Università Ca' Foscari di Venezia

Dottorato di Ricerca in Informatica, XXX ciclo

Ph.D. Thesis

Extensions of Dominant Sets and Their Applications in Computer Vision

Eyasu Zemene

Supervisor

Marcello Pelillo

PhD Coordinator

Ricardo Focardi

February, 2018

Author's Web Page: http://informatica.dais.unive.it/

Author's e-mail: eyasu201011@gmail.com/eyasu.zemene@unive.it

Author's address:

Dipartimento di Informatica
Università Ca' Foscari di Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348411
fax. +39 041 2348419
web: `http://www.dsi.unive.it`

This thesis is dedicated to my family

for their endless love, support and encouragement

Abstract

Many problems in computer vision can be formulated as a clustering problem, a problem that aims to organize a collection of data objects into groups or clusters, such that objects within a cluster are more "similar" to each other than they are to objects in the other groups.

Assuming the feature-based representation, a computer vision problem can be formulated as follows: Given a set of data points in a 'd' dimensional space, find the best partition of the space which gives us meaningful groups. The points in the representative metric space correspond to the feature vectors extracted from the object with their distances reflecting the dissimilarity relations. On the other hand, objects could also be described indirectly by their respective similarity relations, an approach which is more natural than feature-based technique as there are numerous application domains where it is not possible to find satisfactory features, but it is more natural to provide a measure of similarity.

This work proposes similarity based data clustering framework, based on extensions of the dominant sets framework using theories and mathematical tools inherited from graph theory, optimization theory and game theory, that could be adapted "flexibly" in a wide range of vision applications, thereby combining the research domain of computer vision and that of machine learning. In our system, clusters are in one-to-one correspondence with Evolutionary Stable Strategies (ESS) - a classic notion of equilibrium in evolutionary game theory field - of a so-called "clustering game". The clustering game is a non-cooperative game between two-players, where the objects to cluster form the set of strategies, while the affinity matrix provides the players' payoffs.

The dominant sets framework, a well-known graph-theoretic notion of a cluster which generalizes the concept of a maximal clique to edge-weighted graphs, has proven itself to be relevant in many computer vision problems such as action recognition, image segmentation, tracking, group detection and others. Its regularized counterpart, determining the global shape of the energy landscape as well as the location of its extrema, is able to organize the data to be clustered in a hierarchical manner. It generalizes the dominant sets framework in that putting the regularization parameter to zero results local solutions that are in one-to-one correspondence with dominant sets. In this thesis we propose constrained dominant sets, parameterized family of quadratic programs that generalizes both formulations, the dominant

sets framework and its regularized counterpart, in that here, only a subset of elements in the main diagonal is allowed to take the parameter, the other ones being set to zero. In particular, we show that by properly controlling a regularization parameter which determines the structure and the scale of the underlying problem, we are in a position to extract groups of dominant sets clusters which are constrained to contain user-specified elements. We provide bounds that allow us to control this process, which are based on the spectral properties of certain submatrices of the original affinity matrix.

Thanks to the many sensors, which generate a large amount of data every day, distributed in the society, there is a large amount of data for training and testing many computer vision systems. However, real data collected through those sensors is contaminated by outliers, and many computer vision tasks involve processing the available large amounts of data without any assumption on the existence of outliers. Recently, very few computer vision systems have shown that considering presence of outliers while solving computer vision problems help boosting the state-of-the-art results. However, most of the systems either try just to detect outliers from the computer vision datasets or solve their problems by detecting and rejecting outliers before applying the method on the dataset. Our proposed work is robust to outliers, and since we also believe that it is important for clustering methods to detect data consensuses and isolated outliers simultaneously in a clustering task, this thesis proposes a method for simultaneous clustering and outliers detection.

Evolutionary game theory offers a whole class of simple dynamical systems to solve quadratic (constrained) optimization problems like ours. It envisages a scenario in which pairs of players are repeatedly drawn at random from a large population of individuals to play a symmetric two player game. One of the best known class of game dynamics to extract (constrained) dominant set from a graph is the so-called replicator dynamics whose computational complexity is quadratic per step which makes it handicapped for large-scale applications. In this thesis, we propose a fast algorithm, based on dynamics from evolutionary game theory, which is efficient and scalable to large-scale real-world applications.

In general, the clustering algorithm proposed in this thesis has many interesting properties, suitable for solving many computer vision problems, such as: it does clustering while obliterating outliers in simultaneous fashion, it doesn't need any a prior knowledge on the number of clusters, able to deal with compact clusters and with situations involving arbitrarily-shaped clusters in a context of heavy background noise, able to extract clusters which are constrained to contain user-specified elements, does not have any assumptions with the structure of the affinity matrix, it is fast and scalable to large scale problems, and others. This increased flexibility leads to an efficient method that we apply on different computer vision problems: interactive image segmentation, co-segmentation, multi-target multi-camera tracking in multiple non overlapping cameras, to extract dense neighbors which we have used to constrain the diffusion process locally, geo-localization and for person re-identification.

Acknowledgments

Thank you God for making my Ph.D. journey "simpler".

I then have to thank my parents, Adoye and Zeme, for their endless love and support through out my life, for always understanding the things I said, the things I didn't say, and the things I never planned on telling to them. My brothers and sisters also deserve my wholehearted thanks as well. Special thanks goes to my little brother, Dave, who sacrifice to support not only me but also our family.

I would like to sincerely thank my supervisor prof. Marcello Pelillo, for his scientific guidance during my Ph.D. studies. I consider my self very fortunate for being able to work with a very considerate and encouraging professor like him. Without his offering to start and accomplish this research, I would not be able to start and finish my study at Ca' Foscari.

I would also like to thank my co-authors, prof. Andrea Prati for his constant kind help and Dr. Mubarak Shah for granting me a visiting research scholar position to join the Center for Research in Computer Vision (CRCV) group, for his great advice, encouraging and fruitful discussion and deep scientific guidance which helped me deepen my knowledge in computer vision. I am grateful to my external reviewers prof. Richard Wilson and prof. Lamberto Ballan for the time they spent on carefully reading the thesis and for their useful comments and suggestions. Thanks to the Department of Computer Science of Ca'Foscari University of Venice for financing my studies with a three years Ph.D. grant, and I would also like to thank prof. Ricardo Focardi and Nicola Miotello for their coordination of the Ph.D. study.

Loving thanks to all my long standing friends Yoni, Leule, Tinsu, Tewo, Sure, Seyum ... The funny moments we spent together, 'ye buna ereft worie ena tinkosa, ye campo kefita ...' make my studies simple and full of unforgettable experiences. Yoni, thank you for accompanying me on my studies. The time we spent together played important roles along the journey as we mutually engaged in making sense of the various challenges we faced and in providing encouragement to each other at those times when it seemed impossible to continue.

Eyasu Zemene,
Venice, February 2018

Contents

List of Figures

List of Tables

List of Algorithms

Notations

$G = (V, E, w)$	A graph G with vertex set V, edge set E and weight function w
R_+	Set of non-negative reals
$\sigma(\mathbf{x})$	Support of $\mathbf{x}$
Δ	Standard simplex
Δ_S	Face of Δ corresponding to S
$.^\top$	Transposition
$\hat{I}_S$	Digonal matrix whose diagonal elements are set to 1 in correspondece to vertices contained in V\ S and to 0 otherwise
$\mathtt{A} = a_{ij}$	Matrix in $R^{n\times n}$ with $(i, j)^{th}$ element a_{ij}
$\mathbf{e}$	column vector of all ones
$\mathbf{e}_i$	a zero column vector whose i^{th} element is one
$\mathbf{x}$	Column vectors in R^n
I	Identity matrix
x_i	The i^{th} element of vector $\mathbf{x}$

Preface

The dissertation is submitted for the degree of doctor of philosophy at Ca' Foscari University of Venice. It presents a work in the area of pattern recognition and machine learning with various applications in computer vision. The first chapter is devoted to the introduction of the generalization of the well known clustering framework, dominant sets, and its different extensions which have been published in [12][13][14]. Their various computer vision applications are presented in the rest of the chapters. The second chapter presents the first application of constrained dominant sets to interactive image segmentation, which has been presented in [12], and its extension to image co-segmentation (both in supervised and unsupervised flavor) which is under review on Transactions on Pattern Analysis and Machine Intelligence (TPAMI). The third chapter presents the application of constrained dominant sets to retrieval which has been appeared in [15]. One of the interesting results of the thesis is the effective application of constrained dominant sets to multi-target tracking in multiple non-overlapping cameras which is under review on TPAMI (arXiv version is in [16]). The last chapter of the thesis presents how the problem of large-scale image geo-localization is addressed using (constrained) dominant sets. The result has been appeared in TPAMI [17]. The last two chapters are the results of the fruitful collaboration with Dr. Mubarak Shah during my two consecutive visits of Center for Research in Computer Vision at University of Central Florida. Finally, the Appendix of the thesis presents the different results which support the theories behind constrained dominant sets.

Introduction

The beginning is the most
important part of the work.

Plato

Computer vision is the science and technology of machines that see. As a scientific discipline, it is concerned with the theory and technology for building artificial systems that obtain information from images or multi-dimensional data. Its applications include industrial, biomedical, scientific, environment exploration, surveillance, document understanding, video analysis, graphics, games and entertainment. Computer vision systems have been designed that can control robots or autonomous vehicles, inspect machine parts, detect and recognize human faces, retrieve images from large databases according to content, reconstruct large objects or portions of cities from multiple photographs, track suspicious people or objects in videos, in remote sensing, and more.

Many problems in computer vision can be formulated as a clustering problem, a problem that aims to organize a collection of data objects into groups or clusters such that objects within a cluster are more “similar” to each other than they are to objects in the other groups. Clustering arises naturally in many fields; whenever one has a set of objects, it is natural to seek a method for grouping ”similar objects” together based on an underlying measure of similarity. The usual approach is to represent the set of objects as a set of abstract points (figure 1 middle), and try to follow some ”feature based” approaches where the abstract points are points in a metric space with distances reflecting the (dis)similarity — the closer the points, the more similar they are, (figure 1 right). Thus, clustering is centered around an intuitively compelling but vaguely defined goal: given an underlying set of points, partition them into a collection of clusters [18].

Though its study is unified only at this very general level of description, at the level of concrete methods and algorithms, one quickly encounters several different clustering techniques, including hierarchical, spectral, information-theoretic, and centroid-based, as well as those arising from combinatorial optimization and from probabilistic generative models. These techniques are based on diverse underlying principles, and they often lead to qualitatively different results. The very nature of the problem being ill-posed, since any given collection of data items can be clustered in drastically different ways, forces the community have very many different methods to improve existing approaches on specific applications. The very many clustering methods, developed in several communities, to support its extensive use in computer vision, pattern recognition, information retrieval, data mining, etc., put

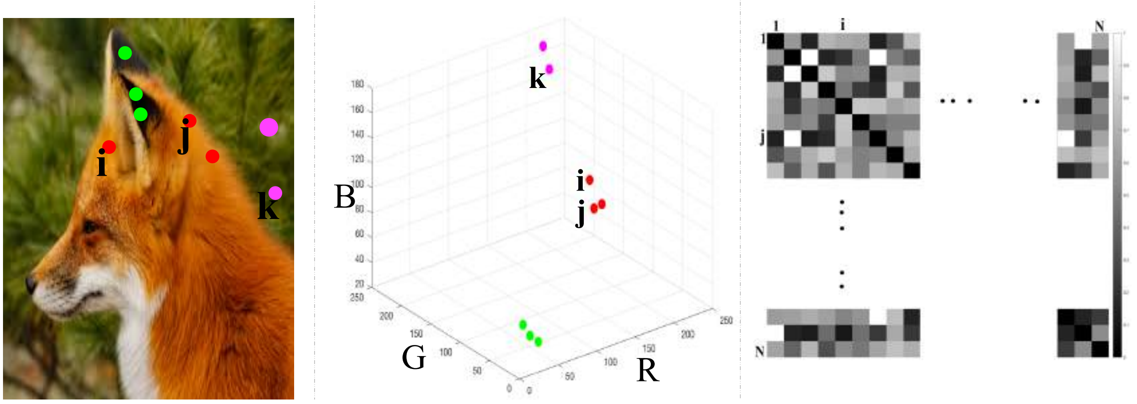


Figure 1: **Left:** An examplar image with sample representative pixels from different regions **Middle:** The points in the representative metric space which correspond to the feature vectors (RGB values of the pixels) extracted from the object with their distances reflecting the dissimilarity relations. Similar dotted colors represent some of the similar pixels of the image. **Right:** Similarity representation of N pixels; as can be seen pixel 'i' and pixel 'j' (pixels from similar region) have a similarity which is closer to one.

some criteria that provide significant distinctions between clustering methods and can help selecting appropriate candidate methods for one's problem: Objective of clustering and, nature of provided data items, nature of available information and the clusters' nature provide one with important information about the clustering model which one wants to build.

Clustering based computer vision algorithms are developed either using a novel clustering method or using an existing one modified in such a way as to fulfill some specific requirements to one's problem. There is no unified clustering scheme for solving a computer vision problem, several problems have been addressed using different kinds of clustering frameworks; there are lots of different clustering frameworks which are designed for solving tracking, other several algorithms for addressing the problem of segmentation, etc. Some of the off-the-shelf clustering applications in computer vision include object tracking [19][20][21], applications of segmentation in medicine [22], action localization [23] [24], text-to-video alignment [25][26], object co-localization in videos and images [27] or instance-level segmentation [28], in semantic segmentation which is the task of clustering parts of images together which belong to the same object class, and this type of algorithm has several use cases such as detecting road signs [29] in tumor detection [30] detecting medical instruments in operations [31] colon crypts segmentation [32], land use and land cover classification [33], object tracking [19][20][21], and others.

This work proposes similarity based clustering technique, based on extensions of the dominant sets framework, which could be used for solving wide range of computer vision problems in a flexible manner. The proposed clustering framework, as introduced below and detailed in the first chapter of the thesis, is with many interesting properties, suitable for solving several computer vision problems, such as: it does clustering while obliterating outliers in simultaneous fashion, able to deal with overlapping clusters, it doesn't need any a prior knowledge on the number of clusters, able to deal with compact clusters and with situations involving arbitrarily-shaped clusters in a context of heavy background noise, able to extract clusters which are constrained to contain user-specified elements, does not have any assumptions with the structure of the affinity matrix, it is fast and scalable to large scale problems, it provides a principled measure of a cluster's cohesiveness as well as a measure of a vertex participation to its assigned group, and others.

The thesis, proposing such a flexible similarity based clustering framework, applies cluster analysis as a unified approach for a wide range of vision applications, thereby combining the research domain of computer vision and that of machine learning. In our system, a robust similarity based clustering technique inspired by game theory, clusters are in one-to-one correspondence with Evolutionary Stable Strategies (ESS) - a classic notion of equilibrium in evolutionary game theory field - of a so-called "clustering game". The clustering game is a non-cooperative game between two-players, where the objects to be clustered form the set of strategies, while the affinity matrix provides the players' payoffs. Assume, in the above example, the image has N pixels; the similarity among the pixels (figure 1 right) is represented by a square matrix of size N which, in the game setting, represents the payoff the players get while playing the game. Each player simultaneously selects a pixel from the set of N pixels the image has, after having revealed his choice, he receives a payoff according to the similarity that the selected pixel has with respect to the opponents'. From a classical game-theoretic point of view, players are rational individuals with complete knowledge of the game setting, and they act with intent to maximize their personal income. Since the received payoff depends on the similarity of the selected pixels, it is in each players interest to select pixels belonging to a common region, since a cluster exhibits a high internal similarity. Hence, by repeatedly playing this game, the hypothesis of cluster membership of each player are expected to converge to a common solution (or equilibrium), which in turn represents a region from the segmented image.

The increased flexibility in the proposed framework leads to an efficient method that we apply, as introduced below and detailed from the second to the last chapters of the thesis, on different computer vision problems: interactive image segmentation, co-segmentation, multi-target multi-camera tracking in multiple non overlapping cameras, to extract dense neighbors which we have used to constrain the diffusion process locally, geo-localization and for person re-identification.

The dominant sets framework has been proven to be relevant in several computer vision problems, including (automatic) image and video segmentation [34, 35], action recognition [36], tracking [37], geo-localization [17]. To determine a dominant set, as it has an intriguing connection with the solutions of a (continuous) quadratic optimization problem, one can use a straightforward dynamics from evolutionary game theory such as replicator dynamics [38] which can be coded in a few lines of any high-level programming language and can easily be implemented in a parallel network of locally interacting computational units. Moreover, its connection with non-cooperative game theory has opened the door to elegant generalizations to directed graphs [39], to hypergraphs [40] and to multi-graphs, i.e.graphs that are permitted to have multiple edges between vertices [41]. It has many interesting features such as providing a principled measure of a cluster's cohesiveness as well as a measure of a vertex participation to its assigned group.

Hierarchical dominant sets framework is regularized counterpart of dominant sets formulation. By determining the global shape of the energy landscape as well as the location of its extrema, it is able to organize the data to be clustered in a hierarchical manner. It generalizes the dominant sets framework in that putting the regularization parameter to zero results local solutions that are in one-to-one correspondence with dominant sets. It is a parametrized (continuous) quadratic program controlled by a non-negative parameter which is varied during a clustering process. A bigger parameter considers the whole data as a single cluster while a smaller value splits the data in to smaller sets. So, starting with a bigger parameter, it is possible, by properly varying the regularization parameter during the clustering process, to organize the data to be clustered in a hierarchical manner.

This thesis proposes constrained dominant sets (CDS), parameterized family of quadratic programs that generalizes both formulations, the dominant sets framework and its regularized counterpart, in that here, only a subset of elements in the main diagonal is allowed to take the parameter, the other ones being set to zero. The framework has many interesting features that the dominant set formulation does not serve: 1) It can enumerate all the dominant sets without the implicit change of scale of the problem, we have a guarantee that all the enumerated dominant sets are local solutions of the original problem. This is not the case in the non-parametrized dominant sets formulation as the procedure is done in an iterative manner by removing the already identified dominant set from the graph. 2) It allows one to obtain "soft" partitions of the input data, by allowing a point to belong to more than one cluster. Of-course, the original dominant sets formulation also allows "soft" partitions. However, this is done by asymmetric extension of the similarity matrix which may be a problem for the dynamics to converge properly. 3) By properly controlling a regularization parameter which determines the structure and the scale of the underlying problem, we are in a position to extract groups of dominant sets clusters which are constrained to contain user-specified elements. We provide bounds that allow us to control this process, which are based on the spectral properties of certain submatrices of the original affinity matrix. Section 1.3 of the first chapter introduces CDS followed by another two section which details the dynamics used to extract CDS and the proposed fast approach to extract CDS when the size of the graph gets larger. Section 1.4 of chapter 1 deals with outliers, data which does not obey the assumed model. Thanks to the many sensors, which generate a large amount of data every day, distributed in the society, there is a large amount of data for training and testing many computer vision systems. However, real data collected through those sensors is contaminated by outliers. Since outliers severely affect many computer vision systems, it is important to explicitly deal with them. Many computer vision tasks however involve processing the available large amounts of data with out any assumption on the existence of outliers. Recently, very few computer vision systems have shown that considering presence of outliers while solving computer vision problems help boosting the state-of-the-art results. However, most of the systems either try just detect outliers from the computer vision datasets or solve their problems by detecting and rejecting outliers before applying the method on the dataset, they just aim first to come up with the result of the robust estimation which is the *inlier/outlier* dichotomy of the data. Our proposed work is robust to outliers, and since we also believe that it is important for clustering methods to detect data consensuses and isolated outliers simultaneously in a clustering task, this thesis proposes a method for simultaneous clustering and outliers detection. The method utilizes some properties of a family of quadratic optimization problems related to dominant sets, a well-known graph-theoretic notion of a cluster which generalizes the concept of a maximal clique to edge-weighted graphs. Unlike most (all) of the previous techniques, in the proposed framework the number of clusters arises intuitively and outliers are obliterated automatically. The resulting algorithm discovers both parameters from the data. Experiments on real and on large scale synthetic dataset demonstrate the effectiveness of the approach and the utility of carrying out both clustering and outlier detection in a concurrent manner.

Chapter 2 of the thesis introduces the application of the proposed framework to interactive image segmentation and co-segmentation. Image segmentation has come a long way since the early days of computer vision, and still remains a challenging task. Modern variations of the classical (purely bottom-up) approach, involve, e.g., some form of user assistance (interactive segmentation) or ask for the simultaneous segmentation of two or more images (co-segmentation). At an abstract level, all these variants can be thought of as “constrained” versions of the original formulation, whereby the segmentation process is guided by some external source of information. In this thesis, we propose a new approach to tackle this kind of problems in a unified way. In particular, we shall focus on interactive segmentation and co-segmentation (in both the unsupervised and the interactive versions). The proposed algorithm can deal naturally with several type of constraints and input modality, including scribbles, sloppy contours, and bounding boxes, and is able to robustly handle noisy annotations on the part of the user.

In the third chapter of the thesis, constrained dominant sets is plugged in to a neighborhood analysis system to constrain the diffusion process locally to improve the performance of any retrieval systems. Learning new global relations based on an initial affinity of the database objects has shown significant improvements in similarity retrievals. Locally constrained diffusion process is one of the recent effective tools in learning the intrinsic manifold structure of a given data. Existing methods, which constrain the diffusion process locally, have problems - manual choice of optimal local neighborhood size, do not allow for intrinsic relation among the neighbors, fix initialization vector to extract dense neighbor - which negatively affect the affinity propagation. CDS alleviates these issues using a more robust neighborhood structure.

An interesting but difficult computer vision application is presented in chapter 4 of the thesis where a unified three-layer hierarchical approach for solving tracking problems in multiple non-overlapping cameras is proposed. Given a video and a set of detections (obtained by any person detector), we first solve *within-camera tracking* employing the first two layers of our framework and, then, in the third layer, we solve *across-camera tracking* by merging tracks of the same person in all cameras in a simultaneous fashion. To best serve our purpose, a constrained dominant sets clustering (CDSC) technique, a parametrized version of standard quadratic optimization, is employed to solve both tracking tasks. The tracking problem is cast as finding constrained dominant sets from a graph. That is, given a constraint set and a graph, CDSC generates a cluster (or clique), which forms a compact and coherent set that contains a subset of the constraint set. The approach is based on a parametrized family of quadratic programs that generalizes the standard quadratic optimization problem. In addition to having a unified framework that simultaneously solves within- and across-camera tracking, the third layer helps link broken tracks of the same person occurring during within-camera tracking. A standard algorithm to extract constrained dominant set from a graph is given by the so-called replicator dynamics whose computational complexity is quadratic per step which makes it handicapped for large-scale applications. In this thesis, we propose a fast algorithm, based on dynamics from evolutionary game theory, which is efficient and scalable to large-scale real-world applications. We have tested this approach on a very large and challenging dataset (namely, MOTchallenge DukeMTMC) and show that the proposed framework outperforms the current state of the art. We also applied it to solve the *re-identification* problem. Towards that end, we have performed experiments on MARS, one of the largest and challenging video-based person re-identification dataset, and have obtained excellent results. These experiments demonstrate the general applicability of the proposed framework for non-overlapping across-camera tracking and person re-identification tasks.

The last chapter of the thesis presents a new approach for the challenging problem of geo-localization using image matching in a structured database of city-wide reference images with known GPS coordinates. We cast the geo-localization as a clustering problem of local image features. Akin to existing approaches to the problem, our framework builds on low-level features which allow local matching between images. For each local feature in the query image, we find its approximate nearest neighbors in the reference set. Next, we cluster the features from reference images using Dominant Set clustering, which affords several advantages over existing approaches. First, it permits variable number of nodes in the cluster, which we use to dynamically select the number of nearest neighbors for each query feature based on its discrimination value. Second, this approach is several orders of magnitude faster than existing approaches. Thus, we obtain multiple clusters (different local maximizers) and obtain a robust final solution to the problem using multiple weak solutions through constrained Dominant Sets clustering on global image features, where we enforce the constraint that the query image must be included in the cluster. This second level of clustering also bypasses heuristic approaches to voting and selecting the reference image that matches to the query. We evaluate the proposed framework on an existing dataset of 102k street view images as well as a new larger dataset of 300k images, and show that it outperforms the state-of-the-art by 20% and 7%, respectively, on the two datasets.

The main contributions of this thesis are summarized as follows:

- It adopts the dominant sets framework to solve different machine learning and computer vision problems.

- The notion of path-based similarity measures, which exploit connectedness information of the elements to be clustered, is adopted to the dominant sets framework resulting a very simple and efficient algorithm which helps one extract elongated structures in the presence of heavy background noise.

- It extends the dominant sets framework to solve the problem of simultaneous clustering and outlier detection. The algorithm is the first algorithm which solves the problem without any prior knowledge of the number clusters and number of outliers. Previous state-of-the-art approaches need a priori the number of outliers and number of clusters.

- It introduces constrained dominant sets, a formalization which generalizes the dominant sets formulation and its regularized counterpart (hierarchical dominant sets), and have successfully applied it to many interesting computer vision problems.

- It develops an algorithm for automatic dense neighbor selection which improves the performance of any retrieval system. The algorithm is new and is with some proposition which answers some open computer vision problems: automatically selecting a reasonable local neighborhood size is an open computer vision and machine learning issue.

- It develops a novel framework for interactive image segmentation which can deal naturally with any type of input modality, including scribbles, sloppy contours, and bounding boxes, and is able to robustly handle noisy annotations on the part of the user. It extends the framework to solve the problem of image co-segmentation.

- It proposes an original fast algorithm to solve the problem of geo-localization using theories inherited from evolutionary game theory, to localize an image (video) to its GPS-coordinates.

- It involves in the development of an algorithm which solves (simultaneously) multi target multi person tracking and person re-identification in multiple non-overlapping cameras.

1 Extensions of Dominant Sets Clustering

> The only groups I willingly joined were spontaneous, short-lived, and usually game-playing.
>
> Meredith Marple

1.1 Introduction

Clustering refers to the process of extracting maximally coherent groups from a set of objects. Traditional approaches to this problem are based on the idea of partitioning the input data into a predetermined number of classes, thereby obtaining the clusters as a by-product of the partitioning process. Some of the limitations of the partitional paradigm include [42]: the number of classes should be known in advance and all the input data will have to get assigned to some class. There are, however, various applications for which it makes little sense to force all data items to belong to some group, a process which might result either in poorly-coherent clusters or in the creation of extra spurious classes. The other intrinsic limitation is the constraint that each element cannot belong to more than one cluster. There are a variety of important applications, however, where this requirement is too restrictive. Examples abound and include, e.g., clustering micro-array gene expression data (wherein a gene often participate in more than one process), clustering documents into topic categories, perceptual grouping, and segmentation of images with transparent surfaces. The symmetry assumption, namely the requirement that the similarities between the data being clustered be symmetric (and non-negative), is the other major problem. Asymmetric (or, more generally, non-metric) similarities are preferred (in image and video processing) in the presence of partially occluded objects, in pairwise structural alignments of proteins that focus on local similarity and others. In early 2000, the dominant set framework, a formalization of the very notion of a cluster that considers the clustering process as a sequential search, is proposed which deals with all the above limitations. Dominant sets, a well-known graph-theoretic notion of a cluster, generalizes the concept of a maximal clique to edge-weighted graphs and has proven to be relevant in many computer vision problems, including (automatic) image and video segmentation [34, 35], action recognition [36], tracking [37], geo-localization [17]. To determine a dominant set, as it has an intriguing connection with the solutions of a (continuous) quadratic optimization problem, one can use a straightforward dynamics from evolutionary game theory, replicator dynamics [38] which can be coded in a few lines of any high-level programming language and can easily be implemented in a parallel network of locally interacting computational units. Moreover, its connection with non-cooperative game theory has opened the door to elegant generalizations to directed graphs [39], to hypergraphs [40] and to multi-graphs, i.e. graphs that are permitted to have multiple edges between vertices [41]. It has many interesting features such as providing a principled measure of a cluster's cohesiveness as well as a measure of a vertex participation to its assigned group.

The dominant sets clustering framework has been extended in different ways over the last decade. Hierarchical dominant sets [43] is regularized counterpart of dominant-set formulation. By determining the global shape of the energy landscape as well as the location of its extrema, it is able to organize the data to be clustered in a hierarchical manner. It generalizes the dominant-set framework in that putting the regularization parameter to zero results local solutions that are in one-to-one correspondence with dominant-set. Same authors have extend the dominant sets framework to deal with large-scale data (*e.g.* , pixel based image segmentation using dominant sets) [44]. Other extensions include its generalization to high-order similarities [40] and to multi-graphs [41].

The dominant sets framework and its extensions, though have proven themselves to be effective in many applications and serve most of our purpose well, have different limitations: problem in dealing with outliers, which is a problem of many computer vision applications. The dominant set framework, to deal with this problem, should be provided with the number of meaningful compact structures to be extracted so that it leaves clutter elements unassigned to any cluster. Since the framework uses a 'peel-off' strategy (take a graph, run the dynamics to extract a dominant set, remove those selected nodes from the graph, run the dynamics again on the rest of the graph, and continue the process until all the nodes are assigned to one of the clusters), in addition to changing the scale of the problem, it has a problem in dealing with overlapping clusters. In [45] a method is proposed to deal with this problem introducing an approach for enumerating equilibria of two-player symmetric games, by iteratively rendering unstable already visited ones through a particular asymmetric extension of the payoff matrix. The game dynamics which they have used to solve the problem, however, has a convergence problem with the extended asymmetric payoff matrix. Another typical problem associated to dominant sets is that they tend to favor compact clusters, the problem therefore remains as to how to deal with situations involving arbitrarily-shaped clusters in a context of heavy background noise. To deal with the last problem, we feed the dominant sets algorithm with a path-based similarity measure which considers connectivity information of the elements being clustered, thereby transforming clusters exhibiting an elongated structure under the original similarity function into compact ones. We deal with its limitation related to outliers proposing a unified approach for simultaneous clustering and outlier detection in data, a method which controls the importance of the extracted compact sets which enables the proposed clustering algorithm to converge at some point and leaves the clutter background as outliers. Unlike all the previous techniques, in this framework, the number of clusters arises intuitively and outliers are obliterated automatically. The resulting algorithm discovers both parameters from the data. Experiments on real and on large scale synthetic dataset demonstrate the effectiveness of its approach and the utility of carrying out both clustering and outlier detection in a concurrent manner. We alleviates the other issues proposing constrained dominant-set, parameterized family of quadratic programs that generalizes both the dominant sets formulation and its regularized counterpart, in that here, only a subset of elements in the main diagonal is allowed to take the parameter, the other ones being set to zero. The framework has many interesting features that the dominant set formulation does not serve: 1) It can enumerate all the dominant sets without the implicit change of scale of the problem, we have a guarantee that all the enumerated dominant sets are local solutions of the original problem. This is not the case in the non-parametrized dominant set formulation as the procedure is done in an iterative manner by removing the already identified dominant set from the graph. 2) It allows one to obtain ”soft” partitions of the input data, by allowing a point to belong to more than one cluster. Of-course, the original dominant set formulation also allows ”soft” partitions. However, this is done by asymmetric extension of the similarity matrix which may be a problem for the dynamics to converge properly [45]. 3) By properly controlling a regularization parameter which determines the structure and the scale of the underlying problem, we are in a position to extract groups of dominant-set clusters which are constrained to contain user-specified elements. We provide bounds that allow us to control this process, which are based on the spectral properties of certain submatrices of the original affinity matrix. This increased flexibility leads to an efficient method that we apply on different computer vision problems.

Evolutionary game theory offers a whole class of simple dynamical systems to solve quadratic constrained optimization problems like ours. It envisages a scenario in which pairs of players are repeatedly drawn at random from a large population of individuals to play a symmetric two player game. One of the best known class of game dynamics to extract constrained dominant set from a graph is the so-called replicator dynamics whose computational complexity is quadratic per step which makes it handicapped for large-scale applications. In this thesis, we propose a fast algorithm, based on dynamics from evolutionary game theory, which is efficient and salable to large-scale real-world applications.

The rest of the chapter is organized as follows: In the next section, we provide a brief introduction to dominant sets and their characterization, and in section 1.3, we introduce the constrained dominant sets framework. We will show the different possible cases of constraint sets and the bounds that allows to control the process. Next we will show some experimental results on random graph instances and DIMACS benchmark graphs. Our simultaneous clustering and outlier detection is presented in Section 1.4 followed by a section that presents the path based dominant sets clustering.

1.2 Dominant Sets Clustering

The dominant set framework provides a formalization of the very notion of a cluster and considering the clustering process as a sequential search of structures in a data. Dominant sets has intriguing connection with game theory, graph-theory and optimization theory. From the game-theoretic perspective, clusters are in one-to-one correspondence with Evolutionary Stable Strategies (ESS) - a classic notion of equilibrium in evolutionary game theory field - of a so-called “clustering game”; in the graph-theoretic context, they are edge-weighted generalizations of the maximal clique; finally, from an optimization point of view, they can be characterized in terms of solutions to a simplex-constrained quadratic optimization problem.

The framework represents subset of vertices as a cluster (a.k.a dominant set), if they satisfies two basic properties of a cluster, i.e.

- *internal coherence*: elements belonging to the cluster should have high mutual similarities.
- *external incoherence*: introducing external elements will destroy the internal coherency.

In this section, we review the formalization of dominant sets, which supports this claim, from a combinatorial perspective and show the relations it has with standard quadratic optimization and game-theoretic notion of evolutionary stable strategy.

In the dominant sets framework, the data to be clustered are represented as an undirected edge-weighted graph with no self-loops $G = (V, E, w)$, where $V = \{1, ..., n\}$ is the vertex set, $E \subseteq V \times V$ is the edge set, and $w : E \to R_+$ is the (positive) weight function. Vertices in G correspond to data points (objects to be clustered), edges represent neighborhood relationships, and edge-weights reflect similarity between pairs of linked vertices. As customary, we represent the graph G with the corresponding weighted adjacency (or similarity) matrix, which is the $n \times n$ nonnegative, symmetric matrix $\mathtt{A} = (a_{ij})$ defined as $a_{ij} = w(i, j)$, if $(i, j) \in E$, and $a_{ij} = 0$ otherwise. Since in G there are no self-loops, note that all entries on the main diagonal of $\mathtt{A}$ are zero.

In an attempt to formally capture dominant sets’ notion, we present some notations and definitions.

1.2.1 Definitions

For a non-empty subset $S \subseteq V$, $i \in S$, and $j \notin S$, define

$$\phi_S(i,j) = a_{ij} - \frac{1}{|S|} \sum_{k \in S} a_{ik} \ . \tag{1.1}$$

This quantity measures the (relative) similarity between nodes j and i, with respect to the average similarity between node i and its neighbors in S. Note that $\phi_S(i,j)$ can be either positive or negative. Next, to each vertex $i \in S$ we assign a weight defined (recursively) as follows:

$$w_S(i) = \begin{cases} 1, & \text{if} \quad |S| = 1, \\ \sum_{j \in S \setminus \{i\}} \phi_{S \setminus \{i\}}(j,i) w_{S \setminus \{i\}}(j), & \text{otherwise} \ . \end{cases} \tag{1.2}$$

Intuitively, $w_S(i)$ gives us a measure of the overall similarity between vertex i and the vertices of $S \setminus \{i\}$ with respect to the overall similarity among the vertices in $S \setminus \{i\}$. Therefore, a positive $w_S(i)$ indicates that adding i into its neighbors in S will increase the internal coherence of the set, whereas in the presence of a negative value we expect the overall coherence to be decreased. Finally, the total weight of S can be simply defined as

$$W(S) = \sum_{i \in S} w_S(i) \ . \tag{1.3}$$

A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be a *dominant set* if:

1. $w_S(i) > 0$, for all $i \in S$,
2. $w_{S \cup \{i\}}(i) < 0$, for all $i \notin S$.

It is evident from the definition that a dominant set satisfies the two basic properties of a cluster: internal coherence and external incoherence. Condition 1 indicates that a dominant set is internally coherent, while condition 2 implies that this coherence will be destroyed by the addition of any vertex from outside. In other words, a dominant set is a maximally coherent data set.

Example: Let us consider a graph with nodes $\{1,2,3\}$, which forms a coherent group (dominant set) with edge weights 20, 21 and 22 as shown in Fig. 1.1(a). Now, let us try to add a node {4} to the graph which is highly similar to the set {1,2,3} with edge weights of 30, 35 and 41 (see Fig. 1.1(b)). Here, we can see that adding node {4} to the set increases the overall similarity of the new set {1,2,3,4}, that can be seen from the fact that the weight associated to the node {4} with respect to the set {1,2,3,4} is positive, ($W_{\{1,2,3,4\}}(4) > 0$). On the contrary, when adding node {5} which is less similar to the set {1,2,3,4} (edge weight of 1 - Fig. 1.1(c)) the

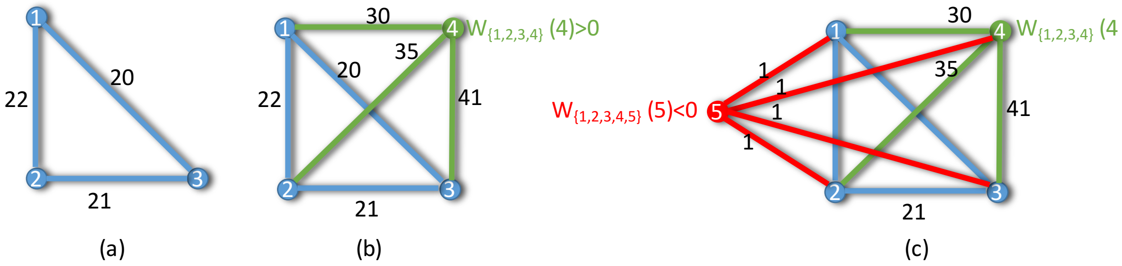


Figure 1.1: Dominant set example: (a) shows a compact set (dominant set), (b) node 4 is added which is highly similar to the set {1,2,3} forming a new compact set. (c) Node 5 is added to the set which has very low similarity with the rest of the nodes and this is reflected in the value $W_{\{1,2,3,4,5\}}(5)$.

overall similarity of the new set {1,2,3,4,5} decreases, since we are adding to the set something less similar with respect to the internal similarity. This is reflected by the fact that the weight associated to node {5} with respect to the set {1,2,3,4,5} is less than zero ($W_{\{1,2,3,4,5\}}(5) < 0$).

From the definition of a dominant set, we could say the set {1,2,3,4} (Fig. 1.1 (b)) forms a dominant set, as it satisfies both criteria (internal coherence and external incoherence). While the weight associated to the node out side of the set (dominant set) is less than zero, $W_{\{1,2,3,4,5\}}(5) < 0$.

1.2.2 Link to Optimization Theory

Consider the following linearly-constrained quadratic optimization problem:

$$\begin{array}{ll} \text{maximize} & f(\mathbf{x}) = \mathbf{x}^\top \mathtt{A}\mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array} \tag{1.4}$$

where $\mathbf{x}^\top$ denotes transposition of vector $\mathbf{x}$ and

$$\Delta = \left\{ \mathbf{x} \in R^n \ : \ \sum_{i=1}^{n} x_i = 1, \ \text{and } x_i \geq 0 \text{ for all } i = 1 \ldots n \right\}$$

is the standard simplex of R^n.

The main result presented in [34, 35], assuming the affinity $\mathtt{A}$ symmetric, provides a one-to-one relation between dominant sets and the local solutions of (1.4). In particular, it is shown that if S is a dominant set then its "weighted characteristics vector," which is the vector of Δ defined as,

$$x_i = \begin{cases} \frac{w_S(i)}{W(s)}, & \text{if} \quad i \in S, \\ 0, & \text{otherwise} \end{cases}$$

is a strict local solution of (1.4). Conversely, under mild conditions, it turns out that if $\mathbf{x}$ is a (strict) local solution of program (1.4) then its "support"

$$\sigma(\mathbf{x}) = \{i \in V \ : \ x_i > 0\}$$

is a dominant set.

By virtue of this result, we can find a dominant set by first localizing a solution of program (1.4) with an appropriate continuous optimization technique, and then picking up the support set of the solution found. In this sense, we indirectly perform combinatorial optimization via continuous optimization. A generalization of these ideas to hypergraphs has recently been developed in [40].

1.2.3 Link to Game Theory

From an optimization point of view, dominant sets are in one-to-one correspondence with solutions to a simplex-constrained quadratic optimization problem. This relation to optimization theory, however, exists as long as we have a symmetric affinity matrix. Interestingly, there is a theory beyond optimization which serves the purpose well with asymmetric affinity, it is game theory.

The relation with game theory exists assuming the players play non-cooperative symmetric game. Symmetricity in a game setting implies that the two players share same payoff matrix, and non-cooperative mean that both players have perfect knowledge of the game and take independent decisions about the strategies to play. From the dominant sets formulation, the objects to be clustered (vertices V) represent the strategies to be chosen while the affinity matrix $\mathtt{A}$ represents the payoff matrix which summarizes the gain each of the players obtains while a pair of strategies are played, i.e, if player 1 chooses the i^{th} strategy (from the set of objects that we have) to play while player 2 chooses strategy j ((i, j) is chosen from the set $V \times V$), a_{ij} will be the gain player 1 obtains and player 2 will receive a_{ji}. While we play a game, we should not show any preference for our opponent, if our opponent knows what we are going to play, he knows what to play to be the winner. A mixed strategy is a randomization in the selection of the pure strategies which models a stochastic playing strategy of a player. If player 1 plays a mixed strategy $\mathbf{x}_1$ while player 2 chooses to play mixed strategy $\mathbf{x}_2$, *i.e.* , ($(\mathbf{x}_1, \mathbf{x}_2)$ is chosen from the simplex $\Delta \times \Delta$), then the expected payoff the players receive is computed as $\mathbf{x}_1^\top \mathtt{A}\mathbf{x}_2$ (for player 1) and $\mathbf{x}_2^\top \mathtt{A}\mathbf{x}_1$(for player 2).

From the clustering perspective, mixed strategies can be regarded as a cluster hypothesis, where x_i represents the probability of having the i^{th} object in the cluster. In any game setting, players play to maximize their gain. Giving high probability, while choosing a mixed strategy to play, for those strategies with higher payoff, they maximize their gain. Considering the payoff matrix as the affinity matrix $\mathtt{A}$ of the dominant set framework, a game, since players give high probability for those strategies which maximizes their payoff, selects objects having high mutual similarities which in turn results a dominant set (a cluster). The zero diagonal in the dominant sets formulation is also consistent with the game setting; if we do not set the diagonal of the payoff matrix to zero, the best strategy for each player would be to coordinate the selection towards exactly the same object, which is useless for the clustering purposes. On the other hand, by penalizing perfect coordination, one forces the players to anti-coordinate, resulting eventually in an equilibrium, where the two players beliefs about the cluster membership are conflicting, which is again unwanted. A third, desired possibility is that the players end up with a so-called symmetric (Nash) equilibrium, where the beliefs about cluster membership coincide.

A mixed strategy $(\mathbf{x}_1, \mathbf{x}_2) \in \Delta \times \Delta$ is said to be a *Nash equilibrium* if no player has an incentive to unilaterally deviate from it, *i.e.* , given the opponent's strategy being fixed the following holds

$$\mathbf{y}_2^\top \mathtt{A}\mathbf{x}_1 < \mathbf{x}_2^\top \mathtt{A}\mathbf{x}_1, \qquad \mathbf{y}_1^\top \mathtt{A}\mathbf{x}_2 < \mathbf{x}_1^\top \mathtt{A}\mathbf{x}_2$$

for all $(\mathbf{y}_1, \mathbf{y}_2) \in \Delta \times \Delta$.

In the case of symmetric Nash equilibrium (*i.e.* , when $\mathbf{x}_1 = \mathbf{x}_2$ which can be represented by a single strategy $\mathbf{x} \in \Delta$ played by two players), the above condition reduces to the following

$$\mathbf{x}^\top \mathtt{A}\mathbf{x} \geq \mathbf{y}^\top \mathtt{A}\mathbf{x} \qquad \text{for all } \mathbf{y} \in \Delta\,. \tag{1.5}$$

This condition implies, from the clustering game perspective, a condition where both players agree on the same hypothesis of cluster membership and no player has incentives to deviate from it. The internal coherency follows from the Nash condition. Indeed, if $\mathbf{x}$ is Nash equilibrium, then Eq. (1.5) implies that $(\mathtt{A}\mathbf{x})_i = \mathbf{x}^\top \mathtt{A}\mathbf{x}$ for all $i \in \sigma(\mathbf{x})$, *i.e.* every element of the cluster ($i \in \sigma(\mathbf{x})$) has the same average similarity with respect to the cluster. The Nash condition, however, does not necessarily guarantee the external incoherency (*a.k.a.* cluster maximality) which follows from a refinement of Nash Equilibrium that is Evolutionary Stable Strategy which is stable and that always implies a Nash equilibrium. A Nash equilibrium is an Evolutionary Stable Strategy(ESS) if the following holds for all strategies $\mathbf{y} \in \Delta \backslash \mathbf{x}$

$$\mathbf{y}^\top \mathtt{A}\mathbf{x} = \mathbf{x}^\top \mathtt{A}\mathbf{x} \implies \mathbf{x}^\top \mathtt{A}\mathbf{y} > \mathbf{y}^\top \mathtt{A}\mathbf{y}$$

We can express this as follows: suppose $\mathbf{x}$ is a Nash equilibrium and suppose that $\mathbf{y}$ is another Nash equilibrium, if we get the same score when $\mathbf{y}$ is played against its opponent and when when $\mathbf{x}$ is played against itself, by changing the role of $\mathbf{x}$ and $\mathbf{y}$ we get the following

$$\mathbf{x}^\top \mathtt{A}\mathbf{y} > \mathbf{y}^\top \mathtt{A}\mathbf{y}$$

.

ESS, which is also Nash equilibrium, satisfies both the internal and external criteria of a cluster, which implies that the problem of clustering becomes the problem of finding ESS-clusters of a clustering game. One of the distinguishing features of this approach is its generality as it allows one to deal in a unified framework with a variety of scenarios, including cases with asymmetric, negative, or high-order affinities. Moreover, when the affinity matrix $\mathtt{A}$ is symmetric, the notion of an ESS-cluster coincides with the original notion of dominant sets, which amounts to finding a (local) maximizer of $\mathbf{x}^\top \mathtt{A}\mathbf{x}$ over the standard simplex Δ [39].

