meijering, Oleh Dzyubachyk, Ihor Smal, et al. Methods for cell and particle tracking. *Methods Enzymol*, 504(9):183–200, 2012. 105

M. Meila. Comparing clusterings: an axiomatic view. In *ICML*, pages 577–584, 2005. 4, 54, 68, 87, 97, 147

Anton Milan, Konrad Schindler, and Stefan Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013. 101

Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198. ACM, 1995. 48

M.Peterson and B.Gibson. Shape recognition contributions to figure-ground organization in three dimensional displays. *Cognitive Psychology*, pages 25 :383–429, 1993. 59, 61, 64, 72, 73

T. Muerle. Experimental evaluation of techniques for automatic segmentation of objects in a complex scene. *Pictorial Pattern Recognition*, pages 3–13, 1968. 39

David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989. 46

Joseph L Mundy. Object recognition in the geometric era: A retrospective. In *Toward category-level object recognition*, pages 3–28. Springer, 2006. 4

Peer Neubert and Peter Protzel. Superpixel benchmark and comparison. In *Proc. Forum Bildverarbeitung*, 2012. 28

Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering1 analysis and an algorithm. *Proceedings of Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press*, 14:849–856, 2001. 30

M. Nguyen, L. Torresani, F. de la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *Computer Vision (ICCV), 2009 IEEE International Conferenc on*, 2009. 137

P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking - linking identities using bayesian network inference. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2187–2194, 2006. 105, 106

Songhwai Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 735–742, 2004. 8, 105

Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002. 22

Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004. 8

Jean-Christophe Olivo-Marin. Extraction of spots in biological images using multiscale products. *Pattern Recognition*, 35(9):1989–1996, 2002. 105

Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988. 46

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975. xiv, xvii, 40, 41, 104, 108, 117, 124

Dirk Padfield, Jens Rittscher, and Badrinath Roysam. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical Image Analysis*, 15(4):650–668, 2011. xvii, 9, 103, 105, 117, 118, 124

J. Palacci, C. Cottin-Bizonne, C. Ybert, and L. Bocquet. Sedimentation and effective temperature of active colloidal suspensions. *Physical review letters*, 105:088304, 2010. 115

Jeremie Palacci, Stefano Sacanna, Asher Preska Steinberg, David J Pine, and Paul M Chaikin. Living crystals of light-activated colloidal surfers. *Science*, 339(6122): 936–940, 2013. 102

M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Computer Vision (ICCV), 2011 IEEE International Conferenc on*, 2011. xii, xviii, 137, 138, 140, 141, 142, 143, 144, 145

Sylvain Paris and Frédo Durand. A topological approach to hierarchical segmentation using mean shift. In *Computer Vision and Pattern Recognition, 2007 IEEE Conference on*, pages 1–8. IEEE, 2007. 44

Hanna Pasula, Stuart Russell, Michael Ostland, and Yaacov Ritov. Tracking many objects with many sensors. In *IJCAI*, volume 99, pages 1160–1171, 1999. 8

P.Bo, Z. Lei, and Z. David. A survey of graph theoretical approaches to image segmentation. *Pattern Recognit.*, 46(3):1020–1038, March 2013. 81

Bo Peng, Lei Zhang, and David Zhang. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3):1020–1038, 2013. 39

Xi Peng, Lei Zhang, and Zhang Yi. Constructing l2-graph for subspace learning and segmentation. *arXiv preprint arXiv:1209.0841*, 2012. 18, 20

Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8. IEEE, 2007. 93

Thomas Pock, Thomas Schoenemann, Gottfried Graber, Horst Bischof, and Daniel Cremers. A convex formulation of continuous multi-label problems. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, 2008. 107, 136

Aubrey B Poore and Sabino Gadaleta. Some assignment problems arising from multiple target tracking. *Mathematical and Computer Modelling*, 43(9):1074–1091, 2006. 105

Robert Clay Prim. Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401, 1957. 45

Zenon W Pylyshyn and Ron W Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism*. *Spatial vision*, 3(3):179–197, 1988. 115

Shankar Rao, Hossein Mobahi, Allen Y. Yang, Shankar Sastry, and Yi Ma. Natural image segmentation with adaptive texture and boundary encoding. In *Asian Conf. on Comput. Vis.*, pages 135–146, 2009. 53, 75, 76

Donald B Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979. 8

Xiaofeng Ren. Multi-scale improves boundary detection in natural images. In *Computer Vision–ECCV 2008*, pages 533–545. Springer, 2008. 41

