Graph-based Inference with Constraints for Object Detection and Segmentation

---

A Dissertation
Submitted to
the Temple University Graduate Board

---

in Partial Fulfillment
of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY

---

by
Tianyang Ma
August, 2013

Examining Committee Members:

Longin Jan Latecki, Advisory Chair, Computer and Information Sciences
Haibin Ling, Computer and Information Sciences
Slobodan Vucetic, Computer and Information Sciences
Xiaolei Huang, External Member, Lehigh University

Abstract

For many fundamental problems of computer vision, adopting a graph-based framwork can be straight-forward and very effective. In this thesis, I propose several graph-based inference methods tailored for different computer vision applications. It starts from studying contour-based object detection methods. Compared to other image cues, the outline contour (silhouette) is invariant to lighting conditions and variations in object color and texture. More importantly, it can efficiently represent image structures with large spatial extents. Because of these advantages, contour information is widely used in object detection and recognition methods. However, the contour-based methods mainly suffer from the fact that the contour is not very distinctive and informative, especially when considered locally. We made several efforts to address this problem. The first effort we made is not directly related to graph-based modeling but rather to increase the distinctness of contour matching. We propose a novel technique that significantly improves the performance of oriented chamfer matching on images with cluttered background. Different to other matching methods, which only measures how well a template fits to an edge map, we evaluate the score of the template in comparison to auxiliary contours, which we call normalizers. We utilize AdaBoost to learn a Normalized Oriented Chamfer Distance (NOCD). Our experimental results demonstrate that it boosts the detection rate of the oriented chamfer distance. The simplicity and ease of training of NOCD on a small number of training samples promise that it can replace chamfer distance and oriented chamfer distance in any template matching application.

While this method can significantly reduce the number of false alarms, the object is still represented by a star-model (a spatial case of graph-based object representation), and hough voting method is adopted to perform the inference. Theoretically this method is still prone to clutter background because no effort has been made to accurately cut and match the contours. We propose a novel framework for contour based object detection, by replacing the hough-voting framework with finding dense subgraph inference. Compared to

previous work, we propose a novel shape matching scheme suitable for partial matching of edge fragments. The shape descriptor has the same geometric units as shape context but our shape representation is not histogram based. The key contribution is that we formulate the grouping of partial matching hypotheses to object detection hypotheses is expressed as maximum clique inference on a weighted graph. Consequently, each detection result not only identifies the location of the target object in the image, but also provides a precise location of its contours, since we transform a complete model contour to the image. We achieve very competitive results on ETHZ dataset, obtained in a pure shape-based framework, demonstrate that our method achieves not only accurate object detection but also precise contour localization on cluttered background.

Similar to the task of grouping of partial matches in the contour-based method, in many computer vision problems, we would like to discover certain pattern among a large amount of data. For instance, in the application of unsupervised video object segmentation, where we need automatically identify the primary object and segment the object out in every frame. We propose a novel formulation of selecting object region candidates simultaneously in all frames as finding a maximum weight clique in a weighted region graph. The selected regions are expected to have high objectness score (unary potential) as well as share similar appearance (binary potential). Since both unary and binary potentials are unreliable, we introduce two types of mutex (mutual exclusion) constraints on regions in the same clique: intra-frame and inter-frame constraints. Both types of constraints are expressed in a single quadratic form. An efficient algorithm is applied to compute the maximal weight cliques that satisfy the constraints. We apply our method to challenging benchmark videos and obtain very competitive results that outperform state-of-the-art methods. We also show that the same maximum weight subgraph with mutex constraints formulation can be used to solve various computer vision problems, such as points matching, solving image jigsaw puzzle, and detecting object using 3D contours.

Graph-based modeling can be also the foundation in semi-supervised learning framework. We propose an approach based on standard graph transduction, semi-supervised learning (SSL) framework. Its key novelty is the integration of global connectivity constraints into this framework. Although connectivity leads to higher order constraints and their number is an exponential, finding the most violated connectivity constraint can be done efficiently in polynomial time. Moreover, each such constraint can be represented as a linear inequality. Based on this fact, we design a cutting-plane algorithm to solve the integrated problem. It iterates between solving a convex quadratic problem of label propagation with linear inequality constraints, and finding the most violated constraint. We demonstrate the benefits of the proposed approach on a realistic and very challenging problem of cosegmentation of multiple foreground objects in photo collections in which the foreground objects are not present in all photos. The obtained results not only demonstrate performance boost induced by the connectivity constraints, but also show a significant improvement over the state-of-the-art methods.

Acknowledgements

It has been a wonderful journey for me, when i look over the past four years studying at Temple University. It is also of great joy reminding me all the friends and family who have helped and supported me along this long but fulfilling road.

I would like to express my heartfelt gratitude to my advisor, Professor Longin Jan Latecki. I have learned so much from him over the past four years. His insights and suggestions helped to improve my research skills. He always encouraged me to explore research topics which i am interested in. He has been inspirational, supportive and patient. Other professors at Temple that I am lucky enough to learn from and interact with: Dr. Haibin Ling and Dr. Slobodan Vucetic. Both are my committee members. I am fortunate enough to have the chance to work together with my lab-mates and friends: Xingwei Yang, Xinggang Wang, Nan Li, Meng Yi, Zhuo Deng, Yinfei Yang. There are also many friends at Temple who makes my life colorful: Erkang Cheng, Liang Du, Chengliang Wang, Zhuang Wang, Qiang Lou, Liang Lan, Haidong Shi, Yi Wu, Yunsheng Wang, Yu Pang, Pengpeng Liang, Xin Li.

I would also like to thank the mentors of my undergraduate study at HUST: Dr. Wenyu Liu and Dr. Xiang Bai who initially inspired me on computer vision research. Also I enjoyed many discussions with Quannan Li, Hairong Liu and Cong Yao.

I would not have contemplated this road if not for my parents, Longqiu Ma and Luchun Song. They have done so much for me, and been always supportive. And finally i would like to thank Le Shu, for her love and care.

To my parents and Le.

Contents

List of Tables

List of Figures

Chapter 1

Contour-based Object Detection

1.1 Boosting Chamfer Matching by Learning Chamfer Distance Normalization

1.1.1 Introduction

Chamfer matching has been widely used for edge based object detection and recognition in computer vision. However, its performance is seriously limited in cluttered images. One of the main drawbacks of chamfer matching is the fact that a given template often fits better to a cluttered background than to the location of a true target object. Oriented chamfer matching (OCD) [127, 125] adds orientation information, which significantly improves the performance of chamfer matching, but the problem still remains, as illustrated in Fig. 1.1. The proposed approach provides a solution to this problem by comparing the matching score of the template to normalizers, which are curve segments of varying but simple shape. There are two key properties of the normalizers. (1) If the target template matches well to a cluttered background, then very likely some of the normalizers match well too. (2) If the template matches well to a true object location, it is very unlikely for any normalizer to match well. Consequently, the normalized oriented chamfer distance (NOCD) significantly improves the discriminative power of OCD. Some examples are shown in Fig. 1.1.

Since it is hard if not impossible to satisfy (1) and (2) with a finite set of normalizers for a given set of target templates, we treat normalized chamfer distances as weak classifiers and employ AdaBoost to learn their weights. The weights provide a soft way of selecting adequate normalizers for a given template. As our experimental results demonstrate, AdaBoost is able to learn the normalizer weights on a small set of training images, which makes the proposed approach suitable for all practical applications currently based on (oriented) chamfer matching.

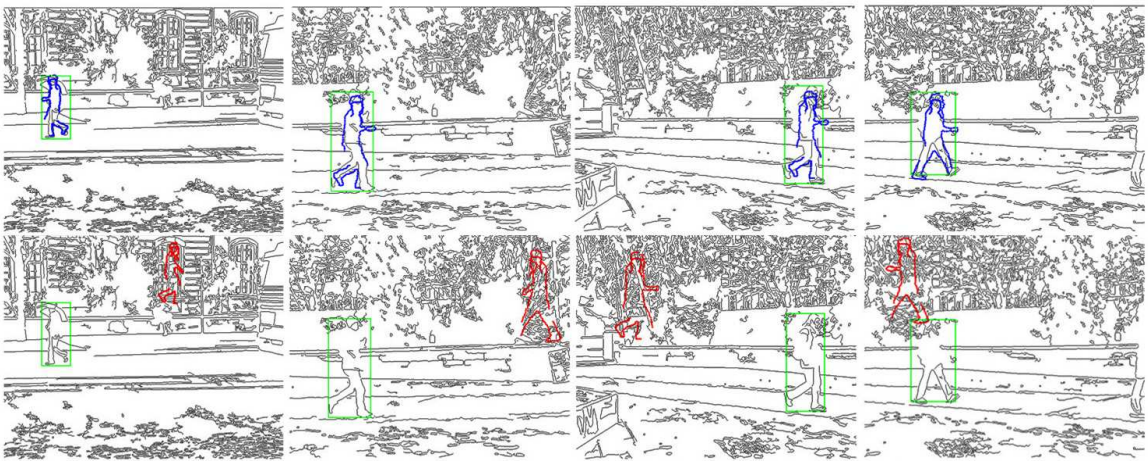

Figure 1.1: Example detection results on 250 test images from TU Darmstadt Pedestrian Dataset. The first row shows the detection results of the proposed NOCD, while the second row shows oriented chamfer matching results. The green rectangle denotes the ground truth bounding box.

The section is structured as follows. In Section 1.1.3, we review basic definitions of chamfer distance and oriented chamfer distance. The new concept of distance normalization is introduced in Section 1.1.4. and AdaBoost learning of their weights is described in Section 1.1.5. Section 1.1.6 describes a simple framework for object detection. Finally, Section 1.1.7 introduces our set of normalizers. The performance of our method is evaluated and compared to OCD in Section 2.7.

1.1.2 Related work

There is a large number of applications of chamfer matching in computer vision and in medical image analysis. Chamfer distance was first introduced by Barrow et al. [5] in 1977 with a goal of matching two collections of contour fragments. Until today chamfer matching is widely used in object detection and classification task due to its tolerance to misalignment in position, scale and rotation. Borgefors [14] introduced a modified chamfer matching method called hierarchical chamfer matching, which could be regarded as a coarse-to-fine

process by matching edge points using a resolution pyramid of the image. This method focuses on alleviating the computational load for chamfer matching. Meanwhile, chamfer matching meets the real-time system requirement due to fast implementations of distance transforms. Gavrila and Munder [53] performed template matching based on chamfer distance transform as a core technique to construct a real-time detection system of pedestrians.

Leibe et al. [79] used chamfer matching to detect pedestrian in crowded scenes, and combined segmentation as a verification to prevent the false alarms that mostly lie in the cluttered background. Stenger et al. [131] introduced a template hierarchy which is formed by bottom-up clustering based on the chamfer distance. In [104], Opelt et al. used chamfer distance to score each boundary fragment for selection of candidate contour fragments. Opelt et al. also compared each boundary fragment from each category to all existing alphabet entries using chamfer distance in [105]. Other methods that utilize chamfer distance as shape similarity metric include [58, 153, 75]. Chamfer distance plays also an important role in medical image analysis, e.g., [143, 99, 41].

However, methods that utilize chamfer distance to measure the similarity between the template and edge maps suffer from mismatching to the cluttered background. It is generally agreed that main negative effect of using chamfer distance is the potential risk of increasing false alarms occurring in background with high level of clutter noise. Thayananthan et al. [134] compared the localization performance of chamfer matching and shape context [9], and concluded that chamfer matching is more robust in clutter than shape context matching even though most failure cases in chamfer matching are still due to false positive matches.

Recently, Shotton et al. [127, 125] proposed an oriented chamfer distance (OCD) that exploits edge orientation information in the form of edge gradients. OCD linearly combines chamfer distance and orientation difference between template points and their closest matches, which leads to reduction of mismatching cases to the noisy background. Trinh and Kimia [140] proposed Contour Chamfer Matching (CCM) to improve OCD. In this

method, based on the observation that the accidental alignment between a contour and the image edges always forms a zig-zagging contour, after finding the corresponding points in edge map, another orientation for edge points is computed based on the new generated curve, and an additional term which is the difference in tangent direction is taken into account when computing the Contour Chamfer Distance.

Since proposed method is not designed specifically for oriented chamfer distance, it could be also used to boost the performance of any distance metric that aims to capture edge support for a model. In particular, it would be possible to apply the proposed method to Hausdorff distance and oriented Hausdorff distance proposed in [62, 101], which is also widely used in computer vision applications. However, in [127] experimental evidence is provide that OCD has better performance than Hausdorff distance.

1.1.3 Oriented Chamfer Distance (OCD)

In this section we define chamfer distance and oriented chamfer distance (OCD), which is a simple linear combination between distance and orientation terms.

Chamfer Distance Chamfer distance was first proposed in [5] as an evaluation of 2D asymmetric distance between two set of edge points. It is tolerant to slight shape distortion caused by shift in location, scale and rotation. Given a template T positioned at location x in an image I and a binary edge map E of the image I, the basic form of chamfer distance is calculated as

$$d_{cham}^{(T,E)}(x) = \frac{1}{|T|} \sum_{x_t \in T} \min_{x_e \in E} ||(x_t + x) - x_e||_2 \ , \tag{1.1}$$

where $||.||_2$ is l_2 norm and $|T|$ denotes number of points in template T. Chamfer distance can be efficiently computed as:

$$d_{cham}^{(T,E)}(x) = \frac{1}{|T|} \sum_{x_t \in T} DT_E(x_t + x) \ , \tag{1.2}$$

where DT_E is a distance transform defined for every image point $x \in I$ as

$$DT_E(x) = \min_{x_e \in E} ||x - x_e||_2 \ . \tag{1.3}$$

Meanwhile, in practice, distance transform is truncated to a constant τ [127]:

$$DT_E^\tau(x) = \min(DT_E(x), \tau) \tag{1.4}$$

This reduces the negative effective due to missing edges in E, and allows normalization to a standard range $[0, 1]$:

$$d_{cham,\tau}^{(T,E)}(x) = \frac{1}{\tau|T|} \sum_{x_t \in T} DT_E^\tau(x_t + x) \ . \tag{1.5}$$

Oriented Chamfer Distance (OCD) Shotton et al. [127] proposed an improved chamfer distance called oriented chamfer distance (OCD), which adds additional robustness by exploiting edge orientation information. To define it, we first need a notation of an argument of a distance transform (ADT) that gives the locations of a closest point.

$$ADT_E(x) = \arg\min_{x_e \in E} ||x - x_e||_2. \tag{1.6}$$

To evaluate a mismatch in orientation, the difference in tangent directions is computed

$$d_{orient}^{(T,E)}(x) = \frac{2}{\pi|T|} \sum_{x_t \in T} |\phi(x_t) - \phi(ADT_E(x_t + x))| \ , \tag{1.7}$$

where $\phi(x)$ denotes tangent direction at point x and ranges between zero and π. $|\phi(x_1) - \phi(x_2)|$ gives the smallest circular difference between $\phi(x_1)$ and $\phi(x_2)$. Using a simple linear combination between the distance and orientation terms, oriented chamfer distance

is defined as

$$OCD_{\lambda}^{(T,E)}(x) = (1-\lambda)\cdot d_{cham,\tau}^{(T,E)}(x) + \lambda \cdot d_{orient}^{(T,E)}(x)\ . \tag{1.8}$$

For clarity, we will omit E and λ below when possible, and use $OCD(T,x) = OCD_{\lambda}^{(T,E)}(x)$ to represent the oriented chamfer distance of template T at location $x \in I$.

1.1.4 Normalization of Oriented Chamfer Distance

Although oriented chamfer matching adds orientation term to avoid mismatching, cluttered background still may match much better to the template than the real object contours. The reason is that cluttered background offers a large variety of edge orientations, consequently, any shape has a large probability of a good oriented chamfer score. This suggests that we need to compare the score of the target template with scores of some random shapes. If both have good OCD score at a given location, then the template match is most likely to be accidental. Based on this insight, we introduce a normalizer as an auxiliary, random shape to evaluate how well the template matches to the edge map at a certain location. For a target template T, we propose to generate K normalizers, denoted by $\mathcal{N} = \{\eta_k |\ k = 1,\ldots,K\}$. A procedure to generate normalizes is described in Section 1.1.7. Instead of only calculating $OCD(T,x)$ at each location x, we also compute $OCD(\eta_k,x)$, and compare the ratios

$$R_k(T,x) = \frac{OCD(T,x)}{OCD(\eta_k,x)}. \tag{1.9}$$

We call $R_k(T,x)$ a **normalized score**.

Now we provide some details about the role of normalizers in improving chamfer score. The analysis is divided into three qualitative cases that illustrate an intended correct

behavior of the normalizers. In practice, not all normalizers will behave in this way, which is addressed in Section 1.1.5.

Case 1: At a correct location containing a target object in a given image, $OCD(T, x)$ is small and $OCD(\eta_k, x)$ is large, so that $OCD(T, x) < OCD(\eta_k, x)$. Consequently, $R_k(T, x)$ will become comparatively smaller than $OCD(T, x)$, which better indicates a correct match.

Case 2: In a cluttered area in which the target object is not present, both $OCD(T, x)$ and $OCD(\eta_k, x)$ are small, but $OCD(T, x) > OCD(\eta_k, x)$, so $R_k(T, x)$ will become comparatively larger than $OCD(T, x)$, which better indicates a wrong match.

Case 3: In an area that is neither cluttered nor contains the target object, both $OCD(T, x)$ and $OCD(\eta_k, x)$ are large, but $OCD(T, x) > OCD(\eta_k, x)$, so $R_k(T, x)$ will become comparatively larger than $OCD(T, x)$, which better indicates a wrong match.

Cases 1 to 3 clearly demonstrate that normalizers increase the discriminate power of OCD. However, they are based on an assumption that we have an ideal set of normalizers $\{\eta_k | \, k = 1, \ldots, K\}$ behaving as described in cases 1 to 3. Even though it may not be possible to find normalizers satisfying cases 1 to 3 for a given template T, we propose to utilize machine learning methods to learn which normalizers yield correct scores $R_k(T, x)$ for a given template T. For a given set of candidate normalizers, we use AdaBoost in Section 1.1.5 to learn the weights of normalized scores $R_k(T, x)$. Thus, we treat each normalized score as a weak classifier. The weights provide a soft selection of a set of normalizers with our intuition being that this selection best approximates the behavior described in cases 1 to 3.

1.1.5 Learning Normalized OCD with AdaBoost

The standard AdaBoost [51] allows us to select a set of normalizers by assigning weights to their normalized scores and to combine them as a weighted linear combination, which yields a more robust matching score. Given is a set of training images with positive and

negative examples, i.e, a set of bounding boxes containing the target object and a set of bounding boxes without the target object. AdaBoost automatically learns the weight for each weak learner and combine them to form a strong learner [137, 149]. We use the ratios $R_k(T, x)$ as weak learners for $k = 1, \ldots, K$. To be precise, a weak learner is defined as

$$h_k(T, x) = \begin{cases} 1 & \text{for} \quad R_k(T, x) < th_k \\ 0 & \text{for} \quad otherwise. \end{cases} \tag{1.10}$$

In each iteration $1, \ldots, K$, we search for a weak learner with the best detection performance on the training set. During the search, the optimal threshold th_k for each weak learner is chosen to minimize the misclassification error (ME). At each iteration of AdaBoost, each training example carries a classification weight. ME is defined as the sum of the classification weights of misclassified training examples (both positives and negatives). As the output we obtain a strong learner

$$H(T, x) = \sum_{k=1}^{K} w_k \cdot h_k(T, x) \tag{1.11}$$

In the AdaBoost terminology, the value of the strong learner indicates how likely a given image location x belongs to the class of template T. The larger the value the most likely this is the case. We propose to replace the oriented chamfer distance of T with the value of $H(T, x)$. We define a **Normalized Oriented Chamfer Distance** as $NOCD(T, x) = H(T, x)$. While OCD is a distance in that the smaller is OCD value the better, NOCD is a similarity measure, i.e., the larger the NOCD value, the most likely the target object is present at location x.

We use a simple strategy to select training examples for AdaBoost. Given is a set of training images with ground truth bounding boxes enclosing target objects. For each training image we select only 5 positive and 5 negative examples. As 5 positive examples we randomly select 5 locations in a small neighborhood around the ground truth locations. We

select as negative examples 5 locations x with locally smallest oriented chamfer distance $OCD(T, x)$ such that the area of the intersection of the bounding box centered at x with any ground truth bounding box is less than 50%.

1.1.6 Object Detection with NOCD

In order to be able to evaluate the performance of NOCD, we describe a very simple approach for object detection in this section. We keep it simple to allow for clear comparison to OCD. However, we use a flexible shape model in our approach in order to be able to evaluate the performance of the proposed $NOCD$ on state-of-the-art test datasets.

Our flexible object model is denoted as $\mathcal{M} = \{B_i | \ i = 1, \ldots, N\}$, where B_i is a part bundle composed of contour parts describing the same location on the contour of a given shape class, e.g., human head or arm, and N is the number of bundles in model $\mathcal{M}$. Contour parts from bundle B_i are represented by c_{ij}, and hence $B_i = \{c_{ij} | j = 1, \ldots, M_i\}$. Since every part bundle B_i describes a specific part of an object, we assume that $B_i \bigcap B_j = \emptyset$ if $i \neq j$. Fig. 1.2 shows an example of human model, here $N = 4$ and $M_i = 5$ for $i = 1, 2, 3, 4$. Our model was manually constructed. Thus, our model contains the total of 20 contour parts c_{ij}. Each part c_{ij} is treated as template T, and $NOCD(c_{ij}, x)$ is learned as describe in Section 1.1.5.

For an input image I, we first use Canny edge detector to compute the edge map E. For each location x in I, we use $NOCD(c_{ij}, x)$ to represent the normalized oriented chamfer distance of model contour part c_{ij} placed at point x. With a simple but efficient sum-max framework, the model fit at point $x \in I$ is defined as:

$$S_I(\mathcal{M}, x) = \sum_{i=1}^{N} \max_{c_{ij} \in B_i} NOCD(c_{ij}, x) \ . \tag{1.12}$$

Thus, we select from each bundle B_i the part with the largest NOCD score and sum the maximal scores over the bundles in the shape model $\mathcal{M}$. Using sliding window we calculate

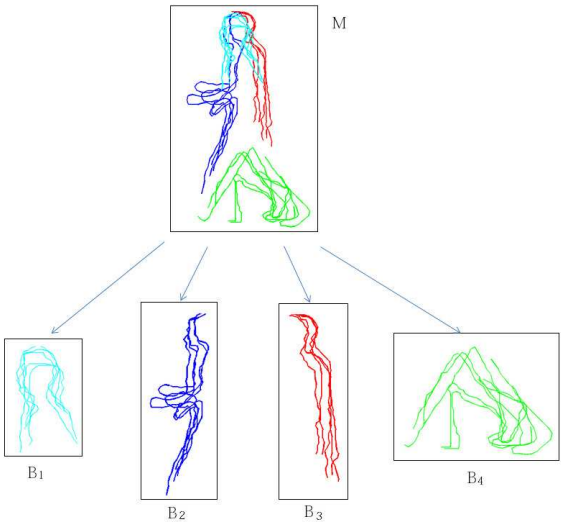

Figure 1.2: Human model $\mathcal{M}$ composed of 4 part bundles B_1, B_2, B_3, B_4 representing head, front, back, and leg parts, respectively. Each bundle has 5 contour parts.

$S_I(\mathcal{M}, x)$ at each point $x \in I$. We define the model fit score as

$$S_I(\mathcal{M}) = \max_{x \in I} S_I(\mathcal{M}, x) \tag{1.13}$$

and the detection center point as point $x^* \in I$ as

$$x^* = \arg\max_{x \in I} S_I(\mathcal{M}, x) \tag{1.14}$$

The detection results for OCD follow the same framework, but with max replaced with min in the above formulas.

1.1.7 Normalizers

It remains to describe how we select a set of normalizers $\{\eta_k |\ k = 1, \ldots, K\}$. We first observe that a good normalizer should be more likely to match to noise than a given contour part. This implies that a normalizer should have a significantly simpler shape than the contour parts of a target shape model. We also want that a normalizer should be less

likely to match to a true object edges in an image than a given contour part. Consequently, normalizers should not be similar to any contour parts in our shape models.

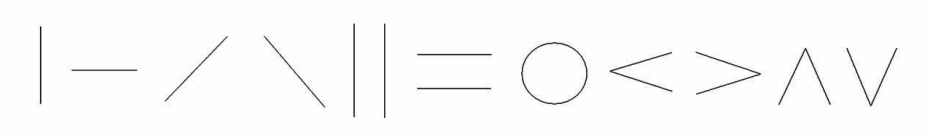

Figure 1.3: **Basic normalizers.** Our set of basic normalizers contains 11 simple shapes.

We satisfy both constrains by first generating a small set of simple geometric curves that are treated as a basic structuring elements to generate a set of normalizers. A set of 11 basic shapes that we have selected is shown in Fig. 1.3. They form the first 11 elements of our set of normalizers $\mathcal{N} = \{\eta_k |\ k = 1, \ldots, K\}$. We obtain further normalizers by pairwise combining the 11 structuring elements, where the combination is simply a union of their aligned images. Since the normalizer combination is symmetric and we only combine different structuring elements, we obtain $55 = (11 \times 10)/2$ additional normalizers. Fig. 1.4 shows a complete set of $K = 66$ normalizers obtained this way. They are ordered according to their weights obtained by the sum of AdaBoost weights of their corresponding weak classifiers by training the AdaBoost strong classifiers on the TU Darmstadt pedestrian dataset [2] (see Section 2.7 for more details). A larger weight indicate that a given normalizer makes more contribution in helping NOCD distinguish true positive from clutter background. The weight order of the normalizers confirms the simplicity principle that guided our design of normalizers in that simpler normalizers are usually more significant. However, the weights of the normalizers are also influence by their ability to match well to noise, which may be image class specific. For example, straight lines in horizontal and vertical direction belong to a common background clutter in inner city images as the images of the TU Darmstadt pedestrian dataset.

For each contour part of a target model c_{ij}, we resize the normalizers to let them have the same bounding box as the contour part c_{ij}. Consequently, the resized normalizers cover

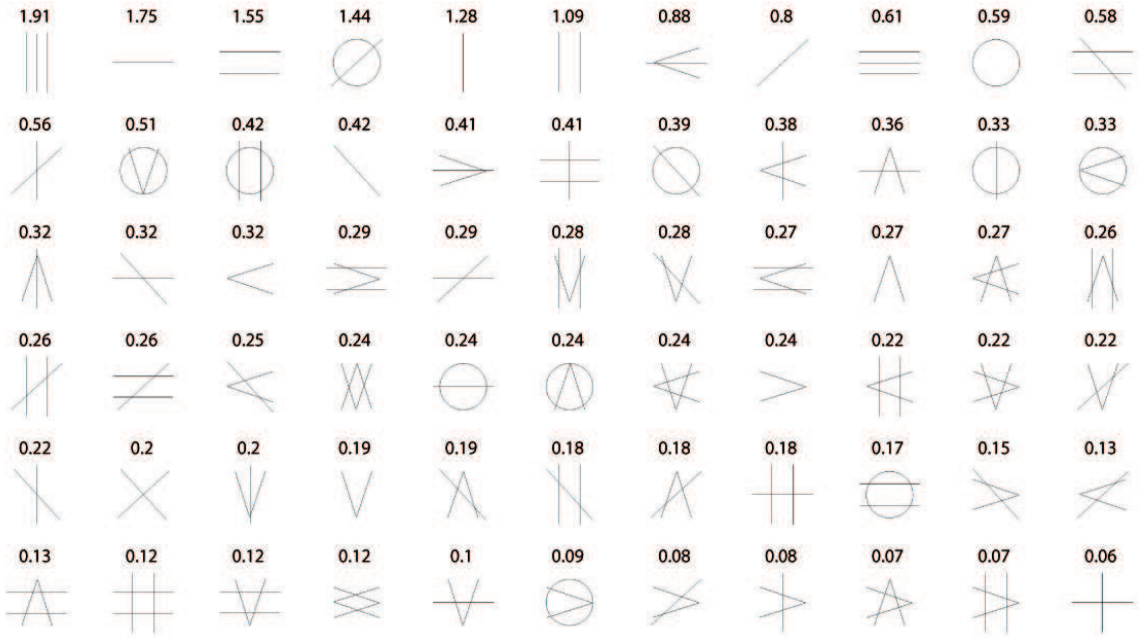

Figure 1.4: Our 66 normalizers displayed in order of their weights.

the same area. Fig. 1.5 shows the resized normalizers generated for each bundle of the human model.

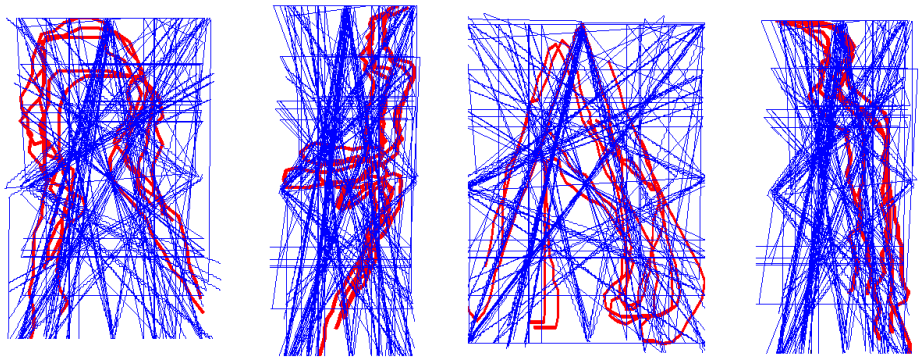

Figure 1.5: **Human model normalizers.** The resized normalizers for four part bundles are shown in blue. The red curves are the original model parts for each bundle.

1.1.8 Experimental Evaluation of Detection Rate

In this section we compare object detection performance of the proposed normalized oriented chamfer distance (NOCD) to the oriented chamfer distance (OCD) and to chamfer

distance on standard test datasets. The detection method is described in Section 1.1.6. We use exactly the same flexible models and the same experimental settings for both methods. In particular, for each image, the edge map was computed by the canny edge detector with the same threshold. The chamfer distance was computed exactly as defined in formula (1.5). The same constants τ and λ were used to truncate the distance transform and linearly combine the distance and orientation terms when calculating the oriented chamfer distance. Results are quantified in terms of detection rate. We use the standard PASCAL criterion to identify correct detections. A detection is regarded as correct if the area of the intersection of the bounding box containing the detected object with the ground truth bounding box is at least 50% of the area of their union.