1.3 Constrained Dominant Sets

Let $G = (V, E, w)$ be an edge-weighted graph with n vertices and let $\mathtt{A}$ denote as usual its (weighted) adjacency matrix. Given a subset of vertices $S \subseteq V$ and a parameter $\alpha > 0$, define the following parameterized family of quadratic programs:

$$\begin{array}{ll} \text{maximize} & f_S^\alpha(\mathbf{x}) = \mathbf{x}^\top(\mathtt{A} - \alpha \hat{I}_S)\mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array} \tag{1.6}$$

where $\hat{I}_S$ is the $n \times n$ diagonal matrix whose diagonal elements are set to 1 in correspondence to the vertices contained in $V \setminus S$ and to zero otherwise, and the 0's represent null square matrices of appropriate dimensions. In other words, assuming for simplicity that S contains, say, the first k vertices of V, we have:

$$\hat{I}_S = \begin{pmatrix} 0 & 0 \\ 0 & I_{n-k} \end{pmatrix}$$

where I_{n-k} denotes the $(n-k) \times (n-k)$ principal submatrix of the $n \times n$ identity matrix I indexed by the elements of $V \setminus S$. Accordingly, the function f_S^α can also be written as follows:

$$f_S^\alpha(\mathbf{x}) = \mathbf{x}^\top \mathtt{A}\mathbf{x} - \alpha \mathbf{x}_S^\top \mathbf{x}_S$$

$\mathbf{x}_S$ being the $(n-k)$-dimensional vector obtained from $\mathbf{x}$ by dropping all the components in S. Basically, the function f_S^α is obtained from f by inserting in the affinity matrix $\mathtt{A}$ the value of the parameter α in the main diagonal positions corresponding to the elements of $V \setminus S$.

Notice that this differs markedly, and indeed generalizes, the formulation proposed in [43] for obtaining a hierarchical clustering in that here, only a subset of elements in the main diagonal is allowed to take the α parameter, the other ones being set to zero. We note in fact that the original (non-regularized) dominant-set formulation (1.4) [35] as well as its regularized counterpart described in [43] can be considered as degenerate version of ours, corresponding to the cases $S = V$ and $S = \emptyset$, respectively. It is precisely this increased flexibility which allows us to use this idea for finding groups of "constrained" dominant-set clusters.

We now derive the Karush-Kuhn-Tucker (KKT) conditions for program (1.6), namely the first-order necessary conditions for local optimality (see, e.g., [46]). For a point $\mathbf{x} \in \Delta$ to be a KKT-point there should exist n nonnegative real constants

$\mu_1, \ldots, \mu_n$ and an additional real number λ such that

$$[(\mathtt{A} - \alpha \hat{I}_S)\mathbf{x}]_i - \lambda + \mu_i = 0$$

for all $i = 1 \ldots n$, and

$$\sum_{i=1}^{n} x_i \mu_i = 0 \ .$$

Since both the x_i's and the μ_i's are nonnegative, the latter condition is equivalent to saying that $i \in \sigma(\mathbf{x})$ implies $\mu_i = 0$, from which we obtain:

$$[(\mathtt{A} - \alpha \hat{I}_S)\mathbf{x}]_i \begin{cases} = & \lambda, \quad \text{if } i \in \sigma(\mathbf{x}) \\ \leq & \lambda, \quad \text{if } i \notin \sigma(\mathbf{x}) \end{cases}$$

for some constant λ. Noting that $\lambda = \mathbf{x}^\top \mathtt{A}\mathbf{x} - \alpha \mathbf{x}_S^\top \mathbf{x}_S$ and recalling the definition of $\hat{I}_S$, the KKT conditions can be explicitly rewritten as:

$$\begin{cases} (\mathtt{A}\mathbf{x})_i - \alpha x_i & = & \mathbf{x}^\top \mathtt{A}\mathbf{x} - \alpha \mathbf{x}_S^\top \mathbf{x}_S, & \text{if } i \in \sigma(\mathbf{x}) \text{ and } i \notin S \\ (\mathtt{A}\mathbf{x})_i & = & \mathbf{x}^\top \mathtt{A}\mathbf{x} - \alpha \mathbf{x}_S^\top \mathbf{x}_S, & \text{if } i \in \sigma(\mathbf{x}) \text{ and } i \in S \\ (\mathtt{A}\mathbf{x})_i & \leq & \mathbf{x}^\top \mathtt{A}\mathbf{x} - \alpha \mathbf{x}_S^\top \mathbf{x}_S, & \text{if } i \notin \sigma(\mathbf{x}) \end{cases} \tag{1.7}$$

We are now in a position to discuss the main results which motivate the algorithm presented in this section. Note that, in the sequel, given a subset of vertices $S \subseteq V$, the face of Δ corresponding to S is given by: $\Delta_S = \{\mathbf{x} \in \Delta : \sigma(\mathbf{x}) \subseteq S\}$.

Proposition 1. *Let $S \subseteq V$, with $S \neq \emptyset$. Define*

$$\gamma_S = \max_{\mathbf{x} \in \Delta_{V \setminus S}} \min_{i \in S} \frac{\mathbf{x}^\top \mathtt{A}\mathbf{x} - (\mathtt{A}\mathbf{x})_i}{\mathbf{x}^\top \mathbf{x}} \tag{1.8}$$

and let $\alpha > \gamma_S$. If $\mathbf{x}$ is a local maximizer of f_S^α in Δ, then $\sigma(\mathbf{x}) \cap S \neq \emptyset$.

Proof. Let $\mathbf{x}$ be a local maximizer of f_S^α in Δ, and suppose by contradiction that no element of $\sigma(\mathbf{x})$ belongs to S or, in other words, that $\mathbf{x} \in \Delta_{V \setminus S}$. By letting

$$i = \arg\min_{j \in S} \frac{\mathbf{x}^\top \mathtt{A}\mathbf{x} - (\mathtt{A}\mathbf{x})_j}{\mathbf{x}^\top \mathbf{x}}$$

and observing that $\sigma(\mathbf{x}) \subseteq V \setminus S$ implies $\mathbf{x}^\top \mathbf{x} = \mathbf{x}_S^\top \mathbf{x}_S$, we have:

$$\alpha > \gamma_S \geq \frac{\mathbf{x}^\top \mathtt{A}\mathbf{x} - (\mathtt{A}\mathbf{x})_i}{\mathbf{x}^\top \mathbf{x}} = \frac{\mathbf{x}^\top \mathtt{A}\mathbf{x} - (\mathtt{A}\mathbf{x})_i}{\mathbf{x}_S^\top \mathbf{x}_S} \ .$$

Hence, $(\mathtt{A}\mathbf{x})_i > \mathbf{x}^\top \mathtt{A}\mathbf{x} - \alpha \mathbf{x}_S^\top \mathbf{x}_S$ for $i \notin \sigma(\mathbf{x})$, but this violates the KKT conditions (1.7), thereby proving the proposition. □

The following proposition provides a useful and easy-to-compute upper bound for γ_S.

Proposition 2. *Let $S \subseteq V$, with $S \neq \emptyset$. Then,*

$$\gamma_S \leq \lambda_{\max}(\mathtt{A}_{V\setminus S}) \tag{1.9}$$

where $\lambda_{\max}(\mathtt{A}_{V\setminus S})$ is the largest eigenvalue of the principal submatrix of $\mathtt{A}$ indexed by the elements of $V \setminus S$.

Proof. Let $\mathbf{x}$ be a point in $\Delta_{V\setminus S}$ which attains the maximum γ_S as defined in (1.8). Using the Rayleigh-Ritz theorem [47] and the fact that $\sigma(\mathbf{x}) \subseteq V \setminus S$, we obtain:

$$\lambda_{\max}(\mathtt{A}_{V\setminus S}) \geq \frac{\mathbf{x}_S^\top \mathtt{A}_{V\setminus S} \mathbf{x}_S}{\mathbf{x}_S^\top \mathbf{x}_S} = \frac{\mathbf{x}^\top \mathtt{A}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \ .$$

Now, define $\gamma_S(\mathbf{x}) = \max\{(\mathtt{A}\mathbf{x})_i \ : \ i \in S\}$. Since $\mathtt{A}$ is nonnegative so is $\gamma_S(\mathbf{x})$, and recalling the definition of γ_S we get:

$$\frac{\mathbf{x}^\top \mathtt{A}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \geq \frac{\mathbf{x}^\top \mathtt{A}\mathbf{x} - \gamma_S(\mathbf{x})}{\mathbf{x}^\top \mathbf{x}} = \gamma_S$$

which concludes the proof. □

The two previous propositions provide us with a simple technique to determine dominant-set clusters containing user-selected vertices. Indeed, if S is the set of vertices selected by the user, by setting

$$\alpha > \lambda_{\max}(\mathtt{A}_{V\setminus S}) \tag{1.10}$$

we are guaranteed that all local solutions of (1.6) will have a support that necessarily contains elements of S. Note that this does not necessarily imply that the (support of the) solution found corresponds to a dominant-set cluster of the original affinity matrix $\mathtt{A}$, as adding the parameter $-\alpha$ on a portion of the main diagonal intrinsically changes the scale of the underlying problem. However, we have obtained extensive empirical evidence which supports a conjecture which turns out to be very useful for our interactive image segmentation application.

To illustrate the idea, let us consider the case where edge-weights are binary, which basically means that the input graph is unweighted. In this case, it is known that dominant sets correspond to maximal cliques [35]. Let $G = (V, E)$ be our unweighted graph and let S be a subset of its vertices. For the sake of simplicity, we distinguish three different situations of increasing generality.

Case 1. The set S is a singleton, say $S = \{u\}$. In this case, we know from Proposition 2 that all solutions $\mathbf{x}$ of f_α^S over Δ will have a support which contains u, that is $u \in \sigma(\mathbf{x})$. Indeed, we conjecture that there will be a unique local (and hence global) solution here whose support coincides with the *union* of all maximal cliques of G which contain vertex u.

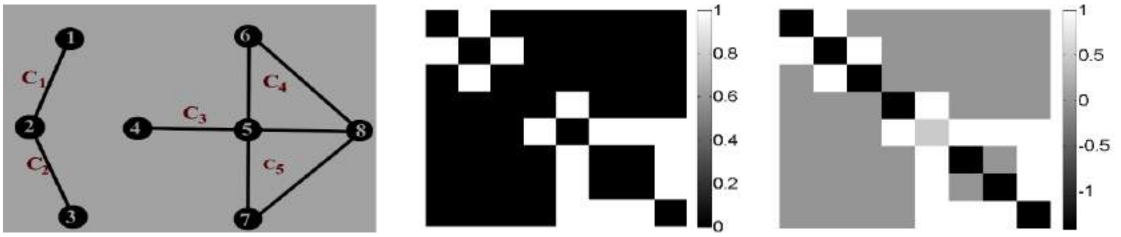


Figure 1.2: An example graph (left), corresponding affinity matrix (middle), and scaled affinity matrix built considering vertex 5 as a user constraint (right). Notation C_i refers to the i^{th} maximal clique.

Case 2. The set S is a clique, not necessarily maximal. In this case, Proposition 2 predicts that all solutions $\mathbf{x}$ of (1.6) will contain at least one vertex from S. Here, we claim that indeed the support of local solutions is the union of the maximal cliques that contain S.

Case 3. The set S is not a clique, but it can be decomposed as a collection of (possibly overlapping) maximal cliques $C_1, C_2, ..., C_k$ (maximal with respect to the subgraph induced by S). In this case, we claim that if $\mathbf{x}$ is a local solution, then its support can be obtained by taking the union of all maximal cliques of G containing one of the cliques C_i in S.

To make our discussion clearer, consider the graph shown in Fig. 1.2. In order to test whether our claims hold, we used as the set S different combinations of vertices, and enumerated all local solutions of (1.6) by multi-start replicator dynamics (see Section 1.3.1). Some results are shown below, where on the left-hand side we indicate the set S, while on the right hand-side we show the supports provided as output by the different runs of the algorithm.

1. $S = \{2\} \quad \Rightarrow \quad \sigma(\mathbf{x}) = \{1, 2, 3\}$
2. $S = \{5\} \quad \Rightarrow \quad \sigma(\mathbf{x}) = \{4, 5, 6, 7, 8\}$
3. $S = \{4, 5\} \quad \Rightarrow \quad \sigma(\mathbf{x}) = \{4, 5\}$
4. $S = \{5, 8\} \quad \Rightarrow \quad \sigma(\mathbf{x}) = \{5, 6, 7, 8\}$
5. $S = \{1, 4\} \quad \Rightarrow \quad \sigma(\mathbf{x}_1) = \{1, 2\}, \quad \sigma(\mathbf{x}_2) = \{4, 5\}$
6. $S = \{2, 5, 8\} \quad \Rightarrow \quad \sigma(\mathbf{x}_1) = \{1, 2, 3\}, \quad \sigma(\mathbf{x}_2) = \{5, 6, 7, 8\}$

The previous observations can be summarized in the following general statement which does comprise all three cases. Let $S = C_1 \cup C_2 \cup \ldots \cup C_k$ $(k \geq 1)$ be a subset of vertices of G, consisting of a collection of cliques C_i $(i = 1 \ldots k)$. Suppose that condition (1.10) holds, and let $\mathbf{x}$ be a local solution of (1.6). Then, $\sigma(\mathbf{x})$ consists of the union of all maximal cliques containing some clique C_i of S.

We conjecture that the previous claim carries over to edge-weighted graphs where the notion of a maximal clique is replaced by that of a dominant set. In section 1.3.3 and A.1, we report the results of an extensive experimentation we have conducted over random instance graphs and over standard DIMACS benchgraphs which provide support to our claim. This conjecture is going to play a key role in our wide range of vision applications.

1.3.1 Finding Constrained Dominant Sets Using Game Dynamics

Evolutionary game theory offers a whole class of simple dynamical systems to solve quadratic constrained optimization problems like ours. It envisages a scenario in which pairs of players are repeatedly drawn at random from a large population of individuals to play a symmetric two-player game. Game dynamics are designed in such a way as to drive strategies with lower payoff to extinction, following Darwin's principle of natural selection [38, 48].

Let $x_i(t)$ is the proportion of the population which plays strategy $i \in J$ (the set of strategies) at time t. The state of the population at any given instant is then given by $\mathbf{x}(t) = (x_1(t), ..., x_n(t))'$ where $'$ denotes transposition and n refers the size of available pure strategies, that is $|J|$.

Let $W = (w_{ij})$ be the $n \times n$ payoff matrix (biologically measured as Darwinian fitness or as profits in economic applications). The payoff for the i^{th}-strategist, assuming the opponent is playing the j^{th} strategy, is given by w_{ij}, the corresponding i^{th} row and the j^{th} column of W. If the population is in state $\mathbf{x}$, the expected payoff earned by an the i^{th}-strategist is:

$$\mathcal{P}_i(\mathbf{x}) = \sum_{j=1}^{n} w_{ij}x_j = (W\mathbf{x})_i$$

and the mean payoff over the whole population is

$$\mathcal{P}(\mathbf{x}) = \sum_{i=1}^{n} x_i\mathcal{P}_i(\mathbf{x}) = \mathbf{x}'W\mathbf{x}$$

The game, which is assumed to be played over and over, generation after generation, changes the state of the population over time until equilibrium is reached. A point $\mathbf{x}$ is said to be a stationary (or equilibrium) point of the dynamical system if $\dot{x} = 0$ where the dot implies derivative with respect to time.

Different formalization of this selection process have been proposed in evolutionary game theory. One of the best-known class of game dynamics is given by the so-called *replicator dynamics*, which prescribes that the average rate of increase $\dot{x}_i/x_i$ equals the difference between the average fitness of strategy i and the mean fitness over the entire population:

$$\dot{x} = x_i\left((W\mathbf{x})_i - \mathbf{x}'W\mathbf{x}\right) \tag{1.11}$$

A well-known discretization of the above dynamics is:

$$x_i^{(t+1)} = x_i^{(t)} \frac{(W\mathbf{x}^{(t)})_i}{(\mathbf{x}^{(t)})'W(\mathbf{x}^{(t)})} \tag{1.12}$$

Now, the celebrated Fundamental Theorem of Natural Selection [48] states that, if $W = W'$, then the average population payoff $\mathbf{x}'W\mathbf{x}$ is strictly increasing along any non-constant trajectory of both the continuous-time and discrete-time replicator dynamics. Thanks to this property, replicator dynamics naturally suggest themselves as a simple heuristics for finding (constrained) dominant sets [35].

In our case, problem (1.6), the payoff matrix W is given by

$$W = \mathtt{A} - \alpha \hat{I}_S$$

which yields:

$$x_i^{(t+1)} = \begin{cases} x_i^{(t)} \frac{(\mathtt{A}\mathbf{x}^{(t)})_i}{(\mathbf{x}^{(t)})'(\mathtt{A}-\alpha\hat{I}_S)(\mathbf{x}^{(t)})}, & \text{if } i \in S \\ \\ x_i^{(t)} \frac{(\mathtt{A}\mathbf{x}^{(t)})_i - \alpha x_i^{(t)}}{(\mathbf{x}^{(t)})'(\mathtt{A}-\alpha\hat{I}_S)(\mathbf{x}^{(t)})}, & \text{if } i \notin S \end{cases} \tag{1.13}$$

Provided that the matrix $\mathtt{A} - \alpha\hat{I}_S$ is scaled properly to avoid negative values, it is readily seen that the simplex Δ is invariant under these dynamics, which means that every trajectory starting in Δ will remain in Δ for all future times.

In addition to the replicator dynamics described above, we mention a faster alternative to solve linearly constrained quadratic optimization problems like ours, namely *Infection and Immunization Dynamics* (InImDyn) [49]. Each step of InImDyn has a linear time/space complexity as opposed to the quadratic per-step complexity of replicator dynamics, and is therefore to be preferred in the presence of large payoff matrices.

The dynamics, inspired by infection and immunization processes summarized in Algorithm (1), finds the optimal solution by iteratively refining an initial distribution $\mathbf{x} \in \Delta$. The process allows for invasion of an infective distribution $\mathbf{y} \in \Delta$ that satisfies the inequality $(\mathbf{y} - \mathbf{x})^\top \mathtt{A}\mathbf{x} > 0$, and combines linearly $\mathbf{x}$ and $\mathbf{y}$ (line 7 of Algorithm (1)), thereby engendering a new population $\mathbf{z}$ which is immune to $\mathbf{y}$ and guarantees a maximum increase in the expected payoff. A selective function, $\mathcal{S}(\mathbf{x})$, returns an infective strategy for distribution $\mathbf{x}$ if it exists, or $\mathbf{x}$ otherwise (line 2 of Algorithm (1)). Selecting a strategy $\mathbf{y}$ which is infective for the current population $\mathbf{x}$, the extent of the infection, $\delta_{\mathbf{y}}(\mathbf{x})$, is then computed in lines 3 to 6 of Algorithm (1).

By reiterating this process of infection and immunization the dynamics drives the population to a state that cannot be infected by any other strategy. If this is the case then $\mathbf{x}$ is an equilibrium or fixed point under the dynamics. The refinement loop of Algorithm (1) controls the number of iterations allowing them to continue until $\mathbf{x}$ is with in the range of the tolerance τ and we emperically set τ to 10^{-7}. The range $\epsilon(\mathbf{x})$ is computed as $\epsilon(\mathbf{x}) = \sum_{i \in J} \min\left\{x_i, (\mathtt{A}\mathbf{x})_i - \mathbf{x}^\top \mathtt{A}\mathbf{x}\right\}^2$.

Algorithm 1 FindEquilibrium($\mathtt{A}$,$\mathbf{x}$,τ)

```
Input: $n \times n$ payoff matrix $\mathtt{A}$, initial distribution $\mathbf{x} \in \Delta$ and tolerance $\tau$.
Output: Fixed point $\mathbf{x}$
while $\epsilon(\mathbf{x}) > \tau$ do
y ← $\mathcal{S}(\mathbf{x})$
$\delta \leftarrow 1$
    if $(\mathbf{y}-\mathbf{x})^\top \mathtt{A}(\mathbf{y}-\mathbf{x}) < 0$ then
        $\delta \leftarrow \min\left\{\frac{(\mathbf{x}-\mathbf{y})^\top \mathtt{A}\mathbf{x}}{(\mathbf{y}-\mathbf{x})^\top \mathtt{A}(\mathbf{y}-\mathbf{x})}, 1\right\}$
    end if
    $\mathbf{x} \leftarrow \delta(\mathbf{y}-\mathbf{x}) + \mathbf{x}$
end while
return x
```

1.3.2 Fast Approach for Solving Constrained Dominant Set Clustering

Though Infection and Immunization Dynamics (InfImDyn) solves our constrained quadratic optimization program in linear time, it needs the whole affinity matrix to extract the compact set which, more often than not, exists in local range of the whole graph. Efficient out-of-sample [44], extension of dominant sets, is the other approach which is used to reduce the computational cost by sampling the nodes of the graph using some given sampling rate that affects the framework efficacy. Liu *et al.* [50] proposed an iterative clustering algorithm, which operates in two steps: Shrink and Expansion. These steps help reduce the runtime of replicator dynamics on the whole data, which might be slow. The approach has many limitations such as its preference of sparse graph with many small clusters and the results are sensitive to some additional parameters. Another approach which tries to reduce the computational complexity of the standard quadratic program (StQP [51]) is proposed by [52].

All the above formulations, with their limitations, try to minimize the computational complexity of StQP using the standard game dynamics, whose complexity is $\mathcal{O}(n^2)$ for each iteration.

In this thesis we propose a fast approach (listed in Algorithm (2)), based on InfImDyn approach which solves StQP in $\mathcal{O}(n)$, for the recently proposed formulation, $\mathbf{x}^\top(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}$, which of-course generalizes the StQP.

InfImDyn is a game dynamics inspired by Evolutionary game theory. The dynamics extracts a dominant set using a two-steps approach (infection and immunization), that iteratively increases the compactness measure of the objective function by driving the (probability) distribution with lower payoff to extinction, by determining an ineffective distribution $\mathbf{y} \in \Delta$, that satisfies the inequality $(\mathbf{y} - \mathbf{x})^\top \mathtt{A}\mathbf{x} > 0$, the dynamics combines linearly the two distributions ($\mathbf{x}$ and $\mathbf{y}$), thereby engendering a new population $\mathbf{z}$ which is immune to $\mathbf{y}$ and guarantees a maximum increase in the expected payoff. In our setting, given a set of instances and their affinity, we first assign all of them an equal probability (a distribution at the centre of the simplex, a.k.a. barycenter). The dynamics then drives the initial distribution with lower affinity to extinction; those which have higher affinity start getting higher, while the other get lower values. A selective function, $\mathcal{S}(\mathbf{x})$, is then run to check if there is any infective distribution; a distribution which contains instances with a better association score. By iterating this process of infection and immunization the dynamics is said to reach the equilibrium, when the population is driven to a state that cannot be infected by any other distribution, that is there is no distribution, whose support contains a set of instances with a better association score. The selective function, however, needs whole affinity matrix, which makes the InfImDyn inefficient for large graphs. We propose an algorithm, that reduces the search space using the Karush-Kuhn-Tucker (KKT) condition of the constrained quadratic optimization, effectively enforcing the user constraints. In the constrained optimization framework [53], the algorithm computes the eigenvalue of the submatrix for every extraction of the compact sets, which contains the user constraint set. Computing eigenvalues for large graphs is computationally intensive, which makes the whole algorithm inefficient.

In our approach, instead of running the dynamics over the whole graph, we localize it on the sub-matrix, selected using the dominant distribution, that is much smaller than the original one. To alleviate the issue with the eigenvalues, we utilize the properties of eigenvalues; a good approximation for the parameter α is to use the maximum degree of the graph, which of-course is larger than the eigenvalue of corresponding matrix. The computational complexity, apart from eigenvalue computation, is reduced to $\mathcal{O}(r)$ where r, which is much smaller than the original affinity, is the size of the sub-matrix where the dynamics is run.

Let us summarize the KKT conditions for quadratic program reported in eq. (1.6). By adding Lagrangian multipliers, n non-negative constants $\mu_1, ..., \mu_n$ and a real number λ, its Lagrangian function is defined as follows:

$$\mathcal{L}(x, \mu, \lambda) = f_{\mathcal{Q}}^{\alpha}(\mathbf{x}) + \lambda \left(1 - \sum_{i+1}^{n} x_i\right) + \sum_{i+1}^{n} \mu_i x_i.$$

For a distribution $x \in \Delta$ to be a KKT-point, in order to satisfy the first-order necessary conditions for local optimality [46], it should satisfy the following two conditions:

$$2 * [(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}]_i - \lambda + \mu_i = 0,$$

for all $i = 1 \ldots n$, and

$$\sum_{i=1}^{n} x_i \mu_i = 0 \ .$$

Since both the x_i and the μ_i values are nonnegative, the latter condition is equivalent to saying that $i \in \sigma(\mathbf{x})$ which implies that $\mu_i = 0$, from which we obtain:

$$[(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}]_i \begin{cases} = & \lambda/2, \quad \text{if } i \in \sigma(\mathbf{x}) \\ \leq & \lambda/2, \quad \text{if } i \notin \sigma(\mathbf{x}) \end{cases} \tag{1.14}$$

We then need to define a *Dominant distribution*

Definition 1. *A distribution* $\mathbf{y} \in \Delta$ *is said to be a* ***dominant distribution*** *for* $\mathbf{x} \in \Delta$ *if*

$$\left\{ \sum_{i,j=1}^{n} x_i y_j a_{ij} - \alpha x_i y_j \right\} > \left\{ \sum_{i,j=1}^{n} x_i x_j a_{ij} - \alpha x_i x_j \right\} \tag{1.15}$$

Let the "support" be $\sigma(\mathbf{x}) = \{i \in V \ : \ x_i > 0\}$ and $\mathbf{e}_i$ the i^{th} unit vector (a zero vector whose i^{th} element is one).

Proposition 3. *Given an affinity* A *and a distribution* $\mathbf{x} \in \Delta$*, if* $(\mathtt{A}\mathbf{x})_i > \mathbf{x}^\top \mathtt{A}\mathbf{x} - \alpha \mathbf{x}_{\mathcal{Q}}^\top \mathbf{x}_{\mathcal{Q}}$*, for* $i \notin \sigma(\mathbf{x})$*,*

1. $\mathbf{x}$ *is not the maximizer of the parametrized quadratic program of (1.6)*

2. $\mathbf{e}_i$ *is a* ***dominant distribution*** *for* $\mathbf{x}$

Proof. To show the first condition holds: Let's assume $\mathbf{x}$ is a KKT point

$$\mathbf{x}^\top (\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x} = \sum_{i=1}^{n} x_i [(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}]_i$$

Since x is a KKT point

$$\mathbf{x}^\top (\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x} = \sum_{i=1}^{n} x_i * \lambda/2 = \lambda/2$$

From the second condition, we have:

$$[(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}]_i \leq \lambda/2 = \mathbf{x}^\top (\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}$$

Since $i \notin \sigma(\mathbf{x})$

$$(\mathtt{A}\mathbf{x})_i \leq \mathbf{x}^\top (\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}$$

Which concludes the proof showing that the inequality does not hold.

For the second condition, if $\mathbf{e}_i$ is a **dominant distribution** for $\mathbf{x}$, it should satisfy the inequality

$$\left\{\mathbf{e}_i^\top(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}\right\} > \left\{\mathbf{x}^\top(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}\right\}$$

Since $i \notin \sigma(\mathbf{x})$

$$(\mathtt{A}x)_i > \left\{\mathbf{x}^\top(\mathtt{A} - \alpha I_{\mathcal{Q}})\mathbf{x}\right\}$$

Which concludes the proof

□

The proposition provides us with an easy-to-compute dominant distribution, and the detail is summarized in Algorithm (2)

Let a function, $\mathcal{S}(\mathtt{A}, x)$, returns a dominant distribution for distribution, x, $\emptyset$ otherwise and $\mathcal{G}(\mathtt{A}, \mathcal{Q}, x)$ returns the local maximizer of program (1.6).

Algorithm 2 Fast CDSC

```
**Input**: Affinity $\mathtt{B}$, Constraint set $\mathcal{Q}$
Initialize $\mathbf{x}$ to the barycenter of $\Delta_{\mathcal{Q}}$
$\mathbf{x}_d \leftarrow \mathbf{x}$, initialize *dominant distribution*
**Output**: Fixed point $\mathbf{x}$
while true do
$\mathbf{x}_d \leftarrow \mathcal{S}(\mathtt{B}, \mathbf{x})$, Find dominant distribution for $x$
    if $\mathbf{x}_d = \emptyset$ then break
    end if
$\mathcal{H} \leftarrow \sigma(\mathbf{x}_d) \cup \mathcal{Q}$, subgraph nodes
$\mathtt{A} \leftarrow \mathtt{B}_{\mathcal{H}}$
$\mathbf{x}_l \leftarrow \mathcal{G}(\mathtt{A}, \mathcal{Q}, x)$
$\mathbf{x} \leftarrow \mathbf{x}$*0
$\mathbf{x}(\mathcal{H}) \leftarrow \mathbf{x}_l$
end while
return $\mathbf{x}$
```

The selected dominant distribution always increases the value of the objective function. Moreover, the objective function is bounded which guaranties the convergence of the algorithm.

1.3.3 Experiments Using CDS

In this section, we report on some empirical evidence which provides support to a conjecture which plays a key role in our computer vision application. For the reader's convenience, we recapitulate our claim below.

Let $G = (V, E)$ be an unweighted graph with adjacency matrix $\mathtt{A}$, and let S be a subset of its vertices. Let $S = C_1 \cup C_2 \cup \ldots \cup C_k$ $(k \geq 1)$ be a subset of vertices of G consisting of a collection of maximal cliques C_i, $i = 1 \ldots k$ (maximal w.r.t. the subgraph induced by S). Let $\mathbf{x}$ be a local solution of the following quadratic program:

$$\begin{array}{ll} \text{maximize} & f_S^\alpha(\mathbf{x}) = \mathbf{x}^\top(\mathtt{A} - \alpha \hat{I}_S)\mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array} \tag{1.16}$$

and suppose that:

$$\alpha > \lambda_{\max}(\mathtt{A}_{V \setminus S}) \ . \tag{1.17}$$

Then, the support of $\mathbf{x}$, defined as

$$\sigma(\mathbf{x}) = \{i \in V \ : \ x_i > 0\}$$

consists of the union of all maximal cliques containing some clique C_i of S.

We also conjecture that the previous claim carries over to edge-weighted graphs where the notion of a maximal clique is replaced by that of a dominant set. Here we include only experiments on DIMACS benchmark graphs. Experiments on random graphs are included in the appendix part of the thesis.

1.3.3.1 Experiments on DIMACS benchmark graphs

We tested our claims over a selection of DIMACS graphs, a standard benchmark dataset used to assess the effectiveness of clique finding algorithms [54]. In particular, we used graphs belonging to the *brock* family, which are constructed so as to camouflage a clique of size larger than expected, and are known to be among the most difficult ones contained in the dataset [55].

We tested our algorithm on the larger *brock* graphs contained in the DIMACS dataset (from 200 to 800 vertices). To evaluate the results we here use the information each graph instance in the dataset contains about the vertices of its maximum clique. Table 1.1 shows the results obtained. The table includes the name of the graph (Name), the number of vertices (#), the graph density (ρ), the size of the maximum clique (ω). Lφ refers to the size, averaged over 20 runs, of the elements of the extracted nodes, say $\mathcal{N}$, which contain φ % of the clique elements that are selected randomly. Fϱ represents the fraction of the clique elements that are members of the extracted set of nodes for a given ϱ % of the clique elements that are selected randomly, as computed as follows:

$$\mathrm{F}\varrho = \frac{|\mathcal{C} \cap \mathcal{N}|}{|\mathcal{C}|}$$

.

We also evaluated the performance of the framework on weighted version of the DIMACS graphs. We generated them by slightly (randomly) perturbing the entries of the adjacency matrices of original (unweighted) graphs used in the previous set of

Name	#	ρ	ω	L20	F20	L50	F50	L75	F75	L90	F90
brock200_1	200	0.75	21	66.5±5.8	1	22.9±1.4	1	21±0	1	21±0	1
brock200_2	200	0.50	12	47.7±6.7	1	12.1±0.3	1	12±0	1	12±0	1
brock200_3	200	0.61	15	40.5±5.3	1	15.5±0.6	1	15±0	1	15±0	1
brock200_4	200	0.66	17	61.8±5.1	1	18.4±1.1	1	17±0	1	17±0	1
brock400_1	400	0.75	27	95.8±6.3	1	28.2±1.3	1	27±0	1	27±0	1
brock400_2	400	0.75	29	95.7±6.4	1	30.1±0.9	1	29±0	1	29±0	1
brock400_3	400	0.75	31	75.5±7.4	1	31.5±0.5	1	31±0	1	31±0	1
brock400_4	400	0.75	33	77.7±5.0	1	33.3±0.7	1	33±0	1	33±0	1
brock800_1	800	0.65	23	129.5±7.2	1	23.7±0.8	1	23±0	1	23±0	1
brock800_2	800	0.65	24	129.9±9.6	1	24.5±0.4	1	24±0	1	24±0	1
brock800_3	800	0.65	25	84.0±7.9	1	25.4±0.4	1	25±0	1	25±0	1
brock800_4	800	0.65	26	87.0±6.4	1	26.2±0.5	1	26±0	1	26±0	1

Table 1.1: Performance of the algorithm on *brock* DIMACS graphs.

experiments, in such a way that their maximal cliques corresponded to the dominant sets in their weighted versions. (This can easily be done using the convergence properties of replicator dynamics [2].) The results for the weighted version is shown in Table 1.2.

As can be observed from Tables 1.1 and 1.2, in both unweighted and weighted versions, as the size of the randomly selected clique elements increases, the extracted version is identical to the optimum clique size, and since the fraction Fϱ is always 1, we can infer that not only the clique size but also the right maximum clique elements are extracted. As the percentage φ increases, the variance drops drastically to zero and the algorithms converges to the right maximal clique which can be verified by Lφ, ω and the fraction Fϱ. In general, given a maximal clique as a constraint set, the algorithms converges to the same maximal clique, and given a subset of a maximal clique, it converges to a super-set of the maximal clique which contains the given set.

1.4 Simultaneous Clustering and Outlier Detection

In the literature, clustering and outlier detection are often treated as separate problems. However, it is natural to consider them simultaneously. The problem of outlier detection is deeply studied in both communities of data mining and statistics [56, 57], with different perspectives.

A classical statistical approach for finding outliers in multivariate data is Minimum Covariance Determinant (MCD). The main objective of this approach is

Name	#	ρ	ω	L20	F20	L50	F50	L75	F75	L90	F90
brock200_1	200	0.75	21	81.2±7.6	1	30.0±4	1	21.25±04	1	21±0	1
brock200_2	200	0.50	12	50.3±4.9	1	13.7±2	1	12±0	1	12±0	1
brock200_3	200	0.61	15	48.5±9.7	1	17.8±2	1	15±0	1	15±0	1
brock200_4	200	0.66	17	70.5±7.0	1	24.5±3	1	17±0	1	17±0	1
brock400_1	400	0.75	27	126.2±18	1	33.2±3	1	27±0	1	27±0	1
brock400_2	400	0.75	29	129.2±21	1	32.8±2	1	29±0	1	29±0	1
brock400_3	400	0.75	31	113.2±18	1	33.2±2	1	31±0	1	31±0	1
brock400_4	400	0.75	33	109.5±14	1	34.0±1	1	33±0	1	33±0	1
brock800_1	800	0.65	23	188.1±43	1	27.8±2	1	23±0	1	23±0	1
brock800_2	800	0.65	24	184.9±51	1	26.2±1	1	24±0	1	24±0	1
brock800_3	800	0.65	25	148.5±36	1	27.3±1	1	25±0	1	25±0	1
brock800_4	800	0.65	26	150.95±34	1	26.7±1	1	26±0	1	26±0	1

Table 1.2: Performance of the algorithm on weighted versions of the *brock* DIMACS graphs.

to identify a subset of points which minimizes the determinant of the variance-covariance matrix over all subsets of size $n-l$, where n is the number of multivariate data points and l is the number of outliers. The resulting variance-covariance matrix can be integrated into the Mahalanobis distance and used as part of a chi-square test to identify multivariate outliers [58]. However, the high computational complexity makes it impractical for high-dimensional datasets.

A distance-based outlier detection is introduced by authors in [59], which does not depend on any distributional assumptions and can easily be generalized to multidimensional datasets. Intuitively, in this approach data points which are far away from their nearest neighbors are considered as an outlier. However, outliers detected by these approaches are global outliers, that is, the outlierness is with respect to the whole dataset.

In [60], the authors introduced a new concept that is local outlier factor (LOF), which shows how isolated an object is with respect to its surrounding neighborhood. In this method, they claim that in some situations local outliers are more important than global outliers which are not easily detected by distance-based techniques. The concept of local outliers has subsequently been extended in several directions [56, 61]. Authors in [62] studied a similar problem in the context of facility location and clustering. Given a set of points in a metric space and parameters k and m, the goal is to remove m outliers, such that the cost of the optimal $k-median$ clustering of the remaining points is minimized. In [63] authors have proposed k-means-- which generalizes $k-means$ with the aim of simultaneously clustering data and discovering outliers. However, the algorithm inherits the weaknesses of the classical $k-means$ algorithm: requiring the prior knowledge of cluster numbers k; and, sensitivity to initialization of the centroids, which leads to unwanted solutions.

More recently, authors in [64] modelled clustering and outlier detection as an integer programming optimization task and then proposed a Lagrangian relaxation to design a scalable subgradient-based algorithm. The resulting algorithm discovers the number of clusters from the data however it requires the cost of creating a new cluster and the number of outliers in advance.

In this section of the thesis, we propose a modified dominant set clustering problem for simultaneous clustering and outlier detection from data (SCOD). Unlike most of the above approaches our method requires no prior knowledge on both the number of clusters and outliers, which makes our approach more convenient for real applications.

A naive approach to apply dominant set clustering is to set a threshold, say cluster size, and label clusters with smaller cluster size than the threshold as outliers. However, in the presence of many cluttered noises (outliers) with a uniformly distributed similarity (with very small internal coherency), the dominant set framework extracts the set as one big cluster. That is, cluster size threshold approaches are handicapped in dealing with such cases. Thus what is required is a more robust technique that can gracefully handle outlier clusters of different size and cohesivenesss.

Dominant set framework naturally provide a principled measure of a cluster's cohesiveness as well as a measure of vertex participation to each group (cluster). On the virtue of this nice feature of the framework, we propose a technique which simultaneously discover clusters and outlier in a computationally efficient manner.

The main contributions of this subsection are:

- we propose a method which is able to identify number of outliers automatically from the data.
- it requires no prior knowledge of the number of clusters since the approach discovers the number of clusters from the data.
- the effectiveness of the SCOD is tested on both synthetic and real datasets.

In the next subsection we detail the approach on enumerating dominant sets while obliterating outliers, while the last part of the section shows experimental results.

1.4.1 Enumerate Dominant Sets Obliterating Outliers

In the dominant set framework, a hard partition of the input data is achieved using a 'peel-off' strategy described as follows.

Initializing the dynamics defined in (1.12) to a point near the barycenter of the simplex, say it converges to a point $\mathbf{x}^*$, which is a strict local solution of (1.4). Let us determine the dominant set $\mathcal{DS}$ as the support of $\mathbf{x}^*$, $\mathcal{DS} = \sigma(\mathbf{x}^*)$. Then, all the vertices corresponding to the extracted dominant set are removed from the similarity graph. This process is repeated on the remaining graph until all data have been covered, but in applications involving large and noisy data sets this makes little sense. In these cases, a better strategy used in [35] is to stop the algorithm using a predefined threshold based on the size of the given data and assign the unprocessed ones to the "nearest" extracted dominant set according to some distance criterion.

This approach has proven to be effective in dealing with situations involving the presence of cluttered backgrounds [14]. However, it lacks an intuitive way to terminate. In fact, either a manual decision on the number of clusters to be extracted stops the 'peel-off' process or all points will be covered in one of the above two ways. It is this limitation which makes the dominant set framework not able to deal with the problem of automated simultaneous clustering and outlier detection.

In this work, we took into account two features which make the dominant set framework able to deal with simultaneous clustering and outlier detection problem, in which the number of clusters arises intuitively while outliers are automatically obliterated: the first one deals with cluster cohesiveness and the second one deals with clusters of different size.

A nice feature of the dominant set framework is that it naturally provides a principled measure of a cluster's cohesiveness as well as a measure of a vertex participation to each group. The degree of membership to a cluster is expressed by the components of the characteristic vector $\mathbf{x}^*$ of the strict local solution of (1.4): if a component has a small value, then the corresponding node has a small contribution to the cluster, whereas if it has a large value, the node is strongly associated with the cluster. Components corresponding to nodes not participating in the cluster are zero. A good cluster is one where elements that are strongly associated with it also have large values connecting one another in the similarity matrix.

The cohesiveness $\mathcal{C}$ of a cluster is measured by the value of equation (1.4) at its corresponding characteristic vector, $\mathcal{C} = f(\mathbf{x}_c)$:

$$\mathcal{C} = f(\mathbf{x}_c) = \mathbf{x}_c^\top \mathtt{A}\mathbf{x}_c$$

A good cluster has high $\mathcal{C}$ value. The average global cohesiveness $\mathcal{GC}$ of a given similarity matrix can be computed by fixing the vector $\mathbf{x}$ to the barycenter of the simplex, specifically $x_i = 1/N$ where N is the size of the data and $i = 1 \dots N$.

If the payoff matrix $\mathtt{A}$ is symmetric, then $f(\mathbf{x}) = \mathbf{x}^\top \mathtt{A}\mathbf{x}$ is a strictly increasing function along any non-constant trajectory of any payoff-monotonic dynamics of which replicator dynamics are a special instance. This property together with cohesiveness measure allowed us modify the dominant set framework for SCOD.

Initializing the dynamics to the barycenter, say $\mathbf{x}_t$ at a time $t = 1$, a step at each iteration implies an increase in $f(\mathbf{x}_{t+i})$ for any $i > 1$, which again entails that at convergence at time $t = c$, the cohesiveness of the cluster, extracted as the support of $\mathbf{x}_c$, is greater than the average global cohesiveness ($\mathcal{GC}$), i.e

$$\mathcal{GC} = \mathbf{x}_1^\top \mathtt{A}\mathbf{x}_1 < \mathbf{x}_c^\top \mathtt{A}\mathbf{x}_c$$

.

In the dominant set framework, there are situations where false positive clusters arise.

First, large number of very loosely coupled objects with similar affinities may arise as a big cluster. This can be handled using the cohesiveness measure as there will not be any big step of the point that initializes the dynamics.

Secondly, a small compact set of outliers form a cluster whose cohesiveness is greater than the average global cohesiveness of the original similarity. In our automated framework, to address these issues, a new affinity ($\mathcal{S}$) is built from the original pairwise similarity ($\mathtt{A}$) based on M-estimation from robust statistics.

To every candidate i a weight which intuitively measures its average relationship with the local neighbors is assigned:

$$\mathcal{S}(i,j) = w(i)w(j)\mathtt{A}(i,j) \tag{1.18}$$

where $w(i) = \frac{1}{|\mathcal{N}_i|} \sum\limits_{j \in \mathcal{N}_i} \mathtt{A}(i,j)$ and $\mathcal{N}_i$ is the set of top $\mathcal{N}$ similar items, based on the pairwise similarity ($\mathtt{A}$), of object i. The framework is not sensitive to the setting of the parameter ($\mathcal{N}$). In all the experiments we fixed it as 10% of the size of the data. One may also choose it based on the average distances among the objects. A similar study, though with a different intention, has been done in [65] and illustrated that this approach makes the system less sensitive to the parameter sigma to built the similarity.

The newly built affinity ($\mathcal{S}$) can be used in different ways: first, we can use it to recheck if the extracted sets are strict local solution of (1.4) setting ($\mathtt{A}$) = ($\mathcal{S}$). Another simpler and efficient way is using it for the cohesiveness measure i.e, an extracted set, to be a cluster, should satisfy the cohesiveness criteria in both affinities $\mathtt{A}$ and $\mathcal{S}$.

Figure 1.3 illustrates the second case. The red points in the middle are a compact outlier sets which forms a dominant set whose cohesiveness ($\mathcal{C} = 0.952$) is greater than the average global cohesiveness ($\mathcal{GC} = 0.580$). However, its cohesiveness in the newly built affinity ($\mathcal{CL} = 0.447$) is less than that of the average global cohesiveness. Observe that the two true positive cluster sets (green and blue) have a cohesiveness measures (in both affinities $\mathcal{C}$ and $\mathcal{CL}$) which are greater than the average global cohesiveness. Algorithm 3 summarizes the detail.