Xiaofeng Ren and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. In *NIPS*, volume 2, page 7, 2012. 42

Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 10–17. IEEE, 2003. 22, 28

Lawrence Gilman Roberts. *MACHINE PERCEPTION OF THREE-DIMENSIONAL soups*. PhD thesis, Massachusetts Institute of Technology, 1963. 41

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004. xiv, 6, 48, 49

Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 993–1000. IEEE, 2006. xiv, 49

CP Royall, ME Leunissen, and A Van Blaaderen. A new colloidal model system to study long-range interactions quantitatively in real space. *Journal of Physics: Condensed Matter*, 15(48):S3581, 2003. 103

Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013. xiv, 50, 52

J.C. Rubio, J. Serrat, A.M. Lopez, and D. Ponsa. Multiple-target tracking for intelligent headlights control. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):594–605, 2012. 101

Bryan C Russell, William T Freeman, Alexei A Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1605–1614. IEEE, 2006. 135

Pekka Ruusuvuori, Antti Lehmussola, Jyrki Selinummi, Tiina Rajala, Heikki Huttunen, and Olli Yli-Harja. Benchmark set of synthetic images for validating cell image analysis algorithms. In *Proceedings of the 16th European Signal Processing Conference, EUSIPCO*, 2008. 104

Ivo F Sbalzarini and Petros Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of structural biology*, 151(2): 182–195, 2005. xvii, 105, 117, 118, 119, 120, 121, 125, 127

Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007. 30

Jean Serra. *Image analysis and mathematical morphology, v. 1.* Academic press, 1982. 104

Mehmet Sezgin et al. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–168, 2004. 41

E. Sharon, A. Brandt, and R. Basri. Segmentation and boundary detection using multiscale intensity measurements. In *Computer Vision and Pattern Recognition (CVPR), 2001 IEEE Conference on*, pages 469–476, 2001. 86

Yaser Ajmal Sheikh, Erum A Khan, and Takeo Kanade. Mode-seeking by medoidshifts. In *Computer Vision (ICCV), 2007 IEEE International Conferenc on*, pages 1–8. IEEE, 2007. 44

J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000. 6, 17, 24, 28, 30, 33, 34, 37, 44, 58, 61, 64, 66, 75, 76, 81, 82, 85, 98

Z. Shi, T. M. Hospedales, and T. Xiang. Transfer learning by ranking for weakly supervised object annotation. In *British Machine Vision Conference (BMVC)*, 2012. 139

J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Computer Vision-ECCV 2006*, pages 1–15, 2006. 4, 5, 51, 52, 92

Guang Shu, Afshin Dehghan, and Mubarak Shah. Improving an object detector and extracting regions using superpixels. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3721–3727. IEEE, 2013. 26

P. Siva and T. Xiang. Weakly supervised object detector learning with model drift detection. In *ICCV*, 2011a. 137

P. Siva and T. Xiang. Weakly supervised object detector learning with model drift detection. In *Computer Vision (ICCV), 2011 IEEE International Conferenc on*, 2011b. 137

P. Siva, C. Russell, and T. Xiang. In defence of negative mining for annotating weakly labelled data. In *Computer Vision-ECCV 2012*, 2012. xii, 137, 138, 141, 142, 144

Ihor Smal, Marco Loog, Wiro Niessen, and Erik Meijering. Quantitative comparison of spot detection methods in fluorescence microscopy. *Medical Imaging, IEEE Transactions on*, 29(2):282–301, 2010a. 104, 105

Ihor Smal, Marco Loog, Wiro Niessen, and Erik Meijering. Quantitative comparison of spot detection methods in fluorescence microscopy. *Medical Imaging, IEEE Transactions on*, 29(2):282–301, 2010b. 105, 117

Peter HA Sneath, Robert R Sokal, et al. *Numerical taxonomy. The principles and practice of numerical classification.* 1973. 43

Raghav Subbarao and Peter Meer. Nonlinear mean shift for clustering over analytic manifolds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1168–1175. IEEE, 2006. 44

Richard Szeliski. *Computer vision: algorithms and applications.* Springer, 2010. 23

I. Theurkauff, C. Cottin-Bizonne, J. Palacci, C. Ybert, and L. Bocquet. Dynamic clustering in active colloidal suspensions with chemical signaling. *Phys. Rev. Lett.*, 108:268303, Jun 2012. 102, 103