TU Darmstadt Pedestrian Dataset Human detection is very challenging for shape-based matching methods, because in many poses the shape of human contours is relatively simple. In surveillance images, there is often a complex background, while humans are relatively small, which also increases the chance for an accidental matching.

TU Darmstadt pedestrian dataset [2] consists of several series of video images containing side-view humans. It provides two training datasets, one has 210 images and another has 400 images. In our experiment, we use training 400 dataset for the training of NOCD. After that, we test both NOCD and OCD on the test dataset with 250 images. The 250 test images are significantly more challenging than the 400 training images. To handle the variance of the human shape caused by people walking in opposite directions, we flip our model with respect to vertical axis, and take the best score of the original and flipped models. Consistent with the results of the λ learning procedure reported in Shotton [127], we also observed that detection accuracy of oriented chamfer distance increases when λ becomes larger. In all human detection experiments, we used $\lambda = 0.8$ for both OCD and NOCD, which was the best performing. As it is often the case in AdaBoost applications, we discarded weak classifiers with very small weights. After training phase, we retained

Chamfer distance	4.4%	HOG [33]	72%
OCD	35.2%	4D-ISM [123]	81%
proposed NOCD	70%	Andriluka et al. [2]	92%

Table 1.1: Detection rate on Test 250 of the TU Darmstadt Pedestrian Dataset. The proposed NOCD doubled the OCD detection rate with exactly the same contour model.

only 37 normalizers with largest weights to form the strong leaner for each model contour part. This allows us to reduce the object detection cost complexity.

The detection rate is shown in Table 1.1. We observe that the proposed NOCD nearly doubled the detection rate of OCD on the 250 test images. The improvement is very significant given the fact that the detection rate of OCD is very low: 35.2%.

Several detection results are displayed in Fig. 1.1. As they illustrate OCD fails when the human contours are broken and distorted while at the same time the background is cluttered. This is exactly when the proposed NOCD performs extremely well. We also report the performance of pure chamfer distance in Table 1.1. in order to show that OCD performs significantly better than chamfer distance on this dataset. Further, we include the detection rates of state-of-the-art approaches estimated form graphs reported in [2]. We observe that our detection rate is compatible to a popular appearance based detector, HOG [33]. We stress that our approach is still a matching approach. Andriluka et al. [2] obtained the currently best performance on this dataset. It is obtained by an approach specifically designed for pedestrian detection that utilizes a sophisticated statistical inference framework and learning to handle articulations; both not present in our approach. Similarly, the approach in [123] is designed to handle articulations for pedestrian detection.

Cow dataset This dataset [77] is from the PASCAL Object Recognition Database Collection. There are 111 images in which cows appear at various positions. Since no training part is provided, we divided the dataset into two parts. We used first 55 images to train our detector, and tested it on the remaining 56 images. Then we trained on the second part, and tested on the first 55 images. This way we are able to report our performance on the

whole dataset. The detection rates are shown in Table 1.2. Again we report a substantial increase in the detection rate by over 17% of NOCD in comparison to OCD. Interestingly, OCD is not able to improve the performance of pure chamfer distance. For this dataset, we used $\lambda = 0.2$, which indicates that the orientation information is not particularly useful. This is most likely due to a particular kind of background clutter present in this dataset as can be seen in the example result images in Fig. 1.6. The areas with dense vertical lines in the edge maps confused oriented chamfer matching. Oriented chamfer matching could not tell the ground truth location from such noise, since most of the false alarms appear in that area. The proposed NOCD was able to learn the difference between such noise and the true targets. For images with little clutter in the background, both OCD and NOCD performed equally well.

The performance of NOCD on this dataset also compares favorably to a very sophisticated learning and inference approach published very recently by Zhu et al. [159]. This comparison may not be quite fair, since this approach uses one-example learning, while our flexible cow model is constructed from 5 cow contours. However, on the other hand our detection algorithm is a simple max-sum. Thus, we do not employ any sophisticated inference in the detection process.

Chamfer distance	73.9%	proposed NOCD	91.0%
OCD	73.9%	Zhu et al. [159]	88.2%

Table 1.2: Detection rate on Cow Dataset.

Infrared images Without extra training, we use the same human model and the same normalizers as for TU Darmstadt Pedestrian dataset to carry out several tests on infrared images. In these images, humans are small, about 60×40 pixels, which increase the possibility of misalignment to background. Some detection results are shown in Fig. 1.7.

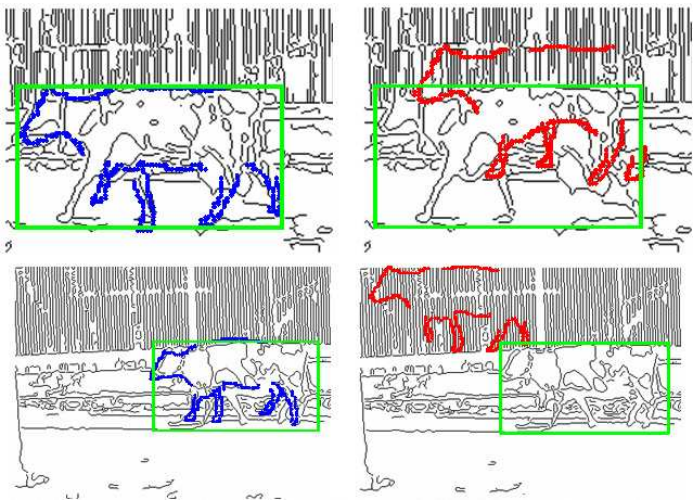

Figure 1.6: Example detection results on the cow dataset. Left column NOCD. Right column OCD. Green rectangle denotes the ground truth object location.

1.1.9 Conclusions

By adding the term of orientation in the evaluation of the score, oriented chamfer distance is more robust to accidental alignment to the background noise than chamfer distance. However, as our experimental results clearly demonstrate this still does not solve the problem of matching to cluttered background, which often leads to a better score than the score at true object location. The proposed NOCD provides a solution to this problem by utilizing AdaBoost to learn normalization of OCD. The key idea is to compare the chamfer matching score of a given template to scores of a set of normalizers. The obtained ratios are interpreted as weak learners, and the strong learner obtained by AdaBoost is interpreted as a normalized OCD. Based on specific application, the proposed method could be modified by replacing oriented chamfer distance with oriented Hausdorff distance, or using sparse logistic regression instead of Adaboost in training phase.

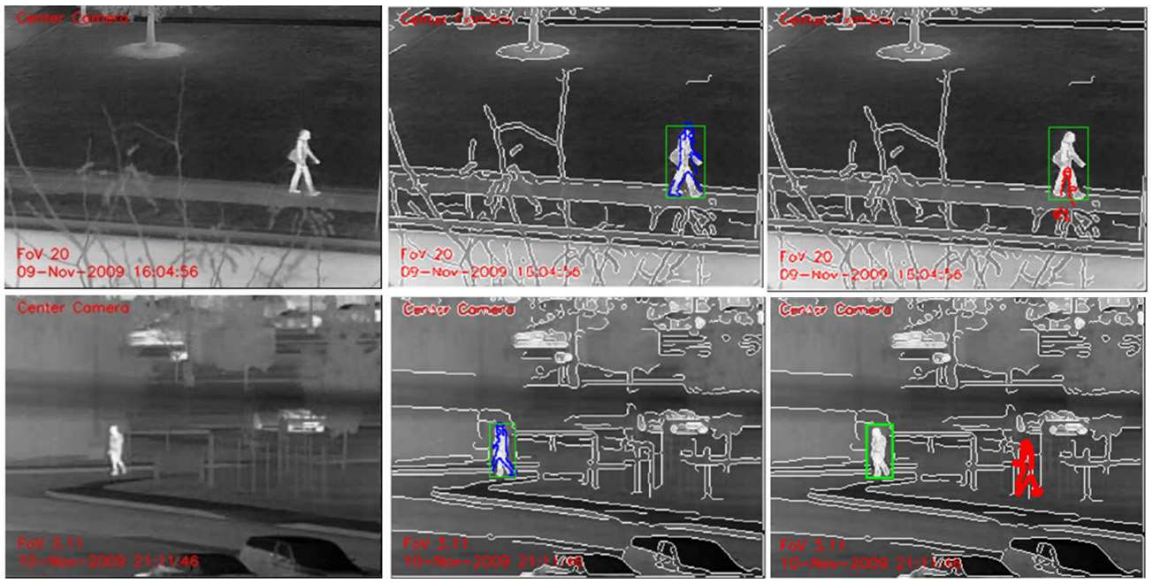

Figure 1.7: Detection result for infrared images. The original images are in the first column. The second column shows result of NOCD while the third column shows the results of OCD. Blue and red dots represent the corresponding parts of the model. Green rectangle denotes the ground truth bounding box. The edge map is overlaid in white on the original images.

1.2 From Partial Shape Matching through Local Deformation to Robust Global Shape Similarity for Object Detection

1.2.1 Introduction

Compared to other image cues, the outline contour (silhouette) is invariant to lighting conditions and variations in object color and texture. More importantly, it can efficiently represent image structures with large spatial extents [126]. Because of these advantages, contour information is widely used in object detection and recognition methods. Recently, several contour-based methods have been demonstrated to work well on the task of object detection and recognition, such as [46], [45], [126] and [130].

Given a gray scale image, edge pixels are obtained by an edge detector, such as Canny [22] or Pb [98]. Then edge pixels are grouped to edge fragments in a bottom up process using an edge-linking algorithm, e.g., [73]. An example of obtained edge fragments is shown in Fig. 1.8(b), where each edge fragment is marked with a different color. These fragments usually form the input to a contour-based object detection algorithm. Given the contour of the target object as a model, the goal of contour-based object detection is to select a small subset of edge fragments that match well to the model contour. The processes of selection and matching are challenged by the following problems with extracted edge fragments in real images: (1) Edge fragments representing part of the target object are missing, e.g., lower part of the legs in Fig. 1.8(b). (2) Edge fragments are broken into several pieces. In our example image in Fig. 1.8(b) both contours of the woman and the swan are broken in many pieces. (3) Part of the true contour of the target object may be wrongly connected to part of a background contour resulting in a single edge fragment. An example is given in Fig. 1.8(c), where the yellow edge fragment contains part of the true

contour of the swan neck and its reflection in water, which obviously does not belong to the true contour of the swan.

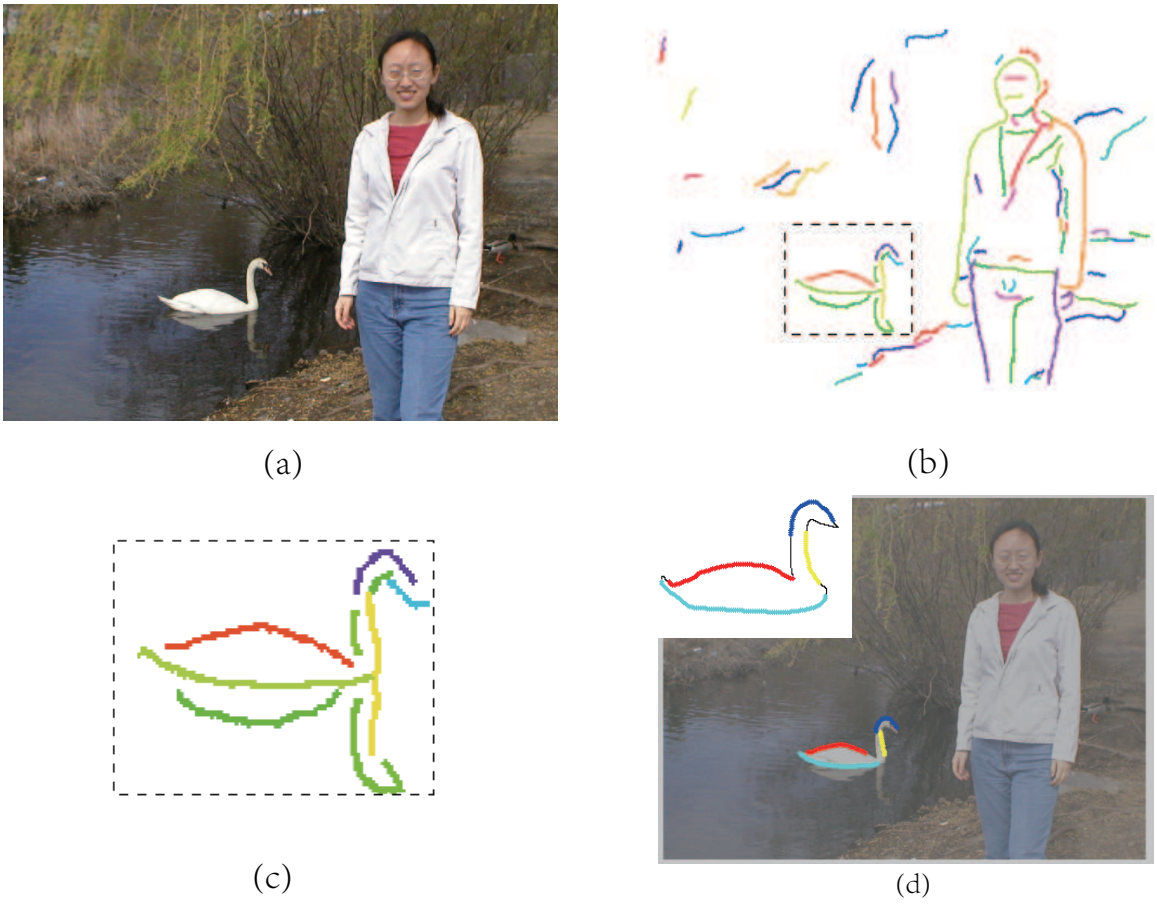

Figure 1.8: (b,c) show edge fragments obtained from (a), which usually are the input to shape based object detection algorithms. (d) shows a detection example of the proposed approach; the corresponding parts in model and image have the same colors.

These problems are unavoidable in real applications, since a perfect edge detector does not exist [98]. In addition (1) may also result from partial occlusion of the target object, which is common in cluttered scenes. Therefore, any object detection approach must address problems (1-3). Assuming that the contour of the target object is given, problems (1, 2) imply that edge fragments can only match parts of the object contour. The situation is significantly more complex due to (3), which implies that only part of an edge fragment may match to part of the object contour. While all recent approaches , e.g. [130] [93],

address the problems (1,2), they suffer from problem (3), since they treat the edge fragments as nonseparable building blocks of the target contours. This may result in missing the target object in the image or locating the object inaccurately, e.g., if the entire yellow fragment is assigned to the swan, the detected bounding box will be larger than the ground truth. To our best knowledge, only the approach in [115] explicitly addresses problem (3) by introducing an efficient partial matching schema based on integral image [149].

However, the final detection evaluation in [115] is appearance based (SVM on HOG features), which demonstrates weakness in the discriminative power of their partial matching schema. There are at least two main reasons for this, one is the selection of the best matching fragments in the integral image framework and the other is simply weak discriminative power of their shape descriptor, which is only angle based.

We utilize the well-known geometric relations of shape context as shape descriptor, but without any histogram representation. One of our main contributions is the selection of the best matching contour fragments in the integral image framework, which by the virtue of the problem is very different from image matching frameworks. As the result we obtain a powerful shape matching framework particularly tailored for partial shape matching. This framework allows us to solve problem (3), since the partial shape matching automatically selects parts of edge fragments that best match to parts of model contour, we essentially generates a new sets of edge fragments. We observe that each of these new edge fragments has a known correspondence to part of the model contour. Thus, partial shape matching is utilized not only to establish the correspondence of edge fragments to model contour parts but also as edge fragment filter.

Given the set of filtered edge fragments and their correspondences to parts of the contour model, our next step is to infer the possible locations of the target object in the image. The inference must simultaneously perform selection and grouping of the edge fragments so that the similarity to the model contour is maximized. We first construct a graph whose nodes are the partial correspondences and edges represent the compatibilities of these cor-

respondences. The location hypotheses are determined as maximal cliques in this graph, i.e., as subgraphs of the weighted graph with maximal affinity of all pairwise connections. To infer the maximal cliques we utilize a recently proposed algorithm [90]. It is very robust in a noisy affinity graph and the number of nodes in a dense-subgraph is automatically determined. This features makes it extremely suitable for our task, because the number of fragments to be grouped is unknown and it varies a lot depending on the quality of edge fragments and the shape of single edge fragments in the image is usually not very discriminative. Each object location hypothesis is identified by several partial correspondences. For example, in Fig. 1.8(d), four partial correspondences identify the target object. We stress that we not only selected the edge fragments in the image but also the corresponding parts of the model contour. Therefore, we can perform a holistic evaluation of the location hypothesis with global shape similarity, i.e., we score each detection hypothesis with a global shape similarity of grouped edge fragments to the model contour.

However, the target object in the image may be distorted, e.g., due to view point change or nonrigid deformation. In addition, as stated above some parts of the model contour do not have any correspondence in image due to missing edge fragments. Therefore, the shape similarity measure must tolerate deformations and missing parts. However, this makes it less discriminative and increases the risk of ”hallucinating” the target object in the background. It follows that it is impossible to tolerate deformations and at the same time keep high discriminative power to avoid hallucinating. This is a very important problem that has not been explicitly addressed by most of the existing approaches.

We address this problem by performing a nonrigid deformation of the model contour according to each detection hypothesis. A nonrigid deformation transformation is obtained by a composition of local affine transformations. Our intuition is that if a detection hypothesis is correct, the deformed model will become more similar to the selected edge fragments, while at the same time it remains similar to the original model. If a detection hypothesis is wrong, the composition of local affine transformations will likely result in a

completely deformed model that resembles neither the original model nor the configuration of the selected edge fragments. However, the key benefit of the proposed local affine transformation is its high capability in estimating the position of missing model parts (i.e., parts that do not correspond to any selected edge fragments). This not only results in a robust scoring of the detection hypotheses but also allows us to put the deformed model contour on the image.

1.2.2 Related Work

In recent years a large number of contour-based object detection and recognition methods has been proposed. Many methods achieve state-of-the-art performance by only utilizing edge information. For example, Shotton et al. [126] and Opelt et al. [103] first learn codebooks of contour fragments, then use Chamfer distance to match learnt fragments to edge images. Ferrari et al. [46] [45] build a network of nearly straight adjacent segments (kAS). In [159], Zhu et al. formulate the shape matching of contour in clutter as a set to set matching problem, and present an approximate solution to the hard combinatorial problem by using a voting scheme. They use a context selection scheme by algebraically encoding shape context into linear programming. Ravishankar et al. [113] use short segments to approximate the outer contour of objects. They decompose the model shapes into segments at high curvature points. Dynamic programming is used to group the matched segments in a multi-stage process which begins with triples of segments. Lu et al. [93] first decompose the model into several part bundles. They use particle filters as inference tool to simultaneously perform selection of relevant contour fragments in edge images, grouping of the selected contour fragments, and matching to the model contours. To address the non-rigid object deformation, Bai et al. [4] use the skeleton information to capture the main structure of an object, and use Oriented Chamfer Matching [126] to match the model parts to images. Most recently, Srinivasan et al. [130] address the contour grouping problem as many-to-one matching, and use this scheme in both training and testing phases. For purpose of improv-

ing detection and score ranking, a sophisticated training process is designed in which latent SVM is used to guarantee the many-to-one score is tuned discriminatively. Besides of literature mentioned above, edge information is also utilized in [115, 96, 102, 10].

1.2.3 Shape Descriptor

We propose a novel shape descriptor that is particularly suitable for shape matching of edge fragments in images to model contours of target objects. Its basic geometric units are the same as in shape context [8]. Shape context (SC) appears to be one of the best performing shape descriptor and definitely the most popular one. Given a planar set X composed of a finite number of points, for every point $x \in X$ we consider both the length and direction of the vector from x to other points in X. However, different from SC, we do not build any histograms representing the lengths and directions.

Given two sequences of points $P = \{p_1 \cdots p_m\}$ and $Q = \{q_1 \cdots q_n\}$ representing two contour fragments in 2D, we compute two matrices, one representing all lengths and the second representing all pairwise orientations of vectors from each $p_i \in P$ to each $q_j \in Q$. As a special case when $P = Q$, the matrices describe the shape of the contour fragment P. The distance $D^{(P,Q)}(i,j)$ from p_i to q_j is defined as Euclidean distance in the log space

$$D^{(P,Q)}(i,j) = \log(1 + ||\vec{p_i} - \vec{q_j}||_2) \tag{1.15}$$

We add one to Euclidean distance to make the $D^{(P,Q)}(i,j)$ positive. The orientation $\Theta^{(P,Q)}(i,j)$ from p_i to q_j is defined as the orientation of vector $\vec{p_i} - \vec{q_j}$:

$$\Theta^{(P,Q)}(i,j) = \angle(\vec{p_i} - \vec{q_j}) \in [-\pi, \pi]. \tag{1.16}$$

The relative geometric relation of two contour fragments P and Q is encoded in two $m \times n$ matrices $D^{(P,Q)}$ and $\Theta^{(P,Q)}$. An example is given in Fig. 1.9.

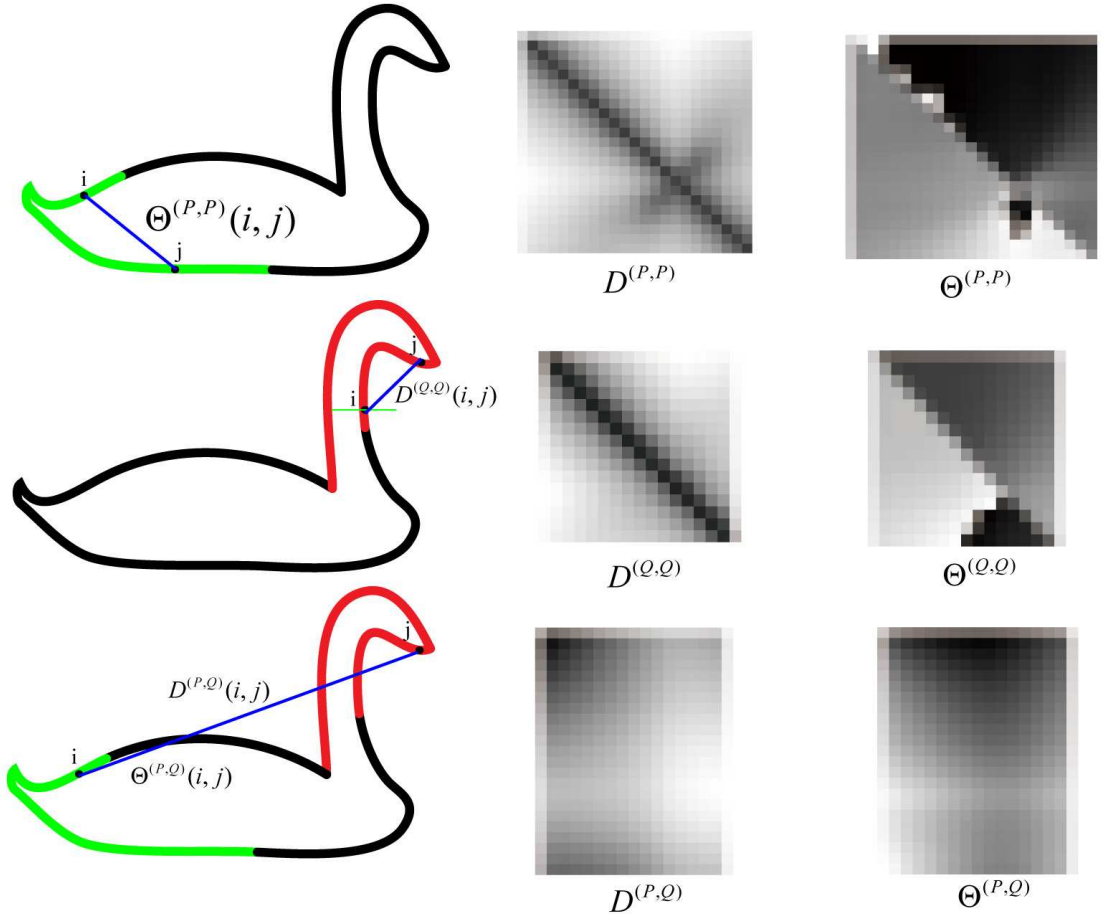

Figure 1.9: Shape descriptor.

Given another two contour fragments T and U consisting of the same number of points as P and Q, respectively, we define two affinity matrices that measure the similarity of the two fragment configuration (P, Q) to the other two fragment configuration (T, U). The first affinity matrix is based on comparison of distances between two pairs of corresponding pairs of points

$$A_D(P, Q, T, U) = \exp(-\frac{(D^{(P,Q)}(i,j) - D^{(T,U)}(i,j))^2}{(D^{(P,Q)}(i,j)\ \sigma)^2}). \tag{1.17}$$

where σ represents the tolerance of distance differences (it is set to 0.2 in all our experiments).

To make the value of $A_D(P,Q,T,U)$ invariant to scale, we divide each distance difference by the distance between the first pair of points. The second affinity matrix is based on angle comparison of vectors connecting the corresponding pairs of points

$$A_\Theta(P,Q,T,U) = \exp(-\frac{(\Theta^{(P,Q)}(i,j) - \Theta^{(T,U)}(i,j))^2}{\delta^2}), \tag{1.18}$$

where the difference of angles is taken modulo π, i.e., it is the angle between vectors $\vec{p_i} - \vec{q_j}$ and $\vec{t_i} - \vec{u_j}$, and δ represents the tolerance of angle differences (it is set to $\frac{\pi}{4}$ in all our experiments). Since both A_D and A_Θ are normalized, we can simply add them to obtain the affinity matrix

$$A(P,Q,T,U) = A_D(P,Q,T,U) + A_\Theta(P,Q,T,U). \tag{1.19}$$

We observe that A is $m \times n$ matrix representing the similarities of corresponding point pairs in (P,Q) and (T,U). The similarity of two configurations of contour fragments (P,Q) and (T,U) is defined as

$$\Psi(P,Q,T,U) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} A(P,Q,T,U). \tag{1.20}$$

As a special case of Eq. (2.24), we obtain a similarity between two contour fragments P and T defined as

$$\Psi(P,T) = \Psi(P,P,T,T) \tag{1.21}$$

(here we slightly abuse the notation for the sake of simplicity). When Q is the same as P in Eq. (1.15) and (1.16), the matrices $D^{(P,P)}$ and $\Theta^{(P,P)}$ represent all pairwise distances between all pair of points of P and corresponding angles of the vector connecting the points. Thus, two matrices form a shape descriptor of the contour fragment P and similarly for T. Hence $\Psi(P,T)$ simply compares the shape descriptor of the contour fragments P and T.

1.2.4 Partial Matching between Edge Fragments and Model Contour

Given an image I, using edge-linking software [73], a set of edge fragments $E = \{e_1 \cdots e_K\}$ is generated. Each fragment e_k is a list of N_k points (i.e., pixels) $\{q_1, \cdots, q_{N_k}\}$. According to our descriptor, the geometry of fragment e_k is encoded in two $N_k \times N_k$ matrices: A_D and A_Θ. Similarly, two $M \times M$ matrices are used to fully represent the contour of a model $\mathcal{M}$ composed of points $\{p_1, \cdots, p_M\}$.

Our goal is to find the best matching between a part of image edge fragment e_k with a part of model fragment $\mathcal{M}$. Thus, we need to find a part $\mathcal{M}(i,l) = \{p_i, \cdots, p_{i+l-1}\} \subseteq \mathcal{M}$, where i is the starting point of the part and l is its length. (The indices are modulo M if the model contour fragment is a closed curve.) Since cannot expect that the whole image fragment participates in the matching, we need to simultaneously select part $e_k(j,l) = \{q_j, \cdots, q_{j+l-1}\} \subset e_k$, where j is the starting point of the fragment part and its length is also l.

Our goal can be expressed as finding two corresponding subblocks of their shape matrices with the maximum similarity Ψ defined in (1.21). To achieve this goal we construct a 4D tensor matrix

$$\Gamma(i,j,l,k) = \Psi(\mathcal{M}(i,l), e_k(j,l)) \tag{1.22}$$

and observe that $\Gamma(i,j,l,k)$ can be computed efficiently by utilizing the integral image algorithm, since it allows to access any element in the 4D matrix in constant time [35, 149].

Intuitively, when very few points are involved in a matching, the shape similarity is neither reliable nor discriminative enough. Therefore, we set a threshold τ on the minimal number of matching points and set $\Gamma(i,j,l,k) = 0$ if $l < \tau$. We then take the maximum of the 4D matrix along different l, and suppress it to

$$S(i,j,k) = \max_l \Gamma(i,j,l,k) \tag{1.23}$$

We observe that the index of the maximal value of S determines a pair of best matching subsegments of $\mathcal{M}$ and e_k:

$$G(i,j,k) = \arg\max_{l} \Gamma(i,j,l,k) = (\mathcal{M}(i,l), e_k(j,l)). \tag{1.24}$$

Based on these local observations, the most popular method to form object location hypothesis is using Hough voting, such as in [115]: local maxima of $S(i,j,k)$ for certain fragment e_k are identified, and corresponding fragment correspondences are used to estimate object location by Hough voting. However, Hough voting seems not to be an optimal choice here. When each part correspondence independently cast a vote, the cluttered background is more likely to get a larger score, since single edge fragments are unlikely to carry discriminative shape information.

More discriminative shape information can be obtained by considering all pairwise shape relations of several edge fragments. We introduce a graph-based clustering method to find location hypothesis through which shape dependency of local edge fragments is naturally captured.

1.2.5 Object Localization as Maximal Clique Computation in a Weighted Graph

Each vertex $v \in V$ of our graph corresponds to a partial match $G(i,j,k)$ (1.24), i.e., v represents a model segment $\mathcal{M}(i,l)$ selected as best matching to part $e_k(j,l)$ of the edge fragment e_k. To limit the number of vertices $G(i,j,k)$, for each point i in model $\mathcal{M}$, we only choose the best K matches as vertices according to their corresponding similarity $S(i,j,k)$. Therefore, for a given model $\mathcal{M}$ contour with M points, the number of vertices is equal to $M \times K$.