Algorithm 3 Cluster Obliterating Outliers

INPUT: Affinity A

```
$\mathit{Outliers} \leftarrow \emptyset$
$\mathit{Clusters} \leftarrow \emptyset$
$\mathcal{S} \leftarrow$ Build new affinity from $\mathtt{A}$ using (1.18)
Initialize $\mathbf{x}$ to the barycenter and $i$ and $j$ to 1
$\mathcal{GC} \leftarrow \mathbf{x}^\top \mathtt{A}\mathbf{x}$
while size of $\mathtt{A} \neq 0$ do $\mathbf{x}_c \leftarrow$ Find local solution of (1.4)
    if $\mathbf{x}_c^\top \mathcal{S}\mathbf{x}_c < \mathcal{GC}$ or $\mathbf{x}_c^\top \mathtt{A}\mathbf{x}_c < \mathcal{GC}$ then
        $\mathcal{O}_j \leftarrow \sigma(x_c)$, find the $j^{th}$ outlier set
        $j \leftarrow j + 1$
    else
        $\mathcal{DS}_i = \sigma(\mathbf{x}_c)$, find the $i^{th}$ dominant set
        $i \leftarrow i + 1$
    end if
Remove $\sigma(x_c)$ from the affinity matrices $\mathcal{S}$ and $\mathtt{A}$
end while
$\mathit{Clusters} = \bigcup_{k=1}^{i} \mathcal{DS}_k$
$\mathit{Outliers} = \bigcup_{k=1}^{j} \mathcal{O}_k$
```

OUTPUT: $\{\mathit{Clusters}, \mathit{Outliers}\}$

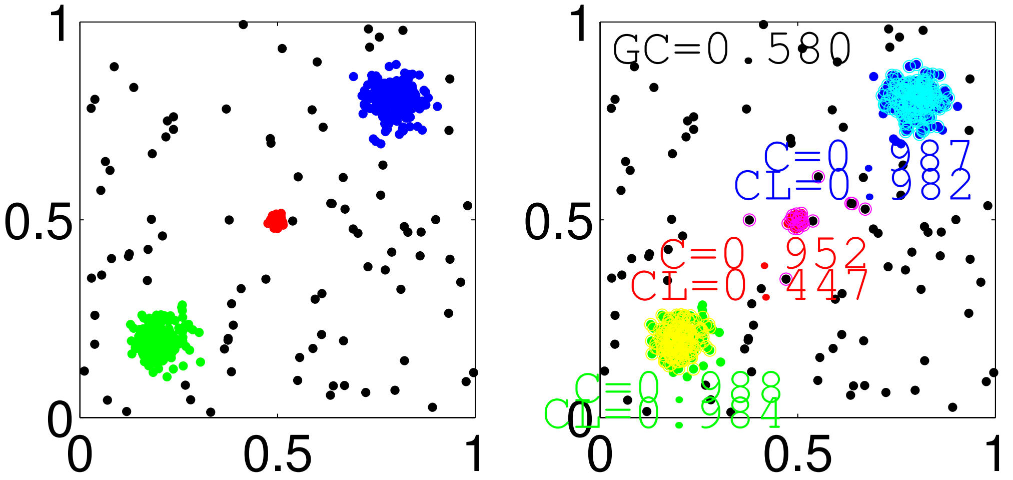


Figure 1.3: Examplar plots: **Left:** Original data points with different colors which show possible clusters. **Right:** Extracted clusters and their cohesiveness measures, $\mathcal{C}$ with affinity ($\mathtt{A}$) and ($\mathcal{CL}$) with the learned affinity ($\mathcal{S}$)

1.4.2 Experiments

In this section we evaluate the proposed approach on both synthetic and real datasets. First we evaluate our method on a synthetic datasets and present quantitative analysis. Then, we present experimental results on real datasets KDD-cup and SHUTTLE.

A pairwise distance $\mathcal{D}$ among individuals is transformed to similarity (edge-weight) using a standard Gaussian kernel

$$\mathtt{A}^{\sigma}_{ij} = \nVdash_{i \neq j} exp(-\mathcal{D}/2\sigma^2)$$

where σ is choosen as the median distance among the items, and $\nVdash_P = 1$ if P is true, 0 otherwise. We compare our Dominant set clustering based approach result with k-means-- [63].

1.4.2.1 Synthetic Data

The synthetic datasets are used to see the performance of our approach in a controlled environment and evaluate between different methods. Similar to [63], we generated synthetic data as follows: K cluster center points are sampled randomly from the space $[0, 1]^d$ and then m cluster member points are generated for each k clusters by sampling each coordinate from the normal distribution $\mathcal{N}(0, \sigma)$. Finally, l outliers are sampled uniformly at random from the space $[0, 1]^d$, where d is the dimensionality of the data.

To assess the performance of our algorithm we use the following metrics:

- The Jaccard coefficient J between the outliers found by our algorithm and the ground truth outliers. It measures how accurately a method selects the ground truth outliers. Computed as :

$$J(O, O^*) = \frac{|O \cap O^*|}{|O \cup O^*|}$$

  where O is the set of outliers returned by the algorithm while O^* are the ground truth outliers. The optimal value is 1.

- V-Measure [66] indicates the quality of the overall clustering solution. The outliers are considered as an additional class for this measure. Similar to the first measure, also in this case the optimal value is 1.

- The *purity* of the results is computed as the fraction of the majority class of each cluster with respect to the size of the cluster. Again, the optimal value is 1.

We evaluate the performance of the algorithm by varying different parameters of the data-generation process. Our objective is to create increasingly difficult settings

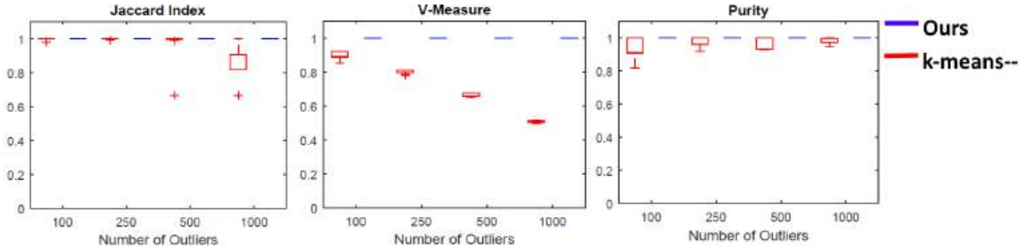


Figure 1.4: Results of the algorithm on synthetic datasets with respect to increasing number of outliers (l). While fixing parameters as k = 10, m = 100, d = 32, and σ = 0.2

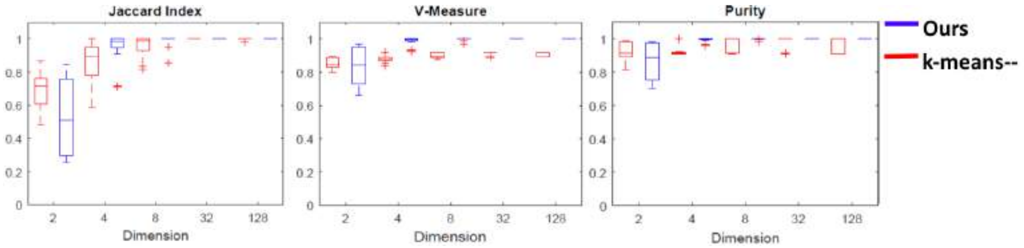


Figure 1.5: Results of the algorithm on synthetic datasets with respect to increasing dimension (d). While fixing parameters as k = 10, m = 100, l = 100, and σ = 0.1

so that the outliers eventually become indistinguishable from the points that belong to clusters. The result of our experiments are shown in Figures 1.4, 1.5 and 1.6 in which we vary the parameters l, d and σ respectively. For each cases, the rest of the parameters will be kept fixed. In each Figure we show the three measures described above Jaccard Index, V-Measure and Purity. Each box-plots indicate results after running each experiment 30 times. To be fair on the comparison, in each case of the experiment we run k-means-- 10 times (with different initialization) and report the best result, since their algorithm depends on the initialization of the centroids.

As we can see from the Figures, the performance of our algorithm is extremely good. In Figure 1.4, were we vary the number of outliers, our approach scored almost 1 in all measurements. This is mainly because, we introduced a robust criteria, that takes in to account the cohesiveness of each extracted clusters, which enables our approach to obliterate outliers in an efficient way. While the results of k-means-- decreases as the number of outliers increases. In Figure 1.5, the case were we vary the dimension, our approach scores relatively low in most of the measurements when the dimension is set to 2. But we gate excellent result as the dimension increases. In the last case, in Figure 1.6, we can see that our method is invariant to the value of standard deviation and it gates almost close to 1 in most of the measurements.

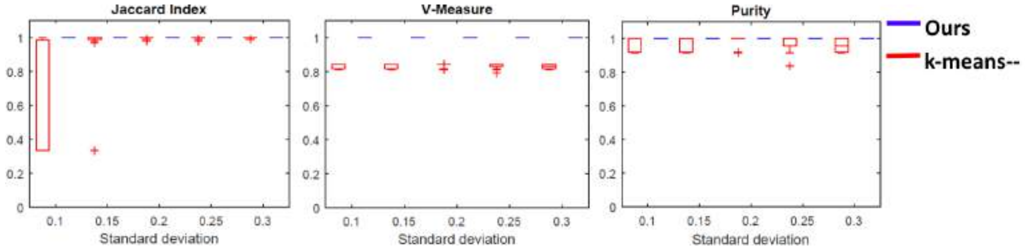


Figure 1.6: Results of the algorithm on synthetic datasets with respect to the standard deviation used to generate the data (σ). While fixing parameters as $k = 10$, $m = 100$, $l = 100$, and $d = 32$

1.4.2.2 Real Datasets

In this section we will discuss the performance of our approach on real datasets.

SHUTTLE: We first consider the "SHUTTLE" dataset which is publicly available on UCI Machine Learning repository [67]. This dataset contains 7 class labels while the main three classes account 99.6% of the dataset, each with 78.4%, 15.5%, and 5.6% of frequency. We took these three classes as non-outliers while the rest (0.4%) are considered as outliers. The dataset contains one categorical attribute, which is taken as class label, and 9 numerical attribute. We use the training part of the dataset, which consists of 43500 instances.

The results of our algorithm on this dataset is shown on Table 1.3 . The *precision* is computed with respect to the outliers found by the algorithm to the ground truth outliers. Since the number of outliers l and cluster k, required by k-mean--, is typically not known exactly we explore how its misspecification affects their results.

To investigate the influence of number of cluster (k) on k-means--, we run the experiments varying values of k while fixing number of outliers to $l = 0.4\%$ (the correct value). As it can be seen from Table 1.3 miss-specification of number of clusters has a negative effect on k-means--. The approach performs worst in all measurements as the number of clusters decreases. Our approach has the best result in all measurements. We can observed how providing k-means-- with different numbers of k results in worst performance which highlights the advantage of our method which is capable of automatically selecting the number of clusters and outliers from the data.

We made further investigation on the sensitivity of k-means-- on the number of outliers (l) by varying the values from 0.2% to 0.8%, while fixing $k = 20$. As the results on table 1.4 shows, as the number of outliers increase the precision of k-means-- decreases, means their algorithm suffers as more outliers are asked to be retrieved the more difficult it will become to separate them from the rest of the data. As we can see from Table 1.4 our approach has stable and prevailing results over k-means-- in all experiments. Our method is prone to such variations in the

Method	K	l	precision	Purity	V-measure
	10	0.4%	0.155	0.945	0.39
k-means--	15	0.4%	0.160	0.957	0.35
	20	0.4%	0.172	0.974	0.33
Ours	n.a.	n.a.	**0.29**	**0.977**	**0.41**

Table 1.3: Results on SHUTTLE dataset with fixed l and varying K

parameters, from the fact that it is able to automatically identify both the number of cluster and outliers from the data.

Method	k	l	Precision	Purity	V-measure
	20	0.2%	0.207	0.945	0.310
k-means--	20	0.4%	0.172	0.957	0.305
	20	0.8%	0.137	0.974	0.292
Ours	n.a.	n.a.	**0.29**	**0.977**	**0.41**

Table 1.4: Results on SHUTTLE dataset with fixed k varying l

KDD-CUP: We further evaluate our approach on 1999 KDD-CUP dataset which contains instances describing connections of tcp packet sequences. Every row is labeled as *intrusion* or *normal* along with their intrusion types. Since the dataset has both categorical and numerical attributes, for simplicity, we consider only 38 numerical attributes after having normalized each one of them so that they have 0 mean and standard deviation equal to 1. Similar to [63], we used 10% of the dataset for our experiment, that is, around 494,021 instances. There are 23 classes while 98.3% of the instances belong to only 3 classes, namely the class *smurf* 56.8%, the class *neptune* 21.6% and the class *normal* 19.6%. We took these three classes as non-outliers while the rest (1.7%) are considered as outliers.

The result of our algorithm on KDD-CUP dataset is reported in Table 1.5. Here also we compared our result with k-means-- while taking different values of k. We see that both techniques perform quit well in purity, that is, they are able to extract clusters which best matches the ground truth labels. While our algorithm better performs in identifying outliers with relatively good precision.

1.5 Path-Based Dominant Sets Clustering

Consider the data points shown in Figure 1.7(a). Despite the heavy background noise, we seem to have no difficulty in extracting a few “natural” clusters represent-

Method	k	l	Precision	Purity
	5	1.7%	0.564	0.987
k-means--	7	1.7%	0.568	**0.990**
	13	1.7%	0.593	0.981
Ours	n.a.	n.a.	**0.616**	**0.990**

Table 1.5: Results on KDD-CUP dataset with fixed number of outliers while varying cluster number

ing the letters of a familiar word. Unfortunately, standard clustering algorithms, such as those based on spectral graph theory, while doing a pretty good job in the noise-free case, perform rather poorly in such situations, as shown in Figure 1.7(c-d). The main reason behind this disappointing behavior is that they are typically all based on the idea of partitioning the input data, and hence the clutter points as well, into coherent classes.

In the last few years, dominant sets have emerged as a powerful alternative to spectral-based and similar methods [35], and are finding applications in a variety of different application domains. By focusing on the question "what is a cluster?" dominant sets overcome some of the classical limitations of partition-based approaches such as the inability to extract overlapping clusters and the need to know the number of clusters in advance [68]. A typical problem associated to dominant sets, however, is that they tend to favor compact clusters. The problem therefore remains as to how to deal with situations involving arbitrarily-shaped clusters in a context of heavy background noise.

In this thesis we propose a simple yet effective approach to solve this problem, which is based on the idea of feeding the dominant-set algorithm with a path-based similarity measure proposed earlier in a different context [69][70][71][72]. This takes into account connectivity information of the elements being clustered, thereby transforming clusters exhibiting an elongated structure under the original similarity function into compact ones. Recently, an approach which combines path-based similarities with spectral clustering has been introduced [72]. It improves the robustness of a spectral clustering algorithm by developing robust path-based similarity based on M-estimation from robust statistics. Instead of applying the spectral analysis directly on the original similarity matrix, they first modify the similarity matrix in such a way that the connectedness information is allowed for and at the same time checking if the sample is an outlier. However, the method is robust only against small number of thinly scattered outliers and, being based on spectral partition-based methods, it cannot safely extract elements from heavy background noise. Indeed, dominant sets and spectral clustering seem to exhibit a complementary features. On the one hand, spectral-based methods do typically a good job at extracting elongated clusters but perform poorly in the presence of clutter noise, on the other

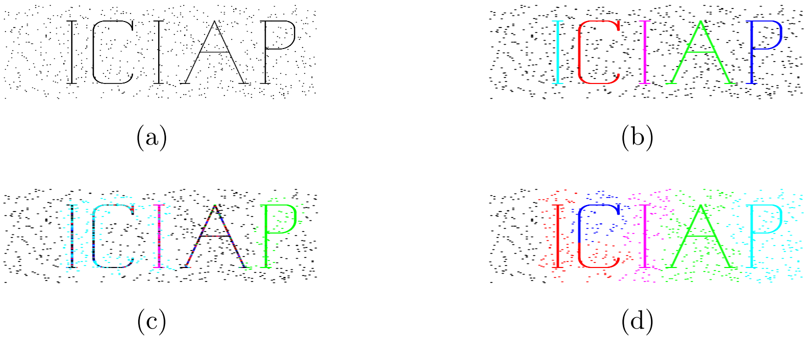


Figure 1.7: Results of extracting characters from clutter (a) Characters with uniformly distributed clutter elements which do not belong to any cluster (Original dataset to be clustered) (b) Result of our method (PBD) (c) The result of Path-based Spectral Clustering (PBS) (d) NJW's algorithms result.

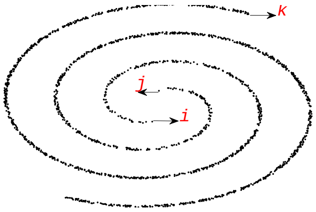


Figure 1.8: Point 'i' and point 'k', even-though they are very far from each other, are more similar than point 'i' and point 'j' as they are connected by a path with denser region.

hand the dominant-set algorithm prefers compact structures but is remarkably robust under heavy background noise. With our simple approach we are able to take the best of the two approaches, namely the ability to extract arbitrarily complex clusters and, at the same time, to deal with clutter noise. A similar attempt, though different objective, was done in [73]. Several experiments conducted over both toy and standard datasets have shown the effectiveness of the proposed approach.

1.5.1 Using Path-Based Similarity

The notion of path based technique, as shown in figure 1.8, is a simple but very effective way to capture elongated structures. It considers the connectedness information to transform elongated structures into compact ones. A path in a graph is a sequence of distinct edges which connects the vertices of the graph. Let the similarity between object 'i' and object 'j' is denoted as $a_{i,j}$, and suppose that two vertices have been connected by a number of different possible paths, which forms a set denoted by $\mathcal{P}_{i,j}$. What we set out to do here, to make objects connected by a path following dense regions, is to define an effective similarity for all the possible paths. The effective similarity between object 'i' and object 'j' along the path $p \in \mathcal{P}$ is set as the minimum edge weight among all the edges contained by the path p. The final best similarity measure between the two objects is chosen as the maximum of all the minimum computed edge weights.

$$a_{i,j}^{p} = \max_{p \in \mathcal{P}} \left\{ \min_{(1 \leq h < |p|)} a_{o[h],o[h+1]} \right\} \tag{1.19}$$

Where $o[h]$ indicates the object at the h^{th} position along the path p and $|p|$ is the number of objects along the path.

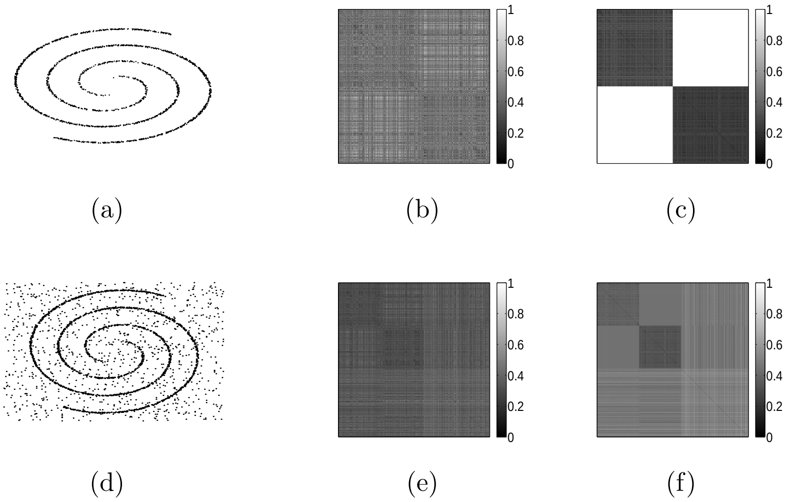


Figure 1.9: Distance matrices of two spiral datasets with and without noise. (a) input spiral data without any noise; (b) original distance matrix of (a); (c) Path-Based dissimilarity matrix of (a); (d) Input spiral data with noise (e) original distance matrix of (d); (f) Path-Based dissimilarity matrix of (d)

To observe how path-based technique is suitable for dominant set clustering, the (dis)similarity measures of the different transitions, for the spiral data set, are displayed as gray scale image. As shown from the figure 1.9 , the framework transforms the data well in such a way that the points of the spiral data set forms two block on the diagonal as a representative of the two clusters. The clusters of the data with out noise forms a clear diagonal block as shown on the first row of figure 1.9 which imply that any simple clustering algorithms such as K-Means can extract the clusters easily. When we come to the second case, it is clear to see, from the second row of figure 1.9, that the two cluster representatives do not form a very clear blocks on the diagonal which can be extracted with simple clustering algorithms. No existing methods are as accurate as our algorithm in extracting the two spirals from the clutter noise. While our algorithm uses dominant set as it easily identifies and extracts the two spirals as two dominant sets leaving the noise as non-dominant sets, other existing algorithms forces the clutter to one of the clusters.

1.5.2 Experiments

In this section we report a number of experimental results that are done on both toy and real datasets from UCI repository [74]. The experiments were conducted in two different ways. The first way of the experiments tests the performance of the different techniques without any clutter noise added. The second approach, which is done by adding a clutter noise samples to the datasets, is performed to see how much the algorithms are robust against background noise. In the first part of the experiment, we applied all algorithms to synthetic datasets of different manually designed structures while in the last part they are tested against real-world datasets.

Our approach was tested against five different approaches: One of the most successful spectral clustering algorithms (Ng-Jordan-Weiss (NJW) algorithm) [75], Path-based Spectral Clustering and Robust Path-based Spectral clustering (RPBS) [72] which outperformed the Path-based Clustering improving its robustness to noise. We compared against the above existing methods as they address similar problems: the problem of clustering algorithms to handle complex separable and elongated structures, and the robustness of clustering algorithms to noisy environments. All the algorithms, as opposed to our method, require the number of clusters. As of the standard clustering algorithms, all the methods also require choosing the scaling parameter σ which has been optimally selected for all the approaches. The forth and fifth approaches which we have used to compare against our algorithm are DBSCAN (DBS) and k-means (KM). We assigned the correct number of clusters for those approaches which require it in advance. For the case of DBSCAN, we used the implementation presented in [76].

1.5.2.1 Synthetic Data Clustering

In this part of the experiment, we applied our algorithm to eight different manually designed datasets which have been used by most of the existing algorithms for testing purpose. As can be seen from figure 1.13, the test had been done on complex separable structures. It has been shown that, classical clustering techniques such as K-means and Spectral Clustering can't solve the clustering problem in most of the data presented here [72]. However, extended version of the classical spectral clustering techniques and our proposed approach, as shown, in figure 1.13 are able to extracts all the clusters.

The robustness of our algorithm against noisy background is shown, using synthetic dataset, here. Similar works have been done to make clustering algorithm robust to noise [72] [77]. Our algorithm, as it uses dominant set framework, has the capability of extracting the best dominant sets leaving the clutter. However, other existing methods consider the background noise as part of the data to be partitioned.

It is clear to see that the existing approaches are vulnerable to applications where data is affected by clutter elements which do not belong to any cluster (as in figure/ground separation problems). Indeed, the only way to get rid of outliers is to

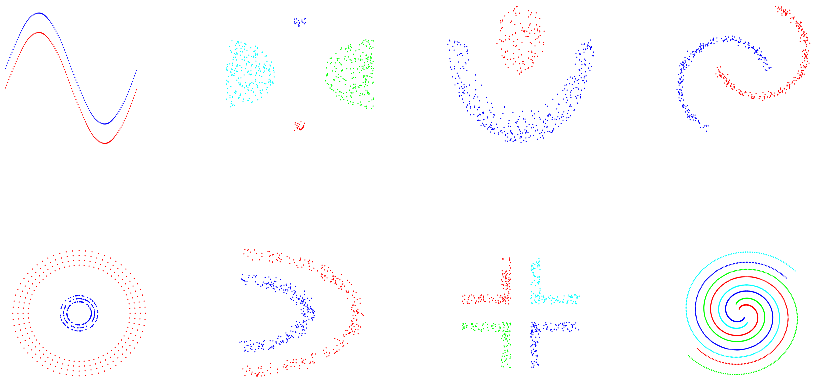


Figure 1.10: Clustering results of NJW algorithm, Path-based Spectral Clustering, Robust Path-based Spectral Clustering, and Path-based Dominant Set clustering. All of the four algorithms perform equally in extracting all the clusters

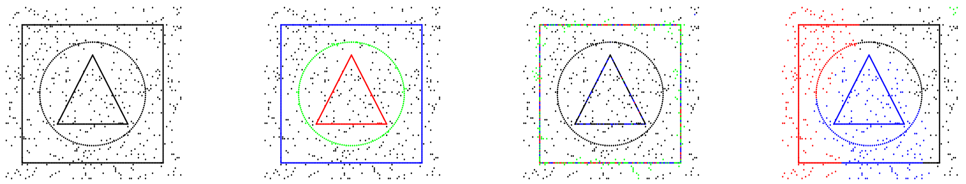
group them in additional clusters. However, since outliers are not mutually similar and intuitively they do not form a cluster, the performance of all the approaches but ours drop drastically as the percentage of noise level increases.

Figure 1.11 shows three shapes (Triangle,Square and Circle) together with uniformly distributed background noise. As we have described above, other methods are not able to extract the right clusters, the three shapes. For the same data of the figure, we have performed an experiment by increasing the level of noise starting from zero. Zero noise implies that we have only the three shapes with out any clutter with which all the four clustering algorithms extract the right clusters. A noise level 'N' implies that a uniformly distributed noise of size of N% of the size of the data is added as an outlier. Figure 1.12 (a) shows that at the zero noise level the accuracy of all the methods is 100 %, however, the performance of all the methods but ours drop drastically as the noise level increases.

Figure 1.12 (b) shows a similar experiment but the noise level which was done on extracting different characters from clutter. A noise level 'n' in this case mean a uniformly distributed n*5 samples put together with the data as a clutter. The result from this experiment also confirms that our approach outperforms all the other approaches.

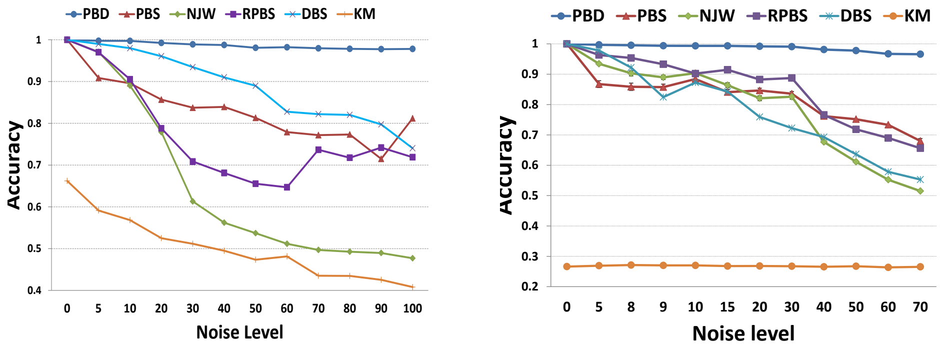
1.5.2.2 Experiments on Real-World Data

We also tested the algorithm on eight commonly used real-world datasets from UCI repository [74]. All the datasets incorporate cluster structures of complex separable, and most of them are with multiple scales. The performance of all the methods tested on the original dataset, refer table 1.6 , is almost comparable.



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

Figure 1.11: Results on three shapes with uniformly distributed clutter elements which do not belong to any cluster (a) Original dataset to be clustered (b) The result of our method (c) The result of Path-based Spectral Clustering (d) NJW's algorithms result. Observe that only our approach is efficient in extracting all the shapes from the background noise



(a) (b)

Figure 1.12: Performance of extracting three shapes (a) and letters (b), as of figure 1.11 and 1.7, from noisy background where the noise level is increased starting from zero.

An experiment has been conducted to show how much our method is robust to clutter noise added to the real-world datasets.

The experimental results, as can be referred from figure 1.13, consistently show that the existing approaches are vulnerable to applications where data is affected by clutter elements which do not belong to any cluster. It is easy to see, from figure 1.13, that the performance of all the approaches but ours drop drastically as noise is added to the datasets.

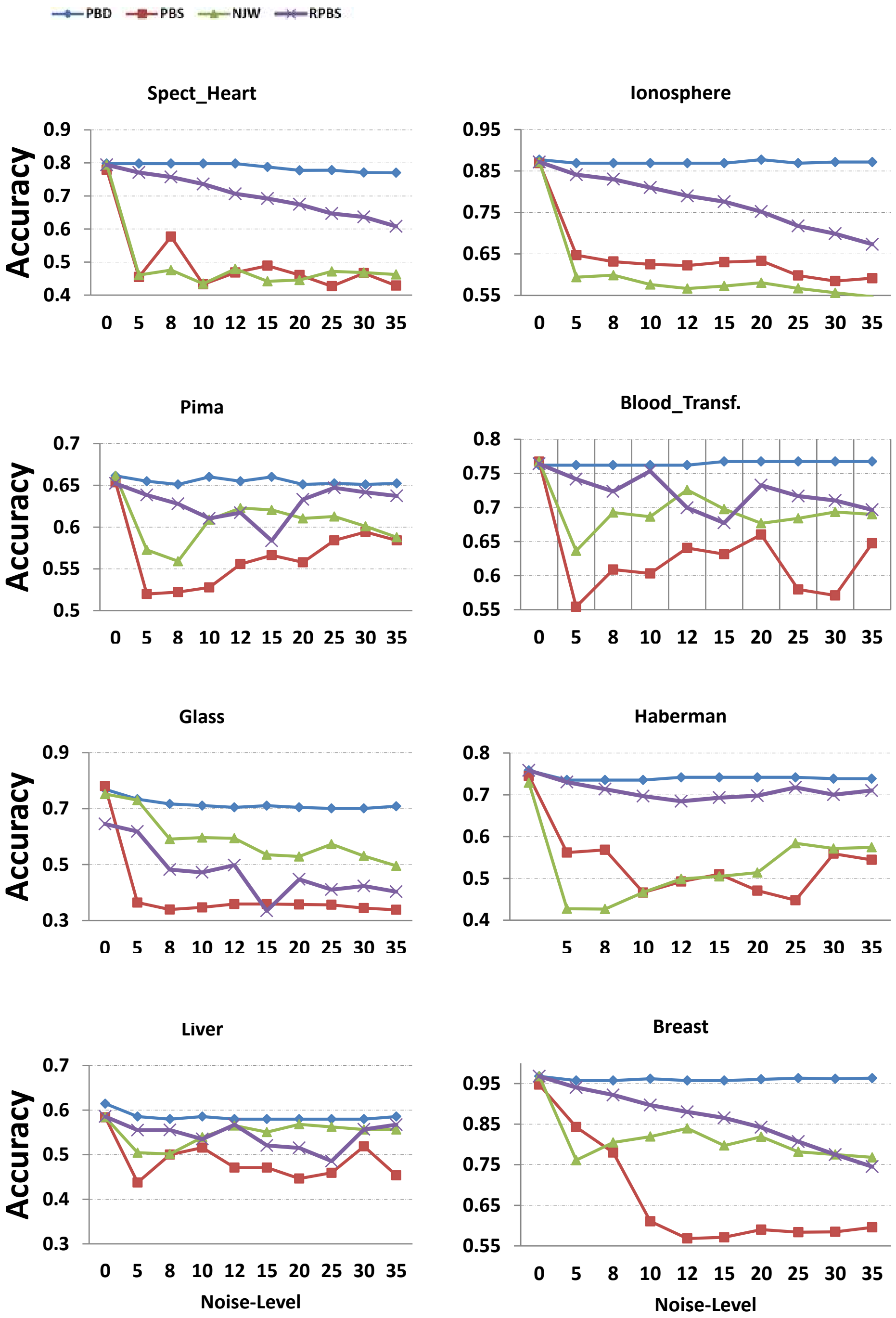


Figure 1.13: Clustering performance of the algorithms when a clutter noise is added to the dataset. Observe that the performance of all the approaches but ours drop drastically as the clutter noise is added.

Data	Inst.	Atr.	PBD	PBS	NJW	RPBS	DBS	KM
Ionosphere	351	33	**0.875**	0.869	0.872	0.863	0.849	0.712
Haberman	306	3	**0.758**	0.745	0.729	**0.758**	0.735	0.508
Spect Heart	267	8	**0.798**	0.779	0.794	0.794	0.749	0.571
Blood Trans.	748	10	0.762	0.767	**0.767**	0.763	0.209	0.730
Pima	768	8	**0.663**	0.654	0.662	0.652	0.634	0.661
Breast	683	9	**0.968**	0.950	**0.968**	**0.968**	0.818	0.961
Glass	214	10	0.766	**0.780**	0.752	0.752	0.467	0.553
Liver	345	6	**0.615**	0.588	0.574	0.586	0.559	0.554

Table 1.6: Accuracy on UCI datasets (Without noise)

1.6 Conclusion

In this chapter, a robust similarity based clustering is proposed. The algorithm has many interesting properties, convenient for solving many computer vision problems, such as: it does clustering while obliterating outliers in simultaneous fashion, it doesn't need any a prior knowledge on the number of clusters, able to deal with compact clusters and with situations involving arbitrarily-shaped clusters in a context of heavy background noise, does not have any assumptions with the structure of the affinity matrix, it is fast and scalable to large scale problems, and others. In section 1.3 of the chapter we have presented a novel approach, constrained dominant-set, that finds a collection of dominant-set clusters constrained to contain user-defined elements. The approach is based on some properties of a family of quadratic optimization problems related to dominant sets which show that, by properly selecting a regularization parameter that controls the structure of the underlying function, we are able to "force" all solutions to contain user specified elements. The performance of the proposed system is tested on random graph instances and DIMACS benchmark graphs. Section 1.4 deals with outlier detection from the data. Unlike most of the previous approaches our method requires no prior knowledge on both the number of clusters and outliers, which makes our approach more convenient for real application. Moreover, our proposed algorithm is simple to implement and highly scalable. We first test the performance of SCOD on large scale of synthetic datasets which confirms that in a controlled set up, the algorithm is able to achieve excellent result in an efficient manner. We conduct further evaluation on real datasets and attain prevailing results. The last section of the chapter shows how the path-based similarity measure, which takes into account connectedness information of the elements to be clustered, is used together with the dominant set framework to deal with the problem of extracting arbitrarily complex clusters under severe noise conditions.

2

Interactive Image (Co-)Segmentation Using Constrained Dominant Sets

I expect of abstraction as much as what imagery does for me... to carry meaning.

Kay WalkingStick

2.1 Introduction

Image segmentation is arguably one of the oldest and best-studied problems in computer vision, being a fundamental step in a variety of real-world applications, and yet remains a challenging task [78] [79]. Besides the standard, purely bottom-up formulation, which involves partitioning an input image into coherent regions, in the past few years several variants have been proposed which are attracting increasing attention within the community. Most of them usually take the form of a "constrained" version of the original problem, whereby the segmentation process is guided by some external source of information.

For example, user-assisted (or "interactive") segmentation has become quite popular nowadays, especially because of its potential applications in problems such as image and video editing, medical image analysis, etc. [80, 2, 81, 82, 83, 84, 85]. Given an input image and some information provided by a user, usually in the form of a scribble or of a bounding box, the goal is to provide as output a foreground object in such a way as to best reflect the user's intent. By exploiting high-level, semantic knowledge on the part of the user, which is typically difficult to formalize, we are therefore able to effectively solve segmentation problems which would be otherwise too complex to be tackled using fully automatic segmentation algorithms.

Existing algorithms fall into two broad categories, depending on whether the user annotation is given in terms of a scribble or of a bounding box, and supporters of the two approaches have both good reasons to prefer one modality against the other. For example, Wu et al. [81] claim that bounding boxes are the most natural and economical form in terms of the amount of user interaction, and develop a multiple instance learning algorithm that extracts an arbitrary object located inside a tight bounding box at unknown location. Yu et al. [86] also support the bounding-box approach, though their algorithm is different from others in that it does not need bounding boxes tightly enclosing the object of interest, whose production of course increases the annotation burden. They provide an algorithm, based on a Markov Random Field (MRF) energy function, that can handle input bounding box that only loosely covers the foreground object. Xian et al. [87] propose a method which avoids the limitations of existing bounding box methods - region of interest (ROI) based methods, though they need much less user interaction, their performance is sensitive to initial ROI.

On the other hand, several researchers, arguing that boundary-based interactive segmentation such as intelligent scissors [85] requires the user to trace the whole boundary of the object, which is usually a time-consuming and tedious process, support scribble-based segmentation. Bai et al. [88], for example, propose a model based on ratio energy function which can be optimized using an iterated graph cut algorithm, which tolerates errors in the user input. In general, the input modality in an interactive segmentation algorithm affects both its accuracy and its ease of use. Existing methods work typically on a single modality and they focus on how to use that input most effectively. However, as noted recently by Jain and Grauman [89], sticking to one annotation form leads to a suboptimal tradeoff between human and machine effort, and they tried to estimate how much user input is required to sufficiently segment a novel input.

Another example of a “constrained” segmentation problem is co-segmentation. Given a set of images, the goal here is to jointly segment same or similar foreground objects. The problem was first introduced by Rother *et al.* [90] who used histogram matching to simultaneously segment the foreground object out from a given pair of images. Recently, several techniques have been proposed which try to co-segment groups containing more than two images, even in the presence of similar backgrounds. Joulin *et al.* [91], for example, proposed a discriminative clustering framework, combining normalized cut and kernel methods and the framework has recently been extended in an attempt to handle multiple classes and a significantly larger number of images [92].

The co-segmentation problem has also been addressed using user interaction [93, 94]. Here, a user adds guidance, usually in the form of scribbles, on foreground objects of some of the input images. Batra *et al.* [93] proposed an extension of the (single-image) interactive segmentation algorithm of Boykov and Jolly [84]. They also proposed an algorithm that enables users to quickly guide the output of the co-segmentation algorithm towards the desired output via scribbles. Given scribbles, both on the background and the foreground, on some of the images, they cast the labeling problem as energy minimization defined over graphs constructed over each image in a group. Dong *et al.* [94] proposed a method using global and local energy optimization. Given background and foreground scribbles, they built a foreground

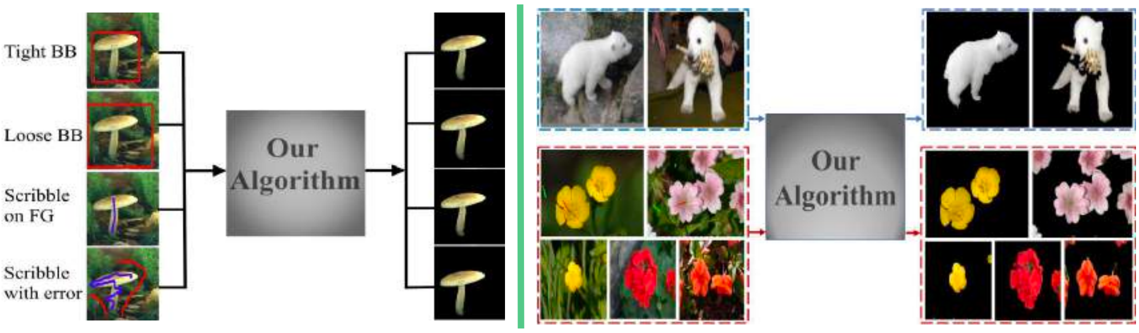


Figure 2.1: **Left:** An example of our interactive image segmentation method and its outputs, with different user annotation. Respectively from top to bottom, tight bounding box (Tight BB), loose bounding box (Loose BB), a scribble made (only) on the foreground object (Scribble on FG) and scribbles with errors. **Right:** Blue and Red dash-line boxes, show an example of our unsupervised and interactive co-segmentation methods, respectively.

and a background Gaussian mixture model (GMM) which are used as global guide information from users. By considering the local neighborhood consistency, they built the local energy as the local smooth term which is automatically learned using spline regression. The minimization problem of the energy function is then converted into constrained quadratic programming (QP) problem, where an iterative optimization strategy is designed for the computational efficiency.

In this chapter, we propose a unified approach, based on constrained dominant sets framework, to address this kind of problems which can deal naturally with various type of input modality, or constraints, and is able to robustly handle noisy annotations on the part of the external source. In particular, we shall focus on interactive segmentation and co-segmentation (in both the unsupervised and the interactive versions).

The resulting algorithm has a number of interesting features which distinguishes it from existing approaches. Specifically: 1) it is able to deal in a flexible manner with *both* scribble-based and boundary-based input modalities (such as sloppy contours and bounding boxes); 2) in the case of noiseless scribble inputs, it asks the user to provide *only* foreground pixels; 3) it turns out to be *robust* in the presence of input noise, allowing the user to draw, e.g., imperfect scribbles (including background pixels) or loose bounding boxes.

Experimental results on standard benchmark datasets demonstrate the effectiveness of our approach as compared to state-of-the-art algorithms on a wide variety of natural images under several input conditions. Figure 2.1 shows some examples of how our system works in both interactive segmentation, in the presence of different input annotations, and co-segmentation settings.

2.2 Interactive Image Segmentation using Constrained Dominant Sets

In this section, we apply our model (CDS) to the interactive image segmentation problem. As input modalities we consider scribbles as well as boundary-based approaches (in particular, bounding boxes) and, in both cases, we show how the system is robust under input perturbations, namely imperfect scribbles or loose bounding boxes.

In this application the vertices of the underlying graph G represent the pixels of the input image (or superpixels, as discussed below), and the edge-weights reflect the similarity between them. As for the set S, its content depends on whether we are using scribbles or bounding boxes as the user annotation modality. In particular, in the case of scribbles, S represents precisely those pixels that have been manually selected by the user. In the case of boundary-based annotation instead, it is taken to contain only the pixels comprising the box boundary, which are supposed to represent the background scene. Accordingly, the union of the extracted dominant sets, say $\mathcal{L}$ dominant sets are extracted which contain the set S, as described in the previous section and below, $\mathbf{UDS} = \mathcal{D}_1 \cup \mathcal{D}_2 ..... \cup \mathcal{D}_{\mathcal{L}}$, represents either the foreground object or the background scene depending on the input modality. For scribble-based approach the extracted set, **UDS**, represent the segmentation result, while in the boundary-based approach we provide as output the complement of the extracted set, namely $\mathbf{V} \setminus \mathbf{UDS}$.

Figure 2.2 shows the pipeline of our system. Many segmentation tasks reduce their complexity by using superpixels (a.k.a. over-segments) as a preprocessing step [81, 86, 95] [96, 97]. While [81] used SLIC superpixels [98], [86] used a recent superpixel algorithm [99] which considers not only the color/feature information but also boundary smoothness among the superpixels. In this work, we used the over-segments obtained from Ultrametric Contour Map (UCM) which is constructed from Oriented Watershed Transform (OWT) using globalized probability of boundary (gPb) signal as an input [1].

We then construct a graph G where the vertices represent over-segments and the similarity (edge-weight) between any two of them is obtained using a standard Gaussian kernel

$$\mathtt{A}^{\sigma}_{ij} = \mathbb{1}_{i \neq j} exp(\|\mathbf{f}_i - \mathbf{f}_j\|^2 / 2\sigma^2)$$

where $\mathbf{f}_i$, is the feature vector of the i^{th} over-segment, σ is the free scale parameter, and $\mathbb{1}_P = 1$ if P is true, 0 otherwise.

Given the affinity matrix $\mathtt{A}$ and the set S as described before, the system constructs the regularized matrix $M = \mathtt{A} - \alpha \hat{I}_S$, with α chosen as prescribed in (1.10). Then, the replicator dynamics (1.12) are run (starting them as customary from the simplex barycenter) until they converge to some solution vector $\mathbf{x}$. We then take the support of $\mathbf{x}$, remove the corresponding vertices from the graph and restart the replicator dynamics until all the elements of S are extracted.

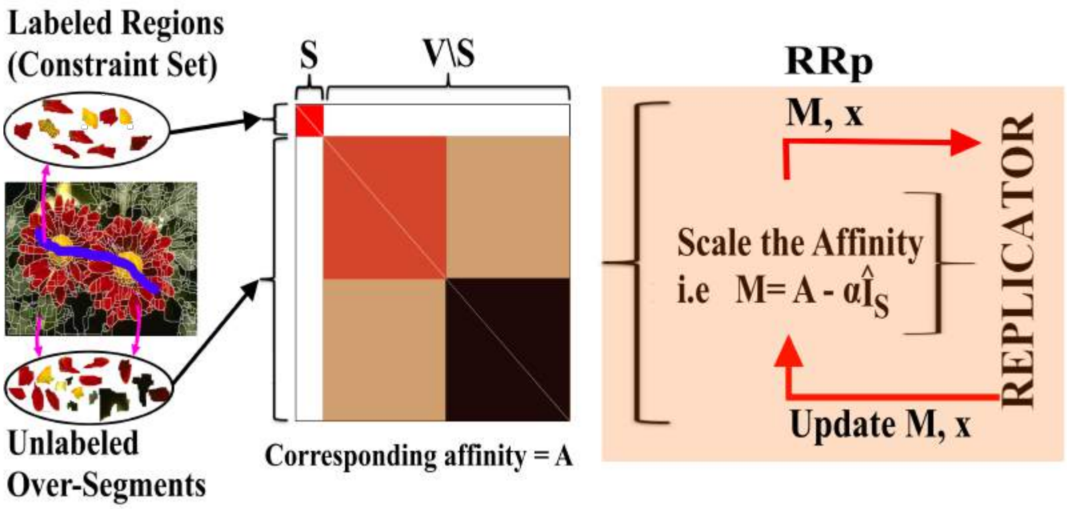


Figure 2.2: Overview of our interactive segmentation system. **Left:** Over-segmented image (output of the UCM-OWT algorithm [1]) with a user scribble (blue label). **Middle:** The corresponding affinity matrix, using each over-segments as a node, showing its two parts: S, the constraint set which contains the user labels, and $V \setminus S$, the part of the graph which takes the regularization parameter α. **Right:** RRp, starts from the barycenter and extracts the first dominant set and update $\mathbf{x}$ and $\mathbf{M}$, for the next extraction till all the dominant sets which contain the user labeled regions are extracted.

2.2.1 Experiments and Results

As mentioned above, the vertices of our graph represents over-segments and edge weights (similarities) are built from the median of the color of all pixels in RGB, HSV, and L*a*b* color spaces, and Leung-Malik (LM) Filter Bank [100]. The number of dimensions of feature vectors for each over-segment is then 57 (three for each of the RGB, L*a*b*, and HSV color spaces, and 48 for LM Filter Bank).

In practice, the performance of graph-based algorithms that use Gaussian kernel, as we do, is sensitive to the selection of the scale parameter σ. In our experiments, we have reported three different results based on the way σ is chosen: 1) CDS_Best_Sigma, in this case the best parameter σ is selected on a per-image basis, which indeed can be thought of as the optimal result (or upper bound) of the framework. 2) CDS_Single_Sigma, the best parameter in this case is selected on a per-database basis tuning σ in some fixed range, which in our case is between 0.05 and 0.2. 3) CDS_Self_Tuning, the σ^2 in the above equation is replaced, based on [101], by $\sigma_i * \sigma_j$, where $\sigma_i = mean(KNN(f_i))$, the mean of the K_Nearest_Neighbor of the sample f_i, K is fixed in all the experiment as 7.

Datasets: We conduct four different experiments on the well-known GrabCut dataset [80] which has been used as a benchmark in many computer vision tasks [102][2, 103, 104, 81, 86] [105, 106]. The dataset contains 50 images together with manually-labeled segmentation ground truth. The same bounding boxes as those in [2] is used as a baseline bounding box. We also evaluated our scribbled-based approach using the well known Berkeley dataset which contains 100 images.