Hamid R Tizhoosh. Image thresholding using type ii fuzzy sets. *Pattern recognition*, 38(12):2363–2372, 2005. 41

Michael S.Landy Toni P.Saarela. Combination of texture and color cues in visual segmentation. *Vision Research*, pages 58:59–67, 2012. 59, 61, 64, 72, 73

Antonio Torralba. How many pixels make an image? *Visual neuroscience*, 26(01): 123–131, 2009. 83

Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008. 23

JRR Uijlings, KEA van de Sande, T Gevers, and AWM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 5, 24, 26, 30, 134, 135

V.L. Ulrike. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007. 13, 16, 18, 33, 66, 85

R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6): 929–944, June 2007. 68, 87

K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010. 51

Koen EA van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1879–1886. IEEE, 2011. 46

Jan C van Gemert, Jan-Mark Geusebroek, Cor J Veenman, and Arnold WM Smeulders. Kernel codebooks for scene categorization. In *Computer Vision–ECCV 2008*, pages 696–709. Springer, 2008. 93

Manik Varma and Debajyoti Ray. Learning the discriminative power-invariance trade-off. In *Computer Vision (ICCV), 2007 IEEE International Conferenc on*, pages 1–8. IEEE, 2007. 51

Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *Computer Vision–ECCV 2008*, pages 705–718. Springer, 2008. xiv, 28, 45

Ashok Veeraraghavan, Rama Chellappa, and Mandyam Srinivasan. Shape-and-behavior encoded tracking of bee dances. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(3):463–476, 2008. ISSN 0162-8828. 101

Jaco Vermaak, Arnaud Doucet, and Patrick Pérez. Maintaining multimodality through mixture tracking. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1110–1116. IEEE, 2003. 8

Luminita A Vese and Tony F Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International journal of computer vision*, 50(3):271–293, 2002. 46

Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2217–2224. IEEE, 2011. 50

J. Wang, Y. Jia, X. Hua, C. Zhang, and L. Quan. Normalized tree partitioning for image segmentation. In *CVPR*, 2008a. 75, 76

J. Wang, Y. Jia, X. Hua, C. Zhang, and L. Quan. Normalized tree partitioning for image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, 2008b. 26, 87

Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer*

Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 3360–3367, 2010. 93

Shitong Wang, Fu-lai Chung, and Fusong Xiong. A novel image thresholding method based on parzen window estimate. *Pattern Recognition*, 41(1):117–129, 2008c. 41

Song Wang and Jeffrey Mark Siskind. Image segmentation with minimum mean cut. In *Computer Vision (ICCV), 2001 IEEE International Conferenc on*, volume 1, pages 517–524. IEEE, 2001. 33, 37

Song Wang and Jeffrey Mark Siskind. Image segmentation with ratio cut. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(6):675–690, 2003. 6, 34, 37

Xiaofang Wang, Huibin Li, C.-E. Bichot, S. Masnou, and Liming Chen. A graph-cut approach to image segmentation using an affinity graph based on l0-sparse representation of features. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 4019–4023, Sept 2013a. 4, 9, 11, 46, 53, 62, 68, 76, 92, 97, 98, 134, 135

Xiaofang Wang, Huibin Li, Simon Masnou, and Liming Chen. Sparse coding and mid-level superpixel-feature for l0-graph based unsupervised image segmentation. In *Computer Analysis of Images and Patterns*, pages 160–168, 2013b. 11, 25, 52, 53, 135

Xiaofang Wang, Chao Zhu, C.-E. Bichot, and S. Masnou. Graph-based image segmentation using weighted color patch. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 4064–4068, Sept 2013c. 11

Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin. Regionlets for generic object detection. In *ICCV'13: Proc. IEEE 14th International Conf. on Computer Vision*, December 2013d. URL `http://www.imicrov.com/papers/Regionlets.pdf`. 134

Yair Weiss. Segmentation using eigenvectors: a unifying view. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 975–982. IEEE, 1999. 34

Max Wertheimer. *Laws of organization in perceptual forms (partial translation)*. Hayes Barton Press, 1938. 3, 58, 59

Mark Winter, Eric Wait, Badrinath Roysam, Susan K Goderie, Rania Ahmed Naguib Ali, Erzsebet Kokovay, Sally Temple, and Andrew R Cohen. Vertebrate neural stem cell segmentation, tracking and lineaging with validation and editing. *Nature protocols*, 6(12):1942–1952, 2011. 105