Given two pairs of matches, i.e., two vertices $v_i = \{\mathcal{M}(i_1, l_1), e_m(j_1, l_1)\}$ and $v_j = \{\mathcal{M}(i_2, l_2), e_n(j_2, l_2)\}$, if $v_i \neq v_j$ we define the edge weight as

$$A(i, j) = \Psi(\mathcal{M}(i_1, l_1), \mathcal{M}(i_2, l_2), e_m(j_1, l_1), e_n(j_2, l_2)), \tag{1.25}$$

which measures the shape similarity of the configuration of two model segments $\mathcal{M}(i_1, l_1)$ and $\mathcal{M}(i_2, l_2)$ to a corresponding configuration $e_m(j_1, l_1)$ and $e_n(j_2, l_2)$ of two parts of edge fragments. As a special case, we define

$$A(i, i) = \Psi(\mathcal{M}(i_1, l_1), e_m(j_1, l_1)), \tag{1.26}$$

which measures the shape similarity of a single model segment $\mathcal{M}(i_1, l_1)$ to a corresponding edge part $e_m(j_1, l_1)$.

To sparsify the affinity matrix A, we observe that $e_m(j_1, l_1)$ and $e_m(j_2, l_2)$ can only correspond to $\mathcal{M}(i_1, l_1)$ and $\mathcal{M}(i_2, l_2)$ if they are relatively close to each other. In practice, we compare the average value of distance matrix $D^{(e_m(j_1,l_1),e_m(j_2,l_2))}$ to average value of $D^{(\mathcal{M}(j_1,l_1),\mathcal{M}(j_2,l_2))}$. If the difference is larger than a reasonable value, we set $A(i, j) = 0$ (for instance in our experiment, it is the square root of model size multiply the scale).

Meanwhile, partial matching v_i and v_j may refer to the corresponding of the similar position of model only with a few pixels offset. We do not want to have these kind of partial matches co-occur in a solution of clustering, since for a true positive configuration of an object hypothesis, it is impossible that several fragments in image corresponding to the same part of model. Based on $f = \frac{|\mathcal{M}(i_1,l_1)\cap\mathcal{M}(i_2,l_2)|}{|\mathcal{M}(i_1,l_1)\cup\mathcal{M}(i_2,l_2)|}$, we tell if v_i and v_j get the same part of model involved in. If $f < t$, we set $A(i, j) = 0$. In experiment, t equals to 0.5.

The obtained weighted affinity graph is denoted as $G = (V, A)$. Our goal is to find all maximal cliques in this graph. As stated in [107], a maximal clique is a subset of V with maximal average affinity between all pairs of its vertices, which is equivalent to the fact that the overall similarity among internal elements is higher than that between external and

internal elements. In our case, given a shape model and corresponding partial matches in the image, clustering is expected to find several pairs of matches with high values of all pairwise similarities. To formally state our goal, we introduce an indicator vector **x** over the vertices V, i.e., has $M \times K$ coordinates. A vertex $v \in V$ is selected as belonging to a maximal clique if and only if $x_v > 0$, where x_v denotes the v coordinate of x. Then each maximal clique is defined as the solution of the following quadratic program

$$\begin{aligned} &\text{maximize} \;\; f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} \\ &\text{subject to} \;\; \mathbf{x} \in \triangle, \end{aligned} \tag{1.27}$$

where $\triangle = \{\mathbf{x} \in R^{M \times K} : \mathbf{x} \geq 0 \;\text{ and }\; \mathbf{e}^T\mathbf{x} = 1\}$ is the simplex in $R^{M \times K}$.

Each maximal clique corresponds to a local solution of Eq. (1.27). We are using the recently proposed algorithm in [90] to compute the local solutions. Each solution **x**, i.e., maximal clique, is treated as an object detection hypothesis. It consists of several model contour segments and the corresponding parts of edge fragments. The final evaluation of the hypotheses is presented in the next section.

1.2.6 Evaluation of Detection Hypothesis

By considering the partial matches as a whole, a detection hypothesis is expressed as the correspondence between a subset of points on the model and a subset of edge points in image. We denote the subset of model points as $\mathcal{M}_a \subset \mathcal{M}$, and subset of image edge points as $E_a \subset E$. Clearly there exists a bijection T between $\mathcal{M}_a \subset \mathcal{M}$ and $E_a \subset E$, i.e., if $x \in \mathcal{M}_a, T(x) \in E_a$. For each hypothesis, there are usually some points in the model that have no correspondence in the image, i.e., $\mathcal{M}_b = \mathcal{M} \setminus \mathcal{M}_a \neq \varnothing$. The mapping $T : \Re^2 \rightarrow \Re^2$ can be regarded as affine-transformation Z which consists of scaling, translation and rotation. Here, we intend to extend T to conclude the transformation Z for $x \in \mathcal{M}_b$.

Therefore, we define T for $x \in \mathcal{M}$ as following:

$$\begin{aligned} T(x) = &\quad \mathcal{M}_a \to E_a, \ \text{ if } \ x \in \mathcal{M}_a \\ = &\quad xZ, \qquad \text{if } \ x \in \mathcal{M}_b \end{aligned} \tag{1.28}$$

For each point among $\mathcal{M}_b$, our goal is to determine the appropriate affine-transformation based on existing mapping relations $\mathcal{M}_a \to E_a$. We attempt to locally estimate Z for every $x \in \mathcal{M}_b$. This is motivated by the observation that affine transformations of points belong to the same part of model are usually consistent, e.g., the points on swan neck. Based on the distance of indices in the model points sequence, we find the a certain number of close points of $x \in \mathcal{M}_b$, and denote them by $N(x) \subset \mathcal{M}_a$. The reason that we define distance as difference between points indices instead of their geometry closeness is: $\mathcal{M}$ is an ordered points set, point connectedness is more important than the closeness in geometry. Then Z is computed as:

$$Z = \min_{Z^*} \ d(T(N(x)), N(x)Z^*) \tag{1.29}$$

Here, function d is simply computing the accumulate square distance between $T(N(x))$ and $N(x)Z^*$. Thus, Eq. (1.29) is turned into

$$\begin{aligned} T(x) = &\quad \mathcal{M}_a \to E_a, \ \text{ if } \ x \in \mathcal{M}_a \\ = &\quad x \min_{Z^*} \ d(T(N(x)), N(x)Z^*), \qquad \text{if } \ x \in \mathcal{M}_b \end{aligned} \tag{1.30}$$

By applying mapping T on every point $x \in \mathcal{M}$, a set of points $T(\mathcal{M})$ corresponding to model points is obtained. It is used for later scoring.

1.2.7 Scoring and Ranking

As mentioned above, the confidence for a hypothesis is evaluated from two aspects.

$$S(T(\mathcal{M})) = \Psi(\mathcal{M}, T(\mathcal{M})) \times \Psi(T(\mathcal{M}), T'(\mathcal{M})) \tag{1.31}$$

The first score indicates how well M is corresponded to $T(\mathcal{M})$ considering the geometric arrangement, which is simply computed using Eq. (2.24).

Moreover, we also need to measure if $T(\mathcal{M})$ is consistent with the contour cues in image. This is indicated by the second score. For this purpose, we first calculate tangent direction θ for both points in $T(x), x \in \mathcal{M}_b$ and edge points E in image. This makes each point to be 3D data, i.e., $[x, y, \theta]$. In this 3D space, for each point in $T(x), x \in \mathcal{M}_b$, we use kd-tree algorithm to find the closest point in E. All these closest points from E are aggregated, together with the points in E_a, are denoted by $T'(\mathcal{M})$. We measure the similarity between $T(\mathcal{M})$ and $T'(\mathcal{M})$ using Eq. (2.24). Finally, we rank all obtained hypothesis according to the confidence $S(T(\mathcal{M}))$.

1.2.8 Experimental Results

We present results on the ETHZ shape classes [46] which features five diverse classes (bottles, swans, mugs, giraffes, apple-logos) and contains a total of 255 images. For all categories, there are significant inner-class variations, scale changes, and illumination changes. Most importantly, the dataset comes with ground truth gray level edge maps, which is computed by Pb edge detector [98]. This makes it possible to have a fair comparison of contour-based object detection methods.

Depending on the way of selecting shape models for each category, we follow two different experiment protocols. First, we utilize single hand-drawn shape model for each class, and testing is done on all 255 images. Second, we follow the protocol in [45]. We use the first half of images in each class for training, and test on the second half of this class as positive images plus all images in other classes as negative images. In our approach we only use the ground truth outlines of objects present in the first half of images for each class. We apply our shape descriptor to compute pairwise similarity of the outlines, and use affinity propagation clustering algorithm [52] to automatically obtain several prototype shape models. Thus, our training is only used to select prototype contour models.

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Our method	0.881	0.920	0.756	0.868	**0.959**	**0.877**
Srinivasan et al. [130]	0.845	0.916	**0.787**	**0.888**	0.922	0.872
Maji et al. [96]	0.869	0.724	0.742	0.806	0.716	0.771
Felz et al. code [42]	**0.891**	**0.950**	0.608	0.721	0.391	0.712
Lu et al. [93]	0.844	0.641	0.617	0.643	0.798	0.709

Table 1.3: Comparison of interpolated average precision (AP) on ETHZ Shape classes.

For the purpose of detection evaluation, we follow the PASCAL criteria, i.e., a detection is deemed as correct if the intersection of detected bounding box and ground truth over the union of the two bounding boxes is larger than 50%.

To convert the gray level edge map to binary edge map, we set all pixels with their values larger than 0 as edge pixels. This means we do not adjust the threshold to get better edges. During detection, 5 different scales are searched for every image. Non-maximum suppression is used to remove duplicate hypothesis.

We focus on comparison to the state-of-the-art contour-based object detection methods, in particular to [45, 130, 93]. We plot the precision/recall (PR) curves in Fig. 1.10. Table 1.3 shows the interpolated average precision (AP) value for 5 methods. Our method achieves the best mean AP and the best AP for category Swans. Our AP is comparable to the best ones in the other four classes. The mean AP of our method is slightly better than [130] and much better than the other contour-based methods.

We also show the false positives per image (FPPI) vs. detection rate (DR) in Fig. 1.11. Table 1.4 compares the detection rates at 0.3/0.4 FPPI. Our method also achieve comparable result to [130], but the mean value of [130] is slightly better than ours for this measure. We observe that our method is the only one with no difference in detection rates at 0.3 FPPI and 0.4 FPPI. The curve of our methods increases sharply at the beginning and reaches the peak of the detection rate before 0.3/0.4 FPPI.

Besides the presented evaluation of the object detection accuracy, which is based on bounding box intersection, accuracy of localizing the boundary of detected objects is ex-

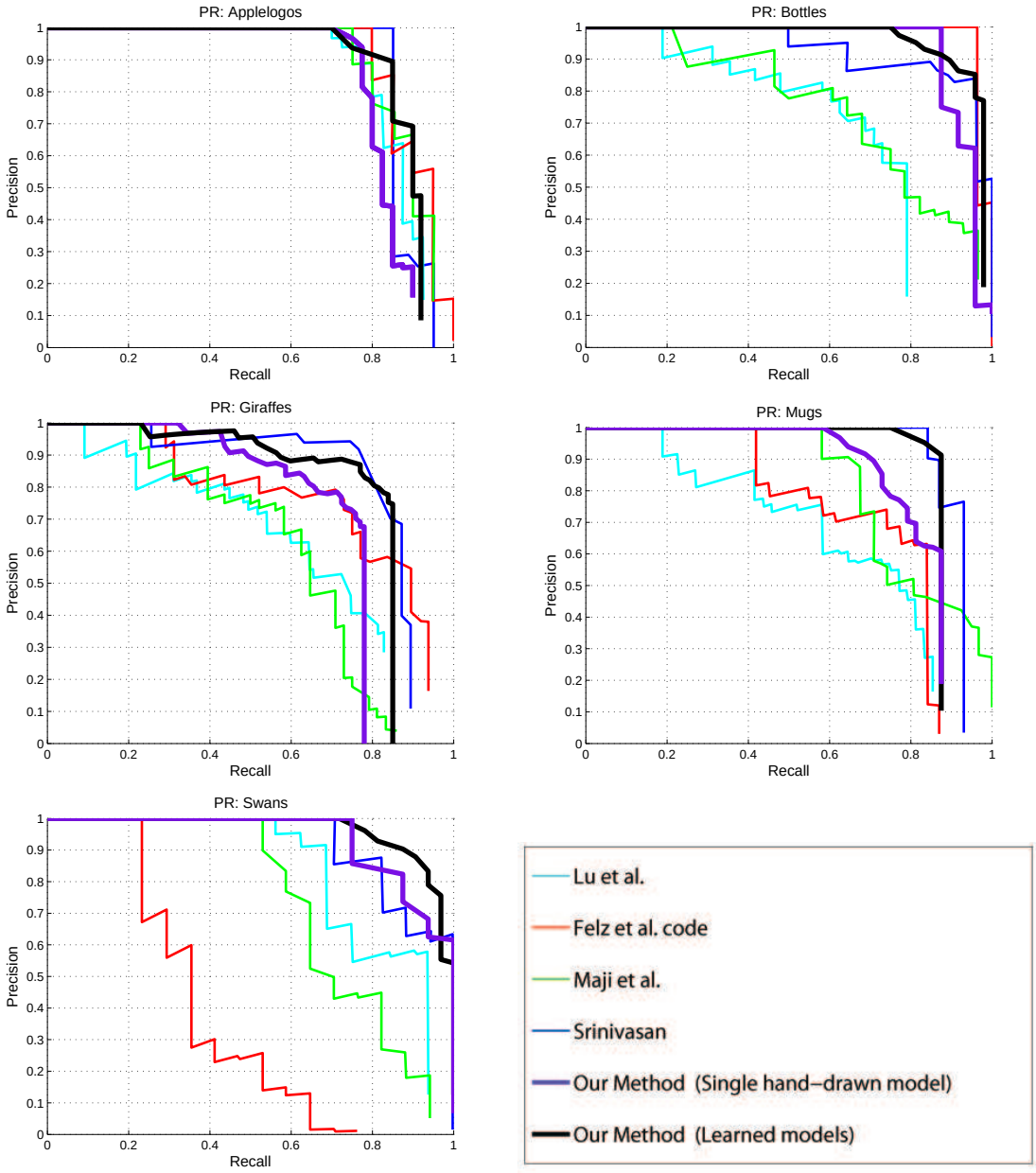

Figure 1.10: Precision/Recall curves of our method compared to Lu et al. [93], Felz et al. [42], Maji et al. [96], and Srinivasan et al. [130] on ETHZ shape classes. We report both the results with single hand-drawn model and with learned models.

tremely important in many applications. Since our final detection evaluation includes non-rigid deformation of a contour model and positioning the deformed model on the edge image, we are able not only to precise localize the boundary but also to complete the missing contours. This fact is illustrated by our example detection results shown in Fig. 1.12.

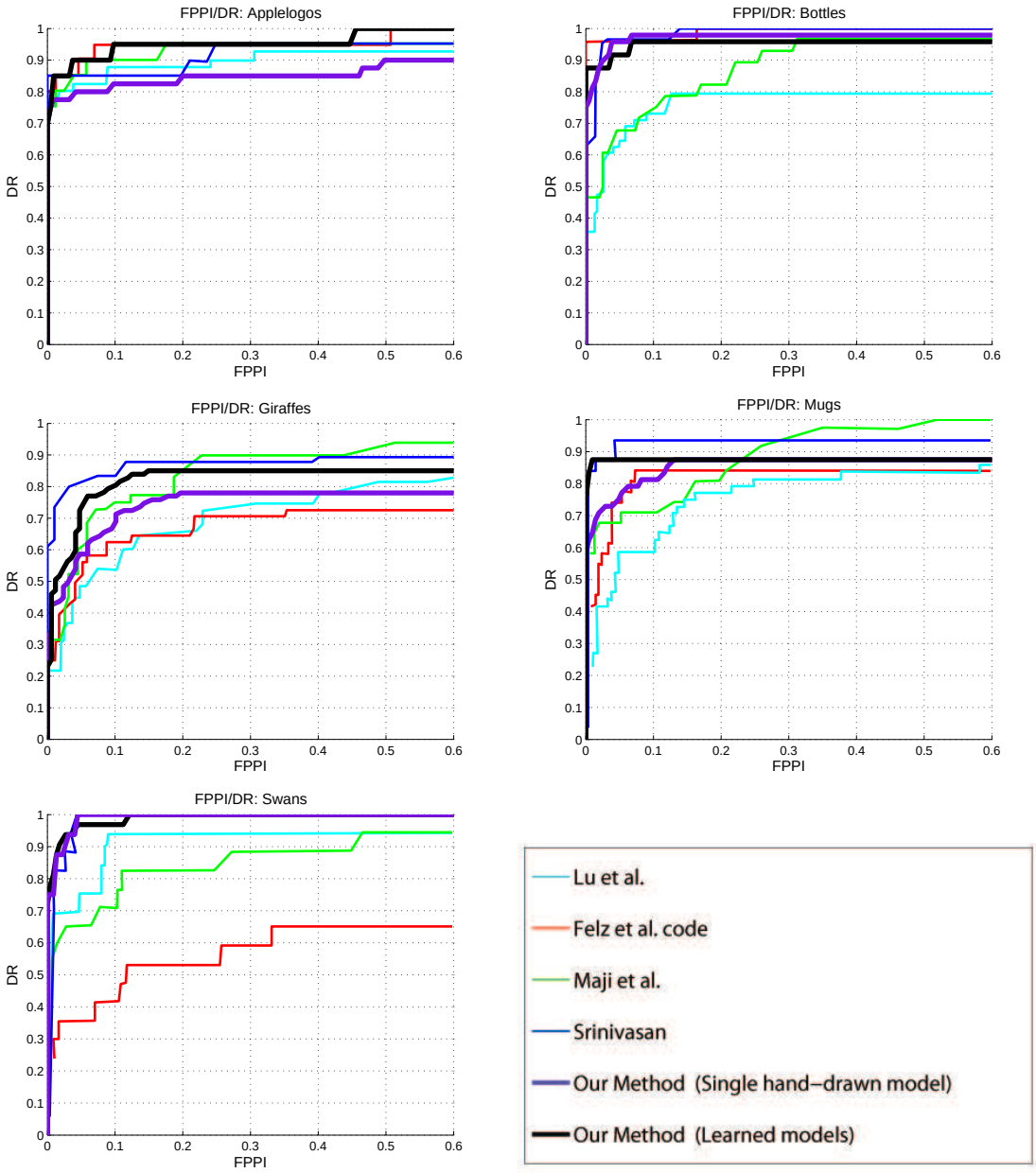

Figure 1.11: Comparison of DR/FPPI curves on ETHZ shape classes.

To qualitatively evaluate the contour detection accuracy, we use the coverage and precision measure defined in [45]. The coverage value shows what percentage of true boundaries have been successfully detected. The precision values measures how many detected edge points are inside the true boundaries. We compare the coverage/precision of our method with [45] in Table 1.5. Our method achieves a higher precision value on all 5 classes, es-

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Our method	0.92/0.92	0.979 / 0.979	0.854/0.854	0.875/0.875	**1 / 1**	0.926 / 0.926
Srinivasan et al. [130]	**0.95/0.95**	**1 / 1**	0.872/0.896	0.936/0.936	**1 / 1**	**0.952 / 0.956**
Maji et al. [96]	0.95/0.95	0.929 / 0.964	**0.896/0.896**	**0.936/0.967**	0.882 / 0.882	0.919 / 0.932
Felz et al. code [42]	0.95/0.95	**1 / 1**	0.729/0.729	0.839/0.839	0.588 / 0.647	0.821 / 0.833
Lu et al. [93]	0.9/0.9	0.792 / 0.792	0.734/0.77	0.813/0.833	0.938 / 0.938	0.836 / 0.851
Riemenschneider et al. [115]	0.933/0.933	0.970 / 0.970	0.792/0.819	0.846/0.863	0.926 / 0.926	0.893 / 0.905
Ferrari et al. [45]	0.777/0.832	0.798 / 0.816	0.399/0.445	0.751/0.8	0.632 / 0.705	0.671 / 0.72
Zhu et al. [159]	0.800/0.800	0.929 / 0.929	0.681/0.681	0.645/0.742	0.824 / 0.824	0.776 / 0.795

Table 1.4: Comparison of detection rates for 0.3/0.4 FPPI on ETHZ Shape classes.

	Our method	Ferrari et al. [46]
Applelogos	**0.923/0.948**	0.916/0.939
Bottles	**0.845/0.903**	0.836/0.845
Giraffes	0.456/0.784	**0.685/0.773**
Mugs	0.735/0.803	**0.844/0.776**
Swans	**0.848/0.909**	0.777/0.772

Table 1.5: Accuracy of boundary localization of the detected objects. Each entry is the average coverage/precision over trials and correct detections at 0.4 FPPI.

pecially there is a big improvement for Applelogos, Bottles, and Swans. For coverage, our method is better on 3 classes, but worse on the classes of Giraffes and Mugs. The reason is that our models for Giraffes and Mugs are very simple, in particular, we do not have the inner contour of the mug handle and the lower part of the giraffe outline as can be seen in Fig. 1.12. Therefore, some part of the true boundaries, such as the internal handle of mugs, are not detected.

1.2.9 Conclusion

We present a novel framework for contour based object detection with three main contributions. First, we introduce a partial shape matching scheme suitable for matching of edge fragments, in which the shape descriptor has the same geometric units as shape context but is not histogram based. Second, we group partial matching hypotheses to object detection hypotheses via maximum clique inference on a weighted graph instead of hough voting. Third, a unique feature of our approach is that we perform nonrigid deformation of

Figure 1.12: Some detection results of ETHZ dataset. The edge map is overlaid in white on the original images. Each detection is shown as the transformed model contour in black. The red framed images in the bottom row show two false positives.

a contour model and position the deformed model on the edge image. Our deformation is based on a local affine-transformation guided by the partial matching to edge fragments. By combining these components, we obtain an effective purely shape-based object detection framework. Our method compares favorable to other state-of-the-art purely shape based methods. In particular, we achieve the best average precision (AP) value averaged over all 5 classes of the ETHZ dataset. The evaluation on the ETHZ dataset demonstrates that the proposed method not only achieves accurate object detection but also precise contour localization on cluttered background.

Chapter 2

Computing Maximum Weight Subgraphs with Mutex Constraints

2.1 Introduction

In many applications mutual exclusion (mutex) constraints can significantly improve the quality of solutions. This is particularly the case when unary and binary potentials are unreliable, which is rather a rule than exception in real applications. As an example let us consider matching feature points between two images in the presence of perspective distortion and occlusion. It is well-known that qualitative spatial relations such as above/below and left/right can significantly improve the quality of solutions. These relations define incompatible matching pairs and as such can be expressed as mutex constraints.

Since mutex constraints are hard pairwise combinatorial constraints, they lead to non-submodular terms with large values of the energy function. When the number of mutex constraints is large, general binary Markov Random Field (MRF) solvers cannot handle it very well. In this section we focus on problems whose adequate modeling requires *global* mutex constraints, meaning that at least one mutex constraint applies to each variable (MRF side). As demonstrated in the experimental results on real applications, the state-of-the-art general MRF solvers LBP (Loopy Belief Propagation), QPBO (Quadratic Pseudo-Boolean Optimization) [15, 71], QPBOP [16], and QPBOI [117] fail to deliver acceptable solutions for such problems. Therefore, we propose a novel algorithm that is tailored for solving problems with global mutex constraints. Our algorithm not only significantly outperforms LBP, QPBO, QPBOP, and QPBOI, but also Integer Projected Fixed Point Method (IPFP) [84] as well as application specific algorithms. It can be viewed as an extension of two recent works, [84], where a quadratic objective is subject to linear constraints, and [20], were a linear objective is subject to quadratic constraints. By contrast, our algorithm has both the objective and constraints in quadratic form.

Since MAP inference in MRF can be expressed as solving a constrained *maximum weight subgraph (MWS)* problem, we use the MWS formulation in this section. Given an undirected graph G with weights on the vertices and edges, the constrained MWS problem is to find a subgraph having the largest total weight subject to some constraints. In its most

general form, the MWS problem is formulated as an integer quadratic program:

$$\begin{aligned} &\text{maximize } \; g(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} \\ &\text{subject to } \; \mathbf{x} \in \{0,1\}^n \; \text{ and } \; \mathbf{x} \in \mathbf{P} \end{aligned} \tag{2.1}$$

where $\mathbf{x}$ is an indicator vector over the vertices of graph G and A is the affinity matrix (a weighted adjacency matrix) with all nonnegative entries, i.e., $A_{ij} \geq 0$ for all $i, j = 1, \ldots, n$. Without loss of generality we also assume that $\sum_{i,j} A_{ij} \leq 1$. In the applications we consider matrix A is usually indefinite.

$\mathbf{P}$ represents additional constraints imposed on $\mathbf{x}$. Usually one-to-one (1-1) constraints are imposed on $\mathbf{x}$ in the case of graph matching problems, and many-to-one constraints are often required in MAP inference problems. The two kinds of constraints can be expressed as linear equality constraints $B\mathbf{x} = \mathbf{1}$, where $\mathbf{1}$ is a column vector of ones. This instance of problem (2.1) can be then formulated following [84] as

$$\begin{aligned} &\text{maximize } \; g(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} \\ &\text{subject to } \; \mathbf{x} \in \{0,1\}^n \; \text{ and } \; B\mathbf{x} = \mathbf{1}. \end{aligned} \tag{2.2}$$

We aim at solving a more general instance of problem (2.2), since we consider quadratic equality constraint:

$$\begin{aligned} &\text{maximize } \; g(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} \\ &\text{s.t. } \; \mathbf{x} \in \{0,1\}^n \; \text{ and } \; \mathbf{x}^{\mathbf{T}} M \mathbf{x} = 0, \end{aligned} \tag{2.3}$$

where $M \in \{0,1\}^{n \times n}$ is a symmetric matrix representing *mutex* (short for mutual exclusion) constraints. If $M(i,j) = 1$, then $x_i \cdot x_j = 0$, meaning that nodes i, j cannot belong to the same MWS. Hence mutex constraints represent incompatible vertices that cannot be selected together as part of the solution. Mutex constraints are very important in many computer vision and machine learning applications. In particular, they make it possible to enforce qualitative spatial relations like left/right and above/below as shown in salient

points matching in Sec. 2.7. Of course, both 1-1 and many-to-one matching constraints can be expressed as mutex constraints.

The goal of (2.3) is to select a subset of vertices of graph G such that g is maximized and the mutex constraints are satisfied. Since g is the sum of unary and binary affinities of the elements of the selected subgraph, the larger is the subgraph, the larger is the value of g. However, the size of the subgraph is limited by mutex constraints. We assume that a discrete vector $\mathbf{x} \in \{0,1\}^n$ exists that satisfies the constraints. We also assume that $\forall i\ M(i,i) = 0$.

The mutex constraints $\mathbf{x^T} M \mathbf{x} = 0$ cannot be expressed as linear equality constraints, but can be expressed as linear inequality constraints, since ($M(i,j) = 1 \Rightarrow x_i \cdot x_j = 0$) is equivalent to $x_i + x_j \leq 1$, given $x_i, x_j \in \{0,1\}$. However, this equivalence does not hold if x is relaxed to the continuous domain, i.e., if $x_i, x_j \in [0,1]$, then the mutex constraint $x_i \cdot x_j = 0$ is stronger than the linear inequality constraint $x_i + x_j \leq 1$. (For instance, $x_i = 0.5$ and $x_j = 0.5$ satisfies $x_i + x_j \leq 1$, but does not satisfy $x_i \cdot x_j = 0$.)

If the sum of each row of M is at least one, then M represents *global mutex constraints*. As stated above, this simply means that at least one constraint applies to each variable. We observe that both 1-1 and many-to-one matching constraints are global mutex constraints.

The first step in our approach is the relaxation of the mutex constraints by moving them to the target function:

$$\begin{aligned} &\text{maximize } \ f(\mathbf{x}) = \mathbf{x^T} W \mathbf{x} = \mathbf{x^T} A \mathbf{x} - \mathbf{x^T} M \mathbf{x} \\ &\text{s.t. } \ \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{2.4}$$

where $W = A - M$. Hence, when $\mathbf{x}$ violates the mutex constraints, $f(\mathbf{x})$ will decrease. Although mutex constraints are relaxed, our goal is to ensure that any solution satisfies $\mathbf{x}^T M \mathbf{x} = 0$.

Problem (2.4) is known as an integer quadratic program (IQP). A lot of effort has been spent on finding good approximate solutions of (2.4) by relaxing the constraints, e.g., in the

case of graph matching problems in [138, 95, 136, 32, 11], and in the case of MAP inference algorithms for MRFs in [112, 81, 31]. This is usually achieved by relaxing the discrete vertex selection vector $\mathbf{x}$ to a continuous vector. We also relax the binary constraints in (2.4) to continuous ones:

$$\begin{aligned} &\text{maximize } \ f(\mathbf{x}) = \mathbf{x}^{\mathbf{T}} W \mathbf{x} = \mathbf{x}^{\mathbf{T}} A \mathbf{x} - \mathbf{x}^{\mathbf{T}} M \mathbf{x} \\ &\text{s.t. } \ \mathbf{x} \in [0,1]^n. \end{aligned} \tag{2.5}$$

Our main contribution is a novel algorithm for solving problem (2.5), presented in Section 2.3. Its key property is the fact that if solution $\mathbf{x}^*$ is discrete, then $\mathbf{x}^*$ is guaranteed to satisfy all mutex constraints, i.e., $(\mathbf{x}^*)^{\mathbf{T}} M \mathbf{x}^* = 0$, as we prove in Section 2.6. Consequently, a discrete solution $\mathbf{x}^*$ of (2.5) is also a solution of the original problem (2.3), since $f(\mathbf{x}^*)$ equals to $g(\mathbf{x}^*)$ in (2.3).

Problem (2.4) can be viewed as a special form of binary MRFs. However, standard optimization techniques, such as Iterative Conditional Modes (ICM) and Simulated Annealing (SA), suffer from mutex constraints (hard pairwise constraints), and do not perform well [117]. The main reason is the fact that mutex constraints introduce large non-submodular terms to the energy function (or equivalently large negative terms in (2.4)).