Metrics: We evaluate the approach using different metrics: error rate, fraction of misclassified pixels within the bounding box, Jaccard index which is given by, following [107], $J = \frac{|GT \cap O|}{|GT \cup O|}$, where GT is the ground truth and O is the output. The third metric is the Dice Similarity Coefficient (DSC), which measures the overlap between two segmented object volume, and is computed as $DSC = \frac{2*|GT \cap O|}{|GT|+|O|}$.

Annotations: In interactive image segmentation, users provide annotations which guides the segmentation. A user usually provides information in different forms such as scribbles and bounding boxes. The input modality affects both its accuracy and ease-of-use [89]. However, existing methods fix themselves to one input modality and focus on how to use that input information effectively. This leads to a suboptimal tradeoff in user and machine effort. Jain et al. [89] estimates how much user input is required to sufficiently segment a given image. In this work as we have proposed an interactive framework, figure 2.1, which can take any type of input modalities we will use four different type of annotations: bounding box, loose bounding box, scribbles - only on the object of interest -, and scribbles with error as of [88].

2.2.1.1 Scribble Based Segmentation

Given labels on the foreground as constraint set, we built the graph and collect (iteratively) all unlabeled regions (nodes of the graph) by extracting dominant set(s) that contains the constraint set (user scribbles). We provided quantitative comparison against several recent state-of-the-art interactive image segmentation methods which uses scribbles as a form of human annotation: [84], Lazy Snapping [83], Geodesic Segmentation [82], Random Walker [108], Transduction [109] , Geodesic Graph Cut [105], Constrained Random Walker [106].

We have also compared the performance of our algorithm againts Biased Normalized Cut (BNC) [110], an extension of normalized cut, which incorporates a quadratic constraint (bias or prior guess) on the solution $\mathbf{x}$, where the final solution is a weighted combination of the eigenvectors of normalized Laplacian matrix. In our experiments we have used the optimal parameters according to [110] to obtain the most out of the algorithm.

Tables 2.1,2.2 and the plots in Figure 2.4 show the respective quantitative and the several qualitative segmentation results. Most of the results, reported on table 2.1, are reported by previous works [86, 81, 2, 105, 106]. We can see that the proposed CDS outperforms all the other approaches.

Error-tolerant Scribble Based Segmentation. This is a family of scribble-based approach, proposed by Bai et. al [88], which tolerates imperfect input scribbles thereby avoiding the assumption of accurate scribbles. We have done experiments using synthetic scribbles and compared the algorithm against recently proposed methods specifically designed to segment and extract the object of interest tolerating

Methods	Error Rate
BNC [110]	13.9
Graph Cut [84]	6.7
Lazy Snapping [83]	6.7
Geodesic Segmentation [82]	6.8
Random Walker [108]	5.4
Transduction [109]	5.4
Geodesic Graph Cut [105]	4.8
Constrained Random Walker [106]	4.1
CDS_Self Tuning (Ours)	**3.57**
CDS_Single Sigma (Ours)	**3.80**
CDS_Best Sigma (Ours)	2.72

Table 2.1: Error rates of different scribble-based approaches on the Grab-Cut dataset.

Methods	Jaccard Index
MILCut-Struct [81]	84
MILCut-Graph [81]	83
MILCut [81]	78
Graph Cut [80]	77
Binary Partition Trees [111]	71
Interactive Graph Cut [84]	64
Seeded Region Growing [112]	59
Simple Interactive O.E[113]	63
CDS_Self Tuning (Ours)	**93**
CDS_Single Sigma (Ours)	**93**
CDS_Best Sigma (Ours)	95

Table 2.2: Jaccard Index of different approaches – first 5 bounding-box-based – on Berkeley dataset.

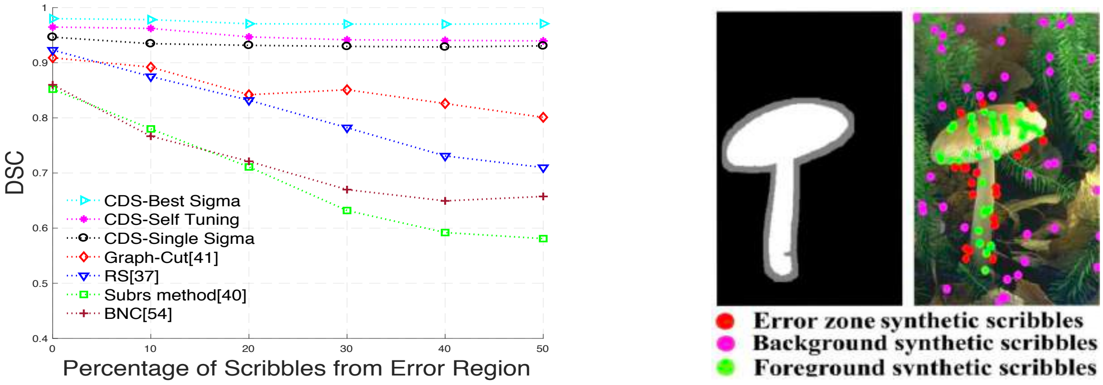


Figure 2.3: **Left:** Performance of interactive segmentation algorithms, on Grab-Cut dataset, for different percentage of synthetic scribbles from the error region. **Right:** Synthetic scribbles and error region

the user input errors [88, 114, 115, 116].

Our framework is adapted to this problem as follows. We give for our framework the foreground scribbles as constraint set and check those scribbled regions which include background scribbled regions as their members in the extracted dominant set. Collecting all those dominant sets which are free from background scribbled regions generates the object of interest.

Experiment using synthetic scribbles. Here, a procedure similar to the one used in [116] and [88] has been followed. First, 50 foreground pixels and 50 background pixels are randomly selected based on ground truth (see Fig. 2.3). They are then assigned as foreground or background scribbles, respectively. Then an error-zone for each image is defined as background pixels that are less than a distance D from the foreground, in which D is defined as 5 %. We randomly select 0 to 50 pixels in the error zone and assign them as foreground scribbles to simulate different degrees of user input errors. We randomly select 0, 5, 10, 20, 30, 40, 50 erroneous sample pixels from error zone to simulate the error percentage of 0%, 10%, 20%, 40%, 60%, 80%, 100% in the user input. It can be observed from figure 2.3 that our approach is not affected by the increase in the percentage of scribbles from error region.

2.2.1.2 Segmentation Using Bounding Boxes

The goal here is to segment the object of interest out from the background based on a given bounding box. The corresponding over-segments which contain the box label are taken as constraint set which guides the segmentation. The union of the extracted set is then considered as background while the union of other over-segments represent the object of interest.

We provide quantitative comparison against several recent state-of-the-art interactive image segmentation methods which uses bounding box: LooseCut [86], GrabCut [80], OneCut [104], MILCut [81], pPBC and [103]. Table 2.3 and the pictures in Figure 2.4 show the respective error rates and the several qualitative segmentation results. Most of the results, reported on table 2.3, are reported by previous works [86, 81, 2, 105, 106].

Segmentation Using Loose Bounding Box. This is a variant of the bounding box approach, proposed by Yu et.al [86], which avoids the dependency of algorithms on the tightness of the box enclosing the object of interest. The approach not only avoids the annotation burden but also allows the algorithm to use automatically detected bounding boxes which might not tightly encloses the foreground object. It has been shown, in [86], that the well-known GrabCut algorithm [80] fails when the looseness of the box is increased. Our framework, like [86], is able to extract the object of interest in both tight and loose boxes. Our algorithm is tested against a series of bounding boxes with increased looseness. The bounding boxes of [2] are used as boxes with 0% looseness. A looseness L (in percentage) means an increase in the area of the box against the baseline one. The looseness is increased, unless it reaches the image perimeter where the box is cropped, by dilating the box by a number of pixels, based on the percentage of the looseness, along the 4 directions: left, right, up, and down.

For the sake of comparison, we conduct the same experiments as in [86]: 41 images out of the 50 GrabCut dataset [80] are selected as the rest 9 images contain multiple objects while the ground truth is only annotated on a single object. As other objects, which are not marked as an object of interest in the ground truth, may be covered when the looseness of the box increases, images of multiple objects are not applicable for testing the loosely bounded boxes [86]. Table 2.3 summarizes the results of different approaches using bounding box at different level of looseness. As can be observed from the table, our approach performs well compared to the others when the level of looseness gets increased. When the looseness $L = 0$, [81] outperforms all, but it is clear, from their definition of tight bounding box, that it is highly dependent on the tightness of the bounding box. It even shrinks the initially given bounding box by 5% to ensure its tightness before the slices of the positive bag are collected. For looseness of $L = 120$ we have similar result with LooseCut [86] which is specifically designed for this purpose. For other values of L our algorithm outperforms all the approaches.

Complexity: In practice, over-segmenting and extracting features may be treated as a pre-processing step which can be done before the segmentation process. Given the affinity matrix, we used replicator dynamics (1.12) to exctract constrained dominant sets. Its computational complexity per step is $O(N^2)$, with N being the total number of nodes of the graph. Given that our graphs are of moderate size (usually less than 200 nodes) the algorithm is fast and converges in fractions of a second, with a code written in Matlab and run on a core i5 6 GB of memory. As for the pre-processing step, the original *gPb-owt-ucm* segmentation algorithm was very slow to be used as a practical tools. Catanzaro et al. [117] proposed a faster alternative, which reduce the runtime from 4 minutes to 1.8 seconds, reducing the computa-

Methods	$L =$ 0%	$L =$ 120%	$L =$ 240%	$L =$ 600%
GrabCut [80]	7.4	10.1	12.6	13.7
OneCut [104]	6.6	8.7	9.9	13.7
pPBC [103]	7.5	9.1	9.4	12.3
MilCut [81]	**3.6**	-	-	-
LooseCut [86]	7.9	**5.8**	6.9	6.8
CDS_Self Tuning (Ours)	7.54	6.78	**6.35**	7.17
CDS_Single Sigma (Ours)	7.48	5.9	**6.32**	**6.29**
CDS_Best Sigma (Ours)	6.0	4.4	4.2	4.9

Table 2.3: Error rates of different bounding-box approaches with different level of looseness as an input, on the Grab-Cut dataset. $L = 0\%$ implies a baseline bounding box as those in [2]

tional complexity and using parallelization which allow *gPb* contour detector and *gPb-owt-ucm* segmentation algorithm practical tools. For the purpose of our experiment we have used the Matlab implementation which takes around four minutes to converge, but in practice it is possible to give for our framework as an input, the GPU implementation [117] which allows the convergence of the whole framework in around 4 seconds.

2.3 Co-Segmentation Using Constrained Dominant Sets

In this section, we describe the application of constrained dominant sets (CDS) to co-segmentation, both unsupervised and interactive. Among the difficulties that make this problem a challenging one, we mention the similarity among the different backgrounds and the similarity of object and background [118] (see, e.g., the top row of Figure 2.5). A measure of "objectness" has proven to be effective in dealing with such problems and improving the co-segmentation results [118][119]. However, this measure alone is not enough, especially when one aims to solve the problem using global pixel relations. One can see from Figure 2.5 (bottom) that the color of the cloth of the person, which of course is one of the objects, is similar to the color of the dog which makes systems that are based on objectness measure fail. Moreover the object may not also be the one which we want to co-segment.

Figure 2.6 and 2.7 show the pipeline of our unsupervised and interactive co-segmentation algorithms, respectively.

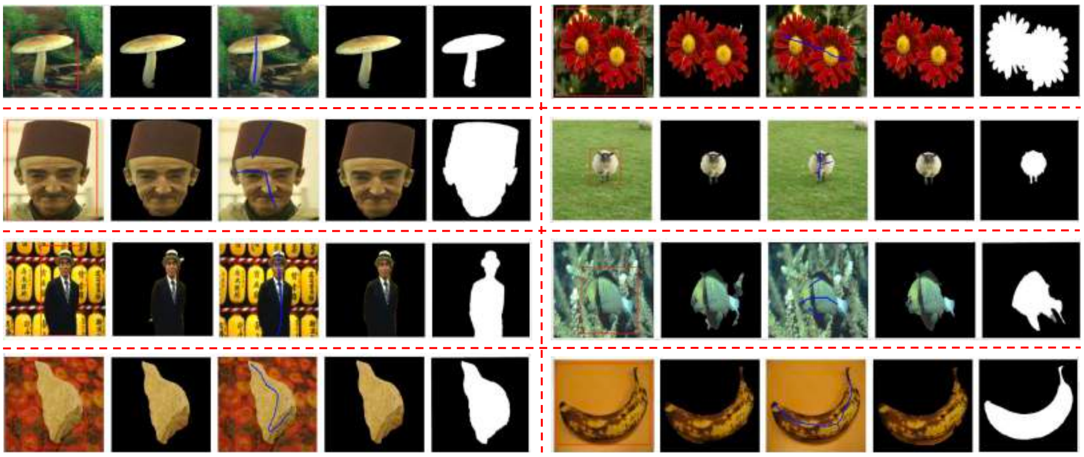


Figure 2.4: Examplar results of the interactive segmentation algorithm tested on Grab-Cut dataset. (In each block of the red dashed line) **Left:** Original image with bounding boxes of [2]. **Middle left:** Result of the bounding box approach. **Middle:** Original image and scribbles (observe that the scribles are only on the object of interest). **Middle right:** Results of the scribbled approach. **Right:** The ground truth.

In figure 2.6, $\mathbf{I}_1$ and $\mathbf{I}_2$ are the given pair of images while $\mathcal{S}_1$ and $\mathcal{S}_2$ represent the corresponding sets of superpixels. The affinity is built using the objectness score of the superpixels and using different handcrafted features extracted from the superpixels. The set of nodes V is then divided into two as the constraint set (S) and the non-constraint ones, $V \backslash S$. We run the CDS algorithm twice: first, setting the nodes of the graph that represent the first image as constraint set and $\mathbb{O}_2$ represents our output. Second we change the constraint set S with nodes that come from the second image and $\mathbb{O}_1$ represents the output. The intersection $\mathbb{O}$ refines the two results and represents the final output of the proposed unsupervised co-segmentation approach.

Our interactive co-segmentation approach, as shown using Figure 2.7, needs user interaction which guides the segmentation process putting scribbles (only) on some of the images with ambiguous objects or background. $\mathbf{I}_1, \mathbf{I}_2, ...\mathbf{I}_n$ are the scribbled images and $\mathbf{I}_{n+1}, ..., \mathbf{I}_{n+m}$ are unscribbled ones. The corresponding sets of superpixels are represented as $\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_n, ...\mathcal{S}_{n+1}, ...\mathcal{S}_{n+m}$. $\mathtt{A}'_{\mathbf{s}}$ and $\mathtt{A}_{\mathbf{u}}$ are the affinity matrices built using handcrafted feature-based similarities among superpixels of scribbled and unscribbled images respectively. Moreover, the affinities incorporate the objectness score of each node of the graph. $\mathcal{B}_{\mathbf{sp}}$ and $\mathcal{F}_{\mathbf{sp}}$ are (respectively) the background and foreground superpixels based on the user provided information. The CDS algorithm is run twice over $\mathtt{A}'_{\mathbf{s}}$ using the two different user provided information as constraint sets which results outputs $\mathbb{O}_1$ and $\mathbb{O}_2$. The intersection of the two outputs, $\mathbb{O}$, help us get new foreground and background sets represented by $\mathcal{B}_{\mathbf{s}}$, $\mathcal{F}_{\mathbf{s}}$. Modifying the affinity $\mathtt{A}'_{\mathbf{s}}$, putting the similarities among elements of the two sets to zero, we get the new affinity $\mathtt{A}_{\mathbf{s}}$. We then build the biggest affinity which incorporates all images'

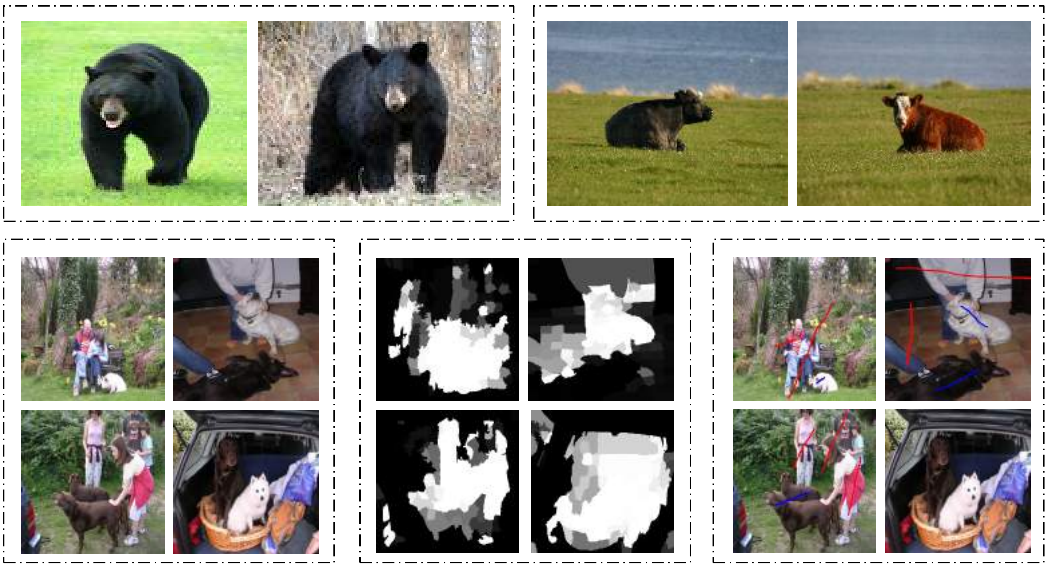


Figure 2.5: The challenges of co-segmentation. Examplar image pairs: **(top left)** similar foreground objects with significant variation in background, **(top right)** foreground objects with similar background. The **bottom** part shows why user interaction is important for some cases. The **bottom left** is the image, **bottom middle** shows the objectness score, and the **bottom right** shows the user label.

superpixels. As our affinity is symmetric, $\mathtt{A}_{\mathbf{us}}$ and $\mathtt{A}_{\mathbf{su}}$ are equal and incorporates the similarities among the superpixels of the scribbled and unscribbled sets of images. Using the new background and foreground sets as two different constraint sets, we run CDS twice which results outputs $\mathbb{O}'_1$ and $\mathbb{O}'_2$ whose intersection ($\mathbb{O}'$) represents the final output.

2.3.1 Experiments and Results

Given an image, we over-segment it to get its superpixels $\mathcal{S}$, which are considered as vertices of a graph. We then extract different features from each of the superpixels. The first features which we consider are features from the different color spaces: RGB, HSV and CIE Lab. Given the superpixels, say size of n, of an image i, $\mathcal{S}_i$, $\mathcal{F}_c^i$ is a matrix of size $n \times 9$ which is the mean of each of the channels of the three color spaces of pixels of the superpixel. The mean of the SIFT features extracted from the superpixel $\mathcal{F}_s^i$ is our second feature. The last feature which we have considered is the rotation invariant histogram of oriented gradient (HoG), $\mathcal{F}_h^i$.

The dot product of the SIFT features is considered as the SIFT similarity among the nodes, let us say the corresponding affinity matrix is $\mathtt{A}_s$. Motivated by [120], the similarity among the nodes of image i and image j ($i \neq j$), based on color, is computed from their Euclidean distance $\mathcal{D}_c^{i \times j}$ as

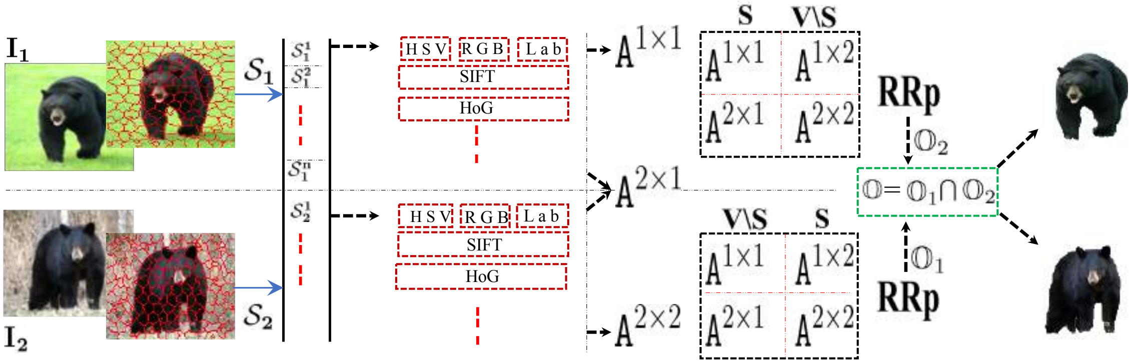


Figure 2.6: Overview of our unsupervised co-segmentation algorithm.

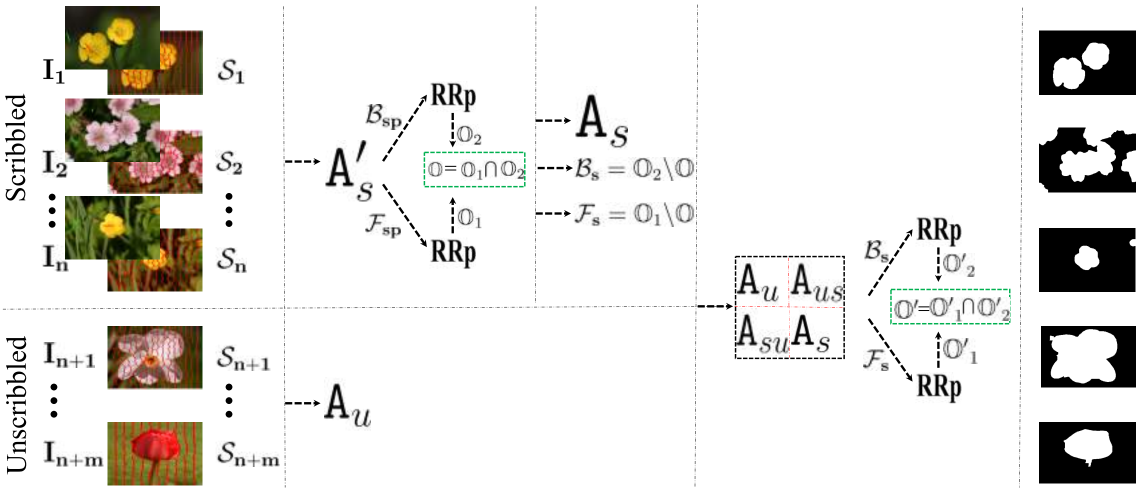


Figure 2.7: Overview of our interactive co-segmentation algorithm.

$$\mathtt{A}_c^{i\times j} = max(\mathcal{D}_c) - \mathcal{D}_c^{i\times j} + min(\mathcal{D}_c)$$

The HoG similarity among the nodes, $\mathtt{A}_h^{i\times j}$, is computed in a similar way , as $\mathtt{A}_c$, from the diffusion distance. All the similarities are then min max normalized.

We then construct the $\mathtt{A}_c^{i\times i}$, the similarities among superpixels of image i, which only considers adjacent superpixels as follows. First, construct the dissimilarity graph using their Euclidean distance considering their average colors as weight. Then, compute the geodesic distance as the accumulated edge weights along their shortest path on the graph, we refer the reader to [14] to see how such type of distances improve the performance of dominant sets. Assuming the computed geodesic distance matrix is $\mathcal{D}_{geo}$, the weighted edge similarity of superpixel p and superpixel q, say $e_{p,q}$, is computed as

$$e_{p,q} = \begin{cases} 0, & \text{if} \quad \text{p and q are not adjacent,} \\ max(\mathcal{D}_{geo}) - \mathcal{D}_{geo}(p,q) + min(\mathcal{D}_{geo}), & \text{otherwise} \end{cases} \tag{2.1}$$

$\mathtt{A}_h^{i \times i}$ for HoG is computed in a similar way while and $\mathtt{A}_s^{i \times i}$ for SIFT is built by just keeping adjacent edge similarities.

Assuming we have I images, the final affinity $\mathtt{A}_\gamma$ (γ can be c, s or h in the case of color, SIFT or HOG respectively) is built as

$$\mathtt{A}_\gamma = \begin{pmatrix} \mathtt{A}_\gamma^{1\times 1} & .. & \mathtt{A}_\gamma^{1\times j} & .. & \mathtt{A}_\gamma^{1\times I} \\ . & . & . & . & . \\ \mathtt{A}_\gamma^{j\times 1} & .. & \mathtt{A}_\gamma^{j\times j} & . & \mathtt{A}_\gamma^{1\times I} \\ . & & . & . & . \\ \mathtt{A}_\gamma^{I\times 1} & .. & \mathtt{A}_\gamma^{I\times j} & .. & \mathtt{A}_\gamma^{I\times I} \end{pmatrix}$$

As our goal is to segment common foreground objects out, we should consider how related backgrounds are eliminated. As shown in the examplar image pair of Figure 2.5 (top right), the two images have a related background to deal with it which otherwise would be included as part of the co-segmented objects. To solve this problem we borrowed the idea from [121] which proposes a robust background measure, called boundary connectivity. Given a superpixel $\mathcal{SP}_i$, it computes, based on the background measure, the backgroundness probability $\mathcal{P}_b^i$. We compute the probability of the superpixel being part of an object $\mathcal{P}_f^i$ as its additive inverse, $\mathcal{P}_f^i$ = 1 - $\mathcal{P}_b^i$. From the probability $\mathcal{P}_f$ we built a score affinity $\mathtt{A}_m$ as

$$\mathtt{A}_m(i,j) = \mathcal{P}_f^i * \mathcal{P}_f^j$$

2.3.1.1 Optimization

We model the foreground object extraction problem as the optimization of the similarity values among all image superpixels. The objective utility function is designed to assign the object region a membership score of greater than zero and the background region zero membership score, respectively. The optimal object region is then obtained by maximizing the utility function. Let the membership score of N superpixels be $\{x_i\}_{i=1}^N$, the (i,j) entry of a matrix $\mathtt{A}_z$ is z_{ij}. Our utility function, combining all the aforementioned terms ($\mathtt{A}_c$,$\mathtt{A}_s$,$\mathtt{A}_h$ and $\mathtt{A}_m$), is thus defined, based on equation (1.6), as:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\left\{\frac{1}{2}\underbrace{x_i x_j m_{ij}}_{\text{objectness score}} + \frac{1}{6} x_i x_j \underbrace{(c_{ij} + s_{ij} + h_{ij})}_{\text{feature similarity}} - \alpha x_i x_j\right\} \tag{2.2}$$

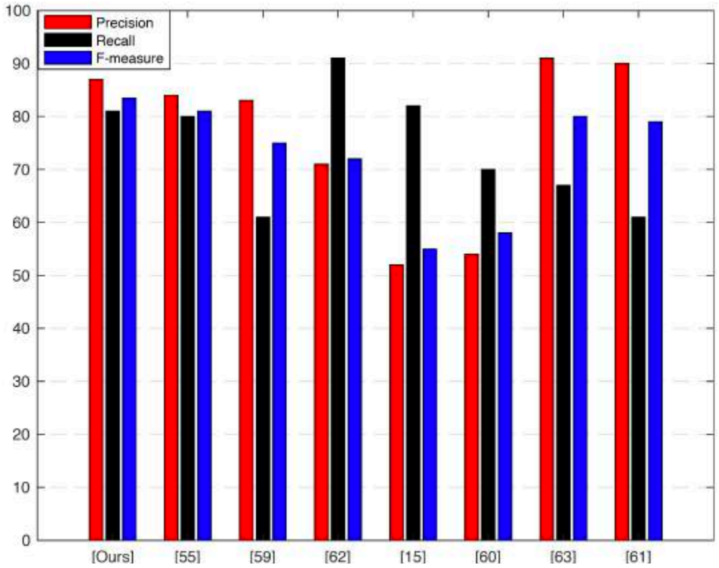


Figure 2.8: Precision, Recall and F-Measure based performance comparison of our unsupervised co-segmentation method with the state-of-the art approaches on image pair dataset

The parameter α is fixed based on the (non-)constraint set of the nodes. For the case of unsupervised co-segmentation, the nodes of the pairs of images are set (interchangeably) as constraint set where the intersection of the corresponding results give us the final co-segmented objects.

In the interactive setting, every node i (based on the information provided by the user) has three states: $i \in FGL$, (i is labeled as foreground label), $i \in BGL$ (i is labeled as background label) or $i \in V\backslash(FGL \cup BGL)$ (i is unlabeled). Hence, the affinity matrix $\mathtt{A} = (a_{ij})$ is modified by setting a_{ij} to zero if nodes i and j have different labels (otherwise we keep the original value).

The optimization, for both cases, is represented in the pipelines by '**RRp**' (replicator dynamics).

To evaluate the performance of our algorithms, we conducted extensive experiments on standard benchmark datasets that are widely used to evaluate the co-segmentation problem: image pairs [122] and MSRC [123]. The image pairs dataset consists 210 images (105 image pairs) of different animals, flowers, human objects, buses, etc. Each of the image pairs contains one or more similar objects. Some of them are relatively simple and some other contains set of complex image pairs, which contain foreground objects with higher appearance variations or low contrast objects with complex backgrounds.

MSRC dataset has been widely used to evaluate the performance of image co-segmentation methods. It contains 14 categories with 418 images in total. We evaluated our interactive co-segmentation algorithm on nine selected object classes of MSRC dataset (bird, car, cat, chair, cow, dog, flower, house, sheep), which contains

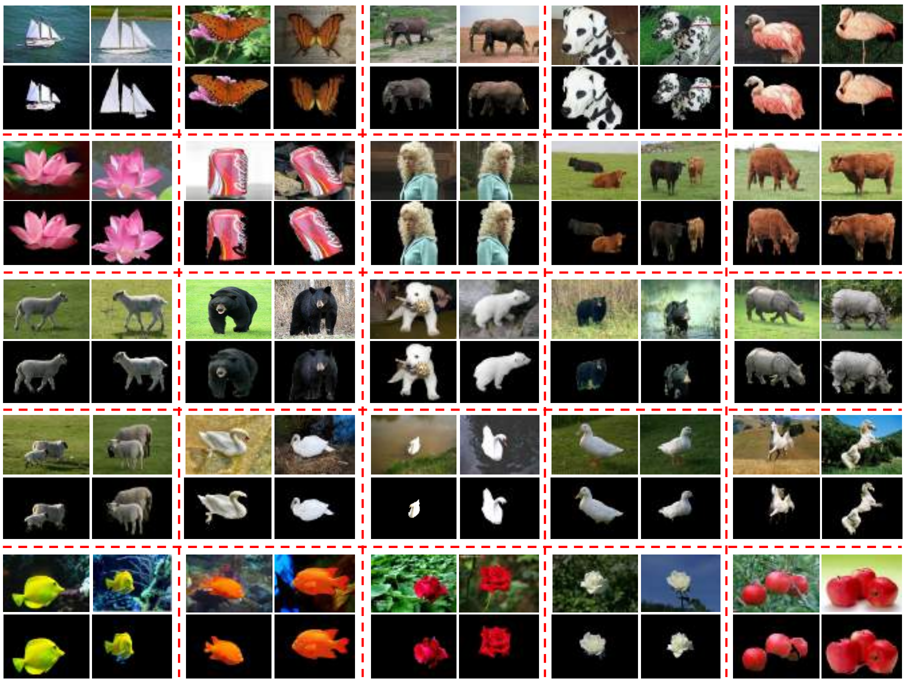


Figure 2.9: Examplar qualitative results of our unsupervised method tested on image pair dataset. **Upper row:** Original image **Lower row:** Result of the proposed unsupervised algorithm.

25∼30 images per class. We put foreground and background scribbles on 15∼20 images per class. Each image was over-segmented to 78∼83 SLIC superpixels using the VLFeat toolbox.

As customary, we measured the performance of our algorithm using precision, recall and F-measure, which were computed based on the output mask and human-given segmentation ground-truth. Precision is calculated as the ratio of correctly detected object pixels to the number of detected object pixels, while recall is the ratio of correctly detected object pixels to the number of ground truth pixels. We have computed the F-measure by setting γ^2 to 0.3 as used in [122][124][119].

We have applied Biased Normalized Cut (BNC) [110] on co-segmentation problem on MSRC dataset by using the same similarity matrix we used to test our method, and the comparison result of each object class is shown in Figure 2.10. As can be seen, our method significantly surpasses BNC and [94] in average F-measure. Furthermore, we have tested our interactive co-segmentation method, BNC and [94] on image pairs dataset by putting scribbles on one of the two images. As can be observed from Table 2.4, our algorithm substantially outperforms BNC and [94] in precision and F-measure (the recall score being comparable among the three competing algorithms).

In addition to that, we have examined our unsupervised co-segmentation algorithm by using image pairs dataset, the barplot in Figure 2.8 shows the quantitative

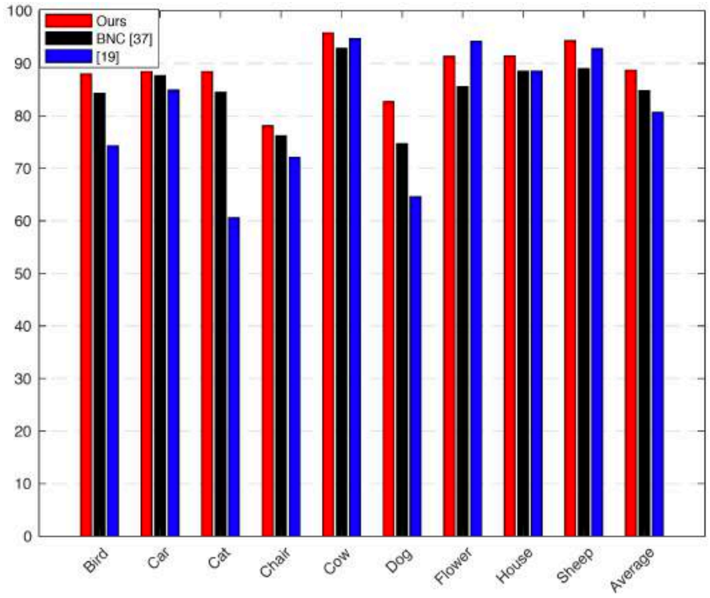


Figure 2.10: F-Measure based performance Comparison of our interactive co-segmentation method with state-of-the-art methods on MSRC dataset.

result of our algorithm comparing to the state-of-the-art methods [119][125][126]. As shown here, our algorithm achieves the best F-measure comparing to all other state-of-the-art methods. The qualitative performance of our unsupervised algorithm is shown in Figure 2.9 on some example images taken from image pairs dataset. As can be seen, Our approach can effectively detect and segment the common object of the given pair of images.

Metrics	$Precision$	$Recall$	$F - measure$
[94]	0.5818	0.8239	0.5971
BNC	0.6421	**0.8512**	0.6564
Ours	**0.7076**	0.8208	**0.7140**

Table 2.4: Results of our interactive co-segmentation method on Image pair dataset putting user scribble on one of the image pairs

2.4 Conclusions

In this chapter, we have demonstrated the applicability of constrained dominant sets to problems such as interactive image segmentation and co-segmentation (in both the unsupervised and the interactive flavor). In our perspective, these can be thought of as "constrained" segmentation problems involving an external source of information (being it, for example, a user annotation or a collection of related images to segment jointly) which somehow drives the whole segmentation process. The proposed method is flexible and is capable of dealing with various forms of constraints and input modalities, such as scribbles and bounding boxes, in the case of interactive segmentation. Extensive experiments on benchmark datasets have shown that our approach considerably improves the state-of-the-art results on the problems addressed. This provides evidence that constrained dominant sets hold promise as a powerful and principled framework to address a large class of computer vision problems formulable in terms of constrained grouping. Indeed, we mention that they are already being used successfully in other applications such as content-based image retrieval [127], multi-target tracking [16] and image geo-localization [17].

3
Constrained Dominant Sets for Retrieval

I think you travel to search and
you come back home to find
yourself there.

Chimamanda Ngozi Adichie

3.1 Introduction

Retrieval has recently attracted considerable attention within the computer vision community, especially because of its potential applications such as database retrieval, web and mobile image search. Given a user provided query, the goal is to provide as output a ranked list of objects that best reflect the user's intent. Classical approaches perform the task based on the (dis)similarity between the query and the database objects. The main limitation of such classical retrieval approaches is that they do not allow for the intrinsic relation among the database objects.

Recently, various techniques, instead of simply using the pairwise similarity, try to learn a better similarities that consider manifold structures of the underlying data. Qin *et al.* [128] try to alleviate the asymmetry problem of the k-nearest neighbor (k-NN) using the notion of k-reciprocal nearest neighbor. In [129] the notion of shared nearest neighbor is used to build secondary similarity measure, which stabilizes the performance of the search, based on the primary distance measure. In [130] shape meta-similarity measure, which is computed as the **L1** distance between new vector representation which considers only the k-NN set of similarities fixing all others to 0, was proposed. Choosing the right size of the neighbor is important. In [131], the notion of shortest path was used to built a new affinity for retrieval.

Diffusion process is one of the recent effective tools in learning the intrinsic manifold structure of a given data [132, 133, 134]. Given data, a weighted graph is built where the nodes are the objects and the edge weight is a function of the affinity between the objects. The pairwise affinities are then propagated following structure of the weighted edge links in the graph. The result of the affinity propagation highly depends on the quality of the pairwise similarity [135, 136]. Inaccurate Pairwise similarity results in a graph with much noise which negatively affects the diffusion process. Constraining the diffusion process locally alleviates this issue [136, 134, 132]. Dominant neighbor (DN) and k-NN are two notions used by the recent existing methods to constrain the diffusion process locally [132, 133, 134]. In [132], it has been shown that affinity learning constraining relation of an object to its neighbors effectively improves the retrieval performance and was able to achieve 100 % bull's eye score in the well known MPEG datset. The author of [132] put automatically selecting local neighborhood size (K) as the main limitation of the approach and is still an open problem. The influence of selecting different K values was also studied which proved that the parameter is a serious problem of the approach. For MPEG7 dataset, the choice is insignificant while for the other two datasets YALE and ORL choosing the reasonable K is difficult which resulted in a decrease in performance for the right value of K. Moreover, it is obvious that the selection of k-NN is prone to errors in the pairwise similarities [134]. Since any k-NN decision procedure relies only on affinities of an object to all other objects, k-NN approach is handicapped in resisting errors in pairwise affinities and in capturing the structure of the underlying data manifold.

Yang *et al.* in [134], to avoid the above issues, proposed the notion of dominant neighbors (DN). Instead of the k-NN, here a compact set from the k-NN which best explains the intrinsic relation among the neighbors is considered to constrain the diffusion process. However, the approach follows heuristic based k-NN initialization scheme. To capture dominant neighbors, the approach first choose a fixed value of K, collect the K nearest neighbors and then initialize the dynamics, the dynamics which extracts dense neighbors, to the barycenter of the face of the simplex which contains the neighbors. It is obvious to see that the approach is still dependent on K. Moreover, as fixing K limits the dynamics to a specified face of the simplex, objects out of k-NN(q) which form a dominant neighbor with q will be loosed. The chosen k-NN may also be fully noisy which might not have a compact structure.

In this chapter, we propose a new approach, using CDS, to retrieval which can deal naturally with the above problems. The resulting algorithm has a number of interesting features which distinguishes it from existing approaches. Specifically: 1) it is able to constrain the diffusion process locally extracting dense neighbors whose local neighborhood size (K) is fixed automatically; different neighbors can have different value of K. 2) it does not have any initialization step; the dynamics, to extract the dense neighbors, can start at any point in the standard simplex 3) it turns out to be *robust* to noisy affinity matrices.

3.2 Diffusion Process

Given the affinity matrix $\mathtt{A}$, a diffusion process starts from a predefined initialization, say $\mathcal{V}$ and propagates the affinity value through the underlying manifold based on a predefined transition matrix, say $\mathcal{T}$, and diffusion scheme ($\mathcal{S}$).

Off-the-shelf diffusion processes, which basically differ based on the choice of $\mathcal{V}$, $\mathcal{T}$ and $\mathcal{S}$, the most related ones to this work are [134, 137]. In both cases, the diffusion process is locally constrained. While in [137] the notion of k-NN is used to constrain the diffusion process locally, dominant neighbor notion ($\mathcal{DN}$) is used by [134].

3.2.1 Nearest Neighbors

In the first case, the edge-weights of the k-NN are kept i.e define locally constrained affinity $\mathcal{L} = (l_{ij})$ defined as $l_{ij} = w(i,j)$, if $(i,j) \in$ k-NN(q), and $l_{ij} = 0$ otherwise. Then the diffusion process, setting $\mathcal{V}$ as the affinity A , is performed by the following update rule.

$$\mathcal{V}_{t+1} = \mathcal{L}\mathcal{V}\mathcal{L}' \tag{3.1}$$

Nearest neighbors constrained diffusion process, alleviating the issue of noisy pairwise similarity, significantly increases the retrieval performance. However, the approach has two serious limitations: First, automatically selecting local neighborhood size (K) is very difficult and is still an open problem [132]. In [132] the influence of selecting different K values was studied which proved that the parameter is a serious problem of the approach. For MPEG7 dataset, the choice was insignificant while for the other two datasets, YALE and ORL, choosing the reasonable K was difficult which even resulted in a decrease in performance, for ORL from 77.30% to 73.40% and for YALE 77.08% to 73.39%, for the right value of K. Moreover, it is obvious that the selection of k-NN is prone to errors in the pairwise similarities [134].

3.2.2 Dominant Neighbors

Yang *et al.* in [134], to avoid the above issues, proposed the notion of dominant neighbors ($\mathcal{DN}$). Instead of the k-NN, here a compact set from the k-NN which best explains the intrinsic relation among the neighbors is considered to constrain the diffusion process. To do so, the author used the dominant set framework by Pavan and Pelillo [35].

A dominant neighbor ($\mathcal{DN}$) is set as a dominant set, say $\mathcal{DS}$, from the k-NN which contains the user provided query q, lets call it $\mathcal{DS}(q)$.

Yang *et al.* in [134], to find a dominant set in the k-NN, $\mathcal{DS}(q)$, initialized (1.12) with the nearest neighbor of q (k-NN(q)). They set, say the initial time is set as $t = 1$, $x_i(1) = 1/K$ if $i \in$ k-NN(q) zero otherwise. After the convergence of (1.12), say to $\mathbf{x}^*$, $\mathcal{DN}(q)$ is set as the support of $\mathbf{x}^*$, $i \in \mathcal{DN}(q)$ if and only if $i \in \sigma(\mathbf{x}^*)$. The edge-weights of the $\mathcal{DN}(q)$ are then kept i.e define locally constrained affinity $\mathcal{L} = (l_{ij})$ defined as $l_{ij} = w(i,j)$, if $(i,j) \in \mathcal{DN}(q)$, and $l_{ij} = 0$ otherwise. Then the diffusion process, setting $\mathcal{V}$ as the affinity $\mathtt{A}$, is performed by the same update rule as in (3.1).

The $\mathcal{DN}$ approach has proven to be more effective than the k-NN approach [133, 134, 138]. The approach, while effective, is rather heuristic in nature and has limitations. The approach initializing (1.12) with the nearest neighbor of q (k-NN(q)) limits the dynamics to the face of the simplex which contains k-NN(q). Moreover, a fixed value of K should be chosen for initializing (1.12), the approach, as it follows k-NN based initializing scheme, is still dependent on K. However number of nearest neighbors K may be different for different objects. As fixing K limits the dynamics to a specified face of the simplex, objects out of k-NN(q) which form a dominant set with q will be loosed. The chosen k-NN may also be fully noisy which might not have a compact structure.

CDS help us develop a locally constrained diffusion process which, as of existing methods, has no problems such as choosing optimal local neighbor size and initializing the dynamics to extract dense neighbor which constrain the diffusion process. The framework alleviates the issues while improving the performance. The two previous propositions provide us with a simple technique to determine dominant-set clusters containing user-selected vertices. Indeed, if S is the user provided query q, by setting

$$\alpha > \lambda_{\max}(\mathtt{A}_{V\setminus S}) \tag{3.2}$$

we are guaranteed that all local solutions of (1.6) will have a support that necessarily contains the user specified object.

Given a query q, we scale the affinity and run the replicator (1.12), say the dynamics converges to $\mathbf{x}^*$. The support of $\mathbf{x}^*$, $\sigma(\mathbf{x}^*)$, is the constrained dominant set which contains the query q, let us call it $\mathcal{CDS}(q)$. The edge-weights of the $\mathcal{CDS}(q)$ are then kept i.e define locally constrained affinity $\mathcal{L} = (l_{ij})$ defined as $l_{ij} = w(i,j)$, if $(i,j) \in \mathcal{CDS}(q)$, and $l_{ij} = 0$ otherwise. The diffusion process is then performed by the same update rule as in (3.1). For the proof of convergence of the update rule we refer the reader to [133].

3.3 Experiments

The performance of the approach is presented in this section. The approach was tested against three well known data sets in the field of retrieval: MPEG7(shape), YALE(faces) and ORL(faces). For all test data sets the number of iterations for the update rule is set to 200. A given pairwise distance $\mathcal{D}$ is transformed to similarity (edge-weight) using a standard Gaussian kernel

$$\mathtt{A}^{\sigma}_{ij} = \mathbb{1}_{i\neq j} exp(-\mathcal{D}/2\sigma^2)$$

where σ is the free scale parameter, and $\mathbb{1}_P = 1$ if P is true, 0 otherwise. $\mathcal{L}$ is then built, from $\mathtt{A}$, using the constrained dominant set framework. The diffusion process is then computed using the update rule (3.1) which resulted in the final learned affinity for ranking.

A similar experimental analysis as of [132] has been conducted. In [132], a generic framework with 72 different variant of diffusion processes was defined which are resulted from three steps: initialization, definition of transition matrix and diffusion process. In our experiment, the update scheme is fixed to (3.1) which has proven to be effective. The four different types of initialization schemes are Affinity Matrix $\mathtt{A}$ (A1) [139], Identity Matrix $\mathbf{I}$ (A2), Transition Matrix $\mathbf{P}$ which is the standard random walk transition matrix (A3) [140] and Transition Matrix $\mathbf{P}_{kNN}$ which is the random walk transition matrix constrained to the k-nearest neighbors (A4) [140]. Including our transition matrix (B6), we have in total 6 different types of transition matrices: $\mathbf{P}$ (B1), Personalized PageRank Transition Matrix $\mathbf{P}_{PPR}$ (B2) [140], $\mathbf{P}_{kNN}$ (B3), Dominant Set Neighbors $\mathbf{P}_{DS}$ [134] (B4), and Affinity Matrix $\mathtt{A}$ (B5)

Metric: The Bull's eye score is used as a measure of retrieval accuracy. It measures the percentage of objects sharing the same class with a query q in the top $\mathcal{R}$ retrieved shapes. Let us say $\mathcal{C}$ is the set of objects in the same class of the query q and $\mathcal{O}$ is the set of top $\mathcal{R}$ retrieved shapes. The Bull's eye score ($\mathcal{B}$) is then computed as $\mathcal{B}=\frac{|\mathcal{O}\cap\mathcal{C}|}{|\mathcal{C}|}$

MPEG7: a well known data set for testing performance of retrieval and shape matching algorithms. It comprises 1400 silhouette shape images of 70 different categories with 20 images in each categories. Articulated Invariant Representation (AIR) [141], best performing shape matching algorithm, is used as the input pairwise distance measure. The retrieval performance is measured fixing $\mathcal{R}$ to 40.