Andrew P Witkin. Scale-space filtering, April 14 1987. US Patent 4,658,372. 41

John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009. 19, 63, 73

John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas S Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010. 17, 18, 19, 20

Zheng Wu, T.H. Kunz, and M. Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1185–1192, 2011. 105, 106

Zhenyu Wu and R Leahy. Tissue classification in mr images using hierarchical segmentation. In *Nuclear Science Symposium, 1990. Conference record: Including Sessions on Nuclear Power Systems and Medical Imaging Conference, 1990 IEEE*, pages 1410–1414. IEEE, 1990. 30, 32, 37

Wei Xia, Csaba Domokos, Jian Dong, Loong-Fah Cheong, and Shuicheng Yan. Semantic segmentation without annotating segments. xiv, 52

Hongyuan Zha Xiaofeng, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. In *Proc. the tenth Int. Conf. on Inf. and Knowl. Manag. (CIKM)*, pages 25–32, 2001. 66

X. Tang Y. Ke and F. Jing. The design of high-level features for photo quality assessment. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*, 2006. 140

Fei Yan, William Christmas, and Josef Kittler. Layered data association using graph-theoretic formulation with application to tennis ball tracking in monocular sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30 (10):1814–1830, 2008. 9

Allen Y. Yang, John Wright, Yi Ma, and Shankar S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008. 26

Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, CVPR, 2009 IEEE Conference on*, pages 1794–1801. IEEE, 2009. 93

Jianchao Yang, Kai Yu, and Thomas Huang. Efficient highly over-complete sparse coding using a mixture model. In *Computer Vision–ECCV 2010*, pages 113–126. Springer, 2010. 93

Xingwei Yang, Lakshman Prasad, and Longin Jan Latecki. Affinity learning with diffusion on tensor product graph. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):28–38, 2013. 75, 76

Alper Yilmaz. Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*. Citeseer, 2007. 44

Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm Computing Surveys (CSUR)*, 38(4):13, 2006. 105

D. Yining and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):800–810, 2001a. 86

D. Yining and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(8):800–810, August 2001b. 53, 75, 76

Kai Yu and Tong Zhang. Improved local coordinate coding using local tangents. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1215–1222, 2010. 94

Kai Yu, Yuanqing Lin, and John Lafferty. Learning image representations from the pixel level via hierarchical sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1713–1720. IEEE, 2011. 93

Stella X Yu. Segmentation using multiscale cues. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, volume 1, pages I–247. IEEE, 2004. 24

Stella X Yu and Jianbo Shi. Multiclass spectral clustering. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 313–319. IEEE, 2003. 24, 30, 33

Zhiding Yu, Ang Li, Oscar C Au, and Chunjing Xu. Bag of textons for image segmentation via soft clustering and convex shift. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 781–788, 2012. xiv, 27, 45, 93

Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision–ECCV 2012*, pages 343–356. Springer, 2012. 9

Chunjie Zhang, Jing Liu, Qi Tian, Changsheng Xu, Hanqing Lu, and Songde Ma. Image classification by non-negative sparse coding, low-rank and sparse decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1673–1680, 2011. 93

Hui Zhang, Jason E Fritts, and Sally A Goldman. Image segmentation evaluation: A survey of unsupervised methods. *computer vision and image understanding*, 110(2):260–280, 2008a. 52

Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8, 2008b. 9, 106, 112, 136

Yu Qian Zhao, Xiao Fang Wang, Frank Y Shih, and Gang Yu. A level-set method based on global and local regions for image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 26(01), 2012. 46

Yu Zhou, Xiang Bai, Wenyu Liu, and Longin J Latecki. Fusion with diffusion for robust visual tracking. In *Advances in Neural Information Processing Systems*, pages 2978–2986, 2012. 75, 76

C. Zhu, C. Bichot, and L. Chen. Multi-scale color local binary patterns for visual object classes recognition. In *Int. Conf. on Pattern Recognit.*, pages 3065–3068, 2010. 69

Liansheng Zhuang, Haoyuan Gao, Zhouchen Lin, Yi Ma, Xin Zhang, and Nenghai Yu. Non-negative low rank and sparse graph for semi-supervised learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2328–2335. IEEE, 2012. 21, 22, 73

Wenbin Zou, Kidiyo Kpalma, and Joseph Ronsin. Semantic image segmentation using region bank. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 922–925. IEEE, 2012. xiv, 51, 52, 92