Recently, several successful binary MRFs solvers focusing on optimizing non-submodular function have been proposed. QPBO via roof duality enables graph cuts algorithm to solve problems with non-submodular terms, but often provides only part of optimal solution. Its usefulness depends on how many variables are labeled, which is still an open question, and can be only examined through experiments [71]. As pointed out in [63], only a very small percentage of variables get assigned labels by QPBO in the presence of large non-submodular terms. This fact is also confirmed by our experimental results.

The most famous extensions of QPBO are QPBOP (probing) [16] and QPBOI (improving) [117]. Both methods address the problem of partial labeling of QPBO. QPBOI shows

excellent results on some applications whose modeling requires a limited number of local hard pairwise constraints, such as Interactive Segmentation [117]. In [117], QPBOP and QPBOI are combined in to a unified method called QPBOP + I, which is also shown to be superior to both methods. Simply speaking, QPBOP + I boils down to first running QPBOP and then initializing QPBOI with the partial solutions obtained by QPBOP. However, QPBOP + I is also unable to handle global mutex constraints, since they lead to large non-submodular terms and there is at least one such term for each variable (each MRF side). The same applies to LBP. This fact is in accord with the observations in [63] and we will also provide a clear experimental evidence supporting it.

Moreover, the proposed algorithm significantly outperforms IPFP [84], even if we restrict mutex constraints to be equivalent to $B\mathbf{x} = \mathbf{1}$. As is the case for IPFP, the proposed algorithm does not guarantee that the obtained solution is discrete, in which we can output the last discrete solution obtained during the computation following [84]. However, in all our experimental results on real data all obtained solutions are discrete. This in turn guarantees that the solutions satisfy the mutex constraints. This is a very important property for practical applications, since mutex constraints give a great flexibility for modeling application specific constraints, which can substantially improve the quality of the solution. As we demonstrate in the experimental results, this is of key importance for applications where the unary and binary potentials are not particularly informative.

The rest of the section is structured as follows. Related work is discussed in Section 3.2. In Sections 2.3 and 2.4, the proposed algorithm for computing maximal subgraphs that satisfy global mutex constraints is described. Its theoretical properties are analyzed in Sections 2.5 and 2.6. Experimental evaluation is presented in Section 2.7.

2.2 Related Work

A special instance of the MWS problem called maximum weight clique (MWC) problem is well-known, where the solution subgraph is also required to be a clique, i.e., any two of its vertices are connected by an edge with a positive weight. The first formulation of a matching problem as a maximal clique in a correspondence (binary) graph introduced for object recognition in [1].

Recently an extension of MWC to weighted graphs was proposed in [108], where a special case of (2.2) is considered in which $B = \mathbf{1}^T$, a row vector of ones. Hence the L_1 norm of $\mathbf{x}$ must equal one, meaning that $\mathbf{x}$ belongs to a simplex. This formulation essentially finds cliques with maximum average weights, and the solution is sparse [108, 13]. [108] showed that when $B = \mathbf{1}^T$, problem (2.2) generalizes the concept of maximum cliques from un-weighted graphs to weighted graphs. Note that, in IPFP [84] and our approach, the solution is not restricted to a simplex, i.e., the L_1 norm of $\mathbf{x}$ is not restricted to equal 1.

Our algorithm is related but very different from the Frank-Wolfe (FW) algorithm [49] and other line search algorithms. We elaborate on this relation in Section 2.5. There is also a big difference in the target function, since FW algorithm is not designed to optimize an indefinite target function and in general performs badly in this case [29].

As stated in the introduction, MWS problem is also related to pseudo-boolean optimization [15], from the point of view of energy function and binary discrete domain. In fact, it is easy to notice that Eq. 2.4 is in a special form of binary MRFs, and the energy function has both submodular and non-submodular terms. While binary MRFs with all submodular terms can be efficiently solved by graph cuts algorithms, energy function with non-submodular term, which arise naturally in real applications, are in general NP-hard. One simple way to deal with non-submodular terms is 'truncating' them, i.e., replacing a function with a submodular approximation and optimize the latter [118]. However, when

the number of non-submodular terms is large, the truncation may not be appropriate [71]. In our MWS problem formulation, non-submodular terms come from mutex constraints.

The proposed algorithm for solving problem (2.5) has been first published by the authors in a conference paper [94]. However, since the focus of this paper was on object video segmentation, neither theoretical properties of the algorithm nor the solutions were analyzed. Moreover, no comparison to the recent MRF solvers was presented. In addition to this new theoretical content, we provide experimental comparison to the MRF solvers on two challenging combinatorial problems feature point matching and image jigsaw puzzle solving. Now we review these applications as well as the works on video object segmentation.

Feature point matching: it is well known that graph matching framework is very powerful and robust when it is used to address the feature correspondence problem [54]. There exist tons of work where graph matching framework is applied to solve computer vision problems related to feature point matching, such as shape matching, object recognition and video analysis [11, 83, 139, 156, 37, 25, 38, 26]. In a graph matching framework, each local feature is represented by a node, and edge attribute is then used to represent the spatial relation between local features. The main drawback of graph matching lies in its NP-hard nature. Existing approaches either propose novel graph matching algorithm based on various approximations [80, 25], or consider a higher-order relation between local features [37, 74, 86]. In some works [85, 82], a better edge attribute affinity is learned to improve the graph matching results.

Different from the existing methods, we consider mutex constraints in a graph matching framework to address the feature point matching problem. In particular, while the affinity between edge attribute is usually utilized in a soft manner in most existing work, we consider the hard, mutual exclusive constraints, such as qualitative spatial relations above/below and left/right, which significantly improve the quality of solutions. These

are global mutex constraints if for each feature point there exists at least one other point above/below or left/right to it, which is the case in real applications.

Image jigsaw puzzle: Shape based jigsaw puzzle problem has been a long standing problem in computer vision [34, 50, 72]. Recently, image jigsaw puzzle problem is revisited in [27], where each image is divided into squared pieces, and the goal is to use these pieces to reconstruct the original image. It is easy to see that this is a combinatorial problem. A MRFs formulation is adopted in [27], and loopy belief propagation is used to solve it. In [155] it is observed that by strictly enforcing the one-to-one correspondence between puzzle pieces and board locations the image reconstruction results can be significantly improved. To ensure that these hard mutex constraints are satisfied a particle filter framework (sequential Monte Carlo) is adopted. The drawback of this method relies in its high computational complexity. (To be more accurate, to achieve a relatively good solution quality, a large number of particles must be utilized.) In this work, we also enforce the one-to-one constraints. However, we formulate the problem as finding MWSs with mutex constraints and solve it with the proposed algorithm. Although image jigsaw puzzle seems not to be very practical by itself, it is shown to be a very effective framework in image segmentation [24] and scene labeling [110].

Video object segmentation: Given an unannotated video, the task is to automatically identify the primary object, and segment that object out in every frame. Unsupervised video object segmentation is important for many potential applications, such as activity recognition and video retrieval. Existing methods explore tracking of regions or keypoints over time [19, 21, 114] or perform low-level grouping of pixels from all frames using appearance and motions cues [61, 56]. However, as pointed out in [76], these methods lack an explicit notion of *what a foreground object should look like* in video. In [147], an object cosegmentaiton problem in a set of static images are solved by selecting one region propos-

al per image in a way that their coherence is maximized. This is a combinatorial problem, and A^* algorithm is utilized in [147]. However, due to the complexity of A^*, it can only solve small scale problems. To address the video object segmentation problem, we use the similar idea as in [145]. However, in addition to the constraint that one region proposal is selected each image, we also enforce the constraint derived from the motion coherence. Finally, we formulate the problem as finding MWS that satisfy these mutex constraints and solve it with the proposed algorithm. As the solution we obtain exactly one foreground object in all frames simultaneously.

Video object segmentation: We propose an approach for view-invariant object detection directly in 3D with following properties: (i) The detection is based on matching of 3D contours to 3D object models. (ii) The matching is constrained with qualitative spatial relations such as above/below, left/right, and front/back. (iii) In order to ensure that any matching solution satisfies these constraints, we formulate the matching problem as finding maximum weight subgraphs with hard constraints, and utilize a novel inference framework to solve this problem. Given a single view of an RGB-D camera, we obtain 3D contours by ”back projecting” 2D contours extracted in the depth map. As our experimental results demonstrate, the proposed approach significantly outperforms the state-of-the-art 2D approaches, in particular, latent SVM object detector, as well as recently proposed approaches for object detection in RGB-D data.

2.3 Algorithm Description

The proposed algorithm has similar properties to IPFP [84] in that it iteratively seeks the solution between discrete domain and continuous domain while keeping the score of the target function climbing. It does not guarantee that the obtained solution is discrete, although it always targets a discrete solution, and the final continuous solution is most often discrete in practice. However, our algorithm differs from IPFP in three key aspects: (A1)

Our final problem formulation (2.5) is very different from IPFP. In particular, we relax the mutex constraints at the beginning and move it to the target function as a penalty term, while the linear constraints are enforced directly in IPFP. (A2) Consequently, the proposed algorithm targets a discrete solution in a different way than IPFP. IPFP handles the linear equality constraints explicitly when finding the discrete solution in each iteration, e.g., Hungarian algorithm is used in the case of one-to-one constraints. Consequently, each intermediate discrete solution must satisfy all the constraints. This significantly narrows the search space, but with a serious danger of losing better solutions. In contrast we do not force the intermediate solutions to satisfy the constraints, which often leads to better solutions as compared to IPFP. In each iteration step we discretize a current continuous solution by performing a local first-order Taylor approximation, which results in a simple discretization step introduced in [20]. While the goal of their algorithm is maximization of a linear function, our goal is maximization of a quadratic function. (A3) In order to gain expressive power, we allow for mutex constraints expressed in a quadratic form $x^T M x = 0$ as opposed to linear equality form. As we elaborate above, the mutex constraints in the relaxed formulation are stronger than both linear equality and inequality constraints.

A weighted graph G is defined as $G = (V, A)$, where $V = \{v_1, \ldots, v_n\}$ is the vertex set, n is the number of vertices, and A is a symmetric $n \times n$ affinity matrix with all nonnegative entries, i.e., $A_{ij} \geq 0$ for all $i, j = 1, \ldots, n$.

In the remainder of this section $f(\mathbf{x}) = \mathbf{x}^\mathbf{T} W \mathbf{x}$ denotes the objective function in (2.5), where W is a symmetric matrix.

The proposed algorithm visits a sequence of continuous points $\{\mathbf{y}_{(k)} \in [0,1]^n\}_{k=1,2,\ldots}$. We assume that the initial assignment $\mathbf{y}_{(0)}$ satisfies the mutex constraints, i.e., ${\mathbf{y}_{(0)}}^T M \mathbf{y}_{(0)} = 0$. This implies that $f(\mathbf{y}_{(0)}) \geq 0$, since all entries in A are nonnegative.

In each iteration k, we have two steps. First, we compute the first-order Taylor approximation of $f(\mathbf{y})$ around $\mathbf{y}_k$ as

$$\begin{aligned} f(\mathbf{y}) &\approx f(\mathbf{y}_{(k)}) + 2(\mathbf{y} - \mathbf{y}_{(k)})^T W \mathbf{y}_{(k)} \\ &= 2\mathbf{y} W \mathbf{y}_{(k)} - f(\mathbf{y}_{(k)}) \end{aligned} \tag{2.6}$$

Since the second term $f(\mathbf{y}_{(k)})$ in (2.6) does not depend on $\mathbf{y}$, the first-order Taylor approximation of $f(\mathbf{y})$ only depends on $\mathbf{y}W\mathbf{y}_{(k)}$, which is a linear function of $\mathbf{y}$. This fact allows an easy computation of a *discrete* maximizer

$$\tilde{\mathbf{x}}_{(\mathbf{k})} = \underset{\mathbf{y} \in [0,1]^n}{\arg\max}\, \mathbf{y}^T W \mathbf{y}_{(k)} \tag{2.7}$$

as

$$(\tilde{\mathbf{x}}_{(\mathbf{k})})_i = \begin{cases} 1, & \text{if } (W\mathbf{y}_{(k)})_i > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.8}$$

In the second step of iteration k, we want to verify whether the obtained $\tilde{\mathbf{x}}_{(\mathbf{k})}$ can be accepted as a valid discrete solution that increases f. In the case that $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) > f(\mathbf{y}_{(k)})$, we set $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})}$. Hence we prefer a discrete solution if it increases the f value, but if $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) \leq f(\mathbf{y}_{(k)})$, we estimate the local maximizer of f in the continuous domain by performing line search, i.e., by maximizing one dimensional function $h(\alpha) = f(\mathbf{y}_{(\mathbf{k})} + \alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)}))$ over the line segment from $\tilde{\mathbf{x}}_{(\mathbf{k})}$ to $\mathbf{y}_{(k)}$. In Section 2.6 we show that $h(\alpha)$ obtains its maximum at α defined in (2.9). We also show that $0 < \alpha < 1$, which guarantees that line search will not reach outside the box $[0, 1]^n$.

$$\alpha = -\frac{(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W \mathbf{y}_{(k)}}{(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})} \tag{2.9}$$

Then we set $\mathbf{y}_{(k+1)} = \mathbf{y}_{(\mathbf{k})} + \alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})$.

In the above two cases, $f(\mathbf{y}_{(k+1)}) > f(\mathbf{y}_{(k)})$ as we will show in Proposition 1 below. Our algorithm stops when the following *stop condition* holds for all coordinates i of vector $\mathbf{x}^* = \mathbf{y}_{(k+1)}$:

$$\begin{aligned} &\text{if } (W\mathbf{x}^*)_i > 0, \quad \text{then } \mathbf{x}_i^* = 1 \\ &\text{if } (W\mathbf{x}^*)_i < 0, \quad \text{then } \mathbf{x}_i^* = 0 \end{aligned} \tag{2.10}$$

We observe that $W\mathbf{x}^* = \frac{1}{2}\nabla f(\mathbf{x}^*)$. Hence $(W\mathbf{x}^*)_i > 0$ means that the direction of the increase of f coincides with the direction of ith coordinate, while $(W\mathbf{x}^*)_i < 0$ means that the direction of the increase of f is opposite to the direction of ith coordinate. Therefore, the stop condition tells us that $f(\mathbf{x}^*)$ already has the maximum possible value for every increase direction of f. In other words, we cannot increase f without leaving our domain $[0,1]^n$, meaning that $\mathbf{x}^*$ is a local maximum of f over $[0,1]^n$.

By Proposition 1 below, $f(\mathbf{y}_{(k)})$ is strictly increasing in each iteration. Consequently, $f(\mathbf{y}_{(k)}) > f(\mathbf{y}_{(0)}) \geq 0$ for $k > 0$. This fact in turn implies that $W\mathbf{y}_{(k)} \neq 0$. Suppose $W\mathbf{y}_{(k)} = 0$. In this case $f(\mathbf{y}_{(k)}) = {\mathbf{y}_{(k)}}^T W\mathbf{y}_{(k)} = 0$, which contradicts $f(\mathbf{y}_{(k)}) > 0$. Hence, the assumption $f(\mathbf{y}_{(0)}) \geq 0$ implies that for every iteration $k > 0$ the gradient of f is a nonzero vector.

2.4 Algorithm

2.5 Relation to Frank-Wolfe Algorithm

In FW and related algorithms [49, 29], after obtaining $\tilde{\mathbf{x}}_{(\mathbf{k})}$ with Eq. (2.8), the maximum value of the target function along the line from $\mathbf{y}_{(k)}$ to $\tilde{\mathbf{x}}_{(\mathbf{k})}$ is always computed, which is also done in lines 8, 9 of our algorithm. However, we prefer a discrete solution $\tilde{\mathbf{x}}_{(\mathbf{k})}$ if it increases the target function (lines 5, 6), even if the value of $f(\tilde{\mathbf{x}}_{(\mathbf{k})})$ is lower than

Input: Matrix W, $f(\mathbf{y}_{(0)}) \geq 0$, and $\epsilon > 0$

1: **repeat**
2: Use (2.8) to find $\tilde{\mathbf{x}}_{(\mathbf{k})} = \arg\max_{\mathbf{y}\in[\mathbf{0},\mathbf{1}]^{\mathbf{n}}} \mathbf{y}\mathbf{W}\mathbf{y}_{(k)}$
3: **if** $\tilde{\mathbf{x}}_{(\mathbf{k})} = \mathbf{y}_{(k)}$ **then**
4: $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})}$
5: **else if** $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) > f(\mathbf{y}_{(k)})$ **then**
6: $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})}$
7: **else**
8: Use (2.9) to compute α.
9: $\mathbf{y}_{(k+1)} = \mathbf{y}_{(\mathbf{k})} + \alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})$
10: **end if**
11: **until** $\mathbf{y}_{(k+1)}$ satisfies (2.10) or $f(\mathbf{y}_{(k+1)}) - f(\mathbf{y}_{(k)}) < \epsilon$

Output: $\mathbf{y}_{(k+1)}$

the maximal value along the line. In contrast, FW always takes the maximal value along the line. The preference for discrete solutions in each step has a dramatic impact on the obtained final solutions. We obtain discrete solutions in all real applications and matrix W in the target function is always indefinite, while FW and related algorithms cannot obtain any reasonable solution in this case. Of course, we need to stress that FW and related algorithms are designed for optimizing the target function defined with a PSD matrix W.

2.6 Properties of the Algorithm

In this section, we are going to establish the main properties of the proposed algorithm. With a symmetric W, we first show that the target function f increases in every iteration in Proposition 1 below. Considering f is an upper bounded function, our algorithm is guaranteed to converge. In Proposition 2, we state the key property of the proposed algorithm: if the algorithm halts with a discrete solution, then the solution is guaranteed to satisfy the mutex constraints.

We begin with a simple observation that if $\tilde{\mathbf{x}}_{(\mathbf{k})} = \mathbf{y}_{(k)}$ in line 3, then the stop condition (2.10) in line 11 is true. In this case the solution $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})}$ is discrete. This holds, since then $\tilde{\mathbf{x}}_{(\mathbf{k})}$ is a fixed point of the operator in Eq. (2.8), and consequently, it satisfies

condition (2.10).

Proposition 1: f increases in every iteration, i.e., if $\mathbf{y}_{(k)}$ does not satisfy (2.10), then $f(\mathbf{y}_{(k+1)}) > f(\mathbf{y}_{(k)})$ for all k.

Proof. In iteration k of the algorithm, if $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) > f(\mathbf{y}_{(k)})$ then the next point visited by algorithm is $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})}$. In this case, the implication is trivially true, since $f(\mathbf{y}_{(k+1)}) = f(\tilde{\mathbf{x}}_{(\mathbf{k})}) > f(\mathbf{y}_{(k)})$.

If $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) \leq f(\mathbf{y}_{(k)})$, we perform line search by $\mathbf{y}_{(k+1)} = \tilde{\mathbf{x}}_{(\mathbf{k})} + \alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})$. Here, according to the algorithm, we have two conditions valid before we execute the line search step: $\tilde{\mathbf{x}}_{(\mathbf{k})} \neq \mathbf{y}_{(k)}$ and $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) \leq f(\mathbf{y}_{(k)})$. Since

$$\begin{aligned} f(\mathbf{y}_{(k+1)}) &= f(\mathbf{y}_{(k)} + \alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})) \\ &= f(\mathbf{y}_{(k)}) + 2\alpha(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W \mathbf{y}(k) \\ &+ \alpha^2(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)}) \end{aligned}$$

we obtain

$$f(\mathbf{y}_{(k+1)}) - f(\mathbf{y}_{(k)}) = h(\alpha) = d\alpha^2 + 2c\alpha \tag{2.11}$$

by setting $c = (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W \mathbf{y}_{(k)}$ and $d = (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})$.

$h(\alpha)$ is a quadratic function in α, and $h'(\alpha) = 2(d\alpha + c)$. Hence $h(\alpha)$ can only have its maximum at $\alpha = -\frac{c}{d}$.

We will show below that $c > 0$ and $d < 0$ and that $0 < -\frac{c}{d} < 1$. This implies that $\alpha = -\frac{c}{d}$ is indeed the maximum of $h(\alpha)$. Since $\alpha > 0$ and $2c + \alpha d = c > 0$, we obtain and that > 0:

$$h(\alpha) = f(\mathbf{y}_{(k+1)}) - f(\mathbf{y}_{(k)}) = \alpha(d\alpha + 2c) > 0 \tag{2.12}$$

Therefore, $f(\mathbf{y}_{(k+1)}) > f(\mathbf{y}_{(k)})$.

In the remainder of the proof we first show that $c = (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W \mathbf{y}_{(k)} > 0$. Since $W\mathbf{y}_{(k)}$ is a nonzero vector due to the initialization, if $\mathbf{y}_{(k)}$ does not satisfy condition (2.10), there either exists i such that $(W\mathbf{y}_{(k)})_i > 0$ and $(\mathbf{y}_{(k)})_i < 1$ or there exists i such that $(W\mathbf{y}_{(k)})_i < 0$ and $(\mathbf{y}_{(k)})_i > 0$.

For every i for which $(W\mathbf{y}_{(k)})_i > 0$ and $(\mathbf{y}_{(k)})_i < 1$ hold, we have $(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})_i (W\mathbf{y}_{(k)})_i > 0$, since $(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})_i > 0$ in this case.

We arrive at the same conclusion if $(W\mathbf{y}_{(k)})_i < 0$ and $(\mathbf{y}_{(k)})_i > 0$ hold, since we have $(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})_i < 0$, and therefore, $(\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})_i (W\mathbf{y}_{(k)})_i > 0$. Since all other coordinates of vector $W\mathbf{y}_{(k)}$ are zero, we obtain that $c = (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W \mathbf{y}_{(k)} > 0$.

In order to show that $d < 0$, we observe

$$\begin{aligned} d &= (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)}) \\ &= f(\tilde{\mathbf{x}}_{(\mathbf{k})}) - 2\tilde{\mathbf{x}}_{(\mathbf{k})}^T W \mathbf{y}_{(k)} + f(\mathbf{y}_{(k)}) \end{aligned} \tag{2.13}$$

Since we have $f(\tilde{\mathbf{x}}_{(\mathbf{k})}) \leq f(\mathbf{y}_{(k)})$, we obtain $d \leq f(\mathbf{y}_{(k)}) - 2\tilde{\mathbf{x}}_{(\mathbf{k})}^T W \mathbf{y}_{(k)} + f(\mathbf{y}_{(k)}) = -2c < 0$.

Since $c > 0$ and $d < 0$, thus $\alpha = -\frac{c}{d} > 0$. In addition, we have $\alpha = -\frac{c}{d} < 1$, because $c + d = (\tilde{\mathbf{x}}_{(\mathbf{k})} - \mathbf{y}_{(k)})^T W \tilde{\mathbf{x}}_{(\mathbf{k})} \leq f(\mathbf{y}_{(k)}) - \mathbf{y}_{(k)}{}^T W \tilde{\mathbf{x}}_{(\mathbf{k})} < 0$, which guarantees the line search will not reach outside the cube. Thus, we have just shown that $c > 0$, $d < 0$, and $0 < \alpha = -\frac{c}{d} < 1$. □

Proposition 2: If the algorithm halts with a discrete solution $\mathbf{x}^* = \mathbf{y}_{(k+1)}$, i.e., the stop condition (2.10) applies to $\mathbf{x}^*$ and $\mathbf{x}^*$ is discrete, then $\mathbf{x}^*$ satisfies the mutex constraints, i.e., $(\mathbf{x}^*)^{\mathbf{T}} M \mathbf{x}^* = 0$.

Proof. Suppose that the proposition is not true, i.e., there exists i with $\mathbf{x}_i^* = 1$ that violates a mutex constraint. Then $(\mathbf{x}^*)^T M \mathbf{x}^* \geq 1$. Because $(\mathbf{x}^*)^T A \mathbf{x}^* \leq 1$, we obtain $f(\mathbf{x}^*) =$

$(\mathbf{x}^*)^T A\mathbf{x}^* - (\mathbf{x}^*)^T M\mathbf{x}^* \leq 0$. A contradiction, since by Proposition 1, $f(\mathbf{x}^*) > f(\mathbf{y}_{(0)}) \geq 0$. □

Complexity: In each iteration, the algorithm in Section 2.4 has complexity of $O(w)$ where w is the number of nonzero entries in matrix W. Complexity is determined by line 8, in which several vector multiplications with W is computed following Eq 2.9. As illustrated in Section 2.7, W is very sparse in many applications. Depending on the problem, in our experimental results the number of iterations was between 10 and 130.

2.7 Experimental Evaluation

A large number of machine learning and computer vision tasks can be expressed as constrained MWSs. First, we consider two challenging tasks, which are known as hard combinatorial optimization problems, matching salient points (Sec 2.8) and solving image jigsaw puzzle (Sec 2.9). We demonstrate that the proposed algorithm significantly outperforms state-of-the-art methods on both tasks. As stated in the introduction, the main reason is the presence of global mutex constraints. In particular, we provide a clear evidence that LBP, QPBO, QPBOP +I cannot handle the resulting non-submodular terms.

In these two experiments, we use the same initialization for all the compared methods. Particularly, for IPFP and our method, the same x_0 is used. For LBP, QPBO, QPBOP +I, we use the same initialization x_0 by performing the operation called "fixing a node" following [117], which essentially manipulates the graph through assigning sufficiently large constant to those nodes whose corresponding element in x_0 is 1.

Furthermore, in Sec 2.10, we demonstrate how MWSs can be used to solve the video object segmentation problem effectively, and how global mutex constraints (GMCs) can be used to significantly improve the quality of the solution.

Finally, in Sec 2.12, we conduct a random matrix test to examine how often the proposed algorithm converges to a discrete solution under extreme conditions.

In all of our experiments ϵ is set to be $1e-6$.

2.8 Matching of Salient Points on Faces

We use the face dataset from [97], e.g., see Fig. 2.1, which consists of 107 face images of 11 different people. There are scale and strong pose variations between different faces. Each face is represented by 7 or less salient points located on the eyes, nose and mouth. Some salient points are missing in some images due to self occlusion. We match each image to all other images. Thus, there are 107×106 face pairs and 59980 points correspondences in total.

Given a query point set P on one face with n_p points and point set Q with n_q points on another face, we aim to find the assignments of points $i \in P$ to $i' \in Q$. In our framework, each assignment (i, i') between two points is represented by a node in the correspondence graph. Therefore, the best configuration of matchings is identified as a constrained maximum weight subgraph.

The weight $A(u, u)$ of vertex $u = (i, i')$ (unary potential) encodes the similarity between local appearance (SIFT) features f_i and $f_{i'}$ of points i and i' in two images. The weight of the edge between $u = (i, i')$ and $v = (j, j')$ (binary potential) encodes the pairwise distance consistency between two assignments, $A(u, v) = \exp(-\frac{(d(i,j)-d(i',j'))^2}{2\sigma_d^2})$, where $d(i, j)$ is the Euclidean distance between i and j in one image, and $d(i', j')$ is the Euclidean distance between i' and j' in the other image. σ_d controls the sensitivity to variations of geometric deformation.

In order to define mutex constraints, we make a few simple observations. We observe that some qualitative geometric relations between face salient points are usually preserved, even under some serious perspective distortion. For example, two corner points of left eye are always on the left of the right eye points. Similarly, points on the nose are always above the mouth points. Hence if two assignments, $u = (i, i')$ and $v = (j, j')$, violate

the qualitative geometric constraints, then u and v should not appear in the same MWS. Formally, we can define the left-right mutex constraint as

$$M_{l/r}(u,v) = \begin{cases} 1, & \text{if } |i_x - j_x| > \theta \text{ , } |i'_x - j'_x| > \theta \\ & \text{and } (i_x - j_x)(i'_x - j'_x) < 0 \\ 0, & \text{otherwise.} \end{cases}$$

where i_x, j_x, i'_x and j'_x are the x-coordinates of points i, j, i', j'. Threshold θ excludes points whose x-coordinates are too close to each other. In other words, θ controls the amount of allowable deformation in the horizontal direction. Similarly, we can define the above-below constraints $M_{a/b}$. Hence $M_{l/r} \vee M_{a/b}$ represents qualitative geometric constraints (QC).

To enforce the standard one-to-one constraints (1-1) in a matching problem, we define the mutex relation between two nodes $u = (i, i')$ and $v = (j, j')$ as $M_{1-1}(u, v) = 1$ if ($i = j$ and $i' \neq j'$) or ($i \neq j$ and $i' = j'$), and $M_{1-1}(u, v) = 0$ otherwise. The mutex matrix is then defined as logical or $M = M_{1-1} \vee M_{l/r} \vee M_{a/b}$. We observe that both matrices M_{1-1} and $M_{l/r} \vee M_{a/b}$ represent global mutex constraints. By setting $W = A - M$, the problem of the face salient point matching is expressed as problem (2.3), and the proposed algorithm (Sec. 2.4) is used to solve it.

Note that, if the only constraints used are 1-1 constraints, then $\mathbf{x^T} M\mathbf{x} = 0$ in our method is equivalent to $B\mathbf{x} = \mathbf{1}$ in [84]. $\mathbf{x^T} M\mathbf{x} = 0$ implies $\sum_{x_i \in \mathcal{C}} x_i \leq 1$, where $\mathcal{C}$ is a maximal set of nodes such that every two nodes in $\mathcal{C}$ are mutually exclusive. Since there are no other constraints, always one vertex from $\mathcal{C}$ is selected in order to increase $\mathbf{x^T} A\mathbf{x}$, which implies that $\sum_{x_i \in \mathcal{C}} x_i = 1$.