MPEG7	B1	B2	B3	B4	B5	B6(Ours)
A1	99.91	99.93	**100**	**100**	99.88	**100**
A2	99.92	99.93	**100**	**100**	99.88	**100**
A3	99.93	99.94	**100**	**100**	99.88	**100**
A4	99.92	99.94	**100**	**100**	99.88	**100**

Table 3.1: Results on MPEG7 dataset. Bull's eye score for the first 40 elements

Table 3.1 shows bull's eye score on MPEG7 dataset. Observe that we were able to achieve 100% bulle's eye score while alleviating serious problems such as the problem of selecting a reasonable local neighborhood size and initializing the dynamics to find dense neighbors.

The retrieval performance has also been tested by varying the first $\mathcal{R}$ returned objects, the set in which instances of the same category are checked in. For the purpose of this experiment we used the best diffusion variants (B3 and B4 initialized with A2). The performance of the algorithms is shown in Table 3.2. As can be observed, in this case our algorithm, besides giving flexibility, shows a small increment in the results.

$\mathcal{R}$	20	25	30	35	40
B3	94.321	97.871	**98.614**	99.357	**100**
B4	94.296	97.846	**98.614**	99.357	**100**
Ours	**94.354**	**97.896**	**98.614**	**99.360**	**100**

Table 3.2: Results on MPEG7 dataset varying the first $\mathcal{R}$ returned objects

MPEG7 has been used, most frequently, for testing retrieval algorithms. Table 3.3 shows the comparison against different state-of-the-art approaches. The proposed approach and [132] achieve 100% bulle's eye score. However, [132] needs to set an optimal neighborhood size whereas in our approach the number of neighbors to individual items arises intuitively.

Methods	[142]	[143]	[141]	[144]	[134]	[132]	Ours
$\mathcal{B}$	85.40	91.61	93.67	95.96	99.99	**100**	**100**

Table 3.3: Retrieval performance comparison on MPEG7 dataset. **Up:** methods, **Down:** Bull's eye score for the first 40 elements

YALE: [145] a popular benchmark for face clustering which consists of 15 unique people with 11 pictures for each under different conditions: normal, sad, sleepy, center light, right light, etc that include variations of pose, illumination and expression. Similar procedures of [146, 132] were followed to build the distance matrix. Down sample the image, normalize to 0-mean and 1-variance, and compute the Euclidean distance between the vectorized representation. The retrieval performance, measured fixing $\mathcal{R}$ to 15, is demonstrated in table 3.4. Our approach shows a small improvement in the retrieval performance except in one where the affinity itself initializes the diffusion process.

YALE	B1	B2	B3	B4	B5	B6(Ours)
A1	71.74	71.24	**75.59**	75.31	70.25	75.15
A2	71.96	70.69	77.30	76.20	69.92	**77.41**
A3	72.07	70.57	74.93	76.14	70.30	**75.37**
A4	72.23	70.74	77.08	76.10	70.25	**77.36**

Table 3.4: Results on YALE dataset. Bull's eye score for the first 15 elements

Results of the algorithm on YALE data set varying $\mathcal{R}$ is shown in Table 3.5.

ORL: face data set of 40 different persons with 10 grayscale images per person with slight variations of pose, illumination, and expression. Similar procedure as of

$\mathcal{R}$	11	13	15	17	20
B3	71.240	**74.105**	77.303	79.559	80.826
B4	70.854	72.176	76.198	77.741	79.063
Ours	**71.350**	74.050	**77.411**	**80.000**	**81.653**

Table 3.5: Results on YALE dataset varying the first $\mathcal{R}$ returned objects

YALE data set was followed and The retrieval performance is measured fixing $\mathcal{R}$ to 15.

ORL	B1	B2	B3	B4	B5	B6(Ours)
A1	72.75	73.48	**74.25**	73.90	70.58	**74.25**
A2	72.75	73.75	**77.42**	74.82	70.15	**77.42**
A3	73.12	73.75	**75.52**	75.35	71.05	**75.52**
A4	73.12	73.75	**77.32**	75.50	71.40	**77.32**

Table 3.6: Results on ORL dataset. Bull's eye score for the first 15 elements

Results of the algorithm on ORL data set varying $\mathcal{R}$ is shown in Table 3.7.

$\mathcal{R}$	10	13	15	17	20
B3	**70.950**	**75.250**	**77.425**	**79.275**	**80.550**
B4	68.850	72.900	74.825	76.775	77.700
Ours	**70.950**	**75.250**	**77.425**	**79.275**	**80.550**

Table 3.7: Results on ORL dataset varying the first $\mathcal{R}$ returned objects

As can be observed from Tables 3.6 and 3.7, the proposed approach and [132] perform equally on the ORL face dataset.

3.4 Conclusion

In this chapter, we have developed a locally constrained diffusion process which, as of existing methods, has no problems such as choosing optimal local neighbor size and initializing the dynamics to extract dense neighbor which constrain the diffusion process. The framework alleviates the issues while improving the performance. Experimental results on three well known data sets in the field of retrieval demonstrate that the approach compares favorably with state-of-the-art algorithms. Future work will focus on applying the framework on other computer vision problems such as action retrieval and video object segmentation and co-segmentation.

4

Multi-Target Tracking in Multiple Non-Overlapping Cameras using Constrained Dominant Sets

> I don't have a problem with stepped-up surveillance as long as we follow the rule of law.
>
> Bob Beckel

4.1 Introduction

As the need for visual surveillance grow, a large number of cameras have been deployed to cover large and wide areas like airports, shopping malls, city blocks etc.. Since the fields of view of single cameras are limited, in most wide area surveillance scenarios, multiple cameras are required to cover larger areas. Using multiple cameras with overlapping fields of view is costly from both economical and computational aspects. Therefore, camera networks with non-overlapping fields of view are preferred and widely adopted in real world applications.

In the work presented in this chapter, the goal is to track multiple targets and maintain their identities as they move from one camera to the another camera with non-overlapping fields of views. In this context, two problems need to be solved, that is, within-camera data association (or tracking) and across-cameras data association by employing the tracks obtained from within-camera tracking. Although there have been significant progresses in both problems separately, tracking multiple target jointly in both within and across non-overlapping cameras remains a less explored topic. Most approaches, which solve multi-target tracking in multiple non-overlapping cameras [147, 148, 149, 150, 151], assume tracking within each camera has already been performed and try to solve tracking problem only in non-overlapping cameras; the results obtained from such approaches are far from been optimal [150].

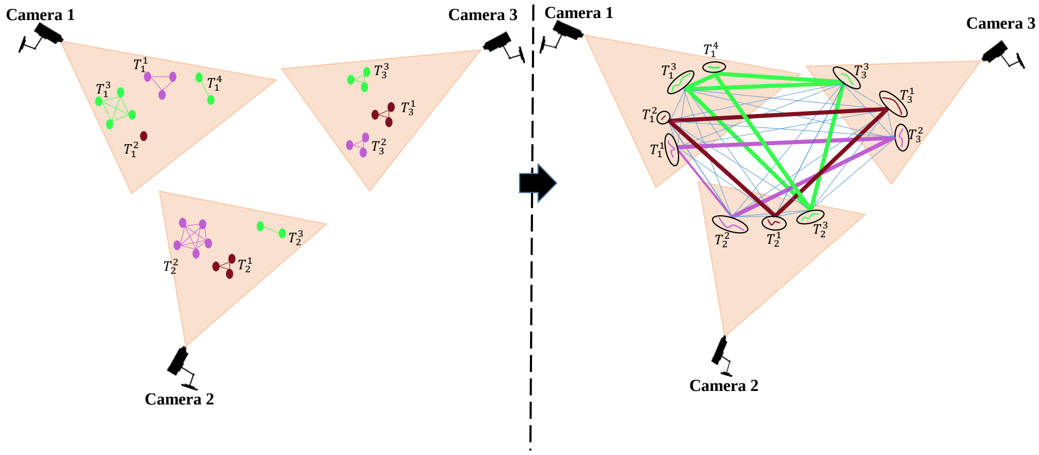


Figure 4.1: A general idea of the proposed framework. (a) First, tracks are determined within each camera, then (b) tracks of the same person from different non-overlapping cameras are associated, solving the across-camera tracking. Nodes in (a) represent tracklets and nodes in (b) represent tracks. The i^{th} track of camera j, T_j^i, is a set of tracklets that form a clique. In (b) each clique in different colors represent tracks of the same person in non-overlapping cameras. Similar color represents the same person. (Best viewed in color)

In this chapter, we propose a hierarchical approach in which we first determine tracks within each camera, (Figure 4.1(a)) by solving data association, and later we associate tracks of the same person in different cameras in a unified approach (Figure 4.1(b)), hence solving the across-camera tracking. Since appearance and motion cues of a target tend to be consistent in a short temporal window in a single camera tracking, solving tracking problem in a hierarchical manner is common: tracklets are generated within short temporal window first and later they are merged to form full tracks (or trajectories) [152, 153, 37]. Often, across-camera tracking is more challenging than solving within-camera tracking due to the fact that appearance of people may exhibit significant differences due to illumination variations and pose changes between cameras.

Therefore, this chapter proposes a unified three-layer framework to solve both within- and across-camera tracking. In the first two layers, we generate tracks within each camera and in the third layer we associate all tracks of the same person across all cameras in a simultaneous fashion.

To best serve our purpose, a constrained dominant sets clustering (CDSC) technique, a parametrized version of standard quadratic optimization, is employed to solve both tracking tasks. The tracking problem is cast as finding constrained dominant sets from a graph. That is, given a constraint set and a graph, CDSC generates cluster (or clique), which forms a compact and coherent set that contains all or part of the constraint set. *Clusters* represent tracklets and tracks in the first and second layers, respectively. The proposed within-camera tracker can robustly handle long-term occlusions, does not change the scale of original problem as it does not remove nodes from the graph during the extraction of compact clusters and is several orders of magnitude faster (close to real time) than existing methods. Also, the proposed across-camera tracking method using CDSC and later followed by refinement step offers several advantages. More specifically, CDSC not only considers the affinity (relationship) between tracks, observed in different cameras, but also takes into account the affinity among tracks from the same camera. As a consequence, the proposed approach not only accurately associates tracks from different cameras but also makes it possible to link multiple short broken tracks obtained during within-camera tracking, which may belong to a single target track. For instance, in Figure 4.1(a) track T_1^3 (third track from camera 1) and T_1^4 (fourth track from camera 1) are tracks of same person which were mistakenly broken from a single track. However, during the third layer, as they are highly similar to tracks in camera 2 (T_2^3) and camera 3 (T_3^3), they form a clique, as shown in Figure 4.1(b). Such across-camera formulation is able to associate these broken tracks with the rest of tracks from different cameras, represented with the green cluster in Figure 4.1(b).

The contributions of this chapter are summarized as follows:

- We formulate multi-target tracking in multiple non-overlapping cameras as finding constrained dominant sets from a graph. We propose a three-layer hierarchical approach, in which we first solve within-camera tracking using the first two layers, and using the third layer we solve the across-camera tracking problem.

- We propose a technique to further speed up our optimization by reducing the search space, that is, instead of running the dynamics over the whole graph, we localize it on the sub graph selected using the dominant distribution, which is much smaller than the original graph.

- Experiments are performed on MOTchallenge DukeMTMCT dataset and MARS dataset, and show improved effectiveness of our method with respect to the state of the art.

The rest of the chapter is organized as follows. In Section 4.2, we review relevant previous works. Overall proposed approach for within- and across-cameras tracking modules is summarized in section 4.3, while sections 4.3.1 and 4.3.2 provide more in details of the two modules. Experimental results are presented in Section 4.4. Finally, section 4.5 concludes the chapter.

4.2 Related Work

Object tracking is a challenging computer vision problem and has been one of the most active research areas for many years. In general, it can be divided in two broad categories: tracking in single and multiple cameras. Single camera object tracking associates object detections across frames in a video sequence, so as to generate the object motion trajectory over time. Multi-camera tracking aims to solve handover problem from one camera view to another and hence establishes target correspondences among different cameras, so as to achieve consistent object labelling across all the camera views. Early multi-camera target tracking research works fall in different categories as follows. Target tracking with partially overlapping camera views has been researched extensively during the last decade [154, 155, 156, 157, 158, 159]. Multi target tracking across multiple cameras with disjoint views has also been researched in [147, 148, 149, 150, 151]. Approaches for overlapping field of views compute spatial proximity of tracks in the overlapping area, while approaches for tracking targets across cameras with disjoint fields of view, leverage appearance cues together with spatio-temporal information.

Almost all early multi-camera research works try to address only across-camera tracking problems, assuming that within-camera tracking results for all cameras are given. Given tracks from each camera, similarity among tracks is computed and target correspondence across cameras is solved, using the assumption that a track of a target in one camera view can match with at most one target track in another camera view. Hungarian algorithm [160] and bipartite graph matching [149] formulations are usually used to solve this problem. Very recently, however, researchers have argued that assumptions of cameras having overlapping fields of view and the availability of intra-camera tracks are unrealistic [150]. Therefore, the work proposed in this chapter addresses the more realistic problem by solving both within- and across-camera tracking in one joint framework.

In the rest of this section, we first review the most recent works for single camera tracking, and then describe the previous related works on multi-camera multi-view tracking.

Single camera target tracking associates target detections across frames in a video sequence in order to generate the target motion trajectory over time. Zamir *et al.* [152] formulate tracking problem as generalized maximum clique problem (GMCP), where the relationships between all detections in a temporal window are considered. In [152], a cost to each clique is assigned and the selected clique maximizes a score function. Nonetheless, the approach is prone to local optima as it uses greedy local neighbourhood search. Deghan *et al.* [153] cast tracking as a generalized maximum multi clique problem (GMMCP) and follow a joint optimization for all the tracks simultaneously. To handle outliers and weak-detections associations they introduce dummy nodes. However, this solution is computationally expensive. In addition, the hard constraint in their optimization makes the approach impractical for large graphs. Tesfaye *et al.* [37] consider all the pairwise relationships between detection responses in a temporal sliding window, which is used as an input to their optimization based on fully-connected edge-weighted graph. They formulate tracking as finding dominant set clusters. Though the dominant set framework is effective in extracting compact sets from a graph [17][35][13] [161] [41], it follows a pill-off strategy to enumerate all possible clusters, that is, at each iteration it removes the found cluster from the graph which results in a change in scale (number of nodes in a graph) of the original problem. In this chapter, we propose a multiple target tracking approach, which in contrast to previous works, does not need additional nodes to handle occlusion nor encounters change in the scale of the problem.

Across-camera tracking aims to establish target correspondences among trajectories from different cameras so as to achieve consistent target labelling across all camera views. It is a challenging problem due to the illumination and pose changes across cameras, or track discontinuities due to the blind areas or miss detections. Existing across-camera tracking methods try to deal with the above problems using appearance cues. The variation in illumination of the appearance cues has been leveraged using different techniques such as Brightness Transfer Functions (BTFs). To handle the appearance change of a target as it moves from one camera to another, the authors in [162] show that all brightness transfer functions from a given camera to another camera lie in a low dimensional subspace, which is learned by employing probabilistic principal component analysis and used for appearance matching. Authors of [163] used an incremental learning method to model the colour variations and [164] proposed a Cumulative Brightness Transfer Function, which is a better use of the available colour information from a very sparse training set. Performance comparison of different variations of Brightness Transfer Functions can be found in [165]. Authors in [166] tried to achieve color consistency using colorimetric principles, where the image analysis system is modelled as an observer and camera-specific transformations are determined, so that images of the same target appear similar to this observer. Obviously, learning Brightness Transfer Functions or color correction models requires large amount of training data and they may not be robust against drastic illumination changes across different cameras. Therefore, recent approaches have combined them with spatio-temporal cue which improve multi-target tracking performance [167, 168, 169, 170, 171, 172]. Chen *et al.* [167] utilized human part configurations for every target track from different cameras to describe the across-camera spatio-temporal constraints for across-camera track association, which is formulated as a multi-class classification problem via Markov Random Fields (MRF). Kuo *et al.* [168] used Multiple Instance Learning (MIL) to learn an appearance model, which effectively combines multiple image descriptors and their corresponding similarity measurements. The proposed appearance model combined with spatio-temporal information improved across-camera track association solving the “target handover” problem across cameras. Gao *et al.* [169] employ tracking results of different trackers and use their spatio-temporal correlation, which help them enforce tracking consistency and establish pairwise correlation among multiple tracking results. Zha *et al.* [170] formulated tracking of multiple interacting targets as a network flow problem, for which the solution can be obtained by the K-shortest paths algorithm. Spatio-temporal relationships among targets are utilized to identify group merge and split events. In [171] spatio-temporal context is used for collecting samples for discriminative appearance learning, where target-specific appearance models are learned to distinguish different people from each other. And the relative appearance context models inter-object appearance similarities for people walking in proximity and helps disambiguate individual appearance matching across cameras.

The problem of target tracking across multiple non-overlapping cameras is also tackled in [4] by extending their previous single camera tracking method [173], where they formulate the tracking task as a graph partitioning problem. Authors in [172], learn across-camera transfer models including both spatio-temporal and appearance cues. While a color transfer method is used to model the changes of color across cameras for learning across-camera appearance transfer models, the spatio-temporal model is learned using an unsupervised topology recovering approach. Recently Chen *et al.* [151] argued that low-level information (appearance model and spatio-temporal information) is unreliable for tracking across non-overlapping cameras, and integrated contextual information such as social grouping behaviour. They formulate tracking using an online-learned Conditional Random Field (CRF), which favours track associations that maintain group consistency. In this chapter, for tracks to be associated, besides their high pairwise similarity (computed using appearance and spatio-temporal cues), their corresponding constrained dominant sets should also be similar.

Another recent popular research topic, video-based person re-identification(ReID) [174, 175, 176, 177, 9, 5, 6, 7, 8], is closely related to across-camera multi-target tracking. Both problems aim to match tracks of the same persons across non-overlapping cameras. However, across-camera tracking aims at 1-1 correspondence association between tracks of different cameras. Compared to most video-based ReID approaches, in which only pairwise similarity between the probes and gallery is exploited, our across-camera tracking framework not only considers the relationship between probes and gallery but it also takes in to account the relationship among tracks in the gallery.

4.3 Overall Approach

In this section, the constrained dominant sets based formulation of within- and across-camera tracking is detailed.

In our formulation, in the first layer, each node in our graph represents a short-tracklet along a temporal window (typically 15 frames). Applying constrained dominant set clustering here aim at determining cliques in this graph, which correspond to tracklets. Likewise, each node in a graph in the second layer represents a tracklet, obtained from the first layer, and CDS is applied here to determine cliques, which correspond to tracks. Finally, in the third layer, nodes in a graph correspond to tracks from different non-overlapping cameras, obtained from the second layer, and CDS is applied to determine cliques, which relate tracks of the same person across non-overlapping cameras.

4.3.1 Within-Camera Tracking

Figure 4.2 shows proposed within-camera tracking framework. First, we divide a video into multiple short segments, each segment contains 15 frames, and generate short-tracklets, where human detection bounding boxes in two consecutive frames with 70% overlap, are connected [153]. Then, short-tracklets from 10 different non-overlapping segments are used as input to our first layer of tracking. Here the nodes are short-tracklets (Figure 4.2, bottom left). Resulting tracklets from the first layer are used as an input to the second layer, that is, a tracklet from the first layer is now represented by a node in the second layer (Figure 4.2, bottom right). In the second layer, tracklets of the same person from different segment are associated forming tracks of a person within a camera.

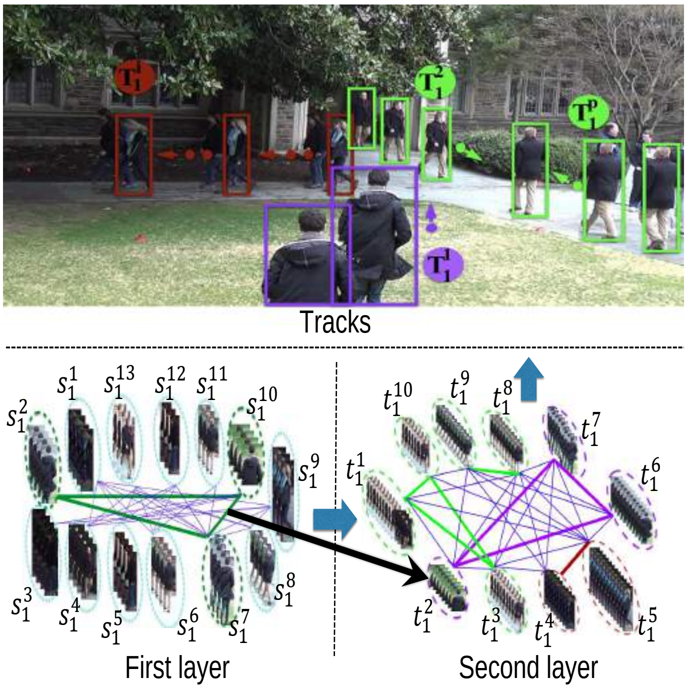

T_1^1
T_1^2
T_1^p
T_1^3
Tracks
s_1^1 s_1^{13} s_1^{12} s_1^{11} s_1^{10}
s_1^2
s_1^9
s_1^3 s_1^4 s_1^5 s_1^6 s_1^7 s_1^8
t_1^{10} t_1^9 t_1^8 t_1^7
t_1^1
t_1^6
t_1^2 t_1^3 t_1^4 t_1^5
First layer
Second layer

Figure 4.2: The figure shows within-camera tracking where short-tracklets from different segments are used as input to our first layer of tracking. The resulting tracklets from the first layer are inputs to the second layer, which determine a tracks for each person. The three dark green short-tracklets (s_1^2, s_1^{10}, s_1^7), shown by dotted ellipse in the first layer, form a cluster resulting in tracklet (t_1^2) in the second layer, as shown with the black arrow. In the second layer, each cluster, shown in purple, green and dark red colors, form tracks of different targets, as can be seen on the top row. tracklets and tracks with the same color indicate same target. The two green cliques (with two tracklets and three tracklets) represent tracks of the person going in and out of the building (tracks T_1^p and T_1^2 respectively)

4.3.1.1 Formulation Using Constrained Dominant Sets

We build an input graph, $G(V, E, w)$, where nodes represent short-tracklet (s_i^j, that is, j^{th} short-tracklet of camera i) in the case of first layer (Figure 4.2, bottom left) and tracklet (t_k^l, that is, l^{th} tracklet of camera k), in the second layer (Figure 4.2, bottom right). The corresponding affinity matrix $\mathtt{A} = \{a_{i,j}\}$, where $a_{i,j} = w(i, j)$ is built. The weight $w(i, j)$ is assigned to each edge, by considering both motion and appearance similarity between the two nodes. Fine-tuned CNN features are used to model the appearance of a node. These features are extracted from the last fully-connected layer of Imagenet pre-trained 50-layers Residual Network (ResNet 50) [178] fine-tuned using the trainval sequence of DukeMTMC dataset. Similar to [152], we employ a global constant velocity model to compute motion similarity between two nodes.

Determining cliques: In our formulation, a clique of graph G represents tracklet(track) in the first (second) layer. Using short-tracklets/tracklets as a constraint set (in eq. 1.6), we enumerate all clusters, using game dynamics, by utilizing intrinsic properties of constrained dominant sets. Note that we do not use peel-off strategy to remove the nodes of found cliques from the graph, this keeps the scale of our problem (number of nodes in a graph) which guarantees that all the found local solutions are the local solutions of the (original) graph. After the extraction of each cluster, the constraint set is changed in such a way to make the extracted cluster unstable under the dynamics. The within-camera tracking starts with all nodes as constraint set. Let us say Γ^i is the i^{th} extracted cluster, Γ^1 is then the first extracted cluster which contains a subset of elements from the whole set. After our first extraction, we change the constraint set to a set $V \backslash \Gamma^1$, hence rendering its associated nodes unstable (making the dynamics not able to select sets of nodes in the interior of associated nodes). The procedure iterates, updating the constraint set at the i^{th} extraction as $V \backslash \bigcup\limits_{l=1}^{i} \Gamma^l$, until the constraint set becomes empty. Since we are not removing the nodes of the graph (after each extraction of a compact set), we may end up with a solution that assigns a node to more than one cluster.

To find the final solution, we use the notion of centrality of constrained dominant sets. The true class of a node j, which is assigned to $\mathtt{K} > 1$ cluster, $\psi = \{\Gamma^1 \dots \Gamma^{\mathtt{K}}\}$, is computed as:

$$\underset{\Gamma^i \in \psi}{\arg\max} \left(|\Gamma^i| * \delta_j^i\right),$$

where the cardinality $|\Gamma^i|$ is the number of nodes that forms the i^{th} cluster and δ_j^i is the membership score of node j obtained when assigned to cluster Γ^i. The normalization using the cardinality is important to avoid any unnatural bias to a smaller set.

Algorithm (4), putting the number of cameras under consideration ($\mathcal{I}$) to 1 and $\mathcal{Q}$ as short-tracklets(tracklets) in the first(second) layer, is used to determine constrained dominant sets which correspond to tracklet(track) in the first (second) layer.

4.3.2 Across-Camera Tracking

4.3.2.1 Graph Representation of Tracks and the Payoff Function

Given tracks (T_i^j, that is, the j^{th} track of camera i) of different cameras from previous step, we build graph $G'(V', E', w')$, where nodes represent tracks and their corresponding affinity matrix $\mathtt{A}$ depicts the similarity between tracks.

Assuming we have $\mathcal{I}$ number of cameras and $\mathtt{A}^{i\times j}$ represents the similarity among tracks of camera i and j, the final track based affinity $\mathtt{A}$, is built as

$$\mathtt{A} = \begin{pmatrix} \mathtt{A}^{1\times 1} & .. & \mathtt{A}^{1\times j} & .. & \mathtt{A}^{1\times \mathcal{I}} \\ . & . & . & . & . \\ \mathtt{A}^{i\times 1} & .. & \mathtt{A}^{i\times j} & . & \mathtt{A}^{i\times \mathcal{I}} \\ . & & . & . & . \\ \mathtt{A}^{\mathcal{I}\times 1} & .. & \mathtt{A}^{\mathcal{I}\times j} & .. & \mathtt{A}^{\mathcal{I}\times \mathcal{I}} \end{pmatrix}.$$

Figure 4.3 shows exemplar graph for across-camera tracking among three cameras. T_j^i represents the i^{th} track of camera j. Black and orange edges, respectively, represent within- and across-camera relations of the tracks. From the affinity $\mathtt{A}$, $\mathtt{A}^{i\times j}$ represents the black edges of camera i if $i = j$, which otherwise represents the across-camera relations using the orange edges.

The colors of the nodes depict the track ID; nodes with similar color represent tracks of the same person. Due to several reasons such as long occlusions, severe pose change of a person, reappearance and others, a person may have more than one track (a *broken track*) within a camera. The green nodes of camera 1 (the second and the p^{th} tracks) typify two *broken tracks* of the same person, due to reappearance as shown in Figure 4.2. The proposed unified approach, as discussed in the next section, is able to deal with such cases.

4.3.2.2 Across-Camera Track Association

In this section, we discuss how we simultaneously solve within- and across-camera tracking. Our framework is naturally able to deal with the errors listed above. A person, represented by the green node from our exemplar graph (Figure 4.3), has two tracks which are difficult to merge during within-camera tracking; however, they belong to clique (or cluster) with tracks in camera 2 and camera 3, since they are highly similar. The algorithm applied to a such across-camera graph is able to cluster all the correct tracks. This helps us linking *broken tracks* of the same person occurring during within-camera track generation stage.

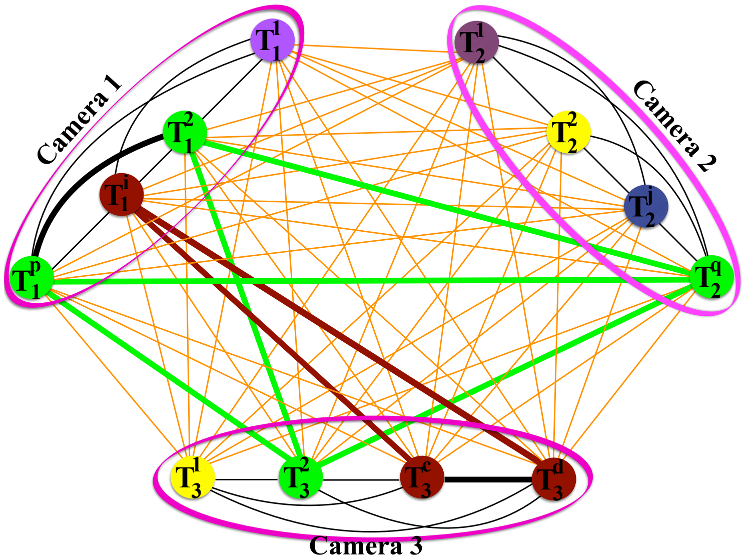


Figure 4.3: Exemplar graph of tracks from three cameras. T_j^i represents the i^{th} track of camera j. Black and colored edges, respectively, represent within- and across-camera relations of tracks. Colours of the nodes depict track IDs, nodes with similar colour represent tracks of the same person, and the thick lines show both within- and across-camera association.

Using the graph with nodes of tracks from a camera as a constraint set, data association for both within- and across-camera are performed simultaneously. Let us assume, in our exemplar graph (Figure 4.3), our constraint set $\mathcal{Q}$ contains nodes of tracks of camera 1, $\mathcal{Q} = \{\ T_1^1, T_1^2, T_1^i, T_1^p\ \}$. $I_{\mathcal{Q}}$ is then $n \times n$ diagonal matrix, whose diagonal elements are set to 1 in correspondence to the vertices contained in all cameras, except camera 1 which takes the value zero. That is, the sub-matrix $I_{\mathcal{Q}}$, that corresponds to $\mathtt{A}^{1\times1}$, will be a zero matrix of size equal to number of tracks of the corresponding camera. Setting $\mathcal{Q}$ as above, we have guarantee that the maximizer of program in eq. (1.6) contains some elements from set $\mathcal{Q}$: i.e., $\mathcal{C}_1^1$=$\{T_1^2, T_1^p, T_2^q, T_3^2\}$ forms a clique which contains set $\{T_1^2, T_1^p\} \in \mathcal{Q}$. This is shown in Figure 4.3, using the thick green edges (which illustrate across-camera track association) and the thick black edge (which typifies the within camera track association). The second set, $\mathcal{C}_1^2$, contains tracks shown with the dark red color, which illustrates the case where within- and across-camera tracks are in one clique. Lastly, the $\mathcal{C}_1^3 = T_1^1$ represents a track of a person that appears only in camera 1. As a general case, C_j^i, represents the i^{th} track set using tracks in camera j as a constraint set and C_j is the set that contains track sets generated using camera j as a constraint set, e.g. $C_1 = \{C_1^1, C_1^2, C_1^3\}$. We iteratively process all the cameras and then apply track refinement step.

Though Algorithm (4) is applicable to within-camera tracking also, here we show the specific case for across-camera track association. Let $\mathcal{T}$ represents the set of

Algorithm 4 Track Association

INPUT: Affinity $\mathtt{A}$, Sets of tracks $\mathcal{T}$ from $\mathcal{I}$ cameras
$C \leftarrow \emptyset$ Initialize the set with empty-set
Initialize $\mathbf{x}$ to the barycenter and i and p to 1

```
while $p \leq \mathcal{I}$ do
    $\mathcal{Q} \leftarrow T_p$, define constraint set
    $\mathcal{X} \leftarrow \mathcal{F}(\mathcal{Q}, \mathtt{A})$
    $C_p^i = \leftarrow \sigma(\mathcal{X}^i)$, compute for all $i = 1 \ldots m$
    $p \leftarrow p + 1$
end while
$C = \bigcup_{p=1}^{\mathcal{I}} C_p$
OUTPUT: $\{C\}$
```

tracks from all the cameras we have and C is the set which contains sets of tracks, as C_p^i, generated using our algorithm. T_p^{ϑ} typifies the ϑ^{th} track from camera p and T_p contains all the tracks in camera p. The function $\mathcal{F}(\mathcal{Q}, \mathtt{A})$ takes as an input a constraint set $\mathcal{Q}$ and the affinity $\mathtt{A}$, and provides as output all the m local solutions $\mathcal{X}^{n \times m}$ of program (1.6) that contain element(s) from the constraint set. This can be accomplished by iteratively finding a local maximizer of equation (program) (1.6) in Δ, e.g. using game dynamics, and then changing the constraint set $\mathcal{Q}$, until all members of the constraint set have been clustered.

4.3.3 Track Refinement

The proposed framework, together with the notion of centrality of constrained dominant sets and the notion of reciprocal neighbours, helps us in refining tracking results using tracks from different cameras as different constraint sets. Let us assume we have $\mathcal{I}$ cameras and $\mathcal{K}^i$ represents the set corresponding to track i, while $\mathcal{K}_p^i$ is the subset of $\mathcal{K}^i$ that corresponds to the p^{th} camera. $\mathcal{M}_p^{li}$ is the membership score assigned to the l^{th} track in the set $\mathcal{C}_p^i$.

We use two constraints during track refinement stage, which helps us refining false positive association.

Constraint-1: *A track can not be found in two different sets generated using same constraint set*, i.e. it must hold that:

$$|\mathcal{K}_p^i| \leq 1$$

Sets that do not satisfy the above inequality should be refined as there is one or more tracks that exist in different sets of tracks collected using the same constraint, i.e. T_p. The corresponding track is removed from all the sets which contain it and is assigned to the right set based on its membership score in each of the sets. Let us say the l^{th} track exists in q different sets, when tracks from camera p are taken as a constraint set, $|\mathcal{K}_p^l| = q$. The right set which contains the track, C_p^r, is chosen as:

$$C_p^r = \underset{C_p^i \in \mathcal{K}_p^l}{\arg\max} \left(|C_p^i| * \mathcal{M}_p^{l^i} \right).$$

where $i = 1, \ldots, |\mathcal{K}_p^l|$. This must be normalized with the cardinality of the set to avoid a bias towards smaller sets.

Constraint-2: *The maximum number of sets that contain track i should be the number of cameras under consideration.* If we consider $\mathcal{I}$ cameras, the cardinality of the set which contains sets with track i, is not larger than $\mathcal{I}$, i.e.:

$$|\mathcal{K}^i| \leq \mathcal{I}.$$

If there are sets that do not satisfy the above condition, the tracks are refined based on the cardinality of the intersection of sets that contain the track, i.e. by enforcing the reciprocal properties of the sets.

If there are sets that do not satisfy the above condition, the tracks are refined based on the cardinality of the intersection of sets that contain the track by enforcing the reciprocal properties of the sets which contain a track. Assume we collect sets of tracks considering tracks from camera q as constraint set and assume a track ϑ in the set C_p^j, $p \neq q$, exists in more than one sets of C_q. The right set, C_q^r, for ϑ considering tracks from camera q as constraint set is chosen as:

$$C_q^r = \underset{C_q^i \in \mathcal{K}_q^\vartheta}{\arg\max} \left(C_q^i \cap C_p^j \right).$$

where $i = 1, \ldots, |\mathcal{K}_q^\vartheta|$.

4.4 Experimental Results

The proposed framework has been evaluated on recently-released large dataset, MOTchallenge DukeMTMC [4, 179, 173]. Even though the main focus of this chapter is on multi-target tracking in multiple non-overlapping cameras, we also perform additional experiments on MARS [11], one of the largest and challenging video-based person re-identification dataset, to show that the proposed across-camera tracking approach can efficiently solve this task also.

DukeMTMC is recently-released dataset to evaluate the performance of multi-target multi-camera tracking systems. It is the largest (to date), fully-annotated and calibrated high resolution 1080p, 60fps dataset, that covers a single outdoor scene from 8 fixed synchronized cameras, the topology of cameras is shown in Fig. 4. The dataset consists of 8 videos of 85 minutes each from the 8 cameras, with 2,700 unique identities (IDs) in more than 2 millions frames in each video containing 0 to 54 people. The video is split in three parts: (1) Trainval (first 50 minutes of the video), which is for training and validation; (2) Test-Hard (next 10 minutes after Trainval sequence); and (3) Test-Easy, which covers the last 25 minutes of the video. Some of the properties which make the dataset more challenging include: huge amount of data to process, it contains 4,159 hand-overs, there are more than 1,800 self-occlusions (with 50% or more overlap), 891 people walking in front of only one camera.

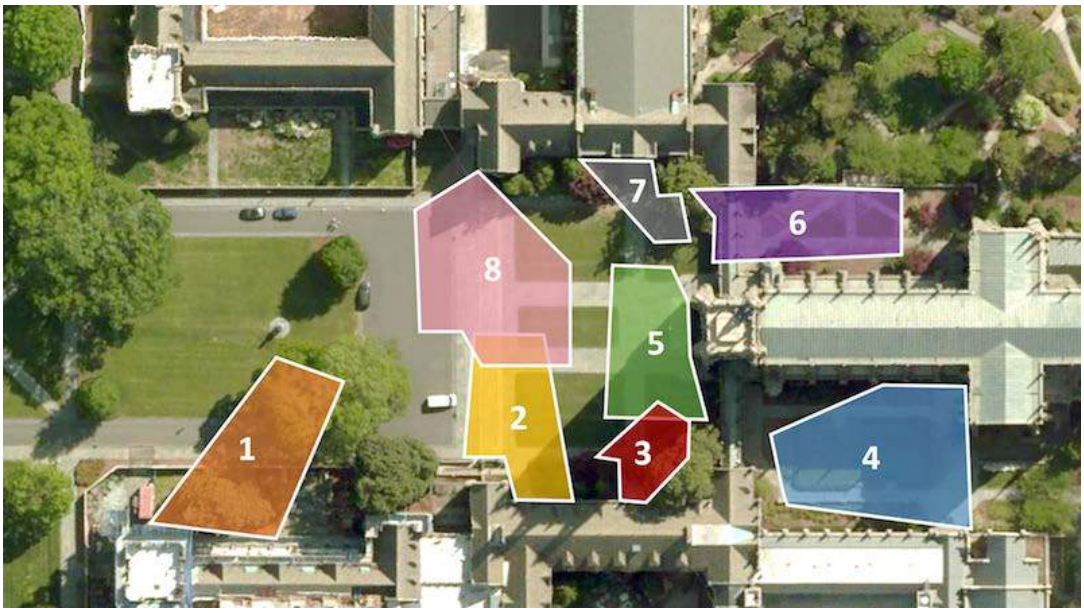


Figure 4.4: Camera topology for DukeMTMC dataset. Detections from the overlapping fields of view are not considered. More specifically, intersection occurred between camera (8 & 2) and camera (5 & 3).

MARS (Motion Analysis and Re-identification Set) is an extension of the Market-1501 dataset [11]. It has been collected from six near-synchronized cameras. It consists of 1,261 different pedestrians, who are captured by at least 2 cameras. The variations in poses, colors and illuminations of pedestrians, as well as the poor image quality, make it very difficult to yield high matching accuracy. Moreover, the dataset contains 3,248 distractors in order to make it more realistic. Deformable Part Model (DPM) [180] and GMMCP tracker [153] were used to automatically generate the tracklets (mostly 25-50 frames long). Since the video and the detections are not available we use the generated tracklets as an input to our framework.

Performance Measures: In addition to the standard Multi-Target Multi-Camera tracking performance measures, we evaluate our framework using additional measures recently proposed in [4]: Identification F-measure (IDF1), Identification Precision (IDP) and Identification Recall (IDR) [4]. The standard performance measures such as CLEAR MOT report the amount of incorrect decisions made by a tracker. Ristani *et al.* [4] argue and demonstrate that some system users may instead be more interested in how well they can determine who is where at all times.

After pointing out that different measures serve different purposes, they proposed the three measures (IDF1, IDP and IDR) which can be applied both within- and across-cameras. These measure tracker's performance not by how often ID switches occur, but by how long the tracker correctly tracks targets.

Identification precision IDP (recall IDR): is the fraction of computed (ground truth) detections that are correctly identified.

Identification F-Score IDF1: is the ratio of correctly identified detections over the average number of ground-truth and computed detections. Since MOTA and its related performance measures under-report across-camera errors [4], we use them for the evaluation of our single camera tracking results.

The performance of the algorithm for re-identification is evaluated employing rank-1 based accuracy and confusion matrix using average precision (AP).

Implementation: In the implementation of our framework, we do not have parameters to tune. The affinity matrix $\mathtt{A}$ adapting kernel trick distance function from [181], is constructed as follows:

$$\mathtt{A}_{i,j} = 1 - \sqrt{\frac{\mathtt{K}(x_i, x_i) + \mathtt{K}(x_j, x_j) - 2 * \mathtt{K}(x_i, x_j)}{2}},$$

where $\mathtt{K}(x_i, x_j)$ is chosen as the Laplacian kernel

$$exp(-\gamma \parallel x_i - x_j \parallel_1).$$

The kernel parameter γ is set as the inverse of the median of pairwise distances.

In our similarity matrix for the final layer of the framework, which is sparse, we use spatio-temporal information based on the time duration and the zone of a person moving from one zone of a camera to other zone of another camera which is learned from the Trainval sequnece of DukeMTMC dataset. The affinity between track i and track j is different from zero , if and only if they have a possibility, based on the direction a person is moving and the spatio-temporal information, to be linked and form a trajectory (across camera tracks of a person). However, this may have a drawback due to *broken tracks* or track of a person who is standing and talking or doing other things in one camera which results in a track that does not meet the spatio-temporal constraints. To deal with this problem, we add, for the across camera track's similarity, a path-based information as used in [14], i.e if a track in camera i and a track in camera j have a probability to form a trajectory, and track j in turn have linkage possibility with a track in camera z, the tracks in camera i and camera z are considered to have a possibility to be linked.

The similarity between two tracks is computed using the Euclidean distance of the max-pooled features. The max-pooled features are computed as the row maximum of the feature vector of individual patch, of the given track, extracted from the last fully-connected layer of Imagenet pre-trained 50-layers Residual Network (ResNet_50) [178], fine-tuned using the Trainval sequence of DukeMTMC dataset. The network is fine-tuned with classification loss on the Trainval sequence, and

	Mthd	MOTA↑	MT↑	ML↓	FP↓	FN↓	IDS↓	IDF1↑	IDP↑	IDR↑
C1	[4]	43.0	24	46	2,713	107,178	39	57.3	91.2	41.8
	Ours	69.9	137	22	5,809	52,152	156	76.9	89.1	67.7
C2	[4]	44.8	133	8	47,919	53,74	60	68.2	69.3	67.1
	Ours	71.5	134	21	8,487	43,912	75	81.2	90.9	73.4
C3	[4]	57.8	52	22	1,438	28,692	16	60.3	78.9	48.8
	Ours	67.4	44	9	2,148	21,125	38	64.6	76.3	56.0
C4	[4]	63.2	36	18	2,209	19,323	7	73.5	88.7	62.8
	Ours	76.8	45	4	2,860	10,689	18	84.7	91.2	79.0
C5	[4]	72.8	107	17	4,464	35,861	54	73.2	83.0	65.4
	Ours	68.9	88	11	9,117	36,933	139	68.3	76.1	61.9
C6	[4]	73.4	142	27	5,279	45,170	55	77.2	87.5	69.1
	Ours	77.0	136	11	4,868	38,611	142	82.7	91.6	75.3
C7	[4]	71.4	69	13	1,395	18,904	23	80.5	93.6	70.6
	Ours	73.8	64	4	1,182	17,411	36	81.8	94.0	72.5
C8	[4]	60.7	102	53	2,730	52,806	46	72.4	92.2	59.6
	Ours	63.4	92	28	4,184	47,565	91	73.0	89.1	61.0
Av	[4]	59.4	665	234	68,147	361,672	**300**	70.1	83.6	60.4
	Ours	**70.9**	**740**	**110**	**38,655**	**268,398**	693	**77.0**	**87.6**	**68.6**

Table 4.1: The results show detailed (for each camera C1 to C8) and average performance (Av) of our and state-of-the-art approach [4] on the Test-Easy sequence of DukeMTMC dataset.

activations of its last fully-connected layer are extracted, L2-normalized and taken as visual features. Cross-view Quadratic Discriminant Analysis (XQDA) [9] is then used for pairwise distance computation between instances. For the experiments on MARS, patch representation is obtained using CNN features used in [11]. The pairwise distances between instances are then computed in XQDA, KISSME [182] and euclidean spaces.