We observe that the qualitative geometric constraints cannot be formulated in a linear equality form, which implies that they cannot be utilized by IPFP. Therefore, we ran two versions of our algorithm, one without QC constraints, i.e., only with 1-1 constraints, called Alg. w/o QC, in order to directly compare to IPFP, and one with 1-1 and QC constraints, called Alg., in order to illustrate the performance gain obtained by the modeling flexibility

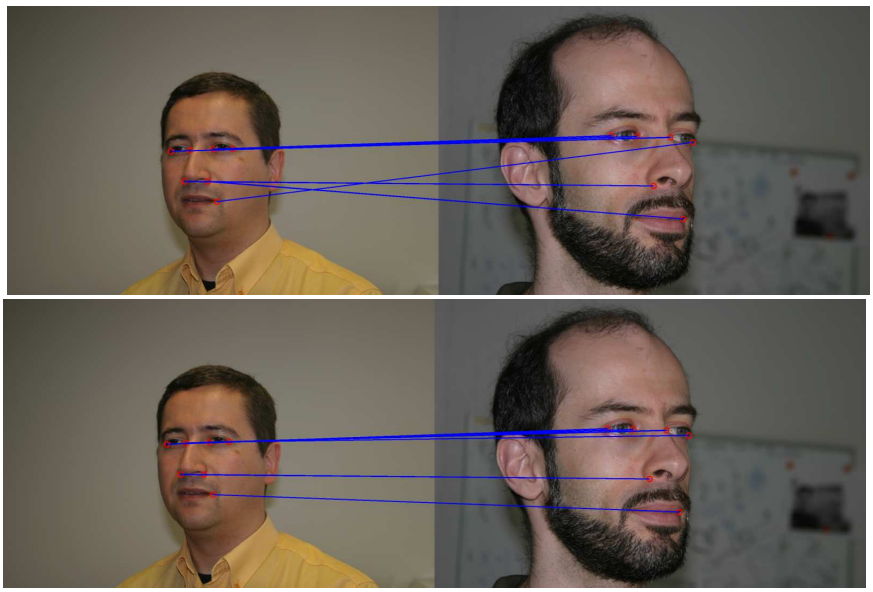

Figure 2.1: Example of face salient points matching. The first row is obtained by our method without qualitative constraints (QC). The second row is obtained with QC.

of mutex constraints in the quadratic form. An example that compares their results is shown in Fig. 2.1.

The results are reported in Table 2.1 as the percentage of correct matching pairs (i.e., recall), the value of the target function f (the higher the better), and the running time in seconds. With the same 1-1 constraints, our method outperforms IPFP by 24%. With 1-1 and QC our method increases the performance by 6%, and outperforms other state-of-the-art methods: [97], which is directly focused on modeling geometric constraints under view point change for point matching, and [91], which solves (2.1) on a simplex, i.e., computes MWCs with maximal average weights. The results of these two methods are quoted from [91]. The two methods ([97] and [91]) optimize different target functions, hence we do not report their target function value. Note that, with only 1-1 constraints, the size of the subgraph is larger than with qualitative constraints (QC), therefore, the f-values can be larger than the values of the last four approaches. LBP, QPBO, QPBOP + I [1] maximize

[1] `http://www.robots.ox.ac.uk/~ojw/files/imrender_v2.4.zip`

method	recall	f-value	time
[97]	95.7	-	-
[91]	97.1	-	-
IPFP [84]	67.1	26.92	0.03s
Our Alg. w/o QC	91.3	27.54	0.02s
LBP	80.6	24.81	0.46s
QPBO	14.7	0.21	0.01s
QPBOP + I	64.8	19.97	0.23s
Our Alg.	97.3	26.88	0.02s
CPLEX	**97.4**	**26.89**	7.20s

Table 2.1: Results on face dataset [97].

the same energy function as our Alg. with 1-1 and QC (to be precise, they minimize the negative of our target function f).

We observe that our method significantly outperforms LBP, QPBO, QPBOP + I in both recall and the value of the target function f. Our running time is only slightly worse than QPBO. However, QPBO assigned values to only 31.7% of variables, which also explains why its scores are extremely low. In contrast, our method, LBP and QPBOP + I, assigned values to all variables.

In addition, we compare to CPLEX with exact the same quadratic target function and constraints. IBM CPLEX (v12.4) which is able to solve mixed-integer quadratic (linear) programming problems using a branch-and-bound algorithm (b&b), a well-known method aiming for a global optimal solution for non-convex problems. The fact that both the recall and the f-value of our algorithm are nearly identical to those of CPLEX demonstrates that the proposed algorithm can get as close to globally optimal solutions as CPLEX. While CPLEX takes 7.20s on average to solve one problem, it takes 0.02s for our method, which is 360 times faster. This shows that our algorithm can scale up to larger problems that are prohibitive for CPLEX if they can be formulated as MWS with global mutex constraints, for example, the image jigsaw puzzle problem considered in the next section.

2.9 Solving image jigsaw puzzle

While the problem considered in Section 2.8 is a small problem, we move to a significantly larger problem now, which is the problem of solving image jigsaw puzzles. It is defined as reconstructing an image from a set of square and non-overlapping image patches [27]. Since the original image is not given, this problem is very challenging, even for humans, e.g., see Fig. 2.2.

Given k different puzzle pieces $(p_1, \ldots, p_k)$ and k board locations $(l_1, \ldots, l_k)$, our goal is to assign puzzle pieces to board locations. Each possible assignment is a node in a correspondence graph, and a solution of the jigsaw puzzle problem is identified as a MWS that satisfies 1-1 mutex constraints.

Formally, each node v in the association graph G is defined as (p_i, l_m), which means puzzle piece p_i is assigned to location l_m. Therefore, if there are k locations and k different puzzle pieces, the total number of nodes in graph is $k \times k$. Two nodes $v = (p_i, l_m)$ and $u = (p_j, l_n)$ are adjacent if and only if l_m and l_n are adjacent board locations. For each location l_m, we define its 4-neighbors (if they exist): left, right, top, bottom, as its adjacent locations. If nodes u and v are not adjacent, we set $A(v, u) = 0$. We also ensure that the affinity matrix A has positive values when nodes u and v are adjacent. We observe that the number of nonzero entries in A is not larger than $4k^2$, since each of k board locations is adjacent to at most 4 other locations, each of which can be assigned k puzzle pieces. In other words, each of k graph nodes has no more than $4k$ adjacent nodes. This implies the $k^2 \times k^2$ affinity matrix A is sparse. In order to compute the affinity value in matrix A for a pair of adjacent nodes, we follow exactly the same computation as in [27]. In fact we use the code released by the authors of this paper to compute the affinity matrix A.

Of course, the main constraint required for a good solution of the image jigsaw puzzle problem is 1-1 correspondence, i.e., 1) a puzzle piece should not be assigned to multiple locations, and 2) multiple puzzle pieces should not assigned to the same location. The 1-1 mutex matrix M is defined as in Sec. 2.8. We again recall that M represents global mutex

method	DC	f-value	time
LBP	0.05	50.565	70.3s
QPBO	0.02	0.001	1.90s
QPBOP + I	0.04	12.824	45.67s
IPFP [84]	0.83	148.705	1.52s
Our Alg.	**0.92**	157.298	0.32s

Table 2.2: Image Jigsaw Puzzle Results on MIT dataset with 48 patches [27].

constraints. By setting $W = A - M$, the problem of the image jigsaw puzzle is expressed as problem (2.3), and the proposed algorithm (Sec. 2.4) is used to solve it.

To evaluate the performance of solution for puzzle reconstruction, we use **Direct Comparison (DC)** following [27]. The inferred reconstruction result is compared directly to the ground-truth. DC is defined as the ratio of the correctly placed puzzle pieces to the total number of locations.

We initialized all methods with one anchor patch. We assigned the correct image patch to the top left corner of puzzle image. For all methods we used exactly the same pairwise potentials, which were computed by the code released by the authors of [27]. The diagonal of matrix A is set to zero, meaning that there is no prior on the expected image layout, i.e., no knowledge of either the target image or its image class is assumed. Since 1-1 constraints can be expressed as linear constraints, we ran IPFP with exactly the same mutex constraints as our algorithm. Also the same 1-1 constraints are used for LBP, QPBO, and QPBOP + I, and CPLEX. Hence all methods optimize the same target function.

The results on the MIT dataset [27] with 48 puzzle pieces are shown in Table 2.2. The proposed algorithm significantly outperforms the other methods in both the quality of the solution (DC) and the values of the target function. Moreover, it is at least few orders of magnitude faster than the other algorithms. QPBO was only able to assign values to a very small percentage of variables, only 2.08% of variables received values. The extremely low values of LBP, QPBO, and QPBOP + I clearly demonstrate that those methods are unable to handle global mutex constraints even on moderate size problems.

method	DC	f-value	time
LBP	0.03	124.669	1518.1s
QPBO	0.01	0.001	21.7s
QPBOP + I	0.02	27.263	544.8s
IPFP [84]	0.71	266.627	70.2s
Our Alg.	**0.91**	326.347	10.4s

Table 2.3: Image Jigsaw Puzzle Results on MIT dataset with 108 patches [27].

Some example reconstructions are shown in Fig. 2.2. The black squares represent image puzzle locations that were assigned no label (puzzle piece). They demonstrate another difficulty of LBP and QPBOP+I for problems with global mutex constraints. Although LBP and QPBOP+I assign labels to all variables, representing pairs (puzzle patch, puzzle location), they do not select all valid puzzle locations in their solutions, i.e., they assign value 0 to all pairs that involve the same puzzle location l_0. Hence this location results in a black patch at l_0. Since not all patches are assigned, there exists at least one patch p_i, such that adding the pair (p_i, l_0) to the solution would decrease the energy function (or equivalently increase the f-value), since no mutex constraint is violated then. This fact clearly demonstrates that LBP and QPBOP+I are unable to reach globally optimal solutions, which severely impairs the solutions of QPBOP+I in our experiment.

Although an image jigsaw puzzle with 48 puzzle pieces is a relatively small puzzle, CPLEX was not able to complete it (we ran CPLEX on a PC with 3.4Ghz CPU, it did not finish the computation for reconstructing one image in 12 hours). To evaluate the scale-up ability to larger problems, we applied the same algorithms as in Table 2.2 to solve larger puzzles with 108 puzzle pieces. The results are presented in Table 2.3.

For 108 puzzle pieces, our method achieves perfect reconstruction, i.e., 100% accuracy in direct comparison on 11 out of 20 test images from [27]. On average, our algorithm needs 128 iterations to converge. All experiments were ran on a quad core 3.4Ghz PC. IPFP and our algorithm are implemented in Matlab. LBP, QPBO, QPBOP + I are implemented in C++.

2.10 Video Object Segmentation

In this section, we address the problem of video object segmentation, which is to automatically identify the primary object and segment the object out in every frame. An example is shown in Fig 2.3. The selection of object region candidates simultaneously in all frames, is formulated as finding a maximum weight subgraph in a weighted region graph. The selected regions are expected to have high objectness score (unary potential) as well as share similar appearance (binary potential). Since both unary and binary potentials are unreliable, we introduce two types of mutex (mutual exclusion) constraints on regions in the same subgraphs: intra-frame and inter-frame constraints. Both types of constraints are expressed in a single quadratic form, and algorithm introduced in Sec 2.3 is applied to compute the maximal weight subgraphs. that satisfy the constraints.

In Sections 2.10, 2.10, and 2.10, we introduce the edge weights in the region graph, the mutex constraints on regions, and express region selection as finding constrained MWSs, respectively. In Section 2.10, we utilize the regions selected in Section 2.10 to achieve a more accurate pixel-level foreground object segmentation. The experimental comparison to state-of-the-art methods is presented in Section 3.6.

Region Graph Construction

Our goal is to segment a foreground object in video without any model of the target. Since we assume no prior knowledge on the size, location, shape or appearance of the target object, we first produce a bag of object ”proposals” in each frame using [39]. The model used in [39] is learned for a generic object from Berkeley Segmentation data, and therefore, it is category independent. Each proposal is a region in the image, an example is shown in Fig 2.4.

For each frame in the video, we retrieve K regions. (We set $K = 300$ in all experiments.) Given a video consisting of N frames, we have $K \times N$ regions in total. Our goal is to discover a small subset of regions that contain the same foreground object across all

the frames. We construct a weighted graph $G = (V, A)$, in which each node corresponds to one of the $K \times N$ regions, and A is its adjacency matrix. The weight $A(u, u)$ of the node u represents the "objectness" of the region u, while the weight $A(u, v)$ between two nodes u and v represents the similarity between the two regions. Both are defined below.

We follow the computation of the region "objectness" in [76]. Specifically, for a region u

$$A(u, u) = ob(u) = sob(u) + mob(u), \tag{2.14}$$

combines its static intra-frame objectness score $sob(u)$ and motion inter-frame objectness score $mob(u)$. The static score $sob(u)$ is computed using [39]. It reflects the confidence that a region contains a generic object. Several static cues are used to compute this score, such as the probability of a surrounding occlusion boundary, and color differences with nearby pixels.

In [76], the motion objectness $mob(u)$ is introduced to complement to the static score in the case of videos. It measures the confidence that region u corresponds to a coherently moving object in the video. Optical flow histograms are computed for the region u and the pixels $\overline{u}$ around it within a loosely fit bounding box. The score is computed as:

$$mob(u) = 1 - \exp(-\chi^2_{flow}(u, \overline{u})), \tag{2.15}$$

where $\chi^2_{flow}(u, \overline{u})$ is the χ^2-distance between L_1-normalized optical flow histograms. The motion score essentially describes how the motion of the region differs from its closest surrounding regions. Both static score and motion scores are normalized using the distributions of scores across all regions in the video.

Each region is also described using its Lab color histogram. The similarity between two regions u and v is computed as:

$$A(u, v) = \exp(-\frac{1}{\Omega}\chi^2_{color}(u, v)), \tag{2.16}$$

where$\chi^2_{color}(u, v)$ is the χ^2-distance between unnormalized color histograms of u and v, and Ω denotes the mean of the χ^2-distance among all the regions. Consequently, if two regions have similar color and similar size, their affinity is high.

Mutex Constraints between Regions

One of the key contributions of the proposed work to video segmentation lies in the utilization of hard, mutex (short for mutual exclusion) constraints. They specify which regions cannot be simultaneously selected as part of the segmentation solution. They allow us to eliminate unreasonable configurations of regions, which otherwise have large joint potentials, since both the unary $A(u, u)$ and binary potentials $A(u, v)$ are unreliable. Furthermore, the utilized inference framework allows us to enforce that the solutions satisfy all the constraints. The proposed mutex constraints are based on the following two insights.

Intra-frame mutex constraint: We assume that a true object should appear in *every* frame, and within each frame, only *one* proposal region should be selected. However, the object may be partially occluded or self occluded. This constraint implies that only one object regions candidate produced by [39] is selected for each frame. The same constraint is also utilized in the problem of object co-segmentation from static images [147]. The fact that exactly one object region candidate is selected in each frame is essential for a good selection of candidates mainly for two reasons: 1) Since many regions in the same frame overlap, their affinities are usually much higher than affinities of true object regions in different frames due to inter-frame variations, such as illumination change. Hence, by excluding affinities of regions from the same frame from consideration in a single subgraph, the comparison of affinities from different frames becomes more informative. 2) Since we guarantee to select one region for *every* frame, the region selected can be further used as location prior.

Inter-frame proximity constraint: two regions selected in two neighboring frames should be not spatially far away from each other, since the change of the location of the same ob-

ject in adjacent frames should be smooth.

We encode these two constraints through a binary mutex matrix M defined over all vertices of graph G as

$$M(u,v) = \begin{cases} 1, & \text{if } u \text{ and } v \text{ are in the same frame} \\ & \text{or (if } u \text{ and } v \text{ are in adjacent frames} \\ & \text{and } d(C(u), C(v)) > \tau \text{)} \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

where $C(u)$ and $C(v)$ are the centroid of two regions, and d is their Euclidean distance in pixels. τ reflects the maximum spatial displacement allowed between u and v. We set $\tau = 100$ for all the experiments in order to allow for fast moving objects.

Finding Objects as Constrained MWSs

We formulate a region selection problem as finding constrained maximum weight subgraphs in graph. For the affinity matrix A, the diagonal elements (unary potentials) are objectness measure, with off-diagonal elements (pairwise potentials) are object proposals coherence measure. For the mutex matrix M, if $M(i,j) = 1$ then the two vertices i, j cannot belong to the same maximum subgraph. $M(i,i) = 0$ for all vertices i. In other words, mutex represents incompatible vertices that two object proposals cannot be selected together.

We use the algorithm introduced in Sec 2.3 to solve the MWSs problem with mutex constraints. Since the maximal subgraph seeking algorithm we use converges to a local optimum, multiple initializations are required to promise a better performance. We rank the regions in graph G according to their unary score $A(u,u)$, and find the top-L best regions. Each time, we use one region u selected from those top-L best regions to initialize the maximal weight subgraph seeking algorithm. We denote the initialization as $\mathbf{x}_{(0)}$, then we set $(\mathbf{x}_{(0)})_u = 1$ and $(\mathbf{x}_{(0)})_i = 0$ for all $i \neq u$. Starting from the $\mathbf{x}_{(0)}$, we obtain a

maximal subgraph indicated by a binary vector $\mathbf{x}^*$. $\mathbf{x}^*$ is a local maximizer of $\mathbf{x}^T A\mathbf{x}$ while satisfying ${\mathbf{x}^*}^T M\mathbf{x}^* = 0$.

Therefore, we obtain L maximal subgraphs in total. We select the best one according to $\mathbf{x}^T A\mathbf{x}$. We find the selected regions as one entries in the indicator vector of this solution. Since the solution satisfies the constraints M defined in Sec 2.10, we select only one region in each frame, and guarantee every two regions selected in neighboring frames are relatively close to each other. These regions reflect the rough appearance and location of the object in each frame.

In all video segmentation experiments, the obtained solutions are discrete, and thus, they satisfy all mutex constraints. We also observe that both matrices A and W are indefinite for all test videos.

Foreground Object Segmentation

The obtained segmentation of the object in video in form of selected regions is not very precise. In particular, the segmentation error is lower-bounded by the object region candidates produced by [39]. The error may due to the inaccuracy of the original superpixel extraction or merging. Therefore, we follow the strategy of utilizing the selected regions to learn the appearance model for both foreground and background, e.g., [76, 147]. In addition, we also utilize the location priors. It is particularly easy in our framework, since we have exactly one object region in each frame. Finally, we use GrabCut [116] to infer a more accurate pixel-level object segmentation. For efficiency, rather than labeling pixels in three consecutive frames at once by constructing a space-time graph as in [76], we simply run the GrabCut [116] for each frame separately. This is possible in our framework, since the data term, defined below, which is obtained by our constrained MWSs is very informative.

We denote the pixels in each frame as $S = \{p_1, \ldots, p_n\}$, and their labels $\gamma = \{\gamma_1, \ldots, \gamma_n\}$, $\gamma_i \in \{0, 1\}$ with 0 for background and 1 for foreground. Then the energy

function for minimization is:

$$E(\gamma) = \sum_{i \in S} D_i(\gamma_i) + \delta \sum_{i,j \in \mathcal{N}} V_{i,j}(\gamma_i, \gamma_j) \tag{2.18}$$

where $\mathcal{N}$ consists of 8 spatially neighboring pixels.

For the smoothness term V, we use the standard contrast-dependent function defined in [116], which favors assigning the same label to neighboring pixels that have similar color.

Similar to [76], our data term $D_i(\gamma_i)$ defines the cost of labeling pixel i with label γ_i as:

$$D_i(\gamma_i) = -\log(1 - P_i^c(\gamma_i) \cdot P_i^l(\gamma_i)) \tag{2.19}$$

where $P_i^c(\gamma_i)$ is the probability of labeling pixel i with label γ_i based on the appearance (color) cues, $P_i^l(\gamma_i)$ is the probability based on location prior. Both are defined below.

To compute $P_i^c(\gamma_i)$, we first estimate two Gaussian Mixture Models (GMM) in RGB color space to model the foreground (fg) and background (bg) appearance. Since the color may vary significantly over the video frames, we need to learn the color models over all video frames, which is an easy task since we have the object regions inferred as the constrained MWSs. The foreground GMM model fg^{color} is learned from pixels in the regions selected in the constrained MWSs computation. The background GMM model bg^{color} is learned from pixels contained in the complement of selected regions in all the frames. Then given these two color distributions fg^{color} and bg^{color}, we define for each pixel p_i:

$$P_i^c(\gamma_i) = \begin{cases} P(p_i|fg^{color}), & \text{if } \gamma_i = 1 \\ P(p_i|bg^{color}), & \text{if } \gamma_i = 0 \end{cases} \tag{2.20}$$

For the computation of location probability $P_l^i(\gamma_i)$, we utilize the object regions selected in the constrained MWSs. Given the selected region (we have only one region per frame), we first compute its distance transform. Let $d(p_i)$ denotes the distance of pixel p_i to the

selected object region. We compute

$$P_i^l(\gamma_i) = \begin{cases} \exp(-\frac{d(p_i)}{\sigma}), & \text{if } \gamma_i = 1 \\ 1 - \exp(-\frac{d(p_i)}{\sigma}), & \text{if } \gamma_i = 0 \end{cases} \tag{2.21}$$

where σ indicates the confidence of the location prior, the larger is σ, the lower is the confidence. We compute $P_i^c(\gamma_i) \cdot P_i^l(\gamma_i)$ as the probability of foreground (γ_i = 1) and background (γ_i = 0). As illustrated in Fig 2.5(b), the color probability is not particularly informative in a global scale of the whole frame, and the main information comes from the probably map of the location shown in Fig. 2.5(c). However, the color information is informative if constrained by the location probability as illustrated by the joint probability shown in Fig 2.5(d).

After obtaining the data term D and smoothness term V, we use the popular method in [18] to find the optimal f that minimizes the energy function (2.18), and obtain the final foreground objects in each video frame.

Segmentation Results

We first examine our method on the SegTrack dataset [141]. There are six videos (*monkey-dog, bird, girl, birdfall, parachutte, penguin*). For each video, a pixel-level segmentation ground-truth is provided for the primary foreground object. This enables a statistical evaluation of our method. Object segmentation in these videos are extremely challenging due to several facts, such that the primary object are with large shape deformation and foreground and background color has overlap. Same as [76], we do not evaluate our method on *penguin* video since only a single penguin is labeled as the foreground object among a group of penguins.

Given a video, we first produce [39] 300 object candidate regions per frame. We use Lab space histograms to describe color for each region. Each Lab channel has 20 bins. For the color model of the foreground and background, we use RGB color space, and two

GMMs with 5 component are learned. Same as [76], we describe motion using optical flow histograms computed from [89] with 60 bins per x and y direction. The region's bounding box is dilated by 30 pixels when computing the background histograms. To initialize the maximal weight subgraph computation, each time we select one from the best 50 object regions candidates according to $A(u,u) = ob(u)$. In the graph cut energy function (2.18), $\delta = 1$ in all our experiments.

Due to the efficiency of the proposed constrained MWSs algorithm, on a PC with 3.4Ghz and 8GB RAM, it only takes 2 minutes to select regions by constrained MWSs with 50 different initializations. The binary graph cut on single frame takes about 0.1s in average.

We compare the results with three state-of-the art methods [76], [141] and [28]. The method in [76] and our method are unsupervised. They automatically discover the primary object in image as well as segment the object out. The methods in [141] and [28] require minor supervision with the object labeled in first frame. The results are shown in Table 2.4. Our method has the lowest average per frame segmentation error over the 5 test videos. It also achieves the lowest segmentation error on 3 out of 5 videos. Compared to [76], which also does not require manual object initialization, we achieve better results on 4 out of 5 videos. Some segmentation results are shown in Fig. 3.4.

The results in Table 2.4 report the average per-frame, pixel error rate computed in comparison to the ground-truth segmentation. Specially, it is computed as [141]:

$$error = \frac{\mathbf{XOR}(\gamma, GT)}{F} \tag{2.22}$$

where f is the label for every pixel in a given video, GT is the ground-truth label, and F is the total number of frames in a given video. Since all videos are roughly of the same size, the average error rate over the 5 videos is computed as average over all frames in all videos, i.e., we treat all 5 videos as a single video and apply (2.22).

Video (No. frames)	Ours	[76]	[141]	[28]
birdfall (30)	**189**	288	252	454
cheetah (29)	**806**	905	1142	1217
girl (21)	1698	1785	**1304**	1755
monkeydog (71)	**472**	521	563	683
parachute (51)	221	**201**	235	502
Average	**542**	592	594	791
Manual seg.:	No	No	Yes	Yes

Table 2.4: Segmentation error as measured by the average number of incorrect pixels per frame. Lower values are better.

	Ours	constrained MWS	Lower bound
birdfall	189	311	295
cheetah	806	1258	700
girl	1698	3063	2973
monkeydog	472	497	493
parachute	221	803	680

Table 2.5: Segmentation error comparison. We compare our entire proposed method (Ours) to the region segmentation results obtained by the region selection as constrained MWSs. The lower bound error is the lowest possible error of regions produced by [39].

As we mentioned above, even without the pixel-based object segmentation described in Secion 2.10, the object regions selected by constrained MWSs in Section 2.10 alone can be regarded as the segmentation result. In Table 2.5, we report the pixel error of the constrained MWSs regions segmentation results, although it is lower-bounded by the accuracy of the region candidates produced by [39]. The lower-bound error is computed as the error of the region candidate with the lowest error as compared to the ground-truth pixels. This reflect the lowest segmentation pixel error we could achieve by only selecting regions from computing the constrained MWSs.

We can see that, for videos *birdfall*, *monkeydog*, the results are very good merely using regions selected by constrained MWSs. Moreover, with the exception of *cheetah*, the pixel error is rather close to the lower bound. This demonstrates that the proposed region selection scheme as constrained MWSs is a powerful tool for video segmentation.

	constrained MWS	w/o constraints
birdfall	311	589
cheetah	1258	1772
girl	3063	3742
monkeydog	497	2024
parachute	803	883

Table 2.6: Segmentation error comparison of the constrained MWSs optimization with and without the mutex constraints.

The proposed algorithm for constrained MWSs computation converged after 207 iterations on average. Moreover, for all videos, the proposed algorithm converged to a discrete solution. This is extremely important, since it implies that the mutex constraints are satisfied.

As shown in Table 2.6, the segmentation error increases significantly if inter-frame proximity mutex constraints, which express spatial coherency, are not taken as input to the constrained MWS optimization. We also provide a visual illustration of the importance of this mutex constraints in Fig. 2.7. We compare the trajectories of the constrained MWSs region centroids computed with and without this mutex constraints. They are shown overlaid over the first video frame. We can see that with the constraints, the trajectory of the centroid is very smooth, and the selected regions are always focusing on the primary object, i.e., the monkey in the example video. This shows that the mutex constraints significantly increase the robustness of the constrained MWSs optimization. They allow us to eliminate unreasonable region selection hypotheses, which result from unreliable region affinity relations, and consequently, play a critical role in selecting correct object regions.

We also examine our method on two videos *Yu-Na Kim* and *Waterski* from [56]. While [56] focus on labeling every pixel in image using motion and appearance cues, we automatically identify the primary object, i.e., ice skater and water skier, and segment them out in every frame. Qualitative results are shown in Fig 2.3.

2.11 View-Invariant Object Detection by Matching 3D Contours

Introduction

Since the beginning of computer vision, the researchers have realized that 3D information makes object detection and recognition simpler and more robust than using 2D image information only. In particular, contours of 3D objects have been utilized in object recognition many decades ago, e.g., [6, 92], since they offer a view invariant representation of 3D objects. Moreover, in contrast to 3D surfaces, 3D contours offer a simpler 1D like representation of complex shapes in 3D like chairs or other man-made objects. However, extraction of 3D contours from single 2D images or stereo image pairs turned out to be a challenging problem. Only due to recent progress of RGB-D sensors, robust extraction of 3D contours became possible. However, we still face the problem of matching of 3D contours. The main challenges are intra class object variance, e.g., everyday objects like chairs come in different sizes and shapes, and occlusion.

Contour is an important cue for human to recognize objects, and has been widely used in 2D single-view object detection in [47, 128, 106]. While contour has certain advantages, such as its low computation cost and its invariance to color and texture changes, it varies significantly under different viewpoints. This challenges most of current state-of-the-art shape-based detection approaches on a multi-view object detection task. As early computer vision approaches, we address this challenge by directly working with contours of 3D objects instead of their 2D projections. In our approach, we still utilize the fact that contours of 3D objects project to 2D contours. It allows us for efficient recovery of 3D contours from 2D contours extracted from depth maps. This is possible thanks to Kinect, which is the most popular RGB-D camera. Since depth information can be obtained from a single view of a given scene, it is possible to recover 3D point cloud representing object

surfaces. Depth map certainly provides more information that a single RGB image, and has proved to boost the performance of object recognition methods [12].

Object detection in 3D point clouds is an active research topic in the robotic community, e.g., see [**?**] for an overview. There objects are recognized by directly matching 3D point clouds or by fitting surfaces to 3D point clouds. While surfaces are appropriate models for certain object classes, e.g., a ball, it is very hard if impossible to model object classes like chairs with surfaces alone. Contours appear to be a very suitable representation for RGB-D images. We observe that contours of 3D objects project to contours in 2D images. This in particular means that we can obtain 3D contours by lifting back contours from 2D images to 3D.

The processing flow of the proposed approach is illustrated in Fig. 2.8. After obtaining an RGB and depth images of a single view of a scene with Kinect, we first run Canny edge detector on the depth map. By linking the edge pixels, we obtain 2D edge fragments shown overlaid on the depth map in Fig. 2.8(b) with different colors. Since for each pixel in the depth map we can recover the 3D point that projects to it (with exception of out of range readings), we can ”back project” each edge fragment to a set of 3D points, which we call 3D contour fragment. In Fig. 2.8(c) we see the 3D points recovered form the depth map in (b); for clarity of visualization the floor points are not shown. In Fig. 2.8(d) we show the 3D contour fragments in different colors. Each 3D contour fragment is represented with a set of 3D line segments fitted to the 3D points ”back projected” from the corresponding 2D edge fragment. While one can recognize there the 3D contours of the two chairs and the stand, there are also many other contours present. They represent edges of walls and the background clutter.

After this preprocessing phase, we are ready for the proposed object detection. The 3D contours that belong to two detected chairs are shown in red and green in Fig. 2.8(e). All other 3D contours are shown in cyan. The detection is obtained by matching the model chair shown in Fig. 2.8(f) to all 3D contours in (e). In our system we used only one

extremely simplistic model chair, as shown in (f), in order to demonstrate the power of matching 3D contours. The main challenges addressed by the proposed approach are intra class variability of 3D contours and occlusion. Occlusion and self-occlusion results in missing parts of 3D contours, which makes their matching challenging. To address these challenges we utilize the fact that geometric relations between 3D contours have more expressive power, and consequently, are less ambiguous compared to 2D.

We propose to solve the object detection by 3D matching problem by finding maximal weight subgraphs (MWSs) that satisfy mutex constraints. An example result is shown in Fig. 2.9. There for each of the three detected chairs, we mark with the same color their 3D segments and the corresponding model segments. We observe that the three chairs vary in shape and size, and all are substantially different form our single model chair. Moreover, due to self-occlusion, and since some edge fragments are not detected in the 2D depth images, all three chairs have some missing parts. The proposed matching approach is able to robustly deal with these challenges. This is possible due to our inference framework for finding MWSs that allows us to enforce hard, mutual exclusion (mutex) constrains. The mutex constraint, which express qualitative spatial relations such as above/below as well as prohibit grouping 3D contours that are too far from each other, eliminate the majority of impossible matching configurations. This allows us to obtain correct detections with weak shape similarity relations, which in turn allow us to tolerate a significant shape and size variance of 3D contours representing objects in the same shape class. In particular, we use only one chair exemplar in our experiments on chair detection.

We compute the MWSs on the correspondence graph composed of all pairs (model segment, 3D scene segment). As shown in Fig. 2.8(f), our exemplar chair is composed of 11 line segments. If we have 200 segments in a given 3D scene, for example, then the correspondence graph has 2200 nodes. In order to detect MWSs in this graph, we initialize with one correspondence, and compute a MWS that contains this correspondence, i.e., we have 2200 initializations. Then we sort the MWSs according to their weights. The three

detected chairs in Fig. 2.9 represent MWSs with three highest weights. As can be seen the subgraphs have 8 to 10 nodes. Thus, our inference framework is capable of finding very small MWSs in graphs with a few thousand nodes.

In Sec. 2.11, we review related works. In Sec. 2.11, we introduce our shape representation and matching, also how to formulate the object localization problem as finding maximal weight subgraph with mutex constraints. In Sec. **??**, a formal definition of maximal weight subgraph with mutex constraints will be given and an algorithm we used to solve it is described. Experiment results are shown in Sec. 2.11.

Related Work

There are some recent works utilizing 3D contour information to perform object detections in range images. Stiene et al. [132] proposed a detection method in range images based on silhouettes. Drost et al. [36] use a local hough-like voting scheme that uses pairs of points as features to detect rigid 3D objects in 3D point clouds. Hinterstoisser et al. [59] proposed a multimodal template matching approach based on RGB-D data that is able to detect objects in highly cluttered scenes.

In a very early work, Ponce et al. [111] established a 3D object recognition framework, where objects are collections of small (planar) patches, their invariants, and a description of their 3D spatial relationship. Ferrari et al. [48] proposed a method to compute feature tracks densely connecting multiple model views of a single object. In [135], Implicit Shape Model [78] and [48] are combined, and activation links for transferring votes across views are used to address the object detection from arbitrary viewpoints. Savarese and Fei-Fei [121] propose a compact model of an object by linking together diagnostic parts of the objects from different viewpoints. Instead of recovering a full 3D geometry, parts are mutually connected by homographic transformation in this approach. More recently, a probabilistic approach to learning affine constraints between object parts is introduced in [133]. In [87], discriminative part-based 2D detectors and generative 3D representation of the object class

geometry which can be learned from a few synthetic 3D models are combined. Yan et al. [154] collect patches from viewpoint-annotated 2D training images and map them onto an existing 3D CAD model. In [3], a 3D implicit shape model is obtained via sparsely annotated 2D feature positions. Payet and Todorovic [109] proposed a shape-based 3D object recognition method, in which a few view-dependent shape templates are jointly used for detecting object occurrences and estimating their 3D poses.

A recent work by Janoch et al. [64] explores different options on how to utilize the depth information from RGB-D cameras to improve the detection accuracy of objects seen from different viewpoints. They call Deformable Part Model (DPM) [43] applied to depth images Depth HOG, and conclude that Depth HOG is never better than HOG on the original 2D image. The best performing system on their dataset is a linear combination of DPM running on the original image with the size distribution of a given object class, which is modeled with a single Gaussian. We call this system DPM-SIZE.

View-invariant object detection can also be addressed by directly using single 2D images, i.e., no 3D contour or surface reconstruction is attempted prior to the detection. Recent approaches of this type include [135, 87, 122]. While 2D single-view object detection methods can be used to addressed the task by combining the outputs of classifiers trained for different object views, such approaches are argued to be only effective when there are sufficient single-view detectors to cover all possible viewpoints [135]. However, this strategy requires a lot of training samples, and many independent detectors may lead to a substantial increase in the number of false-positives. In order to obtain a better multi-view object detector, many methods made an effort to learn a generative model by combining 2D appearance and geometric viewpoint information [133, 88, 87]. While promising results are obtained by such methods, they suffer from ambiguous 2D local features and lack of direct modeling of 3D viewpoint geometry.

In general graph matching frameworks [11], while local features' similarity (unary potential) and geometric relations between them (binary potential) are usually considered,

very coarse qualitative geometric constraints such as above/below, or left/right do not draw much attention. We demonstrate in our work that using mutex constraints to enforce these qualitative geometric constraints makes our method more robust to the noise, and therefore, able to generate higher quality solutions.

Object Detection by Matching 3D Contours

In order to obtain contours of 3D objects form a given RGB-D image, we first find edge fragments in the depth map. They are obtained by linking edge pixels obtained by the Canny edge detector to 2D curves. Then we lift each 2D edge fragment back to a 3D curve. Let C be a single edge fragment. We first dilate it with a dilation radius of 2 pixels. Then we find the set of 3D points Z that project to pixels in dilated C. Finally we iteratively fit 3D line segments to points in Z. We run RANSAC to fit a line and identify the inlier points and outlier points. Then we repeat this process for the outlier points until the number of outlier points is lower than a threshold. Hence we represent each 3D curve Z as a set of 3D line segments, and consequently, we represent 3D contours obtained from a given RGB-D image as set of line segments in 3D. An example is shown in Fig. 2.8(d).

Object detection in the proposed approach is formulated as finding configurations of line segments recovered from a given RGB-D image that are similar to the line segment configuration of the exemplar modeling a given shape class. Thus, we need to identify a subset of 3D line segments that best matches the exemplar. This computation is formulated here as finding maximum weight subgraphs (MWS) in a weighted correspondence graph. We begin with definitions of pairwise similarities of line segments.

Similarity of 3D Vectors

We use a set of straight line segments $\mathcal{S} = \{B_1E_1, \cdots, B_nE_n\}$ to approximate object contours in 3D, where B_i is the beginning point and E_i is the endpoint of segment B_iE_i. An example is shown in Fig 2.8 (b). Since the line segments are oriented, they are vectors

in 3D, and from now on we treat them as vectors. For the model contour each line segment is represented with just one vector. In contrast, each contour line segment in 3D image is represented by two vectors that differ by their orientation.

Although we know the exact size of objects in 3D, the size of objects in the shape shape class may still vary significantly. To obtain a size-invariant vector representation, we characterize each B_iE_i by its angle with a reference vector r defined as

$$\angle(B_iE_i, r) = \arccos(\frac{B_iE_i \cdot r}{||B_iE_i||\ ||r||}) \in [0, \pi] \tag{2.23}$$

We take vector $r = [0, 0, 1]$ representing the z-axis as the reference vector. Since 3D objects are supported by the floor, which is represented as xy-plane, the representation in (2.23) is invariant to the rotation around the z-axis. This means it is invariant to object location on the floor, under the assumption that the object is standing on the floor. To simplify the notation, we omit the direction r below when possible, and use $\angle B_iE_i$ to represent the angle of vector B_iE_i with z-axis.

Given the above angle-based segment representation, we treat two vectors as similar if they have similar angles with the z-axis. We compute this similarity value as

$$\psi(B_iE_i, B_jE_j) = \exp(-\frac{(\angle B_iE_i - \angle B_jE_j)^2}{\sigma^2}) \tag{2.24}$$

where σ represents the tolerance of angle differences (it is set to $\frac{\pi}{3}$ in all our experiments).

Similarity of Vector Configurations

Let $\mathcal{E} = \{B_1^eE_1^e, \cdots, B_m^eE_m^e\}$ be 3D vectors that represent an exemplar (model) of a given shape class, and let $\mathcal{S} = \{B_1^sE_1^s, \cdots, B_n^sE_n^s\}$ be 3D vectors representing the vectors of the recovered 3D scene.

We construct a weighted association graph $G = (V, A)$ with $V = \mathcal{E} \times \mathcal{S}$. Hence each node represents a correspondence $u = (B_i^e E_i^e, B_j^s E_j^s)$ between a model vector i and an image vector j. Consequently, there are $N = m \times n$ nodes in the graph.

We define now the entries of the adjacency matrix A. If $u = v = (B_i^e E_i^e, B_j^s E_j^s)$, then $A(u, u) = \psi(B_i^e E_i^e, B_j^s E_j^s)$, which simply the similarity of the angle with z-axis of both vectors. Given a pair of different correspondences $u \neq v$, where $u = (B_i^e E_i^e, B_j^s E_j^s)$ and $v = (B_k^e E_k^e, B_l^s E_l^s)$, the weight $A(u, v)$ between nodes u and v represents the consistency of the their assignments. We measure it by computing the similarity of the spatial configuration of exemplar vectors $B_i^e E_i^e, B_k^e E_k^e$ to the configuration of the 3D scene vectors $B_j^s E_j^s, B_l^s E_l^s$. For this we consider new vectors that join their start points. For example, in Fig. 2.10 vectors $B_i^e E_i^e, B_k^e E_k^e$ are the cyan lines in the model, and the new vector $B_i^e B_k^e$ is marked with the black dashed line while the new vector $E_i^e E_k^e$ is marked with the red dashed line. The same colors are used for the corresponding vectors in the 3D scene. The similarity of this configuration is determined by the similarity of the angles between the corresponding dashed vectors:

$$A(u, v) = \psi(B_i^e B_k^e, B_j^s B_l^s) \cdot \psi(E_i^e E_k^e, E_j^s E_l^s). \tag{2.25}$$

Mutex Constraints between Contour Vectors

Compared to other graph matching frameworks, the key and unique property of our formulation is usage of qualitative spatial constraints, such as above/below or left/right or front/back. For example, if for a given pair $u = (B_i^e E_i^e, B_j^s E_j^s)$ and $v = (B_k^e E_k^e, B_l^s E_l^s)$, the model vector $B_k^e E_k^e$ is above vector $B_i^e E_i^e$, then we require the same for the corresponding vectors in the 3D scene, i.e., $B_k^s E_k^s$ should be above $B_i^s E_i^s$. By enforcing the qualitative geometric relations in the correspondence computation, we can significantly improve the

solution quality. In particular, the matching becomes robust to significant variance in shape and size of objects form a given class.

We define a symmetric mutex relation $M \subseteq V \times V$ between vertices of the graph defined in Section 2.11. It is represented with a binary matrix $M \in \{0,1\}^{N \times N}$. If $M(u,v) = 1$ then the two vertices u, v cannot belong to the same maximum clique. In other words, mutex represents incompatible vertices that cannot be selected together. Since a vertex cannot exclude itself, we set $M(u,u) = 0$ for all vertices $u \in V$.

Given a pair of two vertices representing the correspondences $u = (B_i^e E_i^e, B_j^s E_j^s)$ and $v = (B_k^e E_k^e, B_l^s E_l^s)$, where $u \neq v$, $M(u,v)$ represents the compatibility of the the spatial relations between vectors $B_i^e E_i^e$ and $B_k^e E_k^e$ in the model, and $B_j^s E_j^s$ and $B_l^s E_l^s$ in the 3D scene. For example, if $B_i^e E_i^e$ is above $B_k^e E_k^e$ in the model and $B_j^s E_j^s$ is below $B_l^s E_l^s$ in the scene, then $M(u,v) = 1$. One the other hand, if $B_j^s E_j^s$ is also above $B_l^s E_l^s$, then $M(u,v) = 0$. Similarly, $M(u,v) = 1$ if front/back or left/right spatial relations are violated.

In order to define M without checking different cases, we project the 4 points $B_i^e, E_i^e, B_k^e, E_k^e$ to vectors $B_i^e E_i^e$ and $B_k^e E_k^e$ in the model and the 4 points $B_j^s, E_j^s, B_l^s, E_l^s$ to vectors $B_j^s E_j^s$ and $B_l^s E_l^s$ in the scene. Then we check whether the two 1D orders on the projection lines are compatible. If yes, we set $M(u,v) = 0$, and if not, we set $M(u,v) = 1$. We skip the technical details, since they only require elementary 3D geometry and the limited space.

Experiments

Chair is an icon object class that has gained much attention form the beginning of AI. Although humans have no problem in identifying chairs, until today no artificial system is able to cope with chair detection. Chair detection is a challenging problem for most computer vision, detection algorithms [55], considering that the chair shape in 2D images varies significantly due to different viewpoints and due to resulting perspective distortion. Moreover, chairs come in different shapes and sizes. Therefore, we focus our performance

evaluation on chair detection. We selected a stand as the second object class, since it is visually very similar to the chair in that it usually has 4 legs supporting a flat rectangular surface on top. The main difference is that the stand does not have any back support and its legs are longer, e.g., see the left image in Fig. 2.11.

We collected a dataset containing 109 RGB-D images captured with the Kinect sensor. It contains a total of 213 chairs shown from many different view points and 40 stands. Our dataset also contains other objects that may be confused with chairs and stands like tables and trash cans as can be seen in Fig. 2.11. Moreover, may objects are occluded and are shown in many different views.

In order to demonstrate that our dataset is very challenging and in order to compare to state-of-the-art object detectors, we compare the performance of our approach to DPM by Felzenszwalb et al. [43] and to DPM-SIZE recently proposed in Janoch et al. in [64]. DPM-SIZE augments DPM with depth information. It utilizes the expected object sizes in 3D scenes to boost DPM performance. We also compare to the popular contour based detection method PAS by Ferrari et al. [47]. For a quantitative evaluation, we use recall-precision curves and average precision (AP) computed as described in [40].

The detection results of chairs are summarized in Fig. 3.4. The proposed approach achieves a significantly better AP value compared to DPM and to DPM-SIZE. Our AP is nearly 30% higher than the second best performing method DPM-SIZE [64]. Moreover, the fact that DPM-SIZE, DPM, and PAS have all very low recall clearly demonstrates that these methods cannot cope with significant view changes and perspective distortions. This comes at no surprise for DPM and PAS, since both methods are based on 2D image analysis. In contrast, the direct matching of 3D contours in 3D allows us to overcome the challenges of view changes and of perspective distortion. We stress that our approach does not require any training, as opposed to the other three approaches, and we only have one extremely simplistic chair model. Moreover, our chair model is not extracted from the test dataset.

The significance of the qualitative mutex constraints is demonstrated by the fact that the performance of our method drops by 10% when these constraints are not used. This in turn illustrates the importance of the utilized inference framework.

In Fig. 2.13, we show some detection results. As seen in Fig. 2.13(b), DPM [43], DPM-SIZE [64], PAS [47] missed many chairs. Adding 3D information about expected object sizes in the 3D scenes (DPM-SIZE [64]) is able to improve the performance of DPM, but still some chairs are missed. The main reason is that the initial detection is still performed in the 2D images (using sliding window processing of DPM).

We use the already trained version of DPM, which is publicly available on the authors' webpage. DPM [43] attempts to solve the object detection problem by using a multiple components object model, and each component is aimed to capture the object appearance under certain view-point. The 2D chair appearance model of DPM is trained using images from [40] with thousands of chairs. We also tried to train DPM detector on half of our dataset and test on the other half as opposed to using the trained detector from images in [40]. This process yields a much worse AP of 0.01. However, the DPM detector is able to get 0.96 AP on training images. This again demonstrates how challenging is significant view point variance, and perspective distortion to state-of-the-art 2D object detectors. The expected size of the chair for DPM-SIZE was learned as described in [64]. We trained it on a random half of our dataset and test on the other half. This process was repeated 10 times. We also used the software of the authors of PAS [47] to perform experiments on our chair dataset. A shape is learned automatically using this software, following the same procedure as for size training of DPM-SIZE.

Since there does not exist any trained version of DPM for the class stand and our dataset exhibits too large view variance for training DPM, we only report the result of our detector with mutex constraints on the class stand in Fig. 2.14.

Discussion and Future Work

We only used one simplistic chair model, which differs in both size and shape from the various chairs captured in our dataset. This allows us to demonstrate the robustness of the proposed 3D matching framework. Our matching framework is also robust to occlusion, and of course, it is not influenced by view point changes. Similarly we only used one simplistic stand model.

However, more 3D contour models are needed to capture the intra class variability. In particular, some chairs may only have one leg like the office chair shown in the right image in Fig. 2.11. Such models can be easily learned by clustering training objects using the proposed similarity measure.

One of the biggest challenges of our 3D contour-based object detection are objects without clear 3D contours like humans or sofas. For such objects it is still possible to extract occluding contours from the RGB-D data, and those contours exhibit significantly lower variation than contours extracted form 2D RGB images. Also the contour detection problem in RGB-D images is significantly simpler. However, the 3D occluding contours exhibit larger variation than intrinsic 3D contours of objects like chair or stand. Our future work will focus on matching the occluding 3D contours.

2.12 Random Matrix Tests

We observe that in all the experiments reported above the proposed algorithm converged to a discrete solution. The goal of this test is to examine under extreme conditions how often the proposed algorithm converges to a discrete solution after a reasonable upper bound on the number of iterations. We consider a task of matching two sets of 40 points. We construct a correspondence graph with 1600 nodes representing all pairs of these points. Then we construct a 1600×1600 affinity matrix A of random entries drawn from a uniform distribution. The mutex matrix M represents the one-to-one constraints. The maximum

possible number of iterations is set to 500 for both IPFP and our algorithm. We repeat this experiment 10000 times with different random matrices A.

which is 0.22%, while 9940 solutions were non discrete for IPFP, which is 99.4%.

2.13 Conclusions

As we observed many problems can be solved by finding maximum weight subgraphs that satisfy global mutex constraints expressed in quadratic equality form. This formulation enjoys great modeling flexibility in many applications, because mutex constraints significantly improve the quality of solutions when unary and binary potentials are unreliable, which is rather a rule than exception in real applications. However, many state-of-the-art general solvers cannot handle well global mutex constraints, since they lead to a large number of non-submodular terms with large values of the energy function. Because global mutex constraints are essential for adequately modeling many real problems, the non-submodular terms cannot be ignored. Therefore, we propose a novel algorithm for computing maximum weight subgraphs that satisfy global mutex constraints. As demonstrated by the experimental results it significantly outperforms the state-of-the-art general solvers IPFP, LBP, QPBO, QPBOP, QPBOI, and QPBOP+I as well as application specific algorithms. In addition, we demonstrate the effectiveness of MWSs framework for solving a video object segmentation problem, in which a state-of-the-art segmentation accuracy is achieved.

Figure 2.2: Some image reconstruction results for puzzles with 48 patches: first row: LBP, second row: QPBOP + I, third row: IPFP. Fourth row: our algorithm. The fifth row shows the original images. The anchor patches are marked in red.

Figure 2.3: Our object segmentation results on two videos *Yu-Na Kim* and *Waterski* from [56].

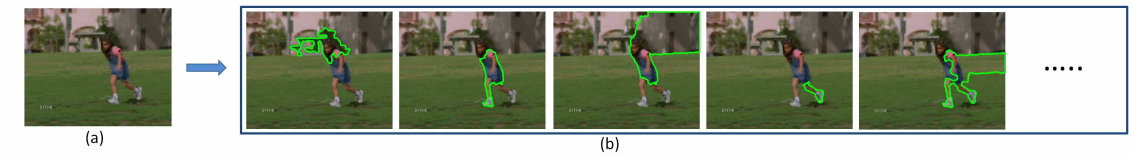

Figure 2.4: Object proposals produced by [39]. (a) A video frame (b) Proposals ranked in order of ”objectness”.

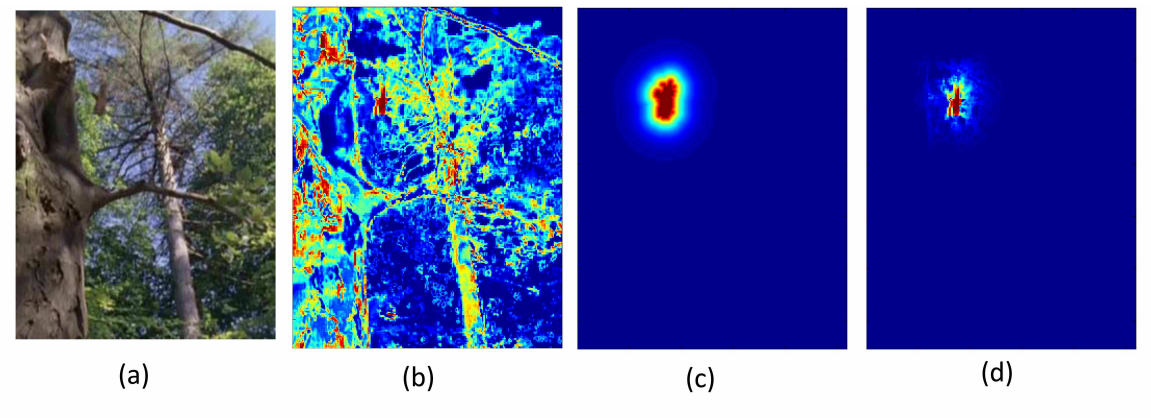

Figure 2.5: (a) A single frame and the probabilities of the foreground object $\gamma_i = 1$. (b) Color prob. $P_i^c(\gamma_i)$. (c) Location prob. $P_i^l(\gamma_i)$. (d) The joint foreground prob. $P_i^c(\gamma_i) \cdot P_i^l(\gamma_i)$

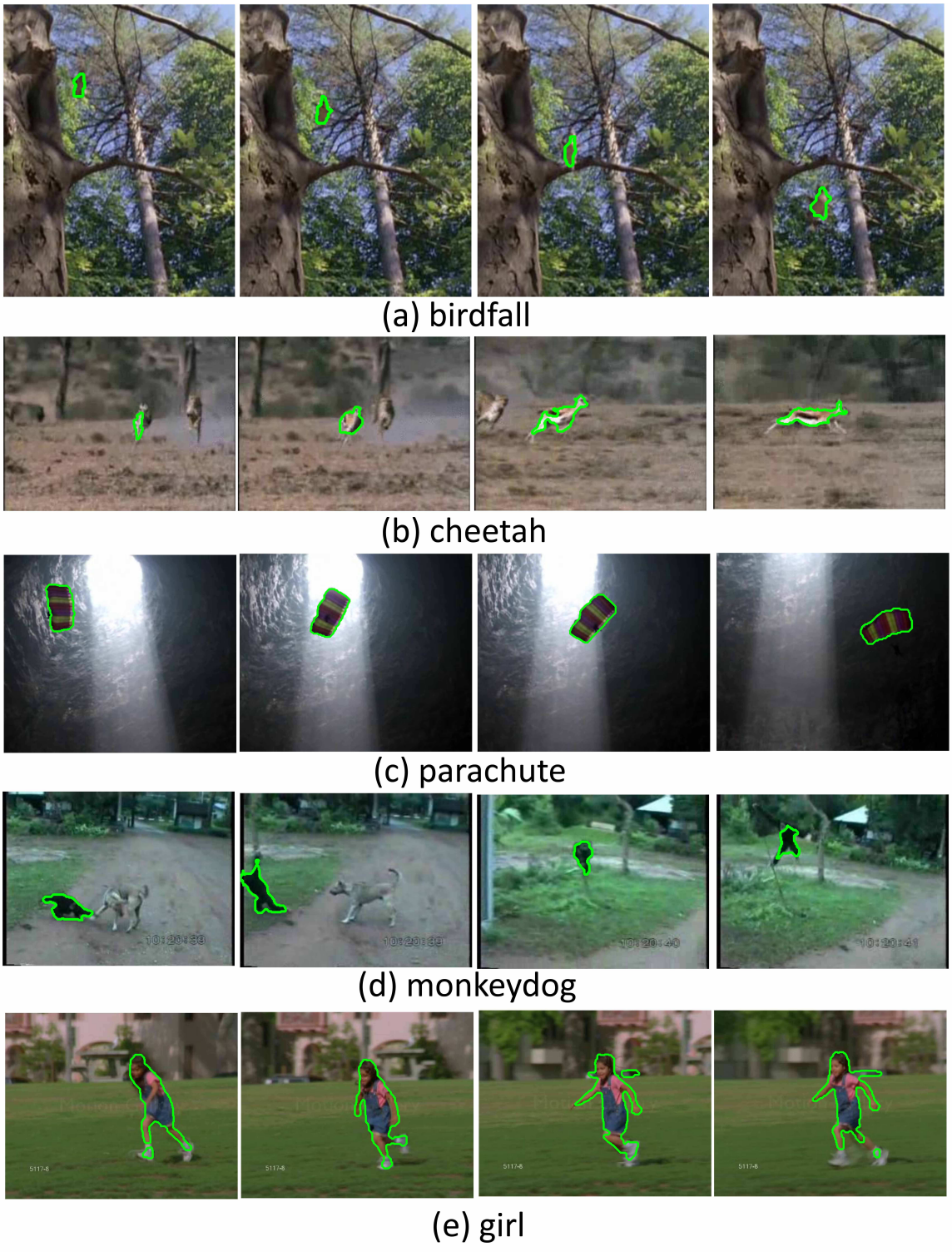

Figure 2.6: Segmentation results. Best viewed in color.

Figure 2.7: The trajectories of centroids of selected regions, green dots connected with red lines, overlaid over the first frame. (a) when inter-frame proximity mutex constraints are used and (b) when inter-frame proximity mutex constraints are *not* used.

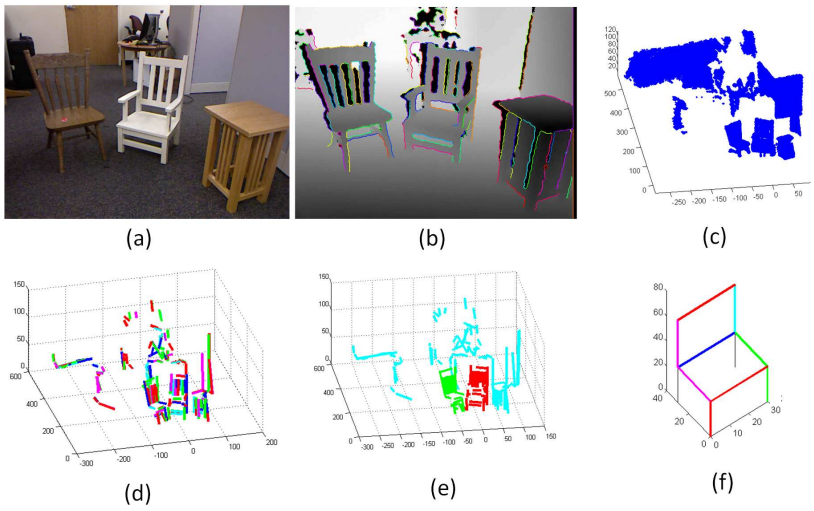

Figure 2.8: An RGB image in (a) and the corresponding depth map in (b). The 3D points recovered from (a) are shown in (c). We recover 3D contour fragments, shown in different colors in (d) from edge fragments in (b). The line segments of two detected chairs in (d) are shown in green and red in (e). They are detected by matching segments of a single model shown in (f) to the segments in (d).

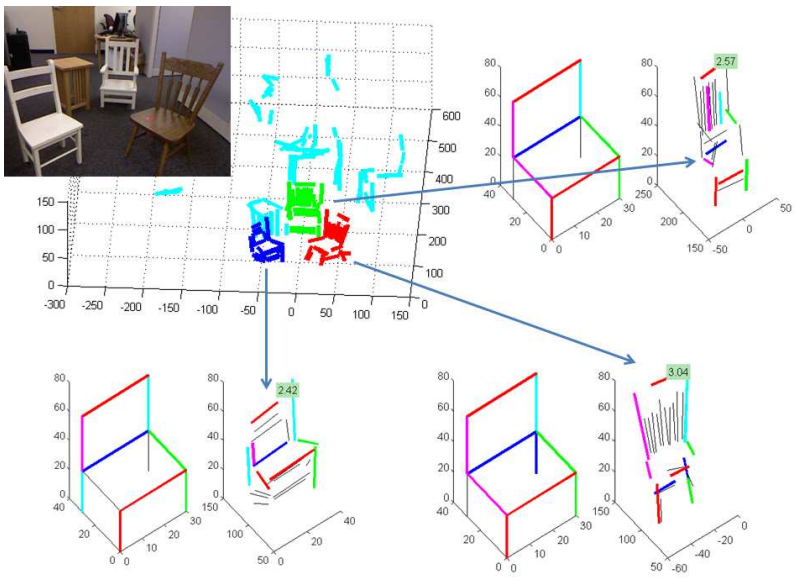

Figure 2.9: A recovered 3D scene from a single RGB-D image. Contours of 3D objects are represented with 3D line segments. Object detection is performed by finding MWSs in the correspondence graph composed of pairs (model segment, 3D scene segment). We mark with the same colors the corresponding segments for three detected chairs shown in red, green, and blue in the 3D scene.

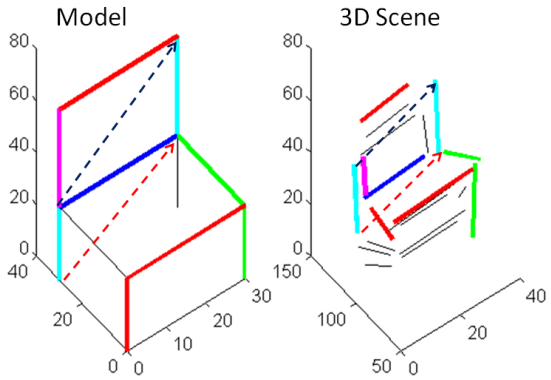

Figure 2.10: Similarity of the two configurations of cyan lines is defined as similarity of the angles between two black dashed vectors and between two red dashed vectors.

Figure 2.11: Example images in our chair-stand dataset.