4.4.1 Evaluation on DukeMTMC Dataset:

In Table 4.1 and Table 4.2, we compare quantitative performance of our method with state-of-the-art multi-camera multi-target tracking method on the DukeMTMC dataset. The symbol ↑ means higher scores indicate better performance, while ↓ means lower scores indicate better performance. The quantitative results of the trackers shown in table 4.1 represent the performance on the Test-Easy sequence, while those in table 4.2 show the performance on the Test-Hard sequence. For a fair comparison, we use the same detection responses obtained from MOTchallenge DukeMTMC as the input to our method. In both cases, the reported results of row 'Camera 1' to 'Camera 8' represent the within-camera tracking performances. The last row of the tables represent the average performance over 8 cameras. Both tabular results demonstrate that the proposed approach improves tracking performance

	Mthd	MOTA↑	MT↑	ML↓	FP↓	FN↓	IDS↓	IDF1↑	IDP↑	IDR↑
C1	[4]	37.8	6	34	1,257	78,977	55	52.7	92.5	36.8
	Ours	63.2	65	17	2,886	44,253	408	67.1	83.0	56.4
C2	[4]	47.3	68	12	26526	46898	194	60.6	65.7	56.1
	Ours	54.8	62	16	8,653	54,252	323	63.4	78.8	53.1
C3	[4]	46.7	24	4	288	18182	6	62.7	96.1	46.5
	Ours	68.8	18	2	2,093	8,701	11	81.5	91.1	73.7
C4	[4]	85.3	21	0	1,215	2,073	1	84.3	86.0	82.7
	Ours	75.6	17	0	1,571	3,888	61	82.3	87.1	78.1
C5	[4]	78.3	57	2	1,480	11,568	13	81.9	90.1	75.1
	Ours	78.6	47	2	1,219	11,644	50	82.8	91.5	75.7
C6	[4]	59.4	85	23	5,156	77,031	225	64.1	81.7	52.7
	Ours	53.3	68	36	5,989	88,164	547	53.1	71.2	42.3
C7	[4]	50.8	43	23	2,971	38,912	148	59.6	81.2	47.1
	Ours	50.8	34	20	1,935	39,865	266	60.6	84.7	47.1
C8	[4]	73.0	34	5	706	9735	10	82.4	94.9	72.8
	Ours	70.0	37	6	2,297	9,306	26	81.3	90.3	73.9
Av	[4]	54.6	338	103	39,599	283,376	**652**	64.5	81.2	53.5
	Ours	**59.6**	**348**	**99**	**26,643**	**260,073**	1637	**65.4**	**81.4**	**54.7**

Table 4.2: The results show detailed (for each camera) and average performance of our and state-of-the-art approach [4] on the Test-Hard sequence of DukeMTMC dataset.

	Methods	IDF1↑	IDP↑	IDR↑
Multi-Camera	[4]	56.2	67.0	48.4
	Ours	**60.0**	**68.3**	**53.5**

Table 4.3: Multi-camera performance of our and state-of-the-art approach [4] on the Test-Easy sequence of DukeMTMC dataset.

for both sequences. In the Test-Easy sequence, the performance is improved by 11.5% in MOTA and 7% in IDF1 metrics, while in that of the Test-Hard sequence, our method produces 5% larger average MOTA score than [4], and 1% improvement is achieved in IDF1. Table 4.3 and Table 4.4 respectively present Multi-Camera performance of our and state-of-the-art approach [4] on the Test-Easy and Test-Hard sequence (respectively) of DukeMTMC dataset. We have improved IDF1 for both Test-Easy and Test-Hard sequences by 4% and 3%, respectively.

Figure 4.5 depicts sample qualitative results. Each person is represented by (similar color of) two bounding boxes, which represent the person's position at some specific time, and a track which shows the path s(he) follows. In the first row, all the four targets, even under significant illumination and pose changes, are successfully tracked in four cameras, where they appear. In the second row, target 714 is successfully tracked through three cameras. Observe its significant illumination and pose changes from camera 5 to camera 7. In the third row, targets that move

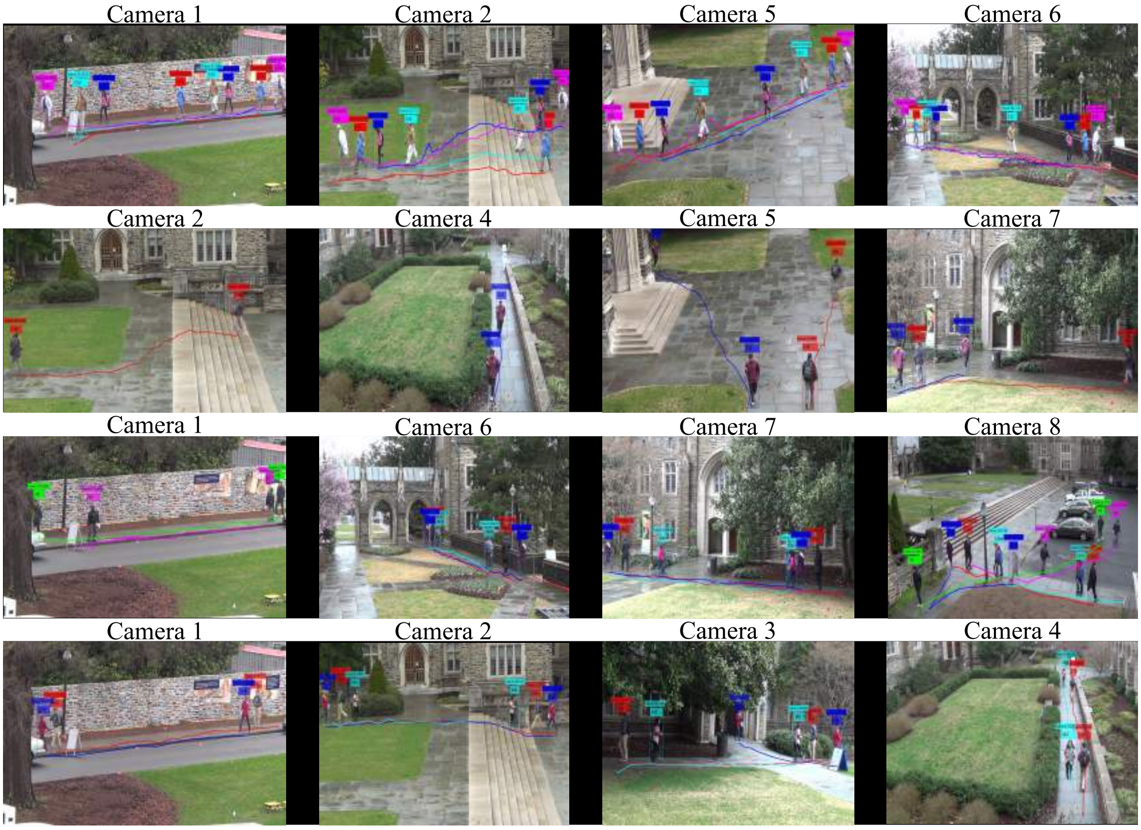


Figure 4.5: Sample qualitative results of the proposed approach on DukeMTMC dataset. Bounding boxes and lines with the same color indicate the same target (Best viewed in color).

through camera 1, target six, seven and eight are tracked. The last row shows tracks of targets that appear in cameras 1 to 4.

4.4.2 Evaluation on MARS Dataset:

In Table 4.5 we compare our results (using the same settings as in [11]) on MARS dataset with the state-of-the-art methods. The proposed approach achieves 3% improvement. In table 4.6 the results show performance of our and state-of-the-art approach [11] in solving the within- (average of the diagonal of the confusion matrix, Fig. 4.6) and across-camera (off-diagonal average) ReID using average precision. Our approach shows up to 10% improvement in the across-camera ReID and up to 6% improvement in the within camera ReID.

To show how much meaningful the notion of centrality of constrained dominant set is, we conduct an experiment on the MARS dataset computing the final ranking using the membership score and pairwise distances. The confusion matrix in Fig. 4.6 shows the detail result of both the within cameras (diagonals) and across cameras

	Methods	IDF1↑	IDP↑	IDR↑
Multi-Camera	[4]	47.3	59.6	39.2
	Ours	**50.9**	**63.2**	**42.6**

Table 4.4: Multi-Camera performance of our and state-of-the-art approach [4] on the Test-Hard sequence of DukeMTMC dataset.

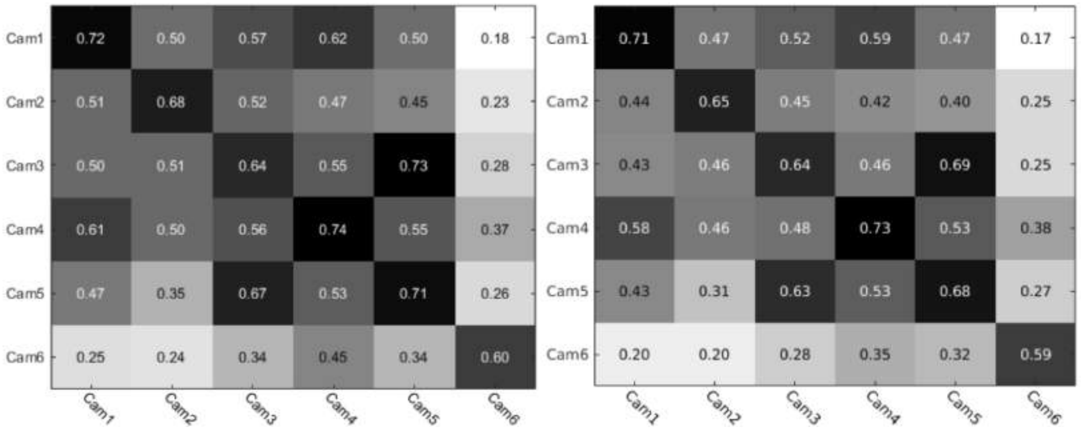


Figure 4.6: The results show the performance of our algorithm on MARS (both using CNN + XQDA) when the final ranking is done using membership score (**left**) and using pairwise euclidean distance (**right**).

Methods	rank 1
HLBP + XQDA	18.60
BCov + XQDA	9.20
LOMO + XQDA	30.70
BoW + KISSME	30.60
SDALF + DVR	4.10
HOG3D + KISSME	2.60
CNN + XQDA [11]	65.30
CNN + KISSME [11]	65.00
Ours	**68.22**

Table 4.5: The table shows the comparison (based on rank-1 accuracy) of our approach with the state-of-the-art approaches: SDALF [5], HLBP [6], BoW [7], BCov [8], LOMO [9], HOG3D [10] on MARS dataset.

(off-diagonals), as we consider tracks from each camera as query. Given a query, a set which contains the query is extracted using the constrained dominant set framework. Note that constraint dominant set comes with the membership scores for all members of the extracted set. We show in Figure 4.6 the results based on the

Feature+Distance	Methods	Within	Across
CNN + Eucl	[11]	0.59	0.28
	Ours (PairwiseDist)	0.59	0.29
	Ours (MembershipS)	**0.60**	**0.29**
CNN + KISSME	[11]	0.61	0.34
	Ours (PairwiseDist)	0.64	0.41
	Ours (MembershipS)	**0.67**	**0.44**
CNN + XQDA	[11]	0.62	0.35
	Ours (PairwiseDist)	0.65	0.42
	Ours (MembershipS)	**0.68**	**0.45**

Table 4.6: The results show performance of our(using pairwise distance and membership score) and state-of-the-art approach [11] in solving within- and across-camera ReID using average precision on MARS dataset using CNN feature and different distance metrics.

final ranking obtained using membership scores (**left**) and using pairwise Euclidean distance between the query and the extracted nodes(**right**). As can be seen from the results in Table 4.6 (average performance) the use of membership score outperforms the pairwise distance approach, since it captures the interrelation among targets.

4.4.3 Computational Time.

Figure 4.7 shows the time taken for each track - from 100 randomly selected (query) tracks - to be associated, with the rest of the (gallery) tracks, running CDSC over the whole graph (CDSC without speedup) and running it on a small portion of the graph using the proposed approach (called FCDSC, CDSC with speedup). The vertical axis is the CPU time in seconds and horizontal axis depicts the track IDs. As it is evident from the plot,our approach takes a fraction of second (red points in Fig. 4.7). Conversely, the CDSC takes up to 8 seconds for some cases (green points in Fig. 4.7). Fig. 5.7 further elaborates how fast our proposed approach is over CDSC, where the vertical axis represents the ratio between CDSC (numerator) and FCDSC (denominator) in terms of CPU time. This ratio ranges from 2000 (the proposed FCDSC 2000x faster than CDSC) to a maximum of above 4500.

4.5 Conclusions

In this chapter we presented a constrained dominant set (CDS) based framework for solving multi-target tracking problem in multiple non-overlapping cameras. The proposed method utilizes a three layers hierarchical approach, where within-camera tracking is solved using first two layers of our framework resulting in tracks for each person, and later in the third layer the proposed across-camera tracker merges

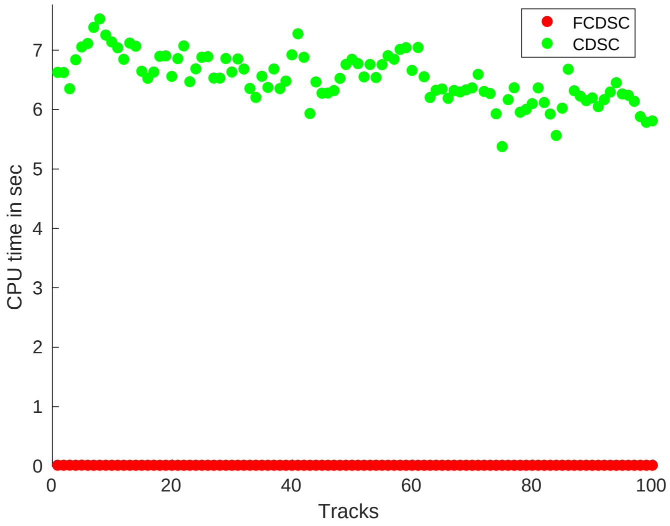


Figure 4.7: CPU time taken for each track association using our proposed fast approach (FCDSC - fast CDSC) and CDSC.

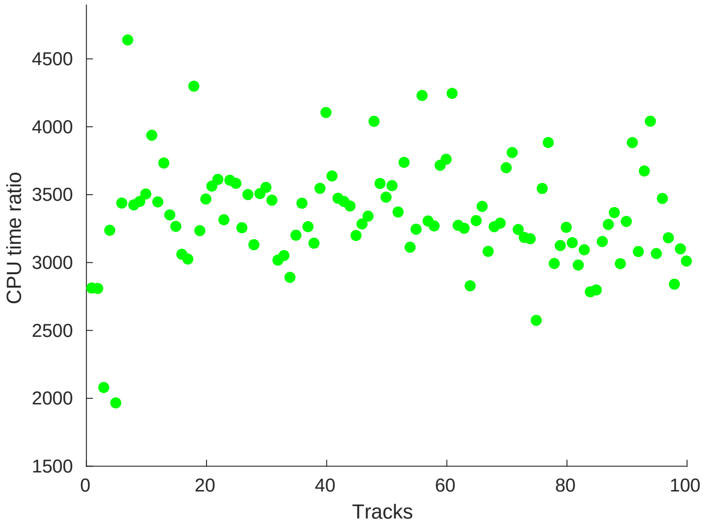


Figure 4.8: The ratio of CPU time taken between CDSC and proposed fast approach (FCDSC), computed as CPU time for CDSC/CPU time for FCDSC.

tracks of the same person across different cameras. Experiments on a challenging real-world dataset (MOTchallenge DukeMTMCT) validate the effectivness of our model.

We further perform additional experiments to show effectiveness of the proposed across-camera tracking on one of the largest video-based people re-identification datasets (MARS). Here each query is treated as a constraint set and its corresponding members in the resulting constrained dominant set cluster are considered as possible candidate matches to their corresponding query.

There are few directions we would like to pursue in our future research. In this work, we consider a static cameras with known topology but it is important for the approach to be able to handle challenging scenario, were some views are from cameras with ego motion (e.g., PTZ cameras or taken from mobile devices) with unknown camera topology. Moreover, here we consider features from static images, however, we believe video features which can be extracted using LSTM could boost the performance and help us extend the method to handle challenging scenarios.

5

Large-scale Image Geo-Localization Using Dominant Sets

> Find your place on the planet. Dig in, and take responsibility from there.
>
> Gary Snyder

5.1 Introduction

Image geo-localization, the problem of determining the location of an image using just the visual information, is remarkably difficult. Nonetheless, images often contain useful visual and contextual informative cues which allow us to determine the location of an image with variable confidence. The foremost of these cues are landmarks, architectural details, building textures and colors, in addition to road markings and surrounding vegetation.

Recently, the geo-localization through image-matching approach was proposed in [183, 3]. In [183], the authors find the first nearest neighbor (NN) for each local feature in the query image, prune outliers and use a heuristic voting scheme for selecting the matched reference image. The follow-up work [3] relaxes the restriction of using only the first NN and proposed Generalized Minimum Clique Problem (GMCP) formulation for solving this problem. However, GMCP formulation can only handle a fixed number of nearest neighbors for each query feature. The authors used 5 NN, and found that increasing the number of NN drops the performance. Additionally, the GMCP formulation selects exactly one NN per query feature. This makes the optimization sensitive to outliers, since it is possible that none of the 5 NN is correct. Once the best NN is selected for each query feature, a very simple voting scheme is used to select the best match. Effectively, each query feature votes for a single reference image, from which the NN was selected for that particular query feature. This often results in identical number of votes for several images from the reference set. Then, both [183, 3] proceed with randomly selecting one reference image as the correct match to infer GPS location of the query image. Furthermore, the GMCP is a binary-variable NP-hard problem, and due to the high computational cost, only a single local minima solution is computed in [3].

In this chapter, we propose an approach to image geo-localization by robustly finding a matching reference image to a given query image. This is done by finding correspondences between local features of the query and reference images. We first introduce automatic NN selection into our framework, by exploiting the discriminative power of each NN feature and employing different number of NN for each query feature. That is, if the distance between query and reference NNs is similar, then we use several NNs since they are ambiguous, and the optimization is afforded with more choices to select the correct match. On the other hand, if a query feature has very few low-distance reference NNs, then we use fewer NNs to save the computation cost. Thus, for some cases we use fewer NNs, while for others we use more requiring on the average approximately the same amount of computation power, but improving the performance, nonetheless. This also bypasses the manual tuning of the number of NNs to be considered, which can vary between datasets and is not straightforward.

Our approach to image geo-localization is based on *Dominant Set clustering* (DSC) - a well-known generalization of maximal clique problem to edge-weighted graphs- where the goal is to extract the most compact and coherent set. It's intriguing connections to evolutionary game theory allow us to use efficient game dynamics, such as replicator dynamics and infection-immunization dynamics (InImDyn). InImDyn has been shown to have a linear time/space complexity for solving standard quadratic programs (StQPs), programs which deal with finding the extrema of a quadratic polynomial over the standard simplex [184, 49]. The proposed approach is on average 200 times faster and yields an improvement of 20% in the accuracy of geo-localization compared to [183, 3]. This is made possible, in addition to the dynamics, through the flexibility inherent in DSC, which unlike the GMCP formulation avoids any hard constraints on memberships. This naturally handles outliers, since their membership score is lower compared to inliers present in the cluster. Furthermore, our solution uses a linear relaxation to the binary variables, which in the absence of hard constraints is solved through an iterative algorithm resulting in massive speed up.

Since the dynamics and linear relaxation of binary variables allow our method to be extremely fast, we run it multiple times to obtain several local maxima as solutions. Next, we use a query-based variation of DSC to combine those solutions to obtain a final robust solution. The query-based DSC uses the soft-constraint that the query, or a group of queries, must always become part of the cluster, thus ensuring their membership in the solution. We use a fusion of several global features to compute the cost between query and reference images selected from the previous step. The members of the cluster from the reference set are used to find the geo-location of the query image. Note that, the GPS location of matching reference image is also used as a cost in addition to visual features to ensure both visual similarity and geographical proximity.

GPS tagged reference image databases collected from user uploaded images on Flickr have been typically used for the geo-localization task. The query images in our experiments have been collected from Flickr, however, the reference images were collected from Google Street View. The data collected through Flickr and Google Street View differ in several important aspects: the images downloaded from Flickr are often redundant and repetitive, where images of a particular building, landmark or street are captured multiple times by different users. Typically, popular or tourist spots have relatively more images in testing and reference sets compared to less interesting parts of the urban environment. An important constraint during evaluation is that the distribution of testing images should be similar to that of reference images. On the contrary, Google Street View reference data used in this chapter contains only a single sample of each location of the city. However, Street View does provide spherical 360° panoramic views, , approximately 12 meters apart, of most streets and roads. Thus, the images are uniformly distributed over different locations, independent of their popularity. The comprehensiveness of the data ensures that a correct match exists; nonetheless, at the same time, the sparsity or uniform distribution of the data makes geo-localization difficult, since every location is captured in only few of the reference images. The difficulty is compounded by the distorted, low-quality nature of the images as well.

The main contributions of this chapter are summarized as follows:

- We present a robust and computationally efficient approach for the problem of large-scale image geo-localization by locating images in a structured database of city-wide reference images with known GPS coordinates.

- We formulate geo-localization problem in terms of a more generalized form of dominant sets framework which incorporates weights from the nodes in addition to edges.

- We take a two-step approach to solve the problem. The first step uses local features to find putative set of reference images (and is therefore faster), whereas the second step uses global features and a constrained variation of dominant sets to refine results from the first step, thereby, significantly boosting the geo-localization performance.

- We have collected new and more challenging high resolution reference dataset (***WorldCities*** dataset) of 300K Google street view images.

The rest of the chapter is structured as follows. We present literature relevant to our problem in Sec. 5.2, followed by technical details of the proposed approach in Sec. 5.3, while constrained dominant set based post processing step is discussed in Sec. 5.4. This is followed by dataset description in section 5.5.1. Finally, we provide results of our extensive evaluation in Sec. 5.5 and conclude in Sec. 5.6.

5.2 Related Work

The computer vision literature on the problem of geo-localization can be divided into three categories depending on the scale of the datasets used: landmarks or buildings [185, 186, 187, 188], city-scale including streetview data [189], and worldwide [190, 191, 192]. Landmark recognition is typically formulated as an image retrieval problem [185, 187, 188, 193, 194]. For geo-localization of landmarks and buildings, Crandall *et al.* [195] perform structural analysis in the form of spatial distribution of millions of geo-tagged photos. This is used in conjunction with visual and meta data from images to geo-locate them. The datasets for this category contain many images near prominent landmarks or images. Therefore, in many works [185, 187], similar looking images belonging to same landmarks are often grouped before geo-localization is undertaken.

For citywide geo-localization of query images, Zamir and Shah [183] performed matching using SIFT features, where each feature votes for a reference image. The vote map is then smoothed geo-spatially and the peak in the vote map is selected as the location of the query image. They also compute 'confidence of localization' using the Kurtosis measure as it quantifies the peakiness of vote map distribution. The extension of this work in [3] formulates the geo-localization as a clique-finding problem where the authors relax the constraint of using only one nearest neighbor per query feature. The best match for each query feature is then solved using Generalized Minimum Clique Graphs, so that a simultaneous solution is obtained for all query features in contrast to their previous work [183]. In similar vein, Schindler *et al.* [196] used a dataset of 30,000 images corresponding to 20 kilometers of streetside data captured through a vehicle using vocabulary tree. Sattler *et al.* [197] investigated ways to explicitly handle geometric bursts by analyzing the geometric relations between the different database images retrieved by a query. Arandjelovic´ *et al.* [198] developed a convolutional neural network architecture for place recognition that aggregates mid-level (conv5) convolutional features extracted from the entire image into a compact single vector representation amenable to efficient indexing. Torii *et al.* [199] exploited repetitive structure for visual place recognition, by robustly detecting repeated image structures and a simple modification of weights in the bag-of-visual-word model. Zeisl *et al.* [200] proposed a voting-based pose estimation strategy that exhibits linear complexity in the number of matches and thus facilitates to consider much more matches.

For geo-localization at the global scale, Hays and Efros [190] were the first to extract coarse geographical location of query images using Flickr collected across the world. Recently, Weyand *et al.* [192] pose the problem of geo-locating images in terms of classification by subdividing the surface of the earth into thousands of multi-scale geographic cells, and train a deep network using millions of geo-tagged images. In the regions where the coverage of photos is dense, structure-from-motion reconstruction is used for matching query images [201, 202, 203, 204]. Since the difficulty of the problem increases as we move from landmarks to city-scale and finally to worldwide, the performance also drops. There are many interesting variations to the geo-localization problem as well. Sequential information such as chronological order of photos was used by [205] to geo-locate photos. Similarly, there are methods to find trajectory of a moving camera by geo-locating video frames using Bayesian Smoothing [206] or geometric constraints [207]. Chen and Grauman [208] present Hidden Markov Model approach to match sets of images with sets in the database for location estimation. Lin *et al.* [209] use aerial imagery in conjunction with ground images for geo-localization. Others [210, 211] approach the problem by matching ground images against a database of aerial images. Jacob *et al.* [212] geo-localize a webcam by correlating its video-stream with satellite weather maps over the same time period. Skyline2GPS [213] uses street view data and segments the skyline in an image captured by an upward-facing camera by matching it against a 3D model of the city.

Feature discriminativity has been explored by [214], who use local density of descriptor space as a measure of descriptor distinctiveness, i.e. descriptors which are in a densely populated region of the descriptor space are deemed to be less distinctive. Similarly, Bergamo *et al.* [215] leverage Structure from Motion to learn discriminative codebooks for recognition of landmarks. In contrast, Cao and Snavely [216] build a graph over the image database, and learn local discriminative models over the graph which are used for ranking database images according to the query. Similarly, Gronat *et al.* [217] train discriminative classifier for each landmark and calibrate them afterwards using statistical significance measures. Instead of exploiting discriminativity, some works use similarity of features to detect repetitive structures to find locations of images. For instance, Torii *et al.* [218] consider a similar idea and find repetitive patterns among features to place recognition. Similarly, Hao *et al.* [219] incorporate geometry between low-level features, termed 'visual phrases', to improve the performance on landmark recognition.

Our work is situated in the middle category, where given a database of images from a city or a group of cities, we aim to find the location where a test image was taken from. Unlike landmark recognition methods, the query image may or may not contain landmarks or prominent buildings. Similarly, in contrast to methods employing reference images from around the globe, the street view data exclusively contains man-made structures and rarely natural scenes like mountains, waterfalls or beaches.

5.3 Image Matching Based Geo-Localization

Fig. 5.1 depicts the overview of the proposed approach. Given a set of reference images, e.g., taken from Google Street View, we extract local features (hereinafter referred as *reference features*) using SIFT from each reference image. We then organize them in a k-means tree [220].

First, for each local feature extracted from the query image (hereinafter referred

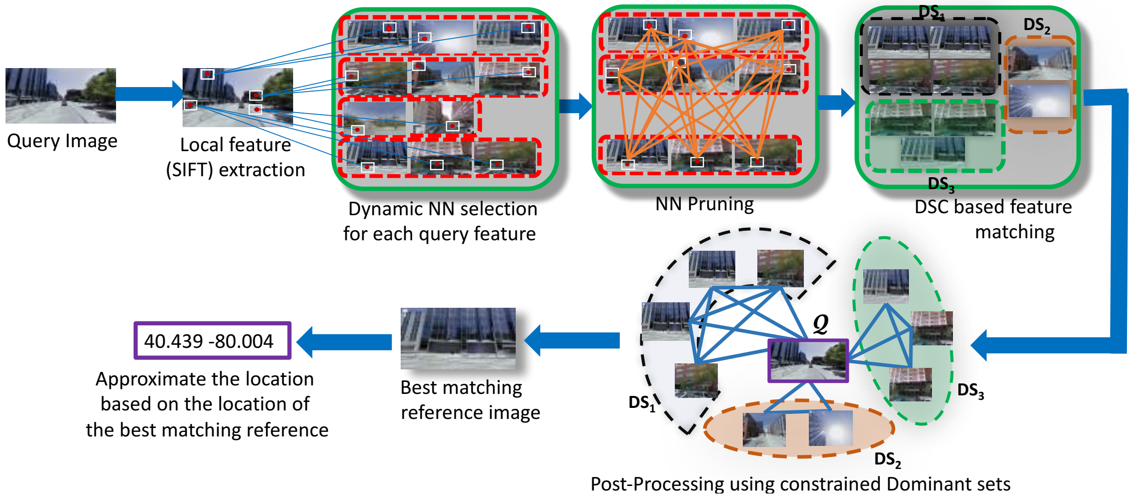

Query Image
Local feature (SIFT) extraction
Dynamic NN selection for each query feature
NN Pruning
DS1
DS2
DS3
DSC based feature matching
Q
DS1
DS3
DS2
Post-Processing using constrained Dominant sets
Best matching reference image
40.439 -80.004
Approximate the location based on the location of the best matching reference

Figure 5.1: Overview of the proposed method.

as *query feature*), we dynamically collect nearest neighbors based on how distinctive the two nearest neighbors are relative to their corresponding query feature. Then, we remove query features, along with their corresponding reference features, if the ratio of the distance between the first and the last nearest neighbor is too large (Sec. 5.3.1). If so, it means that the query feature is not very informative for geo-localization, and is not worth keeping for further processing. In the next step, we formalize the problem of finding matching reference features to query features as a DSC (Dominant Set Clustering) problem, that is, selecting reference features which form a coherent and most compact set (Sec. 5.3.2). Finally, we employ constrained dominant-set-based post-processing step to choose the best matching reference image and use the location of the strongest match as an estimation of the location of the query image (Sec. 5.4).

5.3.1 Dynamic Nearest Neighbor Selection and Query Feature Pruning

For each of N query features detected in the query image, we collect their corresponding nearest neighbors (NN). Let v^i_m be the m^{th} nearest neighbor of i^{th} query feature q^i, and $m \in \mathbb{N} : 1 \leq m \leq \mid NN^i \mid$ and $i \in \mathbb{N} : 1 \leq i \leq N$, where $\mid \cdot \mid$ represents the set cardinality and NN^i is the set of NNs of the i^{th} query feature. In this work, we propose a dynamic NNs selection technique based on how distinctive two consecutive elements are in a ranked list of neighbors for a given query feature, and employ different number of nearest neighbors for each query feature.

As shown in Algorithm (5), we add the $(m + 1)^{th}$ NN of the i^{th} query feature, v^i_{m+1} , if the ratio of the two consecutive NN is greater than θ, otherwise we stop. In other words, in the case of a query feature which is not very discriminative, i.e.,

Algorithm 5 Dynamic Nearest Neighbor Selection for i^{th} query feature (q^i)

Input: the i^{th} query feature (q^i) and all its nearest neighbors extracted from K-means tree $\{v_1^i, v_2^i ....v_{|NN^i|}^i\}$

Output: Selected Nearest Neighbors for the i^{th} query feature ($\mathbb{V}^i$)

```
procedure DYNAMIC NN SELECTION()
    Initialize $\mathbb{V}^i = \{v_1^i\}$ and m=1
    while $m < |NN^i| - 1$ do
        if $\frac{\|\xi(q^i)-\xi(v_m^i)\|}{\|\xi(q^i)-\xi(v_{m+1}^i)\|} > \theta$ then
            $\mathbb{V}^i = \mathbb{V}^i \cup v_{m+1}^i$          ▷ If so, add $v_{m+1}^i$ to our solution
            $m = m + 1$          ▷ Go to the next neighbor
        else
            Break          ▷ If not, stop adding and exit
        end if
    end while
end procedure
```

most of its NNs are very similar to each other, the algorithm continues adding NNs until a distinctive one is found. In this way, less discriminative query features will use more NNs to account for their ambiguity, whereas more discriminative query features will be compared with fewer NNs.

Query Feature Pruning. For the geo-localization task, most of the query features that are detected from moving objects (such as cars) or the ground plane, do not convey any useful information. If such features are coarsely identified and removed prior to performing the feature matching, that will reduce the clutter and computation cost for the remaining features. Towards this end, we use the following pruning constraint which takes into consideration distinctiveness of the *first* and the *last* NN. In particular, if $\|\xi\left(q^i\right) - \xi\left(v_1^i\right)\| \,/\, \left\|\xi\left(q^i\right) - \xi\left(v_{|NN^i|}^i\right)\right\| > \beta$, where $\xi(\cdot)$ represents an operator which returns the local descriptor of the argument node, then q^i is removed, otherwise it is retained. That is, if the *first* NN is similar to the *last* NN (less than β), then the corresponding query feature along with its NNs are pruned since it is expected to be uninformative.

We empirically set both thresholds, θ and β, in Algorithm (5) and pruning step, respectively, to 0.7 and keep them fixed for all tests.

5.3.2 Multiple Feature Matching Using Dominant Sets

5.3.2.1 Similarity Function for Multiple Feature Matching

In our framework, the set of nodes, V, represents all NNs for each query feature which survives the pruning step. The edge set is defined as $E = \{(v_m^i, v_n^j) \mid i \neq j\}$, which signifies that all the nodes in G are connected as long as their corresponding query features are not the same. The edge weight, $\varpi : E \longrightarrow \mathbb{R}^+$ is defined as $\varpi(v_m^i, v_n^j) = exp(-\|\psi(v_m^i) - \psi(v_n^j)\|^2/2\gamma^2)$, where $\psi(\cdot)$ represents an operator which returns the global descriptor of the parent image of the argument node and γ is empirically set to 2^7 . The edge weight, $\varpi(v_m^i, v_n^j)$, represents a similarity between nodes v_m^i and v_n^j in terms of the global features of their parent images. The node score, $\zeta : V \longrightarrow \mathbb{R}^+$, is defined as $\zeta(v_m^i) = exp(-\|\xi(q^i) - \xi(v_m^i)\|^2/2\gamma^2)$. The node score shows how similar the node v_m^i is with its corresponding query feature in terms of its local features.

Matching the query features to the reference features requires identifying the correct NNs from the graph G which maximize the weight, that is, selecting a node (NN) which forms a coherent (highly compact) set in terms of both global and local feature similarities.

Affinity matrix $\mathtt{B}$ represents the global similarity among reference images, which is built using GPS locations as a global feature and a node score $\mathbf{b}$ which shows how similar the reference image is with its corresponding query feature in terms of their local features. We formulate the following optimization problem, a more general form of the dominant set formulation:

$$\begin{aligned} \text{maximize} \quad & f(\mathbf{x}) = \mathbf{x}^\top \mathtt{B}\mathbf{x} + \mathbf{b}^\top \mathbf{x}, \\ \text{subject to} \quad & \mathbf{x} \in \Delta. \end{aligned} \tag{5.1}$$

The affinity $\mathtt{B}$ and the score $\mathbf{b}$ are computed as follows:

$$\mathtt{B}(v_m^i, v_n^j) = \begin{cases} \varpi(v_m^i, v_n^j), & \text{for } i \neq j, \\ 0, & \text{otherwise,} \end{cases} \tag{5.2}$$

$$\mathbf{b}(v_m^i) = \zeta(v_m^i). \tag{5.3}$$

General quadratic optimization problems, like (5.1), are known to be NP-hard [221]. However, in relaxed form, standard quadratic optimization problems can be solved using many algorithms which make full systematic use of data constellations. Off-the-shelf procedures find a local solution of (5.1), by following the paths of feasible points provided by game dynamics based on evolutionary game theory.

Interestingly, the general quadratic optimization problem can be rewritten in the form of standard quadratic problem. A principled way to do that is to follow the result presented in [222], which shows that maximizing the general quadratic problem over the simplex can be homogenized as follows. Maximizing $\mathbf{x}^\top \mathtt{B}\mathbf{x} + \mathbf{b}^\top \mathbf{x}$, subject to $\mathbf{x} \in \Delta$ is equivalent to maximizing $\mathbf{x}^\top \mathtt{A}\mathbf{x}$, subject to $\mathbf{x} \in \Delta$, where $\mathtt{A} = \mathtt{B} + \mathbf{e}\mathbf{b}^\top + \mathbf{b}\mathbf{e}^\top$ and $\mathbf{e} = \sum_{i=1}^{n} \mathbf{e}_i = [1, 1, ...1]$, where $\mathbf{e}_i$ denotes the i^{th} standard basis vector in $\mathbb{R}^n$. This can be easily proved by noting that the problem is solved in the simplex.

$$\mathbf{x}^\top \mathtt{B}\mathbf{x} + 2 * \mathbf{b}^\top \mathbf{x} = \mathbf{x}^\top \mathtt{B}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{b},$$

$$
\begin{aligned}
&= \mathbf{x}^\top \mathtt{B}\mathbf{x} + \mathbf{x}^\top \mathbf{e}\mathbf{b}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{b}\mathbf{e}^\top \mathbf{x}, \\
&= \mathbf{x}^\top (\mathtt{B} + \mathbf{e}\mathbf{b}^\top + \mathbf{b}\mathbf{e}^\top)\mathbf{x}, \\
&= \mathbf{x}^\top \mathtt{A}\mathbf{x}.
\end{aligned}
$$

As can be inferred from the formulation of our multiple NN feature matching problem, DSC can be essentially used for solving the optimization problem. Therefore, by solving DSC for the graph G, the optimal solution that has the most agreement in terms of local and global features will be found.

5.4 Post Processing Using Constrained Dominant Sets

Up to now, we devised a method to collect matching reference features corresponding to our query features. The next task is to select one reference image, based on feature matching between query and reference features, which best matches the query image. To do so, most of the previous methods follow a simple voting scheme, that is, the matched reference image with the highest vote is considered as the best match.

This approach has two important shortcomings. First, if there are equal votes for two or more reference images (which happens quite often), it will randomly select one, which makes it prone to outliers. Second, a simple voting scheme does not consider the similarity between the query image and the candidate reference images at the global level, but simply accounts for local matching of features. Therefore, we deal with this issue by proposing a post processing step, which considers the comparison of the global features of the images and employs *constrained dominant set*, a framework that generalizes the dominant sets formulation and its variant [35, 43].

In our post processing step, the user-selected query and the matched reference images are related using their *global* features and a unique local solution is then searched which contains the union of all the dominant sets containing the query. As customary, the resulting solution will be globally consistent both with the reference images and the query image and due to the notion of *centrality*, each element in the resulting solution will have a membership score, which depicts how similar a given reference image is with the rest of the images in the cluster. So, by virtue of this property of constrained dominant sets, we will select the image with the highest membership score as the final best matching reference image and approximate the location of the query with the GPS location of the best matched reference image.

In this section, we review the basic definitions and properties of constrained dominant sets, as introduced in [53]. Given a user specified query, $\mathcal{Q} \subseteq \hat{V}$, we define the graph $\hat{G} = (\hat{V}, \hat{E}, \hat{w})$, where the edges are defined as $\hat{E} = \{(i,j) | i \neq j, \{i,j\} \in \mathcal{DS}_n \vee (i \in \mathcal{Q} \vee j \in \mathcal{Q})\}$, i.e., all the nodes are connected as long as they do not belong to different local maximizers, $\mathcal{DS}_n$, which represents the n^{th} extracted dominant set. The set of nodes $\hat{V}$ represents all matched reference images (local maximizers) and query image, $\mathcal{Q}$. The edge weight $\hat{w} : \hat{E} \to \mathbb{R}^+$ is defined as:

$$\hat{w}(i,j) = \begin{cases} \rho(i,j), & \text{for } i \neq j,\, i \in \mathcal{Q} \vee j \in \mathcal{Q}, \\ \mathtt{A}_n(i,j), & \text{for } i \neq j,\, \{i,j\} \in \mathcal{DS}_n, \\ 0, & \text{otherwise} \end{cases}$$

where $\rho(i,j)$ is an operator which returns the global similarity of two given images i and j, that is, $\rho(i,j) = exp(-\|\psi(i) - \psi(j)\|^2/2\gamma^2)$, $\mathtt{A}_n$ represents a sub-matrix of $\mathtt{A}$, which contains only members of $\mathcal{DS}_n$, normalized by its maximum value and finally $\mathtt{A}_n(i,j)$ returns the normalized affinity between the i^{th} and j^{th} members of $\mathcal{DS}_n$. The graph $\hat{G}$ can be represented by an $n \times n$ affinity matrix $\hat{\mathtt{A}} = (\hat{w}(i,j))$, where n is the number of nodes in the graph.

Given a parameter $\alpha > 0$, let us define the following parameterized variant of program (1.4):

$$\begin{array}{ll} \text{maximize} & f_{\mathcal{Q}}^{\alpha}(\mathbf{x}) = \mathbf{x}^\top(\hat{\mathtt{A}} - \alpha\hat{I}_{\mathcal{Q}})\mathbf{x}, \\ \text{subject to} & \mathbf{x} \in \Delta, \end{array} \tag{5.4}$$

where $\hat{I}_{\mathcal{Q}}$ is the $n \times n$ diagonal matrix whose diagonal elements are set to 1 in correspondence to the vertices contained in $\hat{V} \setminus \mathcal{Q}$ (a set $\hat{V}$ without the element $\mathcal{Q}$) and to zero otherwise.

Let $\mathcal{Q} \subseteq \hat{V}$, with $\mathcal{Q} \neq \emptyset$ and let $\alpha > \lambda_{\max}(\hat{\mathtt{A}}_{\hat{V}\setminus\mathcal{Q}})$, where $\lambda_{\max}(\hat{\mathtt{A}}_{\hat{V}\setminus\mathcal{Q}})$ is the largest eigenvalue of the principal submatrix of $\hat{\mathtt{A}}$ indexed by the elements of $\hat{V} \setminus \mathcal{Q}$. If $\mathbf{x}$ is a local maximizer of $f_{\mathcal{Q}}^{\alpha}$ in Δ, then $\sigma(\mathbf{x}) \cap \mathcal{Q} \neq \emptyset$. A complete proof can be found in 1.3.

The above result provides us with a simple technique to determine dominant-set clusters containing user-specified query vertices. Indeed, if $\mathcal{Q}$ is a vertex selected by the user, by setting

$$\alpha > \lambda_{\max}(\hat{\mathtt{A}}_{\hat{V}\setminus\mathcal{Q}}), \tag{5.5}$$

we are guaranteed that all local solutions of (5.4) will have a support that necessarily contains elements of $\mathcal{Q}$.

The performance of our post processing may vary dramatically among queries and we do not know in advance which global feature plays a major role. Figs. 5.2 and 5.3 show illustrative cases of different global features. In the case of Fig. 5.2, HSV color histograms and GIST provide a wrong match (dark red node for HSV and dark green node for GIST), while both CNN-based global features (CNN6 and CNN7) matched it to the right reference image (yellow node). The second example, Fig. 5.3, shows us that only CNN6 feature localized it to the right location while the others fail. Recently, fusing different retrieval methods has been shown to enhance the overall retrieval performance [223, 224].

Motivated by [223] we dynamically assign a weight, based on the effectiveness of a feature, to all global features based on the area under normalized score between

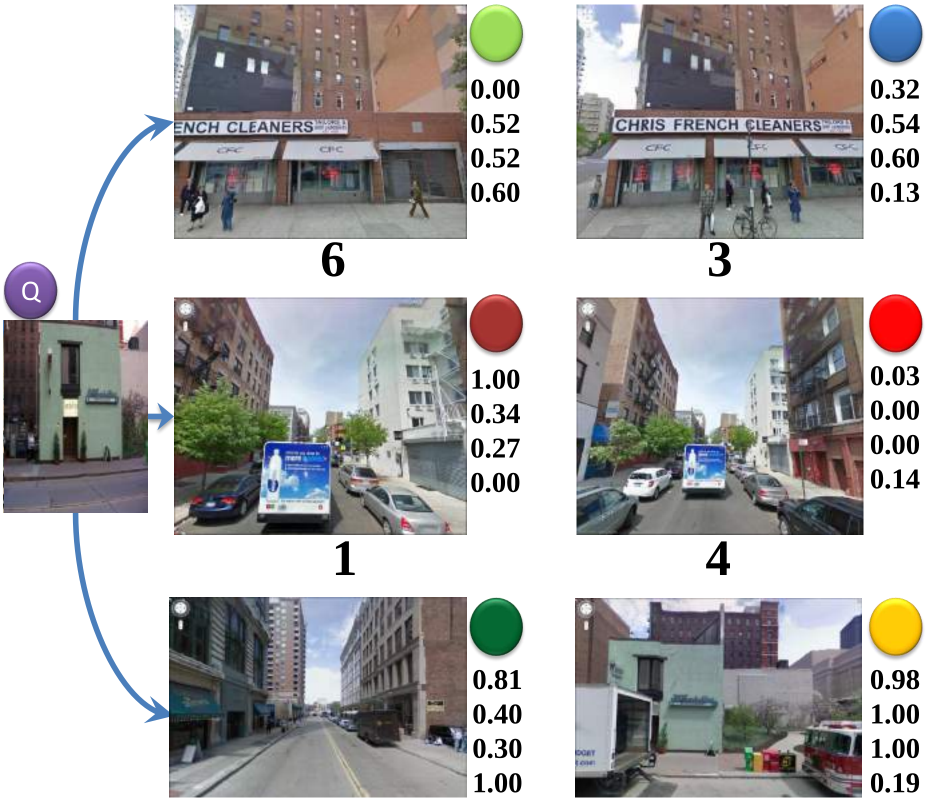


Figure 5.2: Exemplar output of the dominant set framework: **Left:** query, **Right:** each row shows corresponding reference images from the first, second and third local solutions (dominant sets), respectively, from top to bottom. The number under each image shows the frequency of the matched reference image, while those on the right side of each image show the min-max normalized scores of HSV, CNN6, CNN7 and GIST global features, respectively. The filled colors circles on the upper right corner of the images are used as reference IDs of the images.

the query and the matched reference images. The area under the curve is inversely proportional to the effectiveness of a feature. More specifically, let us suppose to have $\mathcal{G}$ global features and the distance between the query and the j^{th} matched reference image ($\mathcal{N}_j$), based on the i^{th} global feature ($\mathcal{G}_i$), is computed as: $f_i^j = \psi_i(\mathcal{Q}) - \psi_i(\mathcal{N}_j)$, where $\psi_i(\cdot)$ represents an operator which returns the i^{th} global descriptor of the argument node. Let the area under the normalized score of f_i be $\mathcal{A}_i$. The weight assigned for feature $\mathcal{G}_i$ is then computed as $w_i = \frac{1}{\mathcal{A}_i} / \sum_{j=1}^{|\mathcal{G}|} \frac{1}{\mathcal{A}_j}$.