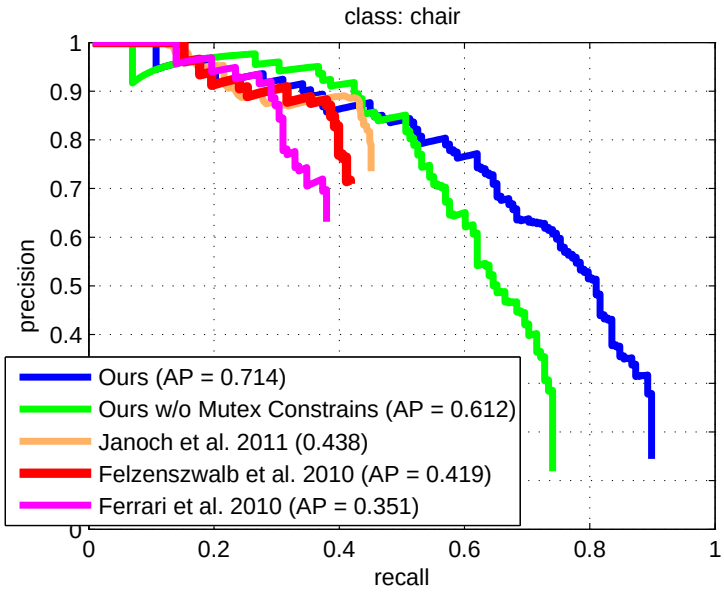

Figure 2.12: Recall-Precision and AP comparison for the class chair.

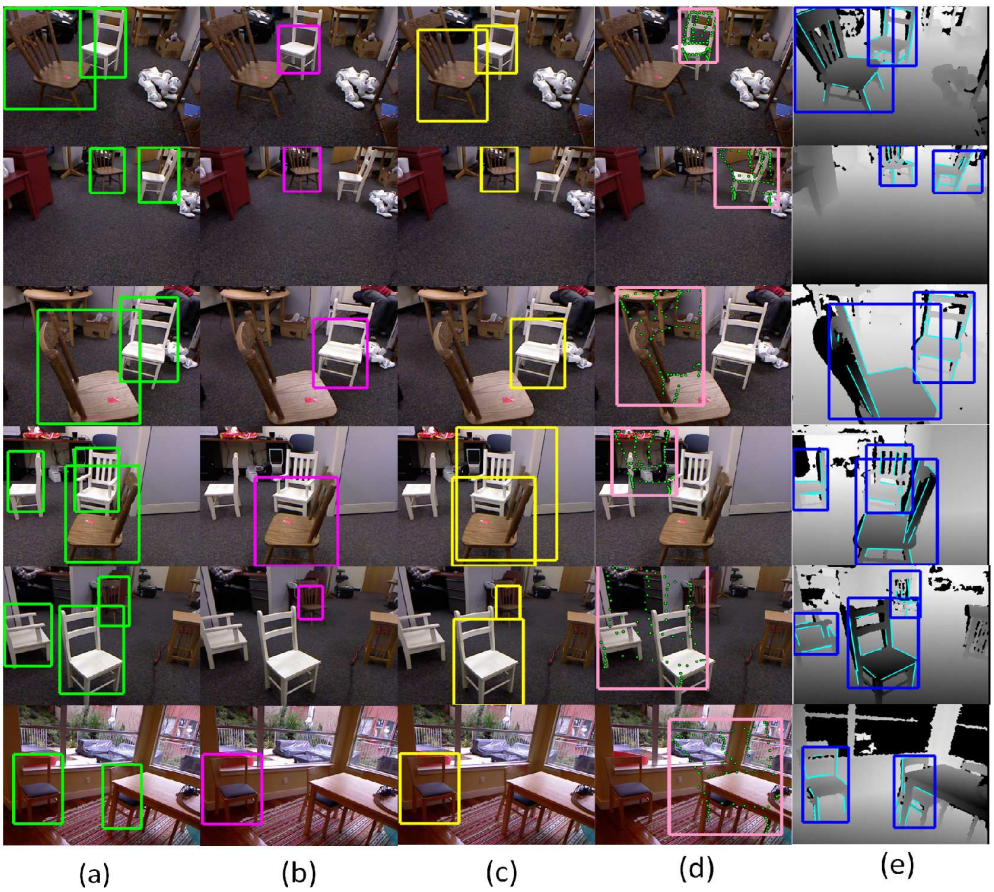

Figure 2.13: Some chair detection results. (a) ground truth, (b) DPM [43], (c) DPM-SIZE [64]. (d) PAS [47] with transformed model shown with dots, and (e) The proposed method with results shown on depth map to stress that they are obtained in 3D.

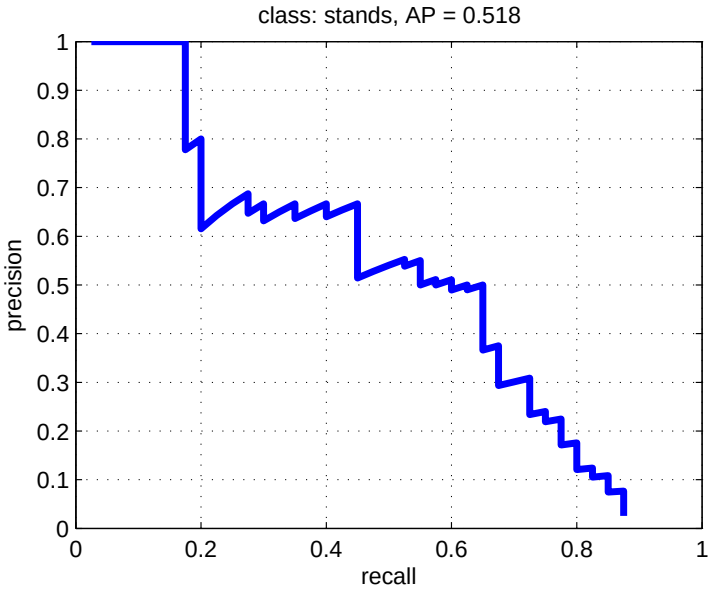

Figure 2.14: Recall-Precision and AP of our detector with mutex constraints on class stand.

Chapter 3

Graph Transduction Learning with Connectivity Constraints with Application to Multiple Foreground Cosegmentation

3.1 Introduction

Given multiple images sharing overlapping contents, the goal of image cosegmentation is to simultaneously divide these images into non-overlapping regions of foreground and background. In an unsupervised setting, foreground is defined as the common regions that repeatedly occur across the input images [119]. In an interactive or supervised setting [7], some foreground objects are explicitly assigned by an user as the regions of interest.

Kim and Xing [69] has recently proposed a multiple foreground cosegmentation (MFC) task, in which K different foreground objects need to be jointly segmented from a group of M input images. This scenario is very realistic, since not all objects need to appear in each image, i.e., each of images contains a different and *unknown* subset of the K objects. Three example images from the same group are shown in the first column of Fig. 3.1. This task contrasts the classical cosegmentation problem dealt with by most existing algorithms [60, 7, 119, 67, 70, 146, 148], where a much simpler and less realistic setting is usually assumed by requiring that the same set of objects occurs in every image. While this assumption provides a relatively strong prior which has been utilized by most of cosegmentation algorithms, it severely limits the application scope of these cosegmentation algorithms, since it is not valid for most real photo collections.

The fact that the MFC problem does not assume that each objects appears in every image, brings serious challenges to the cosegementation algorithms, which is addressed [69]. There are two iterative steps, foreground modeling and region assignment. The region assignment subproblem is solved by assuming foreground model is given. The authors of [69] consider two settings: supervised and unsupervised. In the supervised setting, it is straight forward that foreground model can be built through objects labeled by users in the training images. In the unsupervised setting, foreground model can be initialized by running unsupervised cosegmentation method [70, 66]. As clearly demonstrated in [69], the segmentation results in the supervised setting are significantly better. Their supervised setting is still very realistic from the point of view of real applications, since only a very

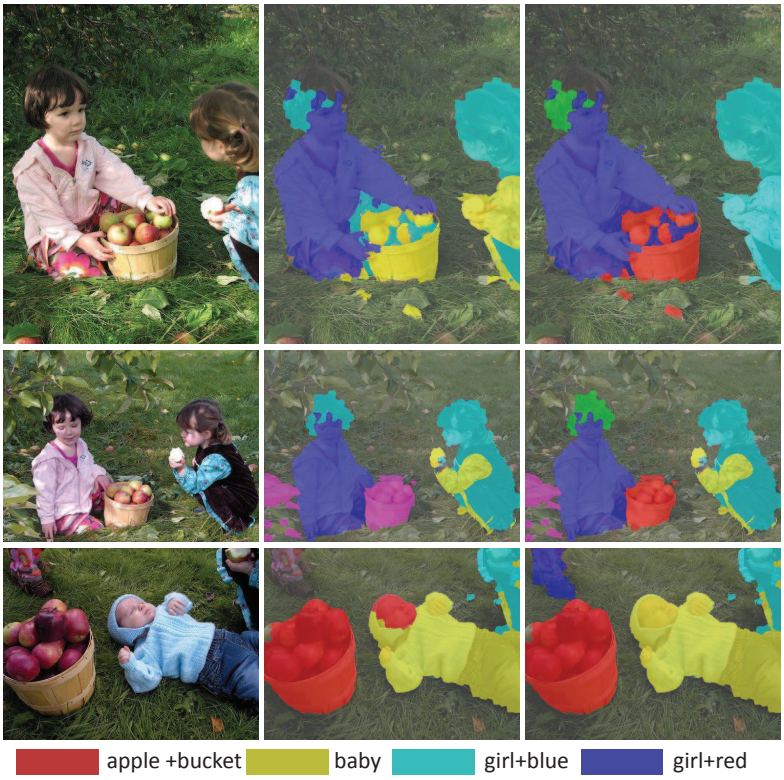

Figure 3.1: Multiple Foreground Cosegmentation results on three images of the scene *Apple+picking*. First Columns: original images. Second Columns: the results of an excellent graph transduction SSL method RLGC [151]. Third Column: results of the proposed GTC. Compared to RLGC, GTC improves the consistency of label assignment by enforcing connectivity of regions with the same label.

small number of objects of interest must be marked by the user. Only 20% of images is used from groups of images containing 10 to 20 images. For example, this means that the user only needs to mark the objects in 2 out of 10 images. Since this supervised setting contains a very small number of training data, which is very challenging for supervised learning methods.

Our contribution is based on the observation that this is an ideal setting for semi-supervised learning (SSL). In particular, we formulate this problem as graph transduction SSL, which has demonstrated impressive results on many tasks, especially when there exists only a small amount of labeled data samples. Compared to supervised methods, its

main advantage relies on using both labeled and unlabeled data during the training process, which yields considerable improvement in labeling accuracy, e.g., [158, 160, 151].

However, the label propagation accuracy in graph transduction SSL highly depends on how reliable the similarity of graph nodes is. Since in the MFC application, the nodes represent image regions (segments or superpixels), their similarity is neither very discriminative nor particularly stable. In particular, due to large appearance variations of the same objects in different images, segments belonging to different objects may accidently have higher similarity than segments belonging to the same object.

To address this problem, we propose to constrain graph transduction SSL framework by integrating global connectivity constraints. In other words, we enforce that segments assigned the same label form connected regions in each image. Connectivity is naturally motivated by the human visual perception, and connectedness is a very intuitive and effective criterium for object segmentation, as has been demonstrated in [144, 100] in the context of supervised image segmentation.

As in [69], for a given set of images containing common objects, we first perform over-segmentation to obtain several segments for each image separately. While [69] uses a spatial pyramid as the objects model, we only utilize colorSIFT and use bag-of-word (BoW) model to represent segments. Although using BoW enjoys some robustness to the object variations, such as changes in shape and orientation, it also makes the similarity between segments not very discriminative, which in turn significantly degrades the labeling results of SSL methods. To demonstrate this, we examine segmentation results by labeling in Fig. 3.1. The second column shows the results of an SSL excellent method introduced in [151]. We call it regularized local and global consistency (RLGC). We can see that many disconnected regions are wrongly assigned the same labels because of their similar color and texture, for example, the face of baby and apple basket. This happens because in standard graph transduction SSL framework, each segment is taken out-of-context and labeled independently. While this is suitable for general SSL inference problem, it is clearly

suboptimal in our application. In particular, while the segment graph encodes the visual similarity between pairs of segments, the spatial information between segments in the same image is totally neglected. This information is expressed as connectivity in the proposed framework.

In our graph-based formulation, if nodes representing segments from the same image share the same class label, they must form a connected subgraph [68]. This is a global property and it introduces high-order constraints. As shown in [100], although it is an exponential problem (with respect to the number of nodes) to examine if two nodes are connected, finding the most violated connectivity constraint can be done efficiently in polynomial time. Moreover, each such constraint can be represented as a linear inequality.

To solve a SSL problem formulated with connectivity constraints in graph transduction formulation, we design a cutting-plane algorithm, in which we iterate between solving a convex problem of label propagation with linear inequality constraints, and finding the most violated constraint. We investigate two versions of our method.

The output of most graph transduction SSL methods, e.g. [158, 151], represents the confidence of assigning data points to all labels. The discretization step is then performed on each unlabeled data point independently, by simply assigning the label with the largest confidence. The first version of our method enforce the connectivity constraints at the final discretization step of label confidences obtained through SSL learning. This can be considered as a postprocessing method, and could be applied to any SSL method. It can be solved as linear programming with linear inequality constraints.

More importantly, in the second version, we integrate the graph transduction formulation with connectivity constraints, and solve it as a convex quadratic programming with linear inequality constraints. We call this method graph transduction with connectivity constraints (GTC). Its segmentation examples are shown in the third column of Fig. 3.1. As can be seen it significantly improves on label assignment of RLGC (second column). In particular, the baby face belongs to the baby not to the basket anymore. It even can correct

wrong labels as can be seen in the first row, where the basket is wrongly labeled as baby by RLGC, which is corrected by GTC. We have a similar case for the basket in the second row. This examples as well as our experimental results in Section 3.6 clearly demonstrate that the connectivity information can be used to increase the robustness of SSL methods.

We evaluate the proposed approach on real world MFC application on FlickrMFC dataset. It significantly outperforms the MFC method in [69] and other state-of-the-art cosegmentation methods.

The remainder of this paper is organized as follow: The related work is introduced in Section 3.2. In Section 3.3, we revisit the standard graph transduction SSL framework. In Sections 3.4 and 3.5, we introduce the proposed integration of connectivity constraints into the graph transduction framework, and derive a method to solve it efficiently.

3.2 Related Work

Many approaches have been proposed to solve the image cosegmentation problem [60, 7, 119, 67, 70, 146, 148]. All these approaches only consider two class (foreground/background) cosegmentation problem. The initial model presented in [119] provides a framework to enforce consistency among two foreground histograms in addition to the Marov Random Field (MRF) segmentation terms for each image. In [67], a discriminative clustering formulation is adopted, in which the goal is to assign foreground/background labels jointly to all images so that a supervised classifier trained with these labels leads to maximal separation of the two classes. Recently, a Random Walker based method is proposed in [30], and is shown to be an effective framework for cosegmentation problem complementary to MRF formulation. While our method shares similar properties as [67] and [30], in the sense that we also have a graph formulation and utilize the normalized graph Laplacian, we have a very different goal for constructing the graph, consequently, the definitions of nodes and edges in the graph are also very

different. In particular, for both [67] and [30], image pixels are taken as nodes, and edges only exist locally between pairs of nearby pixels. This follows the standard framework of spectral clustering for image segmentation [124]. In our method, the graph is constructed using segments as nodes, and the edges exist between every pair of segments, because the graph is used for the purpose of propagating the labels from labeled segments to unlabeled segments following the graph transduction SSL framework.

Semi-supervised learning is the intermediate range of the spectrum between supervised methods and unsupervised methods. It has been widely used to solve many kinds of machine learning and computer vision problems. In [157], Zeisl et al. combined SSL with multiple instance learning to solve the object tracking problem. Fergus et al. [44] introduced a linear SSL method to label tiny images among a gigantic image collections. In [57], SSL method is used to associate keywords (side information) of labeled and unlabeled images, so that a stronger classifier can be obtained for the image classification task. A SSL based hashing method is proposed in [152] for image retrieval. Recently, SSL is used in [129] for solving scene categorization task, where constraints based on mutual exclusion and comparative attributes are imposed. In [150], SSL has been applied to improve the affinity metric for single image segmentation. Our approach is very different from these SSL applications to computer vision problems. To our best knowledge, this is for the first time that connectivity constraints are considered in the SSL framework.

3.3 Semi-supervised Learning (SSL)

In this section, we will first introduce how do we construct the segment graph in Sec 3.3.1 And in Sec 3.3.2, we will review how to use the graph transduction method to solve a standard semi-supervised learning problem. Finally, in Sec 3.5, we focus on how to impose the connectivity constraints under semi-supervised learning framework and how to solve it efficiently.

3.3.1 Segment Graph Construction

Given a set of images which contain multiple common objects, we first divide each image I_m into segments (or superpixels) $S_m = \{s_m^1, \ldots, s_m^K\}$. Set V be the set of the segments in all images. Any segmentation algorithm can be used here. We used submodular image segmentation method introduced in [70]. We assume that segments in a small number of images are labeled with object categories. We are given a small set of labeled segments, and a large majority of unlabeled segments. Our goal is to infer a label for each unlabeled segment.

We define a weighted graph $G = (V, \mathbf{W})$, where is a nonnegative matrix representing the pairwise similarity of image segments, which is defined as follows. For each segment s_i, we compute its ColorSIFT descriptor [142] and quantize them according to a codebook. Then a bag-of-words histogram $\mathbf{x}_i$ is used to represent segment s_i. For two nodes i and j representing two different segments s_i and s_j, the weight w_{ij} is computed using a RBF kernel:

$$w_{ij} = \exp - \frac{d(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2} \tag{3.1}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ computes the $\mathcal{X}^2$ distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and σ is the kernel bandwidth parameter. We follow [23] to compute σ. In particular, $\sigma = \bar{dist}_k/3$, where $\bar{dist}_k$ is the average distance between each sample and its kth nearest neighbor. Since sparsity is important to remove noise and it has been proved that semi-supervised learning algorithms are more robust when run on a sparse graphs [65], we set $w_{ij} = 0$, if $i \notin k\mathrm{NN(j)}$, where kNN denotes the set of k nearest neighbors (k is the same as used in computing σ).

3.3.2 Graph Transduction for SSL

We assign the class labels to unlabeled image segments in a standard graph-based semi-supervised learning framework, which we review here. Let the node degree matrix $\mathbf{D} = diag([d_1, \cdots, d_N])$ be defined as $d_i = \sum_{j=1}^{N} w_{ij}$, where $N = |V|$. The binary label matrix

$\mathbf{Y} \in \{0,1\}^{N\times C}$ is defined as $y_{il} = 1$ if node s_i has label $l \in L$ and $y_{il} = 0$ otherwise, where C is the number of labels in L. We also assume that $\sum_l y_{il} \leq 1$ for every node i meaning that each node can have at most one class label. The normalized graph Laplacian is defined as $\mathbf{L} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}$.

Graph-based semi-supervised learning methods propagate label information from labeled nodes to unlabeled nodes [160]. Most methods define a continuous variable $\mathbf{F} \in \mathbb{R}^{N\times C}$ that is estimated on the graph to minimize a cost function. The cost function typically used has two tradeoff terms. One term is used to measure the smoothness of the function on the graph of both labeled and unlabeled data, with the second term used to measure the fitness between $\mathbf{F}$ and the label information for the labeled nodes. In particular, we follow the formulation introduced in [151]. We call the method regularized local and global consistency (RLGC), since it modifies the cost function from the classic local and global consistency (LGC) method [158] by adding a node regularizer $\mathbf{R}$:

$$\mathcal{Q}(\mathbf{F}) = \text{tr}\{\mathbf{F}^T\mathbf{L}\mathbf{F} + \mu(\mathbf{F} - \mathbf{R}\mathbf{Y}^T)(\mathbf{F} - \mathbf{R}\mathbf{Y}^T)\}, \tag{3.2}$$

where μ is a constant. The matrix $\mathbf{R}$ is used to balance the influence of labels from different classes. It works as node regularizer that normalizes labels within each class based on node degrees. This is very important for the problems with highly unbalanced labeled nodes, which is the case for our application. $\mathbf{R} = diag(\mathbf{r})$ in which $\mathbf{r} = [r_1, \ldots, r_N]$ is computed as

$$r_i = \begin{cases} \frac{1}{C} \cdot \frac{d_i}{\sum_k y_{kl} d_k} & \text{if } \exists_{l\in L}\ y_{il} = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

Due to the convexity of the cost function in (3.2), we obtain a closed form solution by zeroing the partial derivative $\frac{\partial \mathcal{Q}}{\partial \mathbf{F}} = 0$. With simple algebra, we can derive

$$\mathbf{F}^* = (\frac{\mathbf{L}}{\mu} + \mathbf{I})^{-1}\mathbf{R}\mathbf{Y} = \mathbf{P}\mathbf{R}\mathbf{Y} \tag{3.4}$$

where $\mathbf{P} = (\frac{\mathbf{L}}{\mu} + \mathbf{I})^{-1}$ as the propagation matrix [158].

After obtaining the continuous solution $\mathbf{F}^* \in \mathbb{R}^{N \times C}$, we need to binarize it into $\mathbf{Y}^* \in \{0,1\}^{N \times C}$. As is usually the case in graph transduction SSL, this is a simple argmax step: for every node i determine $l^* = \arg\max_l \mathbf{F}^*_{il}$, and then set $\mathbf{Y}^*_{il} = 1$ if $l = l^*$ and $\mathbf{Y}^*_{il} = 0$ if $l \neq l^*$.

3.4 Constrained SSL

According to the cost function defined in (3.2), to solve the SSL problem, we need to solve a QP problem defined on continuous variable $\mathbf{F} \in \mathbb{R}^{N \times C}$. In this section we extend this problem by adding linear constraints to enforce connectivity.

Let $\mathcal{C}$ denotes a set of matrices $\mathbf{M} \in \{-1, 0, 1\}^{C \times N}$ representing linear constraints. We consider the following constrained formulation of Eq. (3.2):

$$\begin{aligned} \mathcal{Q}(\mathbf{F}) \quad &= \quad \text{tr}\{\mathbf{F}^T\mathbf{L}\mathbf{F} + \mu(\mathbf{F} - \mathbf{R}\mathbf{Y}^T)(\mathbf{F} - \mathbf{R}\mathbf{Y}^T)\} \\ s.t. \quad &\text{tr}(\mathbf{M}\mathbf{F}) \leq 1, \ \ \forall\, \mathbf{M} \in \mathcal{C}. \end{aligned} \tag{3.5}$$

With an empty constraints set $\mathcal{C}$, minimizing (3.5) is equivalent to minimizing (3.2). Hence it is a convex QP problem and it has a closed form solution $\mathbf{F}$ as shown (3.4). With a non-empty set of linear constraints, convexity still holds. Although the closed form solution cannot be derived, problem (3.5) can be solved efficiently by many existing solvers. In this work, we use IBM CPLEX (v12.4) to get the optimal solution.

3.5 Enforcing Connectivity Constraints in SSL

Before we give the formal definition of the connectivity constraints, we first introduce a binary adjacency graph $G = (V, \mathbf{A})$ to represent the spatial adjacency of segments, i.e., $\mathbf{A}(i,j) = 1$ if two segments s_i, s_j belong to the same image and are adjacent and $\mathbf{A}(i,j) =$

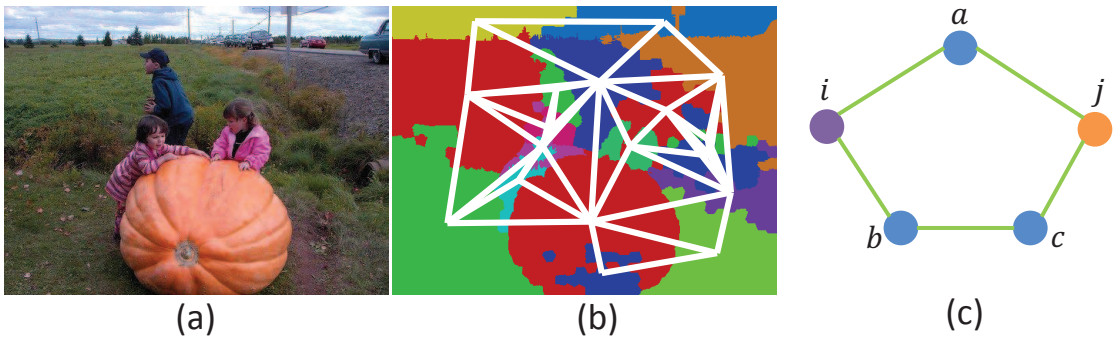

Figure 3.2: (a) Original image (b) Segments and adjacent graph (c) A simple adjacency graph. For a pair of nodes (i, j), there are three vertex-separator sets {a, b}, {a, c} and {a, b, c}. Only {a, b} and {a, c} are essential vertex-separator sets.

0 otherwise. Let $conn(G)$ denotes the set of all connected subgraphs of G. Of course, the nodes of each connected subgraph must represent segments belonging to the same image.

Each subgraph of G can be expressed with an indicator vector $\mathbf{u} \in \{0,1\}^N$. Hence we can identify $conn(G)$ with the set of indicator vectors $\mathbf{u} \in \{0,1\}^N$ representing connected subgraphs of G, i.e., $conn(G) \subset \mathcal{P}(\{0,1\}^N)$. By taking the convex hull of $conn(G)$ we obtain a polytope $Z = conv(conn(G)) \subset [0,1]^N$, where $[0,1]^N$ is the N-dimensional hypercube. We call Z a *connected subgraph polytope* of G.

The most well-known problem defined on Z is finding maximum-weight connected subgraph. As proved in [68], even with a linear target function in this problem, it is NP-hard to optimize. Therefore, to make an optimization problem defined on Z to be polynomially solvable, we have to relax Z. To do this, we follow the method introduced in [100]. It is proved that each facet of Z can be defined by a linear inequality equation. For a better characterization of the facet, we need to define *vertex-separator sets* [100], as follows:

Given an undirected graph $G = (V, \mathbf{A})$, for any pair of vertices $i, j \in V, i \neq j, A(i,j) = 0$, the set $S \subseteq V \setminus \{i,j\}$ is said to be a *vertex-separator set* with respect to {i, j} if the removal of S from G disconnects i and j, which means that there exists no path between i and j in the subgraph with the vertex set $V \setminus S$.

In addition, we define $\bar{S}$ as an *essential vertex-separator set* if it is a vertex-separator set with respect to $\{i, j\}$ and any strict subset $T \subset \bar{S}$ is not. We denote with $\mathcal{S}(i, j)$ the set of all essential vertex-separator sets with respect to $\{i, j\}$. An example of essential vertex-separator sets is shown in Fig 3.2(c).

The proposed SSL algorithm with connectivity constraints is an iterative cutting-plane method. It alternates between solving a convex quadratic programming (QP) with linear inequality constraints (3.5) according to graph $(G, \mathbf{W})$, and adding a new connectivity constraint (facet) according to graph $(G, \mathbf{A})$.

Let $\mathbf{F}^t$ be a solution of (3.5) obtained at iteration t. We need to examine whether $\mathbf{F}^t$ violates the connectivity constraints. In order to do this, we need to define the connectivity constraints as linear constraints. Since our goal is to enforce connectivity of image segments belonging to the same object, i.e., having the same label, for a pair of segments s_i and s_j we only check the connectivity constraints if they are in the same image and have the same label l. We denote with $\mathcal{H}$ a set of all triples (i, j, l) such that s_i and s_j are in the same image, are not adjacent, i.e., $A(i, j) = 0$, and the probability for both segments have label $l \in L$ is positive, i.e., $\mathbf{F}^t_{il}, \mathbf{F}^t_{jl} > 0$. We call $\mathcal{H}$ a *check condition set*, since only for triples in $\mathcal{H}$ the connectivity condition needs to be checked.

As proved in [100], each facet of the polytope containing Z is defined by the following linear inequality for a label $l \in L$ and for all pairs (i, j) such that $(i, j, l) \in \mathcal{H}$:

$$\mathbf{F}^t_{il} + \mathbf{F}^t_{jl} - \sum_{k \in S} \mathbf{F}^t_{kl} - 1 \leq 0, \forall S \in \mathcal{S}(i, j) \tag{3.6}$$

For a triple $(i, j, l) \in \mathcal{H}$, proving that no violated inequality exists or finding the most violated inequality in (3.6), which is given by

$$S^*(i, j, l) = \arg \max_{S \in \mathcal{S}_{(i,j)}} \sum_{k \in S} \mathbf{F}^t_{kl}, \tag{3.7}$$

can be solved efficiently by computing max-flow [1] on an auxiliary directed graph. More details on how to construct the auxiliary directed graph can be found in [100].

Then find $(i^*, j^*, l^*) \in \mathcal{H}$ with the largest violation as

$$(i^*, j^*, l^*) = \arg \max_{(i,j,l) \in H} \sum_{k \in S^*(i,j,l)} \mathbf{F}^t_{kl} \tag{3.8}$$

Let $S^*(i^*, j^*, l^*)$ be the vertex-separator set that yields the maximum value in (3.8). If

$$\mathbf{F}^t_{il} + \mathbf{F}^t_{jl} - \sum_{k \in S^*(i^*,j^*,l^*)} \mathbf{F}^t_{kl} - 1 \leq 0, \tag{3.9}$$

the iterative process stops, since no constraints are violated. Otherwise, there is constraint violated, and it can be represented by the l^*th column in $\mathbf{M}$, with $M_{i^*l^*}, M_{j^*l^*} = 1$, and $M_{kl^*} = -1$ if $k \in S^*(i^*, j^*, l^*)$, and $M_{kl^*} = 0$ otherwise. Then $\mathbf{M}$ is added to the constraint set $\mathcal{C}$, and in next iteration, we solve Eq. (3.5) with the updated $\mathcal{C}$. This iterative process stops when no constraints are violated, or the change between $\mathbf{F}^t$ and $\mathbf{F}^{t+1}$ is smaller than a threshold.

Finally, the output $\mathbf{F}^*$ is binarized to the label indictor $\mathbf{Y}^*$ the same way as at the end of Section 3.3.2: for every node i determine $l^* = \arg\max_l \mathbf{F}^*_{il}$, and then set $\mathbf{Y}^*_{il} = 1$ if $l = l^*$ and $\mathbf{Y}^*_{il} = 0$ if $l \neq l^*$.

We call the proposed method **graph transduction with connectivity constraints (GTC)**, since it integrates RLGC graph transduction formulation and global connectivity constraints. The entire algorithm is described in Alg. 1.

In Fig. 3.3, we visualize some examples of the most violated connectivity constraints discovered by our algorithm. For each left image, we use two green dots to show the pair of segments with the same label that are not connected. Essential vertex-separator set, which corresponds to the violated constraints, is shown using blue dots. We do not show the actual segments for better visualization. The edges are shown as black lines. In the right image,

[1] `http://pub.ist.ac.at/~vnk/software/`[17]

Algorithm 1 Graph Transduction with Connectivity Constraints (GTC)

Input: $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}, \mathbf{A}, \mu, \sigma$
Output: $\mathbf{F}^* = \mathbf{F}^t$

Initial $\mathcal{C}^1$ as an empty set, t = 1
repeat
  obtain $\mathbf{F}^t$ by solving Eq (3.5).
  find the most violated constraints $S^*(i^*, j^*, l^*)$ using Eq (3.8)
  if Eq (3.9) holds for $S^*(i^*, j^*, l^*)$ **then**
    break
  end if
  derive linear equality constraint $\mathbf{M}$ from $S^*(i^*, j^*, l^*)$
  $\mathcal{C}^{t+1} \leftarrow \mathcal{C}^t \cup \mathbf{M}$
until $|\mathbf{F}^t - \mathbf{F}^{t-1}| < \sigma$

we show the result of resolved constraints after the next iteration. In particular, it should be noticed that there are two ways to resolve the constraints. One is to change the label for either of the two green dots so that two segments are no longer with the same label. The other one is to change the labels of some of the separating segments marked in blue dots to the label of the segments with green dots, which makes the two green dots segments connected. As the examples illustrate, our algorithms automatically determines which of the two kinds of solutions is better.

For any semi-supervised learning method that yields a continuous label confidence matrix $\mathbf{F}^*$, it is only possible to impose the connectivity constraints at the final binarization step of $\mathbf{F}^*$. For this we formulate the binarization step as solving a linear MRF problem with the connectivity constraints:

$$\begin{aligned} \mathbf{Y}^* \quad &= \quad \arg\max_{\mathbf{Y}\in[0,1]^{N\times C}} \sum_{i=1}^{N}\sum_{l=1}^{C} \mathbf{Y}_{il}\mathbf{F}^*_{il} \\ &s.t. \quad \mathrm{tr}(\mathbf{M}\mathbf{Y}) \leq 1, \ \forall\, \mathbf{M} \in \mathcal{C}, \quad \sum_{l=1}^{C} \mathbf{Y}_{il} = 1. \end{aligned} \tag{3.10}$$

This constrained problem can be solved using our GTC framework presented above (by only replacing the target function in (3.5) with the linear target function in (3.10)). This

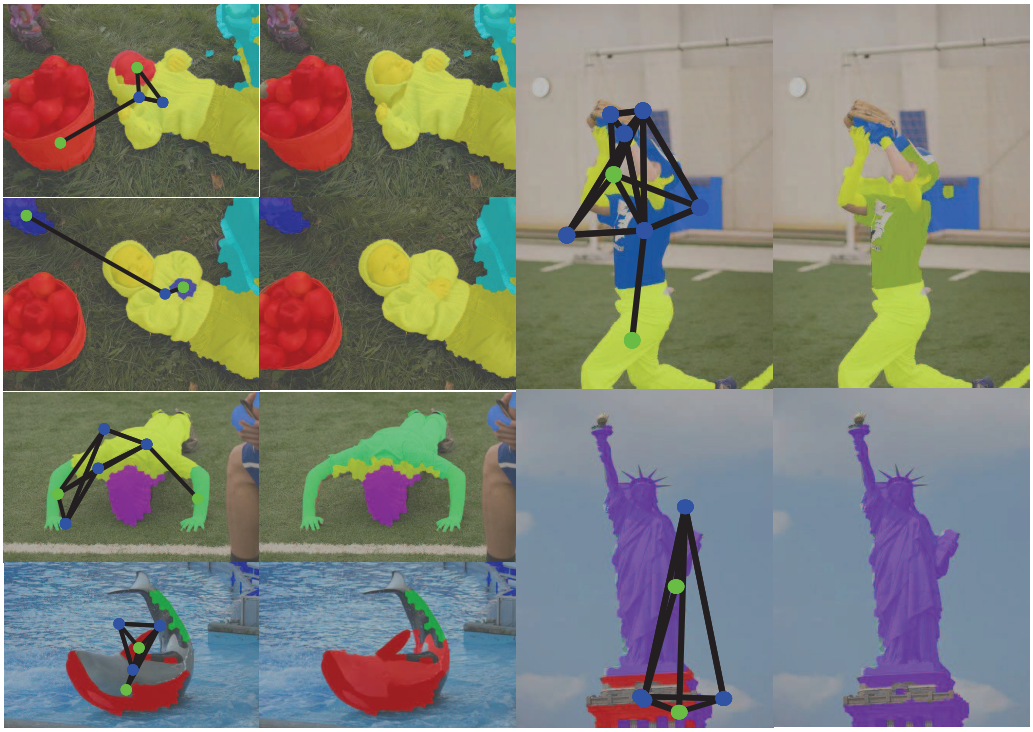

Figure 3.3: Visualization of the most violated connectivity constraints. Green dots: pair of segments with the same label that are not connected. Blue dots: essential vertex-separator set. Adjacency connection between segments is displayed using black lines.

can be considered as a postprocessing step, and it can be applied to any semi-supervised learning method. We name this method as **GTCP**, where *P* stands for postprocessing.

If the constraint set $\mathcal{C}$ is empty, the solution of (3.10) is simply the argmax rule, as described at the end of Section 3.3.2, which is a standard binarization procedure for graph transduction SSL algorithms. Hence the proposed GTCP can be viewed as binarization of SSL solutions with connectivity constraints.

To summarize, RLGC solves the problem under a standard SSL framework, where only affinity graph $(G, \mathbf{W})$ is utilized, and the connectivity between nodes is not considered. In GTCP, the constraints are considered, but only at the final binarization step of label confidences. For GTC, we integrate connectivity with RLGC in an iterative framework. By utilizing the additional information from adjacent graph $(G, \mathbf{A})$, GTC can improve the

label propagation process by increasing its robustness to the unstable affinity measurement in $(G, \mathbf{W})$. This is demonstrated by the experimental results in the next section.

Time Complexity: For the proposed GTC algorithm, in each iteration, solving convex QP with inequality constraints is very efficient. The main computation comes from finding the most violated connectivity constraints. However, this is carried out for each image and for each label independently. Therefore, if there are M images each decomposed into at most K segments, we only need to solve max-flow problem for at most MCK^2 times, where we recall that C is the number of object classes. In our method, K is usually a very small number (we follow [69], and obtain K = 18 segments using [70]). Also, this computation can be easily parallelized, which would further reduce the computation time.

3.6 Experimental Evaluation

We evaluate the proposed approach on a realistic and very challenging dataset called Flickr-rMFC dataset [69][2]. It consists 14 groups of images. Each group has 10 to 20 images, which are sampled from a Flickr photo stream. A finite number of repeating objects is contained in the same group, but the objects are not present in every image.

We follow the protocol of the interactive multiple foreground cosegmentation in [69], in which for each image group, 20% of images are randomly selected as training images, and the objects label in those images are provided. The labels represent a manual input of an user who marks the regions with main objects. The rest of images is used for testing. For each image set, 10 random splits is used, and the segmentation accuracy is averaged.

To evaluate the segmentation accuracy, the standard metric of PASCAL challenges is adopted, in which the intersection-over-union metric is measured. In particular, we follow the evaluation metric used in [69], where the segmentation accuracy is computed as $(\frac{GT_i \cap R_i}{GT_i \cup R_i})$.

[2] `http://www.cs.cmu.edu/~gunhee/r\_mfc.html`

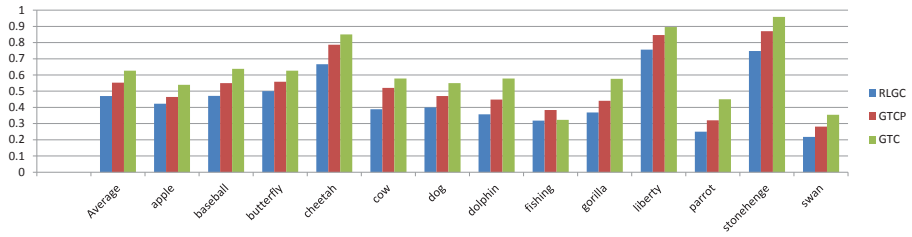

Figure 3.4: Comparison of the segmentation accuracy of RLGC, GTCP and GTC on 14 image groups in FlickrMFC dataset.

We compare our methods GTCP and GTC to the state-of-the-arts methods that have been evaluated on this dataset. The results are reported in Table 3.1 as the average accuracy over all 14 image sets. MFC-S [69] and our method can be viewed as typical SSL methods, since both require a small number of labeled data (labeled foreground objects in training images). The algorithm CoSand (COS) [70] and the discriminative clustering method (DC) [67], are not designed to handle irregularly appearing multiple foreground objects. Hence they require that all images are first manually divided into several subgroups so that the images of each subgroup share the same foreground object. Hence they also require user input, although no label information need to be explicitly provided as in a semi-supervised scenario. Only LDA-based unsupervised localization method (LDA) [120] is truly unsupervised. The results of LDA, DC, COS, MFC-S are copied from [69].

As can be seen in Table 3.1, the performance of RLGC [151], which belongs to classic graph transduction SSL methods, is comparable to MFC-S. This demonstrates the effectiveness of solving MFC problem in SSL framework, and in particular, the benefits of utilizing unlabeled data in addition to labeled data for label inference. Our postprocessing method GTCP applied directly to the label confidence scores of RLGC is able to significantly increase the segmentation accuracy, which demonstrates the benefits of the global connectivity constraints. Finally, our main proposed method GTC significantly outperforms all other methods. In particular, it increased the segmentation accuracy of MFC-S by 14%. Moreover, the fact that GTC outperforms our postprocessing method GTCP by over 7% shows the importance of enforcing the global connectivity constraints directly in

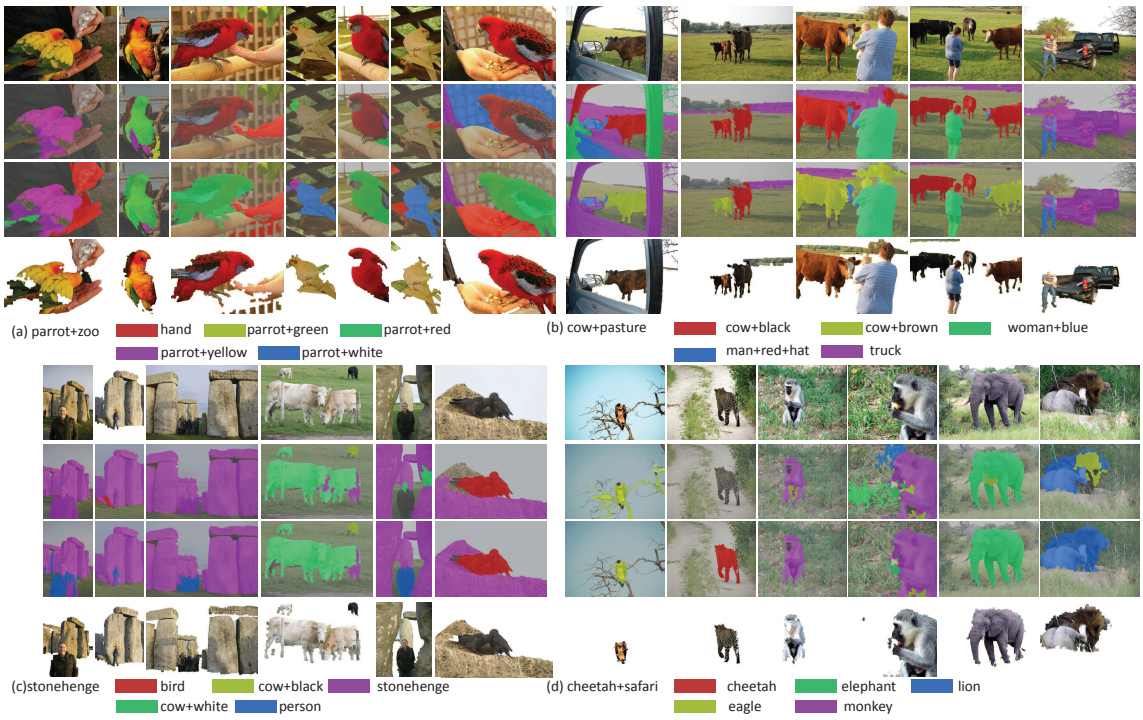

Figure 3.5: Examples of segmentation results on FlickrMFC dataset. First row: original images. Second row: segmentation results by RLGC. Third row: segmentation results by the proposed GTC. Fourth row: figure-ground segmentation results by GTC.

the graph transduction SSL framework. Some example segmentation results of GTC are shown in Fig. 3.5.

LDA [120]	DC [67]	COS [70]	MFC-S [69]	RLGC [151]	GTCP our	GTC our
25.2	31.3	32.1	48.2	47.6	55.0	**62.6**

Table 3.1: Average segmentation accuracy (PASCAL intersection-over-union metric) on FlickerMFC dataset from [69].

We also give a detailed comparison of the segmentation accuracy of RLGC, GTCP and GTC on the 14 image groups in FlickerMFC dataset in Fig. 3.4. GTC outperforms RLGC and GTCP on all 14 groups of images except *fishing*.

3.7 Conclusion

In this work, we integrate the global connectivity constraints with graph transduction learning framework to address a very challenging task: multiple foreground cosegmentation. Connectivity constraints are naturally motivated by human visual perception in that we prefer to identify objects as connected image regions. They play a similar role in our approach by enforcing consistent class label assignment to connected image regions, which significantly improves the segmentation results. State-of-the art results are achieved on the benchmark dataset FlickrMFC, which clearly demonstrates the effectiveness of the proposed approach.

Bibliography

[1] A. P. Ambler, H. G. Barrow, C. M. Brown, R. H. Burstall, and R. J. Popplestone. A versatile computer-controlled assembly system. In *IJCAI*, 1973.

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[3] Mica Arie-Nachimson and Ronen Basri. Constructing implicit 3d shape models for pose estimation. In *ICCV*, pages 1341–1348, 2009.

[4] X. Bai, X. Wang, L. J. Latecki, W. Liu, and Z. Tu. Active skeleton for non-rigid object detection. *ICCV*, 2009.

[5] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, IJCAI'77, 1977.

[6] H.G. Barrow and J.M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116, 1981.

[7] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *IJCV*, 2011.

[8] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24(1):705–522, 2002.

[9] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2001.

[10] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. *CVPR*, 2005.

[11] Alexander C. Berg, Tamara L. Berg, and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005.

[12] Liefeng Bo, Kevin Lai, Xiaofeng Ren, and Dieter Fox. Object recognition with hierarchical kernel descriptors. In *CVPR*, pages 1729–1736, 2011.

[13] I. M. Bomze, M. Pelillo, and V. Stix. Approximating the maximum weight clique using replicator dynamics. In *IEEE Trans. Neural Net*. 2000.

[14] Gunilla Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):849–865, November 1988.

[15] Endre Boros and Peter L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123:155–225, 2002.

[16] Endre Boros, Peter L. Hammer, and Gabriel Tavares. Preprocessing of unconstrained quadratic binary optimization. Technical report, 2006.

[17] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004.

[18] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001.

[19] William Brendel and Sinisa Todorovic. Video object segmentation by tracking regions. In *ICCV*, 2009.

[20] William Brendel and Sinisa Todorovic. Segmentation as maximum-weight independent set. In *NIPS*, 2010.

[21] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010.

[22] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 6:679–698, 1986.

[23] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. 2006.

[24] Xi Chen, Arpit Jain, Abhinav Gupta, and Larry S. Davis. Piecing together the segmentation jigsaw using context. In *CVPR*, 2011.

[25] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. In *ECCV*, 2010.

[26] Minsu Cho and Kyoung Mu Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *CVPR*, 2012.

[27] Taeg Sang Cho, Shai Avidan, and William T. Freeman. A probabilistic image jigsaw puzzle solver. In *CVPR*, 2010.

[28] Prakash Chockalingam, S. Nalin Pradeep, and Stan Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, 2009.

[29] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4), 2010.

[30] Maxwell D. Collins, Jia Xu, Leo Grady, and Vikas Singh. Random walks based multi-image segmentation: Quasiconvexity results and gpu-based solutions. In *CVPR*, 2012.

[31] Timothée Cour and Jianbo Shi. Solving markov random fields with spectral relaxation. *J. of ML Research*, 2, 2007.

[32] Timothée Cour, Praveen Srinivasan, and Jianbo Shi. Balanced graph matching. In *NIPS*, 2006.

[33] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893, 2005.

[34] E. D. Demaine and M. L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 2007.

[35] M. Donoser, H. Riemenschneider, and H. Bischof. Efficient partial shape matchingof outer contours. *ACCV*, 2009.

[36] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:998–1005, 2010.

[37] Olivier Duchenne, Francis Bach, In-So Kweon, and Jean Ponce. A tensor-based algorithm for high-order graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12), 2011.

[38] Olivier Duchenne, Armand Joulin, and Jean Ponce. A graph-matching kernel for object categorization. In *ICCV*, 2011.

[39] Ian Endres and Derek Hoiem. Category independent object proposals. In *ECCV*, 2010.

[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[41] Sylvain Faisan, Nicolas Passat, Vincent Noblet, Renée Chabrier, and Christophe Meyer. Topology preserving warping of binary images: Application to atlas-based skull segmentation. In *MICCAI (1)*, pages 211–218, 2008.

[42] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *CVPR*, 2008.

[43] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010.

[44] Rob Fergus, Yair Weiss, and Antonio Torralba. Semi-supervised learning in gigantic image collections. In *NIPS*, 2009.

[45] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, 2009.

[46] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection with contour segment networks. *ECCV*, 2006.

[47] Vittorio Ferrari, Frédéric Jurie, and Cordelia Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284–303, may 2010.

[48] Vittorio Ferrari, Tinne Tuytelaars, and Luc J. Van Gool. Integrating multiple model views for object recognition. In *CVPR*, pages 105–112, 2004.

[49] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956.

[50] H. Freeman and L. Garder. Apictorial jigsaw puzzles: the computer solution of a problem in pattern recognition. *IEEE TEC*, 13:118–127, 1964.

[51] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

[52] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*.

[53] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. J. Comput. Vision*, 73(1):41–59, June 2007.

[54] Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 377–388, 1996.

[55] Helmut Grabner, Juergen Gall, and Luc J. Van Gool. What makes a chair a chair? In *CVPR*, pages 1529–1536, 2011.

[56] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

[57] Matthieu Guillaumin, Jakob J. Verbeek, and Cordelia Schmid. Multimodal semi-supervised learning for image classification. In *CVPR*, 2010.

[58] Geremy Heitz, Gal Elidan, Benjamin Packer, and Daphne Koller. Shape-based object localization for descriptive classification. *Int. J. Comput. Vision*, 84(1):40–62, August 2009.

[59] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. *Computer Vision, IEEE International Conference on*, 0:858–865, 2011.

[60] Dorit S. Hochbaum and Vikas Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.

[61] Yuchi Huang, Qingshan Liu, and Dimitris N. Metaxas. Video object segmentation by hypergraph cut. In *CVPR*, 2009.

[62] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9):850–863, September 1993.

[63] Hiroshi Ishikawa. Transformation of general binary mrf minimization to the first-order case. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 33(6):861–874, 2011.

[64] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T. Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *ICCV Workshops*, pages 1168–1174, 2011.

[65] Tony Jebara, Jun Wang, and Shih-Fu Chang. Graph construction and *b*-matching for semi-supervised learning. In *ICML*, 2009.

[66] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012.

[67] Armand Joulin, Francis R. Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[68] Richard M. Karp. Maximum-weight connected subgraph problem, 2002.

[69] Gunhee Kim and Eric P. Xing. On multiple foreground cosegmentation. In *CVPR*, 2012.

[70] Gunhee Kim, Eric P. Xing, Fei-Fei Li, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, 2011.

[71] Vladimir Kolmogorov and Carsten Rother. Minimizing non-submodular functions with graph cuts - a review. Technical report, TPAMI, 2007.

[72] W. Kong and B. B. Kimia. On solving 2d and 3d puzzles using curve matching. In *CVPR*, 2001.

[73] P. D. Kovesi. Matlab and octave functions for computer vision and image processing. 2008.

[74] Jungmin Lee, Minsu Cho, and Kyoung Mu Lee. Hyper-graph matching via reweighted random walks. In *CVPR*, 2011.

[75] Yong Jae Lee and Kristen Grauman. Shape discovery from unlabeled image collections. In *CVPR*, pages 2254–2261, 2009.

[76] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[77] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *In ECCV workshop on statistical learning in computer vision*, pages 17–32, 2004.

[78] Bastian Leibe, Ales Leonardis, and Bernt Schiele. An implicit shape model for combined object categorization and segmentation. In *Toward Category-Level Object Recognition*, pages 508–524, 2006.

[79] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 878–885, Washington, DC, USA, 2005. IEEE Computer Society.

[80] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

[81] Marius Leordeanu and Martial Hebert. Efficient map approximation for dense energy functions. In *ICML*, 2006.

[82] Marius Leordeanu and Martial Hebert. Unsupervised learning for graph matching. In *CVPR*, 2009.

[83] Marius Leordeanu, Martial Hebert, and Rahul Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, 2007.

[84] Marius Leordeanu, Martial Hebert, and Rahul Sukthankar. An integer projected fixed point method for graph matching and map inference. In *NIPS*, 2009.

[85] Marius Leordeanu, Rahul Sukthankar, and Martial Hebert. Unsupervised learning for graph matching. *International Journal of Computer Vision*, 96(1):28–45, 2012.

[86] Marius Leordeanu, Andrei Zanfir, and Cristian Sminchisescu. Semi-supervised learning and optimization for hypergraph matching. In *ICCV*, 2011.

[87] Joerg Liebelt and Cordelia Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, pages 1688–1695, 2010.

[88] Joerg Liebelt, Cordelia Schmid, and Klaus Schertler. Viewpoint-independent object class detection using 3d feature maps. In *CVPR*, 2008.

[89] Ce Liu. Beyond pixels: Exploring new representations and applications for motion analysis. *Doctoral Thesis. Massachusetts Institute of Technology.*, 2009.

[90] Hairong Liu, Longin Jan Latecki, and Shuicheng Yan. Robust clustering as ensemble of affinity relations. *NIPS*, 2010.

[91] Hairong Liu and Shuicheng Yan. Common visual pattern discovery via spatially coherent correspondences. In *CVPR*, 2010.

[92] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[93] ChengEn Lu, Longin Jan Latecki, Nagesh Adluru, Xingwei Yang, and Haibin Ling. Shape guided contour grouping with particle filters. *ICCV*, 2009.

[94] Tianyang Ma and Longin Jan Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012.

[95] João Maciel and João Costeira. A global solution to sparse correspondence problems. *PAMI*, 25(2), 2003.

[96] S. Maji and J. Malik. A max-margin hough tranform for object detection. *CVPR*, 2009.

[97] Manuel Marques, Marko Stosic, and João Costeira. Subspace matching: Unique solution to point matching with geometric constraints. In *ICCV*, 2009.

[98] David Martin, Charless Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 2004.

[99] Omaima Nomir and Mohamed Abdel-Mottaleb. Hierarchical contour matching for dental x-ray radiographs. *Pattern Recognition*, 41(1):130 – 138, 2008.

[100] Sebastian Nowozin and Christoph H. Lampert. Global interactions in random field models: A potential function ensuring connectedness. *SIAM J. Img. Sci.*, 2010.

[101] Clark F. Olson and Daniel P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, 1997.

[102] B. Ommer and J.Malik. Multi-scale object detection by clustering lines. *ICCV*, 2009.

[103] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragmentmodel for object detection. *ECCV*, 2006.

[104] Andreas Opelt, Axel Pinz, and Andrew Zisserman. A boundary-fragment-model for object detection. In *Proceedings of the 9th European conference on Computer Vision - Volume Part II*, ECCV'06, pages 575–588, Berlin, Heidelberg, 2006. Springer-Verlag.

[105] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 3–10, Washington, DC, USA, 2006. IEEE Computer Society.

[106] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, 80(1):16–44, 2008.

[107] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *IEEE Trans. PAMI*, 29:167–172, 2007.

[108] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *PAMI*, 29:167-172, 2007.

[109] Nadia Payet and Sinisa Todorovic. From contours to 3d object detection and pose estimation. In *ICCV*, pages 983–990, 2011.

[110] Kontschieder Peter, Samuel Rota Bulo, Michael Donoser, Marcello Pelillo, and Horst Bischof. Semantic image labelling as a label puzzle game. In *BMVC*, 2011.

[111] J. Ponce, S. Lazebnik, F. Rothganger, and C. Schmid. Toward true 3d object recognition. In *Congres de Reconnaissance des Formes et Intelligence Artificielle*, 2004.

[112] Pradeep D. Ravikumar and John D. Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *ICML*, 2006.

[113] S. Ravishankar, A. Jain, and A. Mittal. Multi-stage contour based detection of deformable objects. *ECCV*, 2008.

[114] Amelio Vázquez Reina, Shai Avidan, Hanspeter Pfister, and Eric L. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, 2010.

[115] H Riemenschneider, M. Donoser, and H. Bischof. Using partial edge contour matches for efficient object category localization. *ECCV*, 2010.

[116] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3), 2004.

[117] Carsten Rother, Vladimir Kolmogorov, Victor S. Lempitsky, and Martin Szummer. Optimizing binary mrfs via extended roof duality. In *CVPR*, 2007.

[118] Carsten Rother, Sanjiv Kumar, Vladimir Kolmogorov, and Andrew Blake. Digital tapestry. CVPR, 2005.

[119] Carsten Rother, Tom Minka, Andrew Blake, and Tom Minkaand. Cosegmentation of image pairs by histogram matching incorporating a global constraint into mrfs. In *CVPR*, 2006.

[120] Bryan Russell, William Freeman, Alexei Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[121] Silvio Savarese and Fei-Fei Li. 3d generic object categorization, localization and pose estimation. In *ICCV*, pages 1–8, 2007.

[122] Silvio Savarese, Tinne Tuytelaars, and Luc J. Van Gool. Special issue on 3d representation for object and scene recognition. *Computer Vision and Image Understanding*, 113(12):1181–1182, 2009.

[123] Edgar Seemann and Bernt Schiele. Cross-articulation learning for robust detection of pedestrians. In *Proceedings of the 28th conference on Pattern Recognition*, DAGM'06, pages 242–252, Berlin, Heidelberg, 2006. Springer-Verlag.

[124] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[125] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Efficiently combining contour and texture cues for object recognition. In *BMVC*, pages 1–10, 2008.

[126] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multi-scale categorical object recognition using contour fragments. *IEEE Trans. PAMI*, 2008.

[127] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1270–1281, July 2008.

[128] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1270–1281, 2008.

[129] Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*, 2012.

[130] Praveen Srinivasan, Qihui Zhu, and Jianbo Shi. Many-to-one contour matching for describing and discriminating object shape. *CVPR*, 2010.

[131] Bj?rn Stenger, Arasanathan Thayananthan, Philip H. S. Torr, and Roberto Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28:1372–1384, 2006.

[132] Stefan Stiene, Kai Lingemann, Andreas Nuchter, and Joachim Hertzberg. Contour-based object detection in range image. In *In Third International Symposium on 3D Data Processing, Visualization and Transmission, 2006*, 2006.

[133] Min Sun, Hao Su, Silvio Savarese, and Fei-Fei Li. A multi-view probabilistic model for 3d object classes. In *CVPR*, pages 1247–1254, 2009.

[134] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proceedings of the 2003 IEEE computer society conference on Computer vision and pattern recognition*, CVPR'03, pages 127–133, Washington, DC, USA, 2003. IEEE Computer Society.

[135] Alexander Thomas, Vittorio Ferrari, Bastian Leibe, Tinne Tuytelaars, Bernt Schiele, and Luc J. Van Gool. Towards multi-view object class detection. In *CVPR*, pages 1589–1596, 2006.

[136] P H S Torr. Solving markov random fields using semi definite programming. In *AISTATS*, 2003.

[137] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *CVPR (2)*, pages 762–769, 2004.

[138] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, 2008.

[139] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, 2008.

[140] Nhon H. Trinh and Benjamin B. Kimia. Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model. *Int. J. Comput. Vision*, 94(2):215–240, September 2011.

[141] David Tsai, Matthew Flagg, and James M. Rehg. Motion coherent tracking with multi-label MRF optimization. In *BMVC*, 2010.

[142] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, 2010.

[143] Marcel Van Herk. Handbook of medical imaging. chapter Image registration using Chamfer matching, pages 515–527. Academic Press, Inc., Orlando, FL, USA, 2000.

[144] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.

[145] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation revisited: Models and optimization. In *ECCV*, 2010.

[146] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation revisited: models and optimization. In *ECCV*, 2010.

[147] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[148] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[149] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)*, pages 511–518, 2001.

[150] Bo Wang and Zhuowen Tu. Affinity learning via self-diffusion for image segmentation and clustering. In *CVPR*, 2012.

[151] Jun Wang, Tony Jebara, and Shih fu Chang. Graph transduction via alternating minimization. In *ICML*, 2008.

[152] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, 2010.

[153] Bo Wu and Ram Nevatia. Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *Int. J. Comput. Vision*, 82(2):185–204, April 2009.

[154] Pingkun Yan, Saad M. Khan, and Mubarak Shah. 3d model based object class detection in an arbitrary view. In *ICCV*, pages 1–6, 2007.

[155] Xingwei Yang, Nagesh Adluru, and Longin Jan Latecki. Particle filter with state permutations for solving image jigsaw puzzles. In *CVPR*, 2011.

[156] Ron Zass and Amnon Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008.

[157] Bernhard Zeisl, Christian Leistner, Amir Saffari, and Horst Bischof. Online semi-supervised multiple-instance boosting. In *CVPR*, 2010.

[158] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Schölkopf. Learning with local and global consistency. In *NIPS*, 2004.

[159] Long Zhu, Yuanhao Chen, and Alan L. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6):1029–1043, 2010.

[160] Xiaojin Zhu. Semi-supervised learning literature survey, 2006.