Figs. 5.2 and 5.3 show illustrative cases of some of the advantages of having the post processing step. Both cases show the disadvantage of localization following heuristic approaches, as in [183, 3], to voting and selecting the reference image that matches to the query. In each case, the matched reference image with the highest number of votes (shown under each image) is the first node of the first extracted

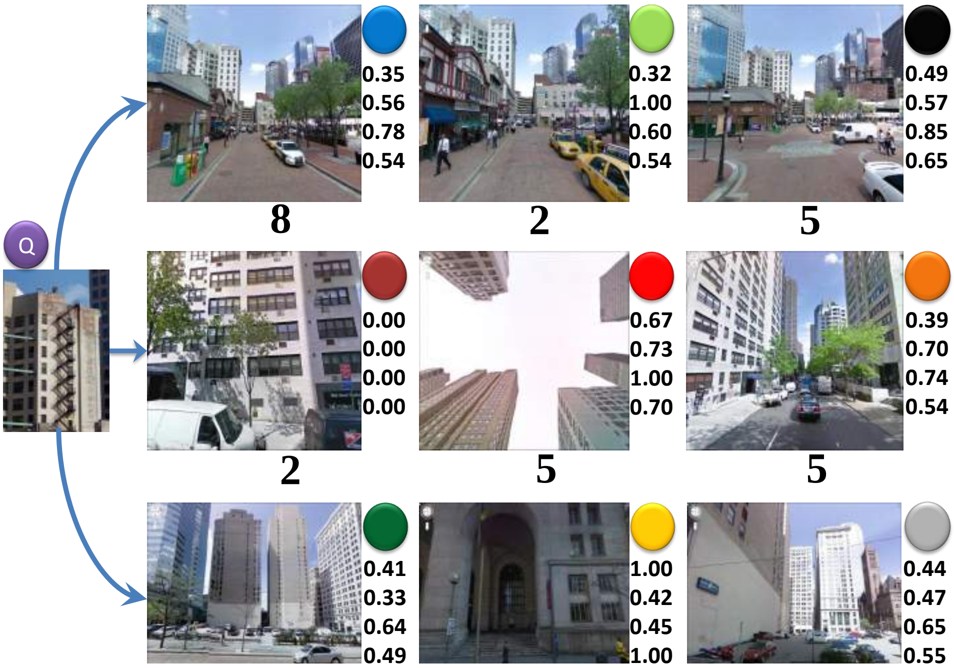
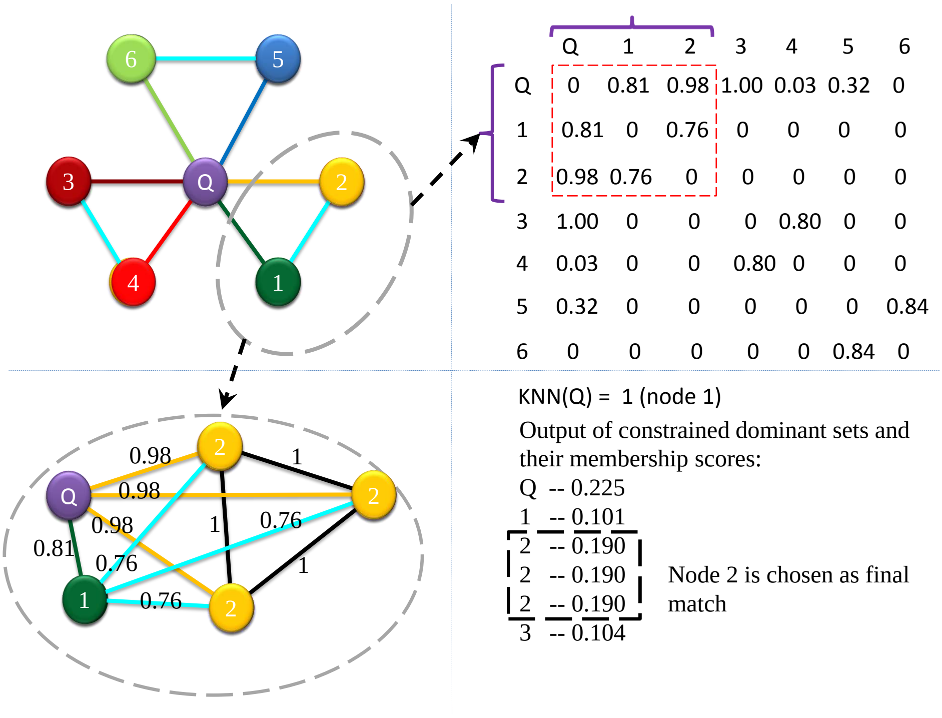


Figure 5.3: Exemplar output of the dominant set framework: **Left:** query, **Right:** each row shows corresponding reference images of the first, second and third local solutions (dominant sets), respectively, from top to bottom. The number under each image shows the mode of the matched reference image, while those on the right of each image show the min-max normalized scores of HSV, CNN6, CNN7 and GIST global features, respectively.

dominant set, but represents a wrong match. Both cases (Figs. 5.2 and 5.3) also demonstrate that the KNN-based matching may lead to a wrong localization. For example, by choosing HSV histogram as a global feature, the KNN approach chooses as best match the dark red node in Fig. 5.2 and the yellow node in Fig. 5.3 (both with min-max value to 1.00). Moreover, it is also evident that choosing the best match using the first extracted local solution (i.e., the light green node in Fig. 5.2 and blue node in Fig. 5.3), as done in [3], may lead to a wrong localization, since one cannot know in advance which local solution plays a major role. In fact, in the case of Fig. 5.2, the third extracted dominant set contains the right matched reference image (yellow node), whereas in the case of Fig. 5.3 the best match is contained in the first local solution (the light green node).

The similarity between query $\mathcal{Q}$ and the corresponding matched reference images is computed using their global features such as HSV histogram, GIST [225] and CNN (CNN6 and CNN7 are Convolutional Neural Network features extracted from ReLU6 and FC7 layers of pre-trained network, respectively [226]). For the different advantages and disadvantages of the global features, we refer interested readers to [3].



	Q	1	2	3	4	5	6
Q	0	0.81	0.98	1.00	0.03	0.32	0
1	0.81	0	0.76	0	0	0	0
2	0.98	0.76	0	0	0	0	0
3	1.00	0	0	0	0.80	0	0
4	0.03	0	0	0.80	0	0	0
5	0.32	0	0	0	0	0	0.84
6	0	0	0	0	0	0.84	0

KNN(Q) = 1 (node 1)

Output of constrained dominant sets and their membership scores:

Q	-- 0.225
1	-- 0.101
2	-- 0.190
2	-- 0.190
2	-- 0.190
3	-- 0.104

Node 2 is chosen as final match

Figure 5.4: Exemplar graph for post processing. **Top left:** reduced graph for Fig. 5.2 which contains unique matched reference images. **Bottom left:** Part of the full graph which contains the gray circled nodes of the reduced graph and the query. **Top right:** corresponding affinity of the reduced graph. **Bottom right:** The outputs of nearest neighbor approach, consider only the node's pairwise similarity, (KNN($\mathcal{Q}$)=node 3 which is the dark red node) and constrained dominant sets approach ($\mathcal{CDS}(\mathcal{Q})$ = node 2 which is the yellow node).

Fig. 5.2 shows the top three extracted dominant sets with their corresponding frequency of the matched reference images (at the bottom of each image). Let $\mathcal{F}_i$ be the number (cardinality) of local features, which belongs to i^{th} reference image from the extracted sets and the total number of matched reference images be $\mathcal{N}$. We build an affinity matrix $\hat{\mathtt{A}}$ of size $\mathcal{S} = \sum_{i=1}^{\mathcal{N}} \mathcal{F}_i + 1$ (e.g., for the example in Fig. 5.2, the size $\mathcal{S}$ is 19). Fig. 5.4 shows the reduced graph for the matched reference images shown in Fig. 5.2. Fig. 5.4 upper left, shows the part of the graph for the post processing. It shows the relation that the query has with matched reference images. The bottom left part of the figure shows how one can get the full graph from the reduced graph. For the example in Fig. 5.4, $\hat{V} = \{\mathcal{Q}, 1, 2, 2, 2, 3, 4, 4.......6\}$.

The advantages of using constrained dominant sets are numerous. First, it provides a unique local (and hence global) solution whose support coincides with the union of all dominant sets of $\hat{G}$, which contains the query. Such solution contains all the local solutions which have strong relation with the user-selected query. As it can be observed in Fig. 5.4 (bottom right), the Constrained Dominant Set which contains the query $\mathcal{Q}$, $\mathcal{CDS}(\mathcal{Q})$, is the union of all overlapping dominant sets (the query, one green, one dark red and three yellow nodes) containing the query as one of the members. If we assume to have no cyan link between the green and yellow nodes, as long as there is a strong relation between the green node and the query, $\mathcal{CDS}(\mathcal{Q})$ will not be affected. In addition, due to the noise, the strong affinity between the query and the green node may be reduced, while still keeping the strong relation with the cyan link which, as a result, will preserve the final result. Second, in addition to fusing all the local solutions leveraging the notion of centrality, one of the interesting properties of dominant set framework is that it assigns to each image a score corresponding to how similar it is to the rest of the images in the solution. Therefore, not only it helps selecting the best local solution, but also choosing the best final match from the chosen local solution. Third, an interesting property of constrained dominant sets approach is that it not only considers the pairwise similarity of the query and the reference images, but also the similarity among the reference images. This property helps the algorithm avoid assignment of wrong high pairwise affinities. As an example, with reference to Fig. 5.4, if we consider the nodes pairwise affinities, the best solution will be the dark red node (score 1.00). However, using constrained dominant sets and considering the relation among the neighbors, the solution bounded by the red dotted rectangle can be found, and by choosing the node with the highest membership score, the final best match is the yellow node which is more similar to the query image than the reference image depicted by the dark red node (see Fig. 5.2).

5.5 Experimental Results

5.5.1 Dataset Description

We evaluate the proposed algorithm using publicly available reference data sets of over 102k Google street view images [3] and a new dataset, ***WorldCities***, of high resolution 300k Google street view images collected for this work. The 360 degrees view of each place mark is broken down into one top and four side view images. The ***WorldCities*** dataset is publicly available. [1]

The 102k Google street view images dataset covers 204 Km of urban streets and the place marks are approximately 12 m apart. It covers downtown and the neighboring areas of Orlando, FL; Pittsburgh, PA and partially Manhattan, NY. The ***WorldCities*** dataset is a new high resolution reference dataset of 300k street view images that covers 14 different cities from different parts of the world: Europe (Amsterdam, Frankfurt, Rome, Milan and Paris), Australia (Sydney and Melbourne), USA (Vegas, Los Angeles, Phoenix, Houston, San Diego, Dallas, Chicago). Exis-

[1] http://www.cs.ucf.edu/~haroon/UCF-Google-Streetview-II-Data/UCF-Google-Streetview-II-Data.zip

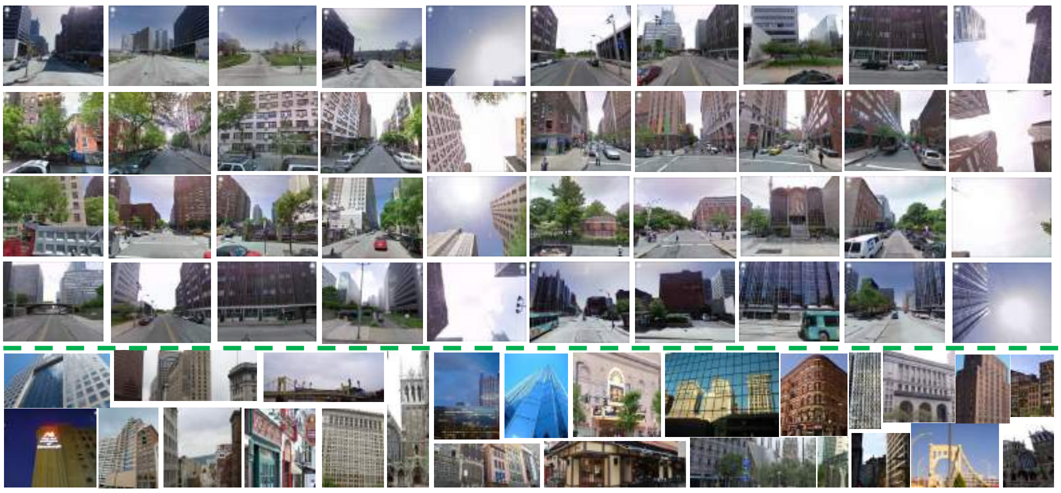


Figure 5.5: The top four rows are sample street view images from eight different places of ***WorldCities*** dataset. The bottom two rows are sample user uploaded images from the test set.

tence of similarity in buildings around the world, which can be in terms of their wall designs, edges, shapes, color etc, makes the dataset more challenging than the other. Fig. 5.5 (top four rows) shows sample reference images taken from different place marks.

For the test set, we use 644 and 500 GPS-tagged user uploaded images downloaded from Picasa, Flickr and Panoramio for the 102k Google street view images and ***WorldCities*** datasets, respectively. Fig. 5.5 (last two rows) shows sample test images. Throughout our experiment, we use the all the reference images from around the world to find the best match with the query image, not just with the ground truth city only.

5.5.2 Quantitative Comparison With Other Methods

5.5.2.1 Performance on the 102k Google Street View Images Dataset

The proposed approach has been then compared with the results obtained by state-of-the-art methods. In Fig. 5.6, the horizontal axes shows the error threshold in meters and the vertical axes shows the percentage of the test set localized within a particular error threshold. Since the scope of this work is an accurate image localization at a city-scale level, test set images localized above 300 meter are considered a failure.

The black (-*-) curve shows localization result of the approach proposed in [196] which uses vocabulary tree to localize images. The red (-o-) curve depicts the results of [183] where they only consider the first NN for each query feature as best matches which makes the approach very sensitive to the query features they select. Moreover, their approach suffers from lacking global feature information. The green (-o-) curve illustrates the localization results of [3] which uses generalized maximum clique problem (GMCP) to solve feature matching problem and follows voting scheme to select the best matching reference image. The black (-o- and $-\Diamond-$) curves show localization results of MAC and RMAC, (regional) maximum activation of convolutions ([227, 228]). These approaches build compact feature vectors that encode several image regions without the need to feed multiple inputs to the network. The cyan (-o-) curve represents localization result of NetVLAD [198] which aggregates mid-level (conv5) convolutional features extracted from the entire image into a compact single vector representation amenable to effici,ent indexing. The cyan ($-\Diamond-$) curve depicts localization result of NetVLAD but finetuned on our dataset. The blue ($-\Diamond-$) curve show localizaton result of approach proposed in [197] which exploits geometric relations between different database images retrieved by a query to handle geometric burstness. The blue (-o-) curve shows results from our baseline approach, that is, we use voting scheme to select best match reference image and estimate the location of the query image. We are able to make a 10% improvement w.r.t the other methods with only our baseline approach (without post processing). The magenta (-o-) curve illustrates geo-localization results of our proposed approach using dominant set clustering based feature matching and constrained dominant set clustering based post processing. As it can be seen, our approach shows about 20% improvement over the state-of-the-art techniques.

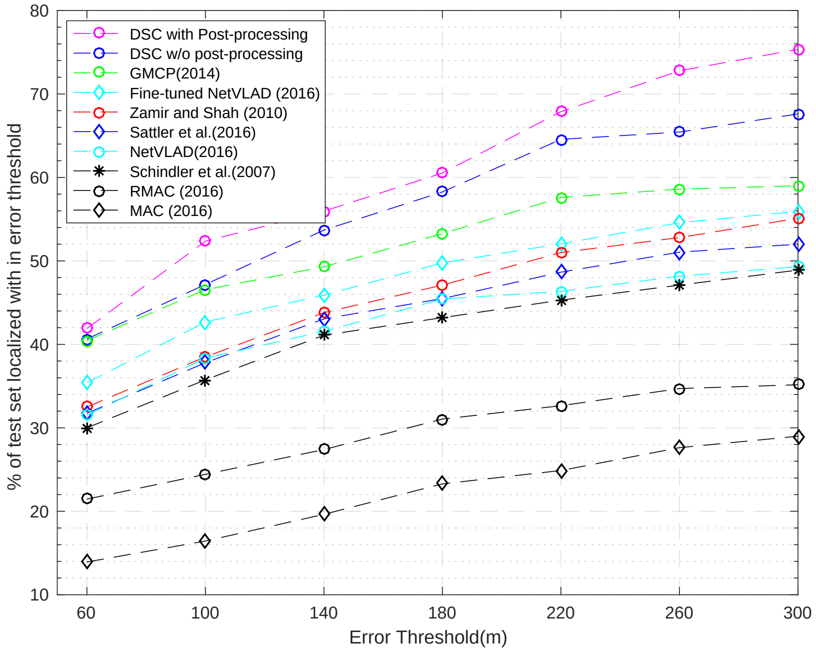


Figure 5.6: Comparison of our baseline (without post processing) and final method, on overall geo-localization results, with state-of-the-art approaches on the first dataset (102K Google street view images).

Computational Time. Fig. 5.7, on the vertical axis, shows the ratio between GMCP (numerator) and our approach (denominator) in terms of CPU time taken for each query images to be localized. As it is evident from the plot, this ratio can range from 200 (our approach 200x faster than GMCP) to a maximum of even 750x faster.

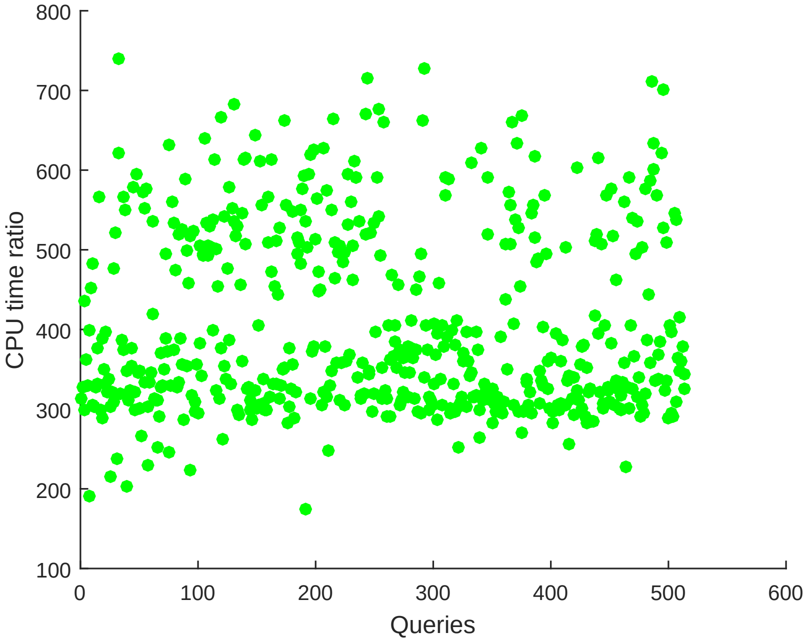


Figure 5.7: The ratio of CPU time taken between GMCP based geo-localization [3] and our approach, computed as CPU time for GMCP/CPU time for DSC.

5.5.2.2 Performance on the WorldCities Dataset

We have also compared the performance of different algorithms on the new dataset of 300k Google street view images created by us. Similarly to the previous tests, Fig. 5.8 reports the percentage of the test set localized within a particular error threshold. Since the new dataset is relatively more challenging, the overall performance achieved by all the methods is lower compared to 102k image dataset.

From bottom to top of the graph in Fig. 5.8 we present the results of [227, 228] black (−◊− and -o-), [197] blue (−◊−), [183] red (-o-), [198] cyan (-o-), fine tuned [198] cyan (−◊−), [3] green (-o-), our baseline approach without post processing blue (-o-) and our final approach with post processing magenta (-o-) . The improvements obtained with our method are lower than in the other dataset, but still noticeable (around 2% for the baseline and 7% for the final approach).

Some qualitative results for Pittsburgh, PA are presented in Fig. 5.9.

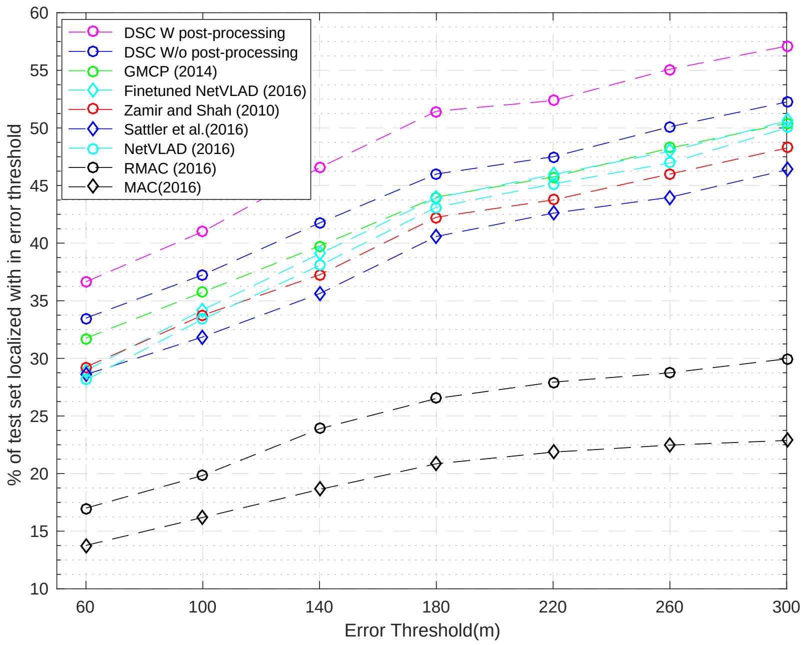


Figure 5.8: Comparison of overall geo-localization results using DSC with and without post processing and state-of-the-art approaches on the ***WorldCities*** dataset.

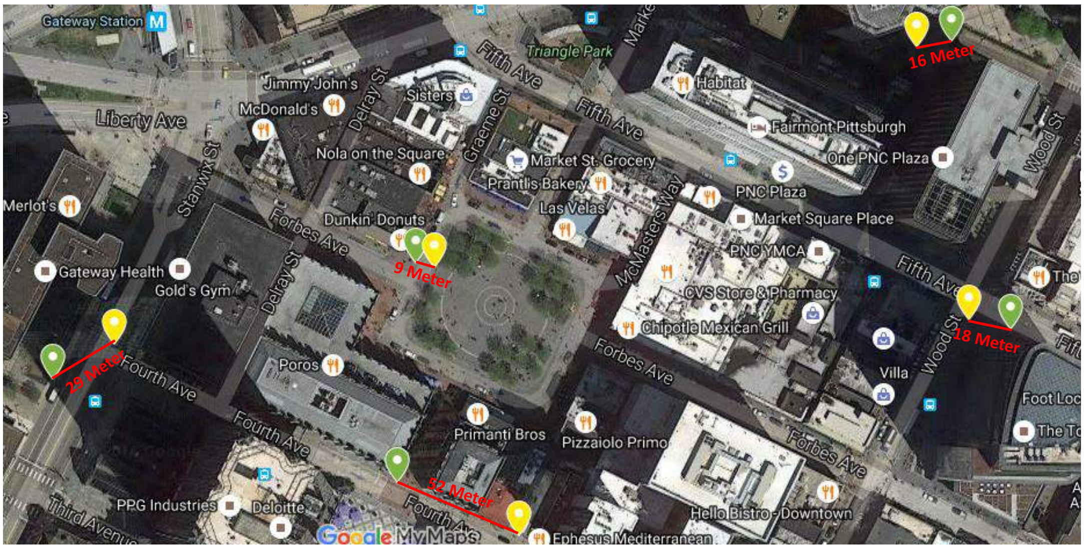


Figure 5.9: Sample qualitative results taken from Pittsburgh area. The green ones are the ground truth while yellow locations indicate our localization results.

5.5.3 Analysis

5.5.3.1 Outlier Handling

In order to show that our dominant set-based feature matching technique is robust in handling outliers, we conduct an experiment by fixing the number of NNs (disabling the dynamic selection of NNs) to different numbers. It is obvious that the higher the number of NNs are considered for each query feature, the higher will be the number of outlier NNs in the input graph, besides the increased computational cost and an elevated chance of query features whose NNs do not contain any inliers surviving the pruning stage.

Fig. 5.10 shows the results of geo-localization obtained by using GMCP and dominant set based feature matching on 102K Google street view images [3]. The graph shows the percentage of the test set localized within the distance of 30 meters as a function of number of NNs. The blue curve shows the results using dominant sets: it is evident that when the number of NNs increases, the performance improves despite the fact that more outliers are introduced in the input graph. This is mainly because our framework takes advantage of the few inliers that are added along with many outliers. The red curve shows the results of GMCP based localization and as the number of NNs increase the results begin to drop. This is mainly due to the fact that their approach imposes hard constraint that at least one matching reference feature should be selected for each query feature whether or not the matching feature is correct.

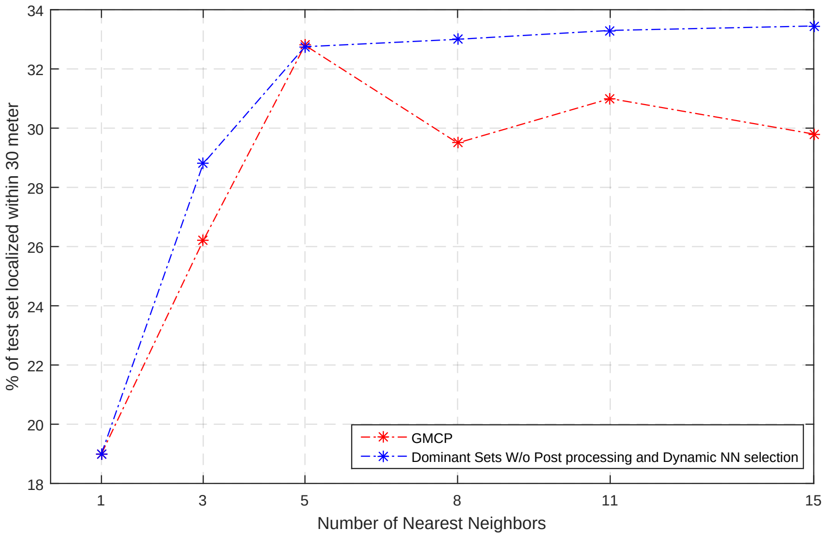


Figure 5.10: Geo-localization results using different number of NN

5.5.3.2 Effectiveness of the Proposed Post Processing

In order to show the effectiveness of the post processing step, we perform an experiment comparing our constrained dominant set based post processing with a simple voting scheme to select the best matching reference image. The GPS information of the best matching reference image is used to estimate the location of the query image. In Fig. 5.12, the vertical axis shows the percentage of the test set localized within a particular error threshold shown in the horizontal axis (in meters). The blue and magenta curves depict geo-localization results of our approach using a simple voting scheme and constrained dominant sets, respectively. The green curve shows the results from GMCP based geo-localization. As it is clear from the results, our approach with post processing exhibits superior performance compared to both GMCP and our baseline approach.

Since our post-processing algorithm can be easily plugged in to an existing retrieval methods, we perform another experiment to determine how much improvement we can achieve by our post processing. We use [199, 227, 228] methods to obtain candidate reference images and employ as an edge weight the similarity score generated by the corresponding approaches. Table 5.1 reports, for each dataset, the first row shows rank-1 result obtained from the existing algorithms while the second row (w_post) shows rank-1 result obtained after adding the proposed post-processing step on top of the retrieved images. For each query, we use the first 20 retrieved reference images. As the results demonstrate, Table 5.1, we are able to make up to 7% and 4% improvement on 102k Google street view images and ***WorldCities*** datasets, respectively. We ought to note that, the total additional time required to perform the above post processing, for each approach, is less than 0.003 seconds on average.

		NetVLAD	NetVLAD*	RMAC	MAC
Dts 1	Rank1	49.2	56.00	35.16	29.00
	w_post	**51.60**	**58.05**	**40.18**	**36.30**
Dts 2	Rank1	50.00	50.61	29.96	22.87
	w_post	**53.04**	**52.23**	**33.16**	**26.31**

Table 5.1: Results of the experiment, done on the 102k Google street view images (Dts1) and ***WorldCities*** (Dts2) datasets, to see the impact of the post-processing step when the candidates of reference images are obtained by other image retrieval algorithms

The NetVLAD results are obtained from the features generated using the best trained model downloaded from the author's project page [199]. It's fine-tuned version (NetVLAD*) is obtained from the model we fine-tuned using images within 24m range as a positive set and images with GPS locations greater than 300m as a negative set.

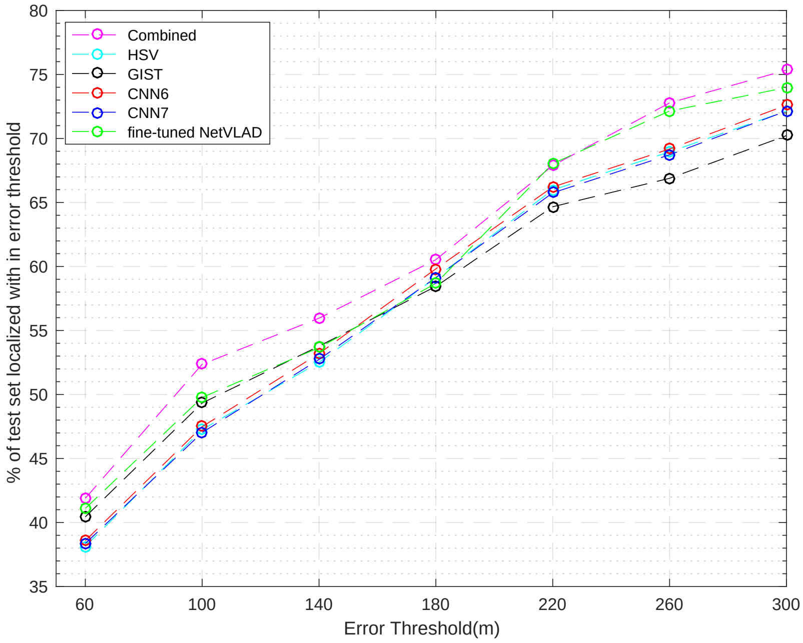


Figure 5.11: Comparison of geo-localization results using different global features for our post processing step.

The MAC and RMAC results are obtained using MAC and RMAC representations extracted from fine-tuned VGG networks downloaded from the authors webpage [227, 228].

5.5.3.3 Assessment of Global Features Used in Post Processing Step

The input graph for our post processing step utilizes the global similarity between the query and the matched reference images. Wide variety of global features can be used for the proposed technique. In our experiments, the similarity between query and the corresponding matched reference images is computed between their global features, using HSV, GIST, CNN6, CNN7 and fine-tuned NetVLAD. The performance of the proposed post processing technique highly depends on the discriminative ability of the global features used to built the input graph.

Depending on how informative the feature is, we dynamically assign a weight for each global feature based on the area under the normalized score between the query and the matched reference images. To show the effectiveness of this approach, we perform an experiment to find the location of our test set images using both individual and combined global features. Fig. 5.11 shows the results attained by using fine-tuned NetVLAD, CNN7, CNN6, GIST, HSV and by combining them together. The combination of all the global features outperforms the individual feature performance, demonstrating the benefits of fusing the global features based on their discriminative abilities for each query.

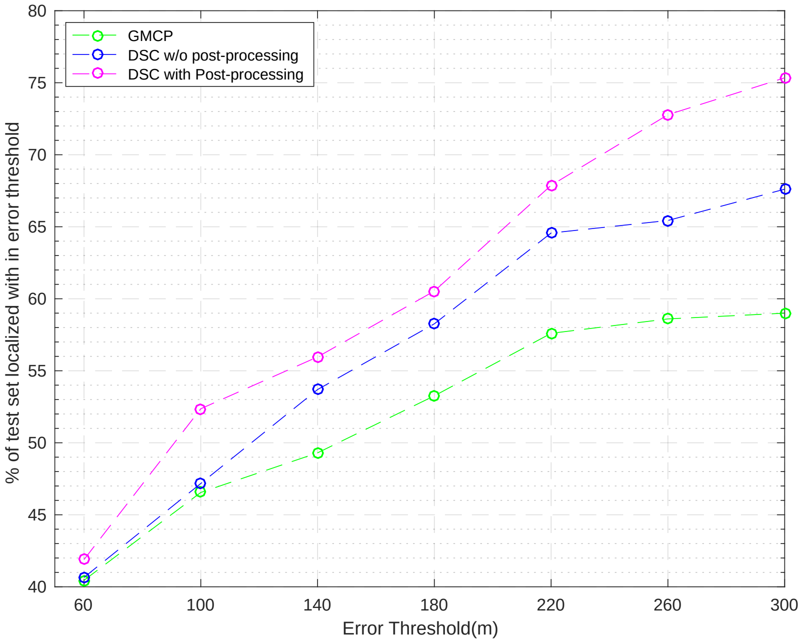


Figure 5.12: The effectiveness of constrained dominant set based post processing step over simple voting scheme.

5.6 Conclusion and Future Work

In this chapter, we proposed a novel framework for city-scale image geo-localization. Specifically, we introduced dominant set clustering-based multiple NN feature matching approach. Both global and local features are used in our matching step in order to improve the matching accuracy. In the experiments, carried out on two large city-scale datasets, we demonstrated the effectiveness of post processing employing the novel constrained dominant set over a simple voting scheme. Furthermore, we showed that our proposed approach is 200 times, on average, faster than GMCP-based approach [3]. Finally, the newly-created dataset (***WorldCities***) containing more than 300k Google Street View images used in our experiments will be made available to the public for research purposes.

As a natural future direction of research, we can extend the results of this work for estimating the geo-spatial trajectory of a video in a city-scale urban environment from a moving camera with unknown intrinsic camera parameters.

6 Conclusion

> I think and think for months and years. Ninety-nine times, the conclusion is false. The hundredth time I am right.
>
> Albert Einstein

This thesis proposes a robust similarity-based clustering technique, based on graph and game theoretic principles, and have demonstrated its applicability to many problems in computer vision such as interactive image segmentation and co-segmentation (in both the unsupervised and the interactive flavor), multi-target tracking in multiple non-overlapping cameras, geo-localization, person re-identification and retrieval. The proposed clustering approach is with many interesting properties such as: it does clustering while obliterating outliers in simultaneous fashion, it doesn't need any a prior knowledge on the number of clusters, able to deal with compact clusters and with situations involving arbitrarily-shaped clusters in a context of heavy background noise, does not have any assumptions with the structure of the affinity matrix, it is fast and scalable to large scale problems, and others.

The proposed *constrained dominant set* clustering technique, which is based on some properties of a family of quadratic optimization problems related to dominant sets, generalizes the dominant sets framework in that putting the regularization parameter to zero results local solutions that are in one-to-one correspondence with dominant sets. In particular, we show that by properly selecting a regularization parameter that controls the structure of the underlying function, we are able to "force" all solutions to contain the constraint elements. We provide bounds that allow us to control this process, which are based on the spectral properties of certain submatrices of the original affinity matrix. This provides us with an increased in flexibility. A modification in adding a stopping criteria in the sequential search of clusters enables us to have an algorithm for simultaneous clustering and outliers detection (SCOD). Unlike most of the previous approaches, the method requires no prior knowledge on both the number of clusters and outliers, which makes our approach more convenient for real application. Moreover, our proposed algorithm is simple to implement and is highly scalable.

One of the best known class of game dynamics to extract (constrained) dominant set from a graph is the so-called replicator dynamics whose computational complexity is quadratic per step which makes it handicapped for large-scale applications. In this thesis, we propose a fast algorithm, based on dynamics from evolutionary game theory, which is efficient and scalable to large-scale real world applications. Running any of the dynamics just extracts one (constrained) dominant set at a time. So, if we want to find more than one (constrained) dominant set, we have to apply a restarting strategy and run the dynamics again from another state in the simplex hoping not to converge again towards the same equilibrium. A further option is to adopt a peeling-off strategy, where we remove the nodes in the support of the extracted equilibrium and run the dynamics again on the rest of the nodes. This way, however, there is no guarantee that the clusters extracted after the very first one still correspond to ESSs of the original problem. In the constrained dominant set formulation, properly selecting a regularization parameter enable us to enumerate the (constrained) dominant sets, and therefore the clusters of the original problem, by iteratively rendering unstable under the evolutionary dynamics the equilibria that have been already extracted without affecting the set of remaining (constrained) dominant sets. Moreover, we apply our idea also to the enumeration of (constrained) maximal cliques of a graph. It enabled us to obtain good performances on both random graph instances and DIMACS benchmark graphs.

The applicability of the proposed similarity-based clustering technique has been shown in different chapters of the thesis with extensive experiments on benchmark datasets which have shown that our approach considerably improves the state-of-the-art results on the many different problems addressed. This provides evidence that the proposed techniques hold promise as a powerful and principled framework to address a large class of vision problems. Chapter four of the thesis shows how CDS is used to constrain the diffusion process locally to improve the performance of any retrieval systems. The framework alleviates the limitations of existing approaches while improving the state-of-the- art results. Moreover, it solves an open computer vision and machine learning issue which is automatically selecting a reasonable local neighborhood size [132]. In chapter three of the thesis, CDS is applied effectively to the problem of ”constrained” segmentation. By exploiting high-level, semantic knowledge on the part of the user, which is typically difficult to formalize, it is able to effectively solve segmentation problems which would be otherwise too complex to be tackled using fully automatic segmentation algorithms. The resulting algorithm, unlike other state-of-the-art approaches, can deal naturally with any type of input modality, including scribbles, sloppy contours, and bounding boxes, and is able to robustly handle noisy annotations on the part of the user, (in the case of interactive co-segmentation) it needs smaller user interactions in that the user can put scribbles only on the object of some of the images. The last two chapters of the thesis use CDS effectively to solve geo-localization, Multi-Target Multi-Person Tracking in Multiple Non-overlapping Cameras and Person Re -Identification. In all the applications we were able to improve the state-of-the-art results by a large margin.

A

Evaluating CDS on Random graphs and DIMACS benchmark graphs

Idleness is an appendix to nobility.

Robert Burton

A.1 Experiments on random graphs

Here, we extend the experiments which evaluate the performance of CDS on both random and DIMACS benchmark graphs.

Our first set of experiments involve randomly generated graphs with varying densities and sizes. In order to verify the correctness of our claim, we needed to output *all* maximal cliques of a given graph and, due to the computational complexity of this problem (which is NP-complete even to approximate), we could work only on moderate-sized graphs. In particular, we generated graphs with sizes ranging from 50 to 80 (in steps of 10) and densities $\delta \in \{0.2, 0.4, 0.5, 0.6\}$. For each pair (n, δ) we generated 30 instances of random graphs with the prescribed size-density value. So, overall, 480 random graphs were generated in this set of experiments. To enumerate all maximal cliques for a given graph, say $\mathcal{C}_1, \mathcal{C}_2, ...\mathcal{C}_\mathcal{L}$, we used the Bron-Kerbosch algorithm, a well-known exact algorithm which uses a recursive backtracking procedure [229].

Note that, as Case 2 is a simple generalization of Case 1, we evaluated our conjecture only for two different situations, namely Case 2 and Case 3.

As for **Case 2**, the procedure we used is as follows:

1. From the $\mathcal{L}$ maximal cliques found by the Bron-Kerbosch algorithm, select one randomly, say $\mathcal{C}_i$, and take a fraction of its elements to form the set S (the parameter φ represents the percentage of selected elements).

2. Collect all the maximal cliques which contain the set S and compute the union $\mathcal{UBK} = \mathcal{K}_1 \cup \mathcal{K}_1 \cup ..... \cup \mathcal{K}_\mathcal{P}$.

3. Run our algorithm using the selected set S, and let $\mathcal{O}$ be its output (namely, the support of the converged solution).

4. Compute **FC** as follows:

$$\mathbf{FC} = \frac{|\mathcal{O} \cap \mathcal{UBK}|^2}{|\mathcal{UBK}| * |\mathcal{O}|}$$

Observe that **FC** $= 1$ if and only if the two sets are identical which implies that our conjecture is verified.

As for **Case 3**, the procedure we used is as follows:

1. From the $\mathcal{L}$ maximal cliques found by the Bron-Kerbosch algorithm, select two randomly, say $\mathcal{C}_i$ and $\mathcal{C}_j$, and take a fraction φ of their elements to form the sets S_i and S_j.

2. Take the set S as the union of the two sets, i.e $S = S_i \cup S_j$.

3. Run our algorithm using the selected set S, and let $\mathcal{O}$ be its output (namely, the support of the converged solution).

4. Denote by S_o the set of elements of $\mathcal{O}$ that are also in S.

5. Collect all the maximal cliques which contain the set S_o and compute the union $\mathcal{UBK} = \mathcal{K}_1 \cup \mathcal{K}_1 \cup ..... \cup \mathcal{K}_{\mathcal{P}}$.

6. Compute **FC** as follows:

$$\mathbf{FC} = \frac{|\mathcal{O} \cap \mathcal{UBK}|^2}{|\mathcal{UBK}| * |\mathcal{O}|}$$

7. Repeat the process setting the set S as $S \setminus S_o$

As noted above, again note that **FC** $= 1$ if and only if the two sets are identical. We also used another measure, **FM**, computed as

$$\mathbf{FM} = \frac{|\mathcal{C}_i|}{|\mathcal{O}|}$$

which has some interesting properties. Increasing φ, the fraction of clique elements chosen randomly, the number of maximal cliques $\mathcal{L}$ which contain the chosen set S decreases. This in turn implies that **FM** goes to 1. Remember that $\mathcal{C}_i$ is the randomly selected maximal clique from which S is built. Further, in our experiments we also measured the number of maximal cliques, denoted by N(cliq), which contain the given set S.

Figures A.1 and A.2 show the average behavior of our measures, together with the corresponding error bars, as a function of the fraction of the vertices chosen randomly from the cliques (namely, φ). Observe that **in all our experiments we got FC** $=1$ (as reflected by the constant behavior of the **FC** plots and the zero variance) which confirms the validity of our claim.

Note also that as expected the **FM** measure goes from small fractions to 1, and the N(cliq) measure and its variance V(N_Cliq) start with very high value which goes down to 1 and 0, respectively, as φ gets increased. These confirm that given a maximal clique, the algorithm converges to the same maximal clique.

Next, we tested our claim over edge-weighted (random) graphs. We generated them by slightly (randomly) perturbing the entries of the adjacency matrices of original (unweighted) graphs used in the previous set of experiments, in such a way that their maximal cliques corresponded to the dominant sets in their weighted versions. (This can easily be done using the convergence properties of replicator dynamics [35].) This allowed us to know all dominant sets of these graphs and hence compute our measures as we did before.

Figures A.3 and A.4 show the average behavior of our measures, together with the corresponding error bars, as a function of the fraction of the vertices chosen randomly from the cliques (namely, φ). Again, bbserve that **in all our experiments we got FC** $=1$ (as reflected by the constant behavior of the **FC** plots and the zero variance) which confirms the validity of our claim. The behavior of the other measures is also consistent with the results obtained on the unweighted case.

As for computational time, given the affinity matrix and the set S, in all the experiments done, the algorithm took only a fraction of milliseconds to converge.

A.2 Experiments on DIMACS benchmark graphs

In order to be able to enumerate all maximal cliques using the Bron-Kerbosch algorithm, we generated a number of moderate-sized *brock* graphs using the generator available on the DIMACS website [1] and repeated the experiments we described before for random (unweighted and weighted) graphs. The results are shown in Figures A.5 and A.6, and again they confirm our conjecture as evinced from the constant value (and zero variance) of the **FC** measure.

[1] http://iridia.ulb.ac.be/∼fmascia/maximum_clique/DIMACS-benchmark

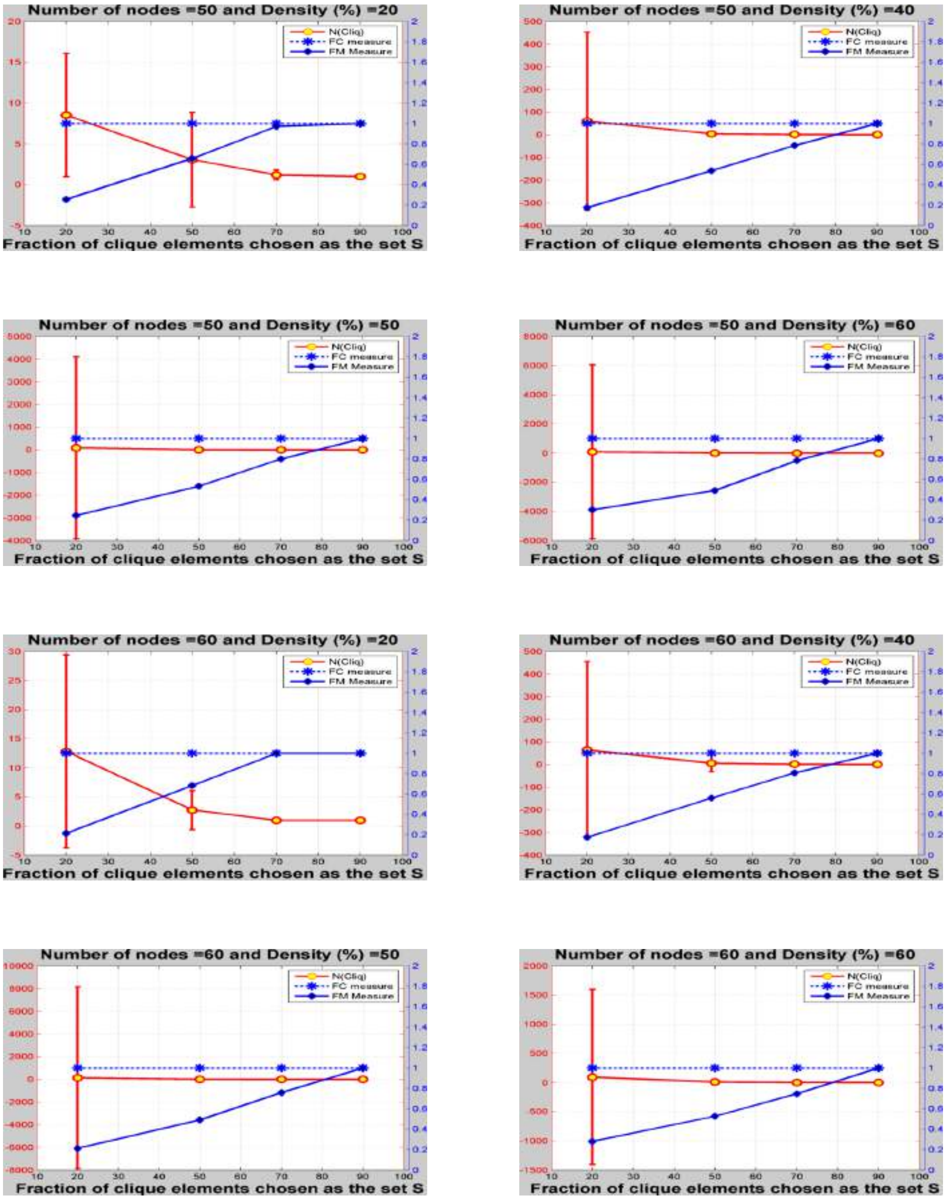


Figure A.1: Performance of the algorithm on randomly generated (unweighted) graphs (part I).

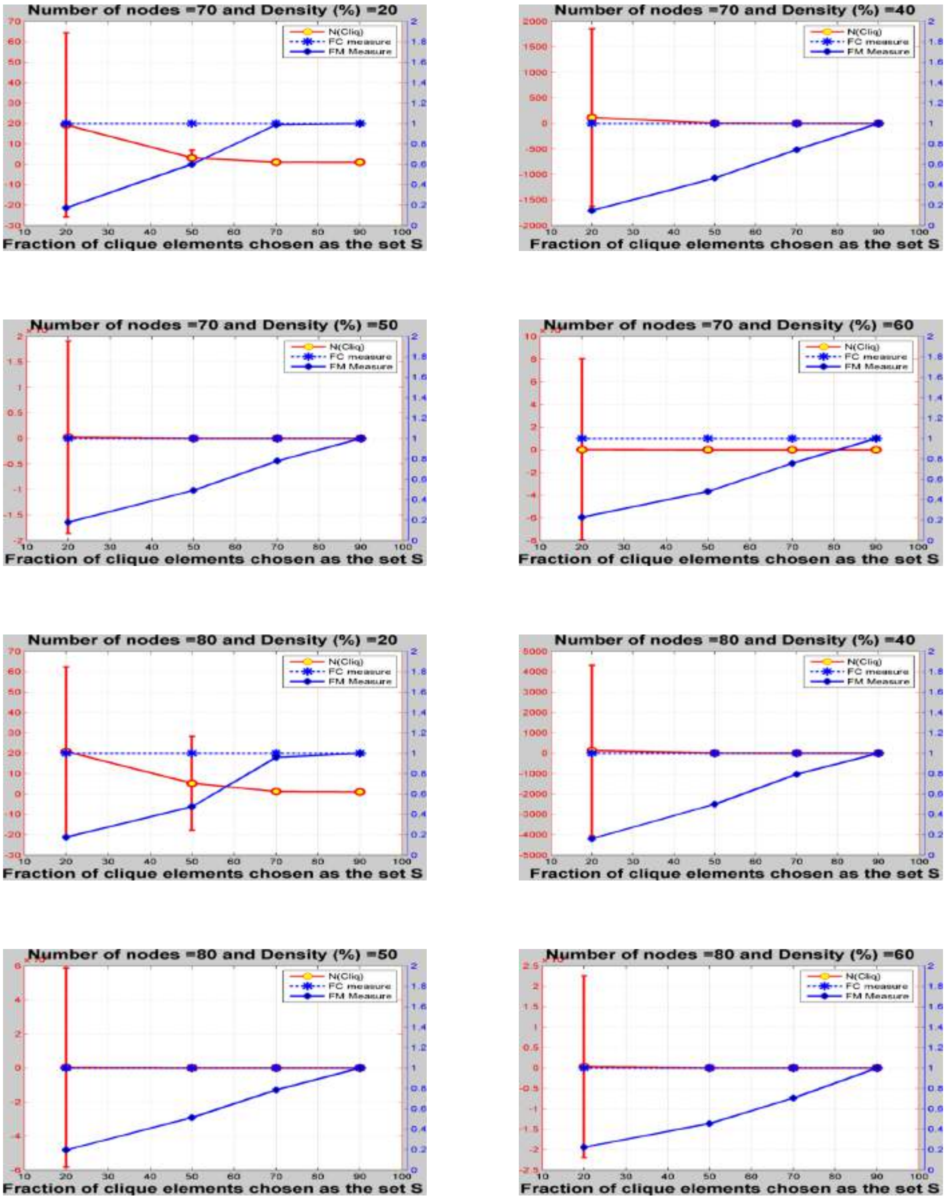


Figure A.2: Performance of the algorithm on randomly generated (unweighted) graphs (part II).

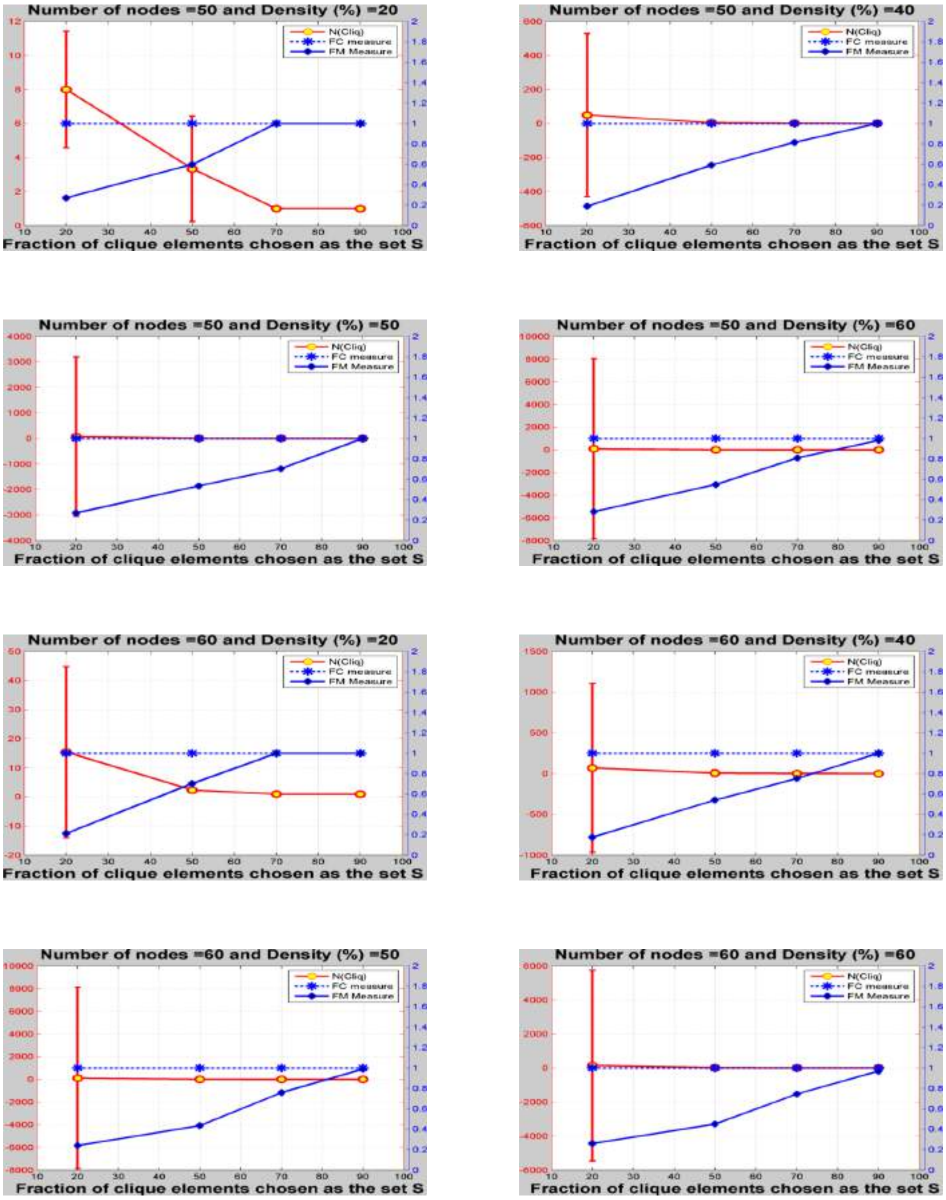


Figure A.3: Performance of the algorithm on randomly generated (weighted) graphs (part I).

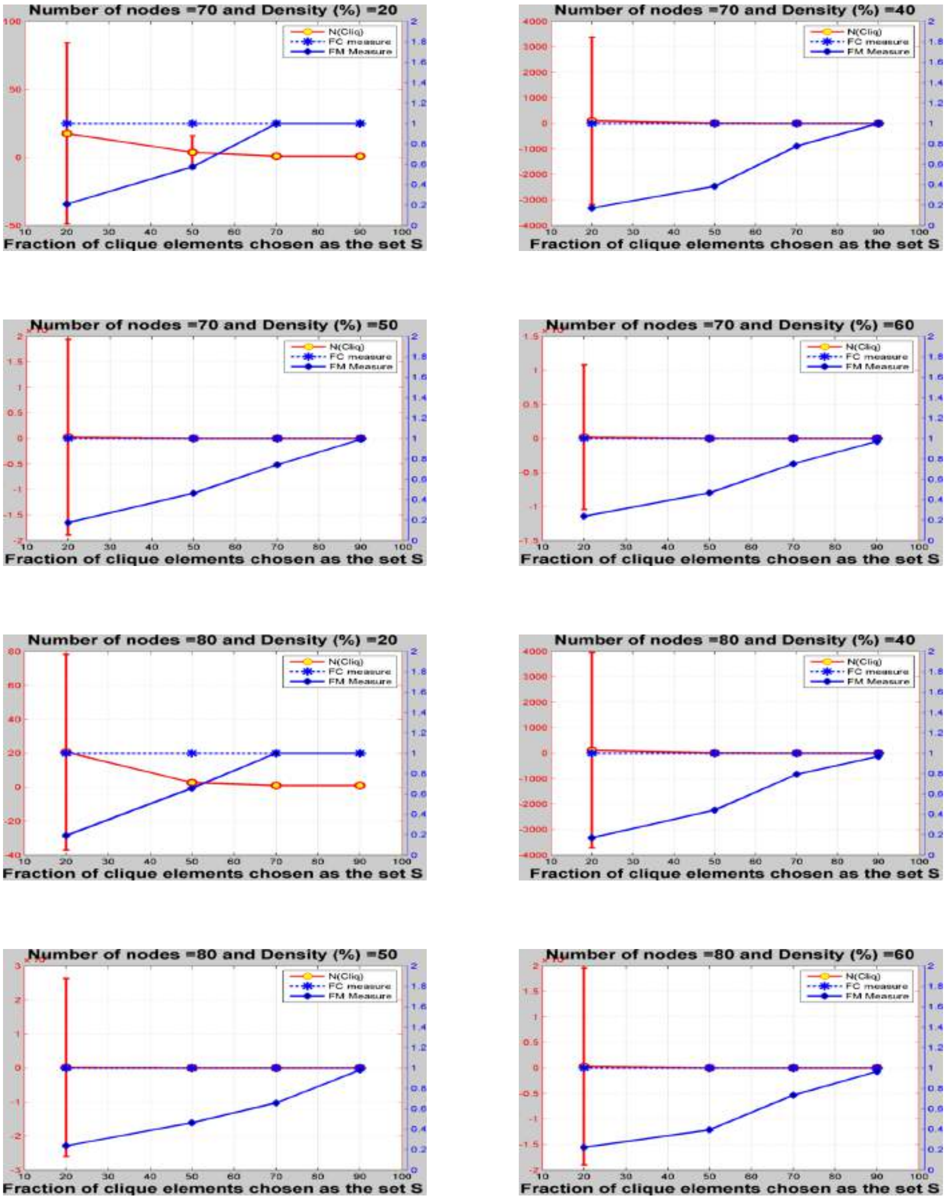


Figure A.4: Performance of the algorithm on randomly generated (weighted) graphs (part II).

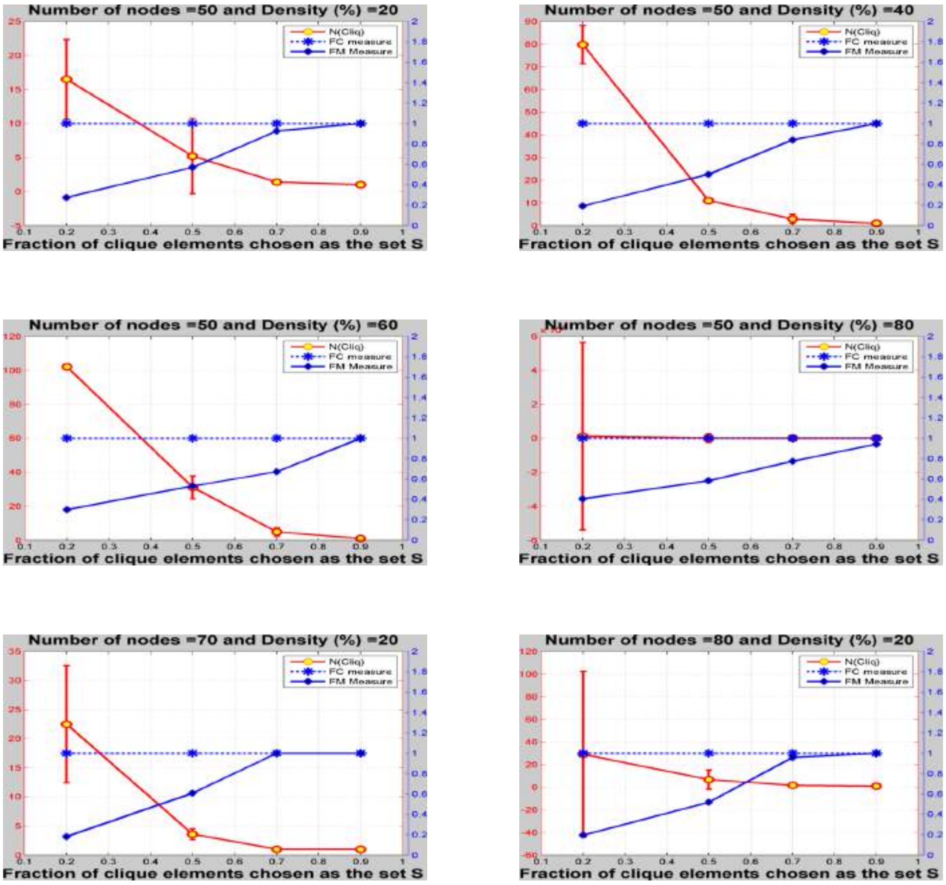


Figure A.5: Performance of the algorithm on moderate-sized *brock* DIMACS graphs.

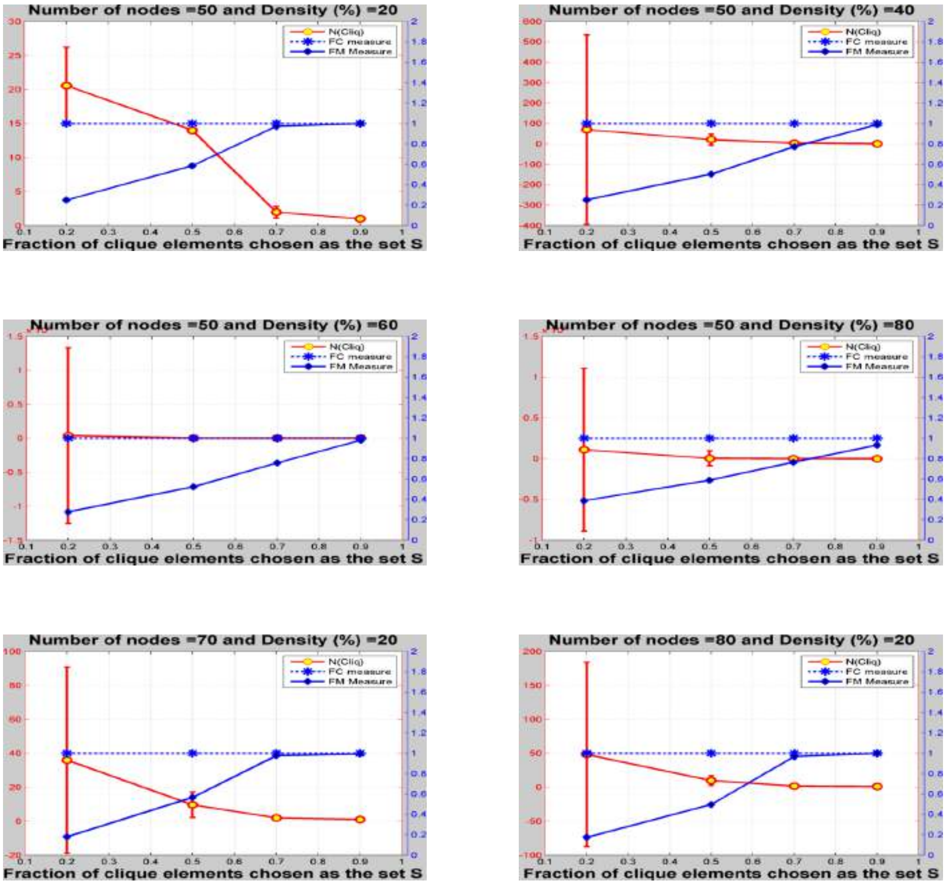


Figure A.6: Performance of the algorithm on moderate-sized *brock* (weighted) DIMACS graphs.

Bibliography

[1] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33:898–916, 2011.

[2] Victor S. Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *ICCV*, pages 277–284, 2009.

[3] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1546–1558, 2014.

[4] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. pages 17–35, 2016.

[5] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. pages 2360–2367, 2010.

[6] Fei Xiong, Mengran Gou, Octavia I. Camps, and Mario Sznaier. Person re-identification using kernel-based metric learning methods. pages 1–16, 2014.

[7] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. pages 1116–1124, 2015.

[8] Bingpeng Ma, Yu Su, and Frédéric Jurie. Covariance descriptor based on bio-inspired features for person re-identification and face verification. *Image Vision Computing*, 32(6-7):379–390, 2014.

[9] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z. Li. Person re-identification by local maximal occurrence representation and metric learning. pages 2197–2206, 2015.

[10] Alexander Kläser, Marcin Marszalek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. pages 1–10, 2008.

[11] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. MARS: A video benchmark for large-scale person re-identification. pages 868–884, 2016.

[12] Eyasu Zemene and Marcello Pelillo. Interactive image segmentation using constrained dominant sets. In *ECCV*, pages 278–294, 2016.

[13] Eyasu Zemene, Yonatan Tariku, Andrea Prati, and Marcello Pelillo. Simultaneous clustering and outlier detection using dominant sets. In *ICPR*, 2016.

[14] Eyasu Zemene and Marcello Pelillo. Path-based dominant-set clustering. In *ICIAP*, pages 150–160, 2015.

[15] Eyasu Zemene, Leulseged Tesfaye Alemu, and Marcello Pelillo. Constrained dominant sets for retrieval. In *ICPR 2016*, pages 76–81, 2014.

[16] Yonatan Tariku Tesfaye, Eyasu Zemene, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets. *arxiv*, abs/1706.06196, 2017.

[17] Eyasu Zemene, Yonatan Tariku, Haroon Idrees, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Large-scale image geo-localization using dominant sets. *arxiv*, abs/1702.01238, 2017.

[18] Jon M. Kleinberg. An impossibility theorem for clustering. In *NIPS*, pages 446–453, 2002.

[19] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, pages 343–356, 2012.

[20] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. GMMCP tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *CVPR*, pages 4091–4099, 2015.

[21] Francesco Solera, Simone Calderara, Ergys Ristani, Carlo Tomasi, and Rita Cucchiara. Tracking social groups within and across cameras. *IEEE Trans. Circuits Syst. Video Techn.*, 27(3):441–453, 2017.

[22] Dzung L Pham, Chenyang Xu, and Jerry L Prince. A survey of current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 1998.

[23] Piotr Bojanowski, Rémi Lajugie, Francis R. Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, pages 628–643, 2014.

[24] Waqas Sultani and Mubarak Shah. Automatic action annotation in weakly labeled videos. *CoRR*, abs/1605.08125, 2016.

[25] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, pages 4575–4583, 2016.

[26] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis R. Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *ICCV*, pages 4462–4470, 2015.

[27] Armand Joulin, Kevin D. Tang, and Fei-Fei Li. Efficient image and video co-localization with frank-wolfe algorithm. In *ECCV*, pages 253–268, 2014.

[28] Guillaume Seguin, Piotr Bojanowski, Rémi Lajugie, and Ivan Laptev. Instance-level video segmentation from object tracks. In *CVPR*, pages 3678–3687, 2016.

[29] Saturnino Maldonado-Bascón, Sergio Lafuente-Arroyo, Pedro Gil-Jiménez, Hilario Gómez-Moreno, and Francisco López-Ferreras. Road-sign detection and recognition based on support vector machines. *IEEE Trans. Intelligent Transportation Systems*, 8(2):264–278, 2007.

[30] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In *Medical Image Understanding and Analysis - 21st Annual Conference, MIUA 2017, Edinburgh, UK, July 11-13, 2017, Proceedings*, pages 506–517, 2017.

[31] Guo-Qing Wei, Klaus Arbter, and Gerd Hirzinger. Automatic tracking of laparoscopic instruments by color coding. In *CVRMed-MRCAS'97, First Joint Conference Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery, Grenoble, France, March 19-22, 1997, Proceedings*, pages 357–366, 1997.

[32] Assaf Cohen, Ehud Rivlin, Ilan Shimshoni, and Edmond Sabo. Memory based active contour algorithm using pixel-level classified images for colon crypt segmentation. *Comp. Med. Imag. and Graph.*, 43:150–164, 2015.

[33] Chengquan Huang, Larry Davis, and John Townshend. An assessment of support vector machines for land cover classification. *International Journal of remote sensing*, 23(4):725–749, 2002.

[34] Massimiliano Pavan and Marcello Pelillo. A new graph-theoretic approach to clustering and segmentation. In *CVPR*, pages 145–152, 2003.

[35] Massimiliano Pavan and M.Pelillo. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):167–172, 2007.

[36] Khai N. Tran, Xu Yan, Ioannis A. Kakadiaris, and Shishir K. Shah. A group contextual model for activity recognition in crowded scenes. In *VISAPP 2015.*, pages 5–12, 2015.

[37] Yonatan T. Tesfaye, Eyasu Zemene, Marcello Pelillo, and Andrea Prati. Multi-object tracking using dominant sets. *IET computer vision*, 10:289–298, 2016.

[38] Jörgen W Weibull. *Evolutionary Game Theory.* MIT press, 1995.

[39] Andrea Torsello, Samuel Rota Bulò, and Marcello Pelillo. Grouping with asymmetric affinities: A game-theoretic perspective. In *(CVPR 2006)*, pages 292–299, 2006.

[40] Samuel Rota Bulò and Marcello Pelillo. A game-theoretic approach to hypergraph clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6):1312–1327, 2013.

[41] Eyasu Zemene Mequanint, Samuel Rota Bulò, and Marcello Pelillo. Dominant-set clustering using multiple affinity matrices. In *SIMBAD*, pages 186–198, 2015.

[42] Marcello Pelillo. What is a cluster? perspectives from game theory. In *Proc. of the NIPS Workshop on Clustering Theory*, 2009.

[43] Massimiliano Pavan and Marcello Pelillo. Dominant sets and hierarchical clustering. In *ICCV*, pages 362–369, 2003.

[44] Massimiliano Pavan and Marcello Pelillo. Efficient out-of-sample extension of dominant-set clusters. pages 1057–1064, 2004.

[45] Andrea Torsello, Samuel Rota Bulò, and Marcello Pelillo. Beyond partitions: Allowing overlapping groups in pairwise clustering. In *ICPR*, pages 1–4, 2008.

[46] David G Luenberger and Yinyu Ye. *Linear and Nonlinear Programming.* Springer, New York, 2008.

[47] Roger A Horn and Charles R Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.

[48] Josef Hofbauer and Karl Sigmund. *Evolutionary Games and Population Dynamics.* Cambridge University Press, 1998.

[49] Samuel Rota Bulò, Marcello Pelillo, and Immanuel M. Bomze. Graph-based quadratic optimization: A fast evolutionary approach. *Computer Vision and Image Understanding*, 115(7):984–995, 2011.

[50] Hairong Liu, Longin Jan Latecki, and Shuicheng Yan. Fast detection of dense subgraphs with iterative shrinking and expansion. 35(9):2131–2142, 2013.

[51] Immanuel M. Bomze. Branch-and-bound approaches to standard quadratic optimization problems. *J. Global Optimization*, 22(1-4):17–37, 2002.

[52] Lingyang Chu, Shuhui Wang, Siyuan Liu, Qingming Huang, and Jian Pei. ALID: scalable dominant cluster detection. *Conference on Very Large DataBases (VLDB)*, 8(8):826–837, 2015.

[53] Eyasu Zemene and Marcello Pelillo. Interactive image segmentation using constrained dominant sets. In *ECCV 2016*, pages 278–294, 2016.

[54] David J. Johnson and Michael A. Trick, editors. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Workshop, October 11-13, 1993.* American Mathematical Society, Boston, MA, USA, 1996.

[55] Samuel Rota Bulò, Andrea Torsello, and Marcello Pelillo. A continuous-based approach for partial clique enumeration. In *Graph-Based Representations in Pattern Recognition, 6th IAPR-TC-15*, pages 61–70, 2007.

[56] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.

[57] Douglas Hawkins. Identification of outliers. *Chapman and Hall,London*, 1980.

[58] Peter J. Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, 1998.

[59] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB, pages 392–403, 1998.

[60] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, May 2000.

[61] Sanjay Chawla and Pei Sun. SLOM: a new measure for local spatial outliers. *Knowl. Inf. Syst.*, 9(4):412–429, 2006.

[62] Ke Chen. A constant factor approximation algorithm for k-median clustering with outliers. SODA, pages 826–835, 2008.

[63] Sanjay Chawla and Aristides Gionis. k-means-: A unified approach to clustering and outlier detection. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA.*, pages 189–197, 2013.

[64] Lionel Ott, Linsey Xiaolin Pang, Fabio Tozeto Ramos, and Sanjay Chawla. On integrated clustering and outlier detection. In *Neural Information Processing Systems, 8-13, Montreal, Canada*, pages 1359–1367, 2014.

[65] Hong Chang and Dit-Yan Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41(1):191–203, 2008.

[66] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL,Prague, Czech Republic*, pages 410–420, 2007.

[67] Andrew J. Frank and Arthur Asuncion. Uci machine learning repository. 2010.

[68] Marcello Pelillo. What is a cluster? perspectives from game theory. *Proc. of the NIPS Workshop on Clustering Theory*, 2009.

[69] Bernd Fischer and Joachim M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):513–518, 2003.

[70] Bernd Fischer and Joachim M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):513–518, 2003.

[71] Bernd Fischer and Joachim M. Buhmann. Bagging for path-based clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(11):1411–1415, 2003.

[72] Hong Chang and Dit-Yan Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41(1):191–203, 2008.

[73] Morteza Haghir Chehreghani. *Information-Theoretic Validation of Clustering Algorithms.* PhD thesis, ETH ZURICH, 2013.

[74] Moshe Lichman. UCI machine learning repository, 2013.

[75] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[76] Michal Daszykowski, Beata Walczak, and DL Massart. Looking for natural patterns in data: Part 1. density-based approach. *Chemometrics and Intelligent Laboratory Systems*, 56(2):83–92, 2001.

[77] Lihi Zelnik-manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 1601–1608. MIT Press, 2004.

[78] Richard Szeliski. *Computer Vision: Algorithms and Applications.* Springer-Verlag, 2011.

[79] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach.* Pearson, 2011.

[80] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. “Grabcut”: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[81] Jiajun Wu, Yibiao Zhao, Jun-Yan Zhu, Siwei Luo, and Zhuowen Tu. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*, pages 256–263, 2014.

[82] Xue Bai and Guillermo Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *Int. J. Computer Vision*, 82(2):113–132, 2009.

[83] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3), 2004.

[84] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, pages 105–112, 2001.

[85] Eric N. Mortensen and William A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384, 1998.

[86] Hongkai Yu, Youjie Zhou, Hui Qian, Min Xian, Yuewei Lin, Dazhou Guo, Kang Zheng, Kareem Abdelfatah, and Song Wang. Loosecut: Interactive image segmentation with loosely bounded boxes. *CoRR*, abs/1507.03060, 2015.

[87] Min Xian, Yingtao Zhang, H. D. Cheng, Fei Xu, and Jianrui Ding. Neutro-connectedness cut. *CoRR*, abs/1512.06285.

[88] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation. In *CVPR*, pages 392–399, 2014.

[89] Suyog Dutt Jain and Kristen Grauman. Predicting sufficient annotation strength for interactive foreground segmentation. In *ICCV*, pages 1313–1320, 2013.

[90] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *(CVPR*, pages 993–1000, 2006.

[91] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, pages 1943–1950, 2010.

[92] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *CVPR*, pages 542–549, 2012.

[93] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, pages 3169–3176, 2010.

[94] Xingping Dong, Jianbing Shen, Ling Shao, and Ming-Hsuan Yang. Interactive cosegmentation using global and local energy optimization. *IEEE Trans. Image Processing*, pages 3966–3977, 2015.

[95] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, pages 654–661, 2005.

[96] Jingdong Wang, Yangqing Jia, Xian-Sheng Hua, Changshui Zhang, and Long Quan. Normalized tree partitioning for image segmentation. In *CVPR*, 2008.

[97] Jianxiong Xiao and Long Quan. Multiple view semantic segmentation for street view images. In *ICCV*, pages 686–693, 2009.

[98] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, 2012.

[99] Youjie Zhou, Lili Ju, and Song Wang. Multiscale superpixels and supervoxels based on hierarchical edge-weighted centroidal voronoi tessellation. In *WACV*, pages 1076–1083, 2015.

[100] Thomas K. Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Computer Vision*, 43(1):29–44, 2001.

[101] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2004.

[102] Hongliang Li, Fanman Meng, and King Ngi Ngan. Co-salient object detection from multiple images. *IEEE Trans. Multimedia*, 15(8):1896–1909, 2013.

[103] Meng Tang, Ismail Ben Ayed, and Yuri Boykov. Pseudo-bound optimization for binary energies. In *ECCV*, pages 691–707, 2014.

[104] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *IEEE International Conference on Computer Vision, ICCV*, pages 1769–1776, 2013.

[105] Brian L. Price, Bryan S. Morse, and Scott Cohen. Geodesic graph cut for interactive image segmentation. In *CVPR*, pages 3161–3168, 2010.

[106] Wenxian Yang, Jianfei Cai, Jianmin Zheng, and Jiebo Luo. User-friendly interactive image segmentation through unified combinatorial user inputs. *IEEE Trans. Image Processing*, 19(9):2470–2479, 2010.

[107] Kevin McGuinness and Noel E. O'Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010.

[108] Leo Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006.

[109] Olivier Duchenne, Jean-Yves Audibert, Renaud Keriven, Jean Ponce, and Florent Ségonne. Segmentation by transduction. In *CVPR*, 2008.

[110] Subhransu Maji, Nisheeth K. Vishnoi, and Jitendra Malik. Biased normalized cuts. In *CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 2057–2064, 2011.

[111] Philippe Salembier and Luis Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Trans. Image Processing*, 9(4):561–576, 2000.

[112] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(6):641–647, 1994.

[113] Gerald Friedland, Kristian Jantz, and Raúl Rojas. SIOX: Simple interactive object extraction in still images. In *(ISM*, pages 253–260, 2005.

[114] Jiangyu Liu, Jian Sun, and Heung-Yeung Shum. Paint selection. *ACM Trans. Graph.*, 28(3), 2009.

[115] Ozan Sener, Kemal Ugur, and A. Aydin Alatan. Error-tolerant interactive image segmentation using dynamic and iterated graph-cuts. In *IMMPD@ACM Multimedia*, pages 9–16, 2012.

[116] Kartic Subr, Sylvain Paris, Cyril Soler, and Jan Kautz. Accurate binary image selection from inaccurate user input. *Comput. Graph. Forum*, 32(2):41–50, 2013.

[117] Bryan C. Catanzaro, Bor-Yiing Su, Narayanan Sundaram, Yunsup Lee, Mark Murphy, and Kurt Keutzer. Efficient, high-quality image contour detection. In *ICCV*, pages 2381–2388, 2009.

[118] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, pages 2217–2224, 2011.

[119] Avik Hati, Subhasis Chaudhuri, and Rajbabu Velmurugan. Image co-segmentation using maximum common subgraph matching and region co-growing. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, pages 736–752, 2016.

[120] Morteza Haghir Chehreghani. Adaptive trajectory analysis of replicator dynamics for data clustering. *Machine Learning*, 104(2-3):271–289, 2016.

[121] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *CVPR*, pages 2814–2821, 2014.

[122] Hongliang Li and King Ngi Ngan. A co-saliency model of image pairs. *IEEE Trans. Image Processing*, 20(12):3365–3375, 2011.

[123] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, pages 1939–1946, 2013.

[124] Huazhu Fu, Xiaochun Cao, and Zhuowen Tu. Cluster-based co-saliency detection. *IEEE Trans. Image Processing*, 22(10):3766–3778, 2013.

[125] Chulwoo Lee, Won-Dong Jang, Jae-Young Sim, and Chang-Su Kim. Multiple random walkers and their application to image cosegmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3837–3845, 2015.

[126] Xiaochun Cao, Zhiqiang Tao, Bao Zhang, Huazhu Fu, and Wei Feng. Self-adaptively weighted co-saliency detection via rank constraint. *IEEE Trans. Image Processing*, 23(9):4175–4186, 2014.

[127] Eyasu Zemene, Leulseged Tesfaye Alemu, and Marcello Pelillo. Constrained dominant sets for retrieval. In *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, pages 2568–2573, 2016.

[128] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc J. Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *IEEE, CVPR*, pages 777–784, 2011.

[129] Michael E. Houle, Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Can shared-neighbor distances defeat the curse of dimensionality? In *SSDBM*, pages 482–500, 2010.

[130] Amir Egozi, Yosi Keller, and Hugo Guterman. Improving shape retrieval by spectral matching and meta similarity. *IEEE Trans. Image Processing*, 19(5):1319–1327, 2010.

[131] Peter Kontschieder, Michael Donoser, and Horst Bischof. Beyond pairwise shape similarity analysis. In *ACCV*, pages 655–666, 2009.

[132] Michael Donoser and Horst Bischof. Diffusion processes for retrieval revisited. In *IEEE, CVPR*, pages 1320–1327, 2013.

[133] Xingwei Yang and Longin Jan Latecki. Affinity learning on a tensor product graph with applications to shape and image retrieval. In *IEEE, CVPR*, pages 2369–2376, 2011.

[134] Xingwei Yang, Lakshman Prasad, and Longin Jan Latecki. Affinity learning with diffusion on tensor product graph. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):28–38, 2013.

[135] Stéphane Lafon and Ann B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1393–1403, 2006.

[136] Martin Szummer and Tommi S. Jaakkola. Partially labeled classification with markov random walks. In *NIPS*, pages 945–952, 2001.

[137] Xingwei Yang, Suzan Köknar-Tezel, and Longin Jan Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *IEEE, CVPR*, pages 357–364, 2009.

[138] Hairong Liu, Xingwei Yang, Longin Jan Latecki, and Shuicheng Yan. Dense neighborhoods on affinity graph. *IJCV*, 98(1):65–82, 2012.

[139] Bo Wang and Zhuowen Tu. Affinity learning via self-diffusion for image segmentation and clustering. In *IEEE CVPR*, pages 2312–2319, 2012.

[140] Cho Minsu and MuLee Kyoung. Authority-shift clustering: Hierarchical clustering by authority seeking on graphs. In *IEEE CVPR*, pages 3193–3200, 2010.

[141] Raghuraman Gopalan, Pavan K. Turaga, and Rama Chellappa. Articulation-invariant representation of non-planar shapes. In *ECCV*, pages 286–299, 2010.

[142] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE TPAMI*, 29(2):286–299.

[143] Xiang Bai, Xingwei Yang, Longin Jan Latecki, Wenyu Liu, and Zhuowen Tu. Learning context-sensitive shape similarity by graph transduction. *IEEE TPAMI*, 32(5):861–874, 2010.

[144] Haibin Ling, Xingwei Yang, and Longin Jan Latecki. Balancing deformability and discriminability for shape matching. In *ECCV*, pages 411–424, 2010.

[145] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997.

[146] Jiayan Jiang, Bo Wang, and Zhuowen Tu. Unsupervised metric learning by self-smoothing operator. In *IEEE, ICCV*, pages 794–801, 2011.

[147] Yinghao Cai, Kaiqi Huang, and Tieniu Tan. Human appearance matching across multiple non-overlapping cameras. pages 1–4, 2008.

[148] Amit Chilgunde, Pankaj Kumar, Surendra Ranganath, and Weimin Huang. Multi-camera target tracking in blind regions of cameras with non-overlapping fields of view. pages 1–10, 2004.

[149] Omar Javed, Zeeshan Rasheed, Khurram Shafique, and Mubarak Shah. Tracking across multiple cameras with disjoint views. pages 952–957, 2003.

[150] Youlu Wang, Senem Velipasalar, and Mustafa Cenk Gursoy. Distributed wide-area multi-object tracking with non-overlapping camera views. *Multimedia Tools and Applications*, 73(1):7–39, 2014.

[151] Xiaojing Chen and Bir Bhanu. Integrating social grouping for multi-target tracking across cameras in a crf model. in press.

[152] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. pages 343–356, 2012.

[153] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. GMMCP tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. pages 4091–4099, 2015.

[154] Nadeem Anjum and Andrea Cavallaro. Trajectory association and fusion across partially overlapping cameras. pages 201–206, 2009.

[155] Simone Calderara, Rita Cucchiara, and Andrea Prati. Bayesian-competitive consistent labeling for people surveillance. 30(2), 2008.

[156] Carlos R. del-Blanco, Raúl Mohedano, Narciso N. García, Luis Salgado, and Fernando Jaureguizar. Color-based 3d particle filtering for robust tracking in heterogeneous environments. pages 1–10, 2008.

[157] Sohaib Khan and Mubarak Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. 25(10):1355–1360, 2003.

[158] Birgit Möller, Thomas Plötz, and Gernot A. Fink. Calibration-free camera hand-over for fast and reliable person tracking in multi-camera setups. pages 1–4, 2008.

[159] Senem Velipasalar, Jason Schlessman, Cheng-Yao Chen, Wayne Hendrix Wolf, and Jaswinder Singh. A scalable clustered camera system for multiple object tracking. *EURASIP J. Image and Video Processing*, 2008, 2008.

[160] Harold W Kuhn. Variants of the hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 3(4):253–258, 1956.

[161] Samuel Rota Bulò, Marcello Pelillo, and Immanuel M. Bomze. Graph-based quadratic optimization: A fast evolutionary approach. *Computer Vision and Image Understanding*, 115(7):984–995, 2011.

[162] Omar Javed, Khurram Shafique, Zeeshan Rasheed, and Mubarak Shah. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2):146–162, 2008.

[163] Andrew Gilbert and Richard Bowden. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. pages 125–136, 2006.

[164] Bryan James Prosser, Shaogang Gong, and Tao Xiang. Multi-camera matching using bi-directional cumulative brightness transfer functions. pages 1–10, 2008.

[165] Tiziana D'Orazio, Pier Luigi Mazzeo, and Paolo Spagnolo. Color brightness transfer function evaluation for non overlapping multi camera tracking. pages 1–6, 2009.

[166] Satyam Srivastava, Ka Ki Ng, and Edward J. Delp. Color correction for object tracking across multiple cameras. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1821–1824, 2011.

[167] De Cheng, Yihong Gong, Jinjun Wang, Qiqi Hou, and Nanning Zheng. Part-aware trajectories association across non-overlapping uncalibrated cameras. *Neurocomputing*, 230:30–39, 2017.

[168] Cheng-Hao Kuo, Chang Huang, and Ram Nevatia. Inter-camera association of multi-target tracks by on-line learned appearance affinity models. pages 383–396, 2010.

[169] Yue Gao, Rongrong Ji, Longfei Zhang, and Alexander G. Hauptmann. Symbiotic tracker ensemble toward A unified tracking framework. 24(7):1122–1131, 2014.

[170] Shu Zhang, Yingying Zhu, and Amit K. Roy-Chowdhury. Tracking multiple interacting targets in a camera network. *Computer Vision and Image Understanding*, 134:64–73, 2015.

[171] Yinghao Cai and Gérard G. Medioni. Exploring context information for inter-camera multiple target tracking. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 761–768, 2014.

[172] Xiaotang Chen, Kaiqi Huang, and Tieniu Tan. Object tracking across non-overlapping views by learning inter-camera transfer models. *Pattern Recognition*, 47(3):1126–1137, 2014.

[173] Ergys Ristani and Carlo Tomasi. Tracking multiple people online and in real time. In *Asian Conference on Computer Vision*, pages 444–459. Springer, 2014.

[174] Jinjie You, Ancong Wu, Xiang Li, and Wei-Shi Zheng. Top-push video-based person re-identification. pages 1345–1353, 2016.

[175] Niall McLaughlin, Jesús Martínez del Rincón, and Paul C. Miller. Recurrent convolutional network for video-based person re-identification. pages 1325–1334, 2016.

[176] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. Person re-identification by video ranking. pages 688–703, 2014.

[177] Dung Nghi Truong Cong, Catherine Achard, Louahdi Khoudour, and Lounis Douadi. Video sequences association for people re-identification across multiple non-overlapping cameras. In *IAPR International Conference on Image Analysis and Processing (ICIAP)*, pages 179–189, 2009.

[178] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016.

[179] Francesco Solera, Simone Calderara, Ergys Ristani, Carlo Tomasi, and Rita Cucchiara. Tracking social groups within and across cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[180] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. 32(9):1627–1645, 2010.

[181] Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2005.

[182] Martin Köstinger, Martin Hirzer, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. pages 2288–2295, 2012.

[183] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *European Conference on Computer Vision*, pages 255–268. Springer, 2010.

[184] Samuel Rota Bulò and Immanuel M. Bomze. Infection and immunization: A new class of evolutionary game dynamics. *Games and Economic Behavior*, 71(1):193–211, 2011.

[185] Yannis Avrithis, Yannis Kalantidis, Giorgos Tolias, and Evaggelos Spyrou. Retrieving landmark and non-landmark images from community photo collections. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 153–162. ACM, 2010.

[186] David M Chen, Georges Baatz, Kevin Köser, Sam S Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 737–744. IEEE, 2011.

[187] Till Quack, Bastian Leibe, and Luc Van Gool. World-scale mining of objects and events from community photo collections. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 47–56. ACM, 2008.

[188] Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: building a web-scale landmark recognition engine. In *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on*, pages 1085–1092. IEEE, 2009.

[189] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Predicting good features for image geo-localization using per-bundle vlad. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1170–1178, 2015.

[190] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[191] James Hays and Alexei A Efros. Large-scale image geolocalization. In *Multimodal Location Estimation of Videos and Images*, pages 41–62. Springer, 2015.

[192] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. *arXiv preprint arXiv:1602.05314*, 2016.

[193] Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. I know what you did last summer: object-level auto-annotation of holiday snaps. In *2009 IEEE 12th International Conference on Computer Vision*, pages 614–621. IEEE, 2009.

[194] Edward Johns and Guang-Zhong Yang. From images to scenes: Compressing an image cluster into a single scene model for place recognition. In *2011 International Conference on Computer Vision*, pages 874–881. IEEE, 2011.

[195] David J Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM, 2009.

[196] Grant Schindler, Matthew A. Brown, and Richard Szeliski. City-scale location recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, Minnesota*, 2007.

[197] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1582–1590, 2016.

[198] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5297–5307, 2016.

[199] Akihiko Torii, Josef Sivic, Masatoshi Okutomi, and Tomás Pajdla. Visual place recognition with repetitive structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(11):2346–2359, 2015.

[200] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2704–2712, 2015.

[201] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th international conference on computer vision*, pages 72–79. IEEE, 2009.

[202] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *European Conference on Computer Vision*, pages 15–29. Springer, 2012.

[203] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE, 2011.

[204] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *European Conference on Computer Vision*, pages 752–765. Springer, 2012.

[205] Evangelos Kalogerakis, Olga Vesselova, James Hays, Alexei A Efros, and Aaron Hertzmann. Image sequence geolocation with human travel priors. In *2009 IEEE 12th International Conference on Computer Vision*, pages 253–260. IEEE, 2009.

[206] Gonzalo Vaca-Castano, Amir Roshan Zamir, and Mubarak Shah. City scale geo-spatial trajectory estimation of a moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1186–1193. IEEE, 2012.

[207] Asaad Hakeem, Roberto Vezzani, Mubarak Shah, and Rita Cucchiara. Estimating geospatial trajectory of a moving camera. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 82–87. IEEE, 2006.

[208] Chao-Yeh Chen and Kristen Grauman. Clues from the beaten path: Location estimation with bursty sequences of tourist photos. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1569–1576. IEEE, 2011.

[209] Tsung-Yi Lin, Serge Belongie, and James Hays. Cross-view image geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2013.

[210] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5007–5015. IEEE, 2015.

[211] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3961–3969, 2015.

[212] Nathan Jacobs, Scott Satkin, Nathaniel Roman, Richard Speyer, and Robert Pless. Geolocating static cameras. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–6. IEEE, 2007.

[213] Srikumar Ramalingam, Sofien Bouaziz, Peter Sturm, and Matthew Brand. Skyline2gps: Localization in urban canyons using omni-skylines. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3816–3823. IEEE, 2010.

[214] Relja Arandjelović and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *Asian Conference on Computer Vision*, pages 188–204. Springer, 2014.

[215] Alessandro Bergamo, Sudipta N Sinha, and Lorenzo Torresani. Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 763–770, 2013.

[216] Song Cao and Noah Snavely. Graph-based discriminative learning for location recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 700–707, 2013.

[217] Petr Gronat, Guillaume Obozinski, Josef Sivic, and Tomas Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 907–914, 2013.

[218] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 883–890, 2013.

[219] Qiang Hao, Rui Cai, Zhiwei Li, Lei Zhang, Yanwei Pang, and Feng Wu. 3d visual phrases for landmark recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3594–3601. IEEE, 2012.

[220] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP 2009 - Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, February 5-8, 2009 - Volume 1*, pages 331–340, 2009.

[221] Reiner Horst, Panos M. Pardalos, and Nguyen Van Thoai. *Introduction to Global Optimization.* Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht/Boston/London, 2000.

[222] Immanuel M. Bomze. On standard quadratic optimization problems. *J. Global Optimization*, 13(4):369–387, 1998.

[223] Liang Zheng, Shengjin Wang, Lu Tian, Fei He, Ziqiong Liu, and Qi Tian. Query-adaptive late fusion for image search and person re-identification. In *CVPR*, pages 1741–1750, 2015.

[224] Shaoting Zhang, Ming Yang, Timothée Cour, Kai Yu, and Dimitris N. Metaxas. Query specific rank fusion for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(4):803–815, 2015.

[225] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[226] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[227] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. 2016.

[228] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *ECCV*, pages 3–20, 2016.

[229] Coen Bron and Joep Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, 1973.