Video object search and discovery

Meng, Jingjing

2016

Meng, J. (2016). Video object search and discovery. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/69414>

<https://doi.org/10.32657/10356/69414>

VIDEO OBJECT SEARCH AND DISCOVERY

JINGJING MENG

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2016

Acknowledgments

First and foremost, I would like to thank Professor Yap-Peng Tan for having been such a supportive advisor and mentor of mine. Thank you for your support and guidance throughout my PhD study, for giving me the intellectual freedom to explore areas of my interests, and for all your insightful input to my research.

I am grateful for the opportunities to collaborate with a bunch of great researchers and students in the past four years. Ying Wu, Jiebo Luo, Zhengyou Zhang, Gang Wang, Hongxing Wang, Yuning Jiang, Zhou Ren, Jiong Yang, thank you. It has been a great learning experience.

My sincere appreciation also goes to my QE examiners, my TAC members, the anonymous reviewers of my thesis and the panel of my PhD Oral Defence. Thank you for your invaluable suggestions to my research and presentation. I am thankful to my fellow colleagues and students at the Rapid Rich Object Search (ROSE) Lab, the Information Systems Lab and Media Technology Lab at Nanyang Technological University: Gangqiang, Terry, Xingxing, Amir, Bing Shuai, Yan Wang, Zhe Wang, Jianyi, Tzu-Yi, Venice, Tan and Joseph. Thank you for your support.

Special thanks also goes to my dear friends, Ying, Xuan, Lin, Chengen, Hui, Yi, Margaret (I miss you) and Art. It is sweet to know that regardless of the ups and downs in my life, you always have my back.

I am greatly indebted to my dear grandparents, parents, in-laws, uncles, aunts,

and cousins. Your unconditional love and support have always been my continuous source of strength and inspiration.

Last but not least, I would like to thank my husband, Junsong, and my son, Duoxuan. You made all the difference. You taught me that it is all about balance and persistence. Life is a journey. I am fortunate to have your company.

It is by no means an exhaustive list. It is the people whom I met along my way made me who I am. Thank you all.

Contents

Acknowledgments	i
Summary	ix
List of Figures	xi
List of Tables	xviii
1 Introduction	1
1.1 Motivation	1
1.2 Background	2
1.2.1 Video Object Search	2
1.2.2 Video Object Discovery	7
1.2.3 Video Object Search vs. Video Object Discovery	11
1.3 Thesis Overview	13
1.3.1 Object Instance Search in Large-scale Videos	13
1.3.2 Discovering Representative Objects from Videos	16

1.3.3	Contributions	17
1.4	Road Map	18
2	Background and Related Work	21
2.1	Object Instance Search in Large-scale Videos	21
2.1.1	Object Representation and Matching	21
2.1.2	Object Localization	26
2.1.3	Object Instance Search vs. Object Detection	29
2.2	Discovering Representative Objects from Videos	31
2.2.1	Object-driven Video Summarization and Object Discovery . .	31
2.2.2	Representative Selection	32
2.2.3	Local Structure Learning	33
3	Spatio-temporal Search of Object Instances in Videos	35
3.1	Overview	35
3.2	Problem Formulation	37
3.2.1	Fast Spatio-Temporal Trajectory Search	39
3.2.2	Efficient Confidence Map Generation	41
3.2.3	Coarse-to-Fine Search	43
3.3	Evaluation Metrics	43

3.3.1	Video mAP	45
3.3.2	Trajectory mAP	45
3.4	Experimental Setup	46
3.4.1	Dataset	46
3.4.2	Queries	48
3.5	Experimental Results	48
3.5.1	Video mAP	48
3.5.2	Trajectory mAP	53
3.5.3	Efficiency	58
3.6	Discussions	59
3.6.1	Measuring Object-Video Relevance by Trajectories	59
3.6.2	Object Instance Search vs. Detection	60
3.7	Summary	61
4	Object Proposals to Speed Up Spatio-Temporal Search	63
4.1	Overview	63
4.2	Problem Formulation	65
4.2.1	Spatio-Temporal Search with Candidate Bounding Boxes	67
4.2.2	Bounding Box Scoring	68

4.2.3	Coarse-to-Fine Search	71
4.3	Experimental Setup	71
4.4	Experimental Results	72
4.4.1	Candidate Selection Methods	72
4.4.2	Video mAP	73
4.4.3	Trajectory mAP	78
4.4.4	Rand-MP	79
4.4.5	Prop-MP	80
4.4.6	Efficiency	82
4.5	Discussions	83
4.5.1	Object Size Prior vs. Objectness Prior	83
4.5.2	Efficiency of Spatio-Temporal Search	88
4.5.3	3D Object Proposals and Spatio-Temporal Search	89
4.6	Summary	89

5 Representative Object Discovery for Video Summarization and Search 93

5.1	Overview	93
5.2	Problem Formulation	96
5.2.1	Preliminaries: sparse dictionary selection	96

5.2.2	Weighted sparse dictionary selection	97
5.2.3	Locally linear reconstruction (LLR) induced sparse dictionary selection	97
5.3	Experiments on Synthetic Data	102
5.4	Experiments on Real Datasets	104
5.4.1	Settings	104
5.4.2	Baseline algorithms	105
5.4.3	Evaluation Metric	106
5.4.4	Topical Video Object Discovery	107
5.4.5	Video Summarization for Object Search	109
5.5	Discussion	115
5.5.1	Object Discovery for Video Search	115
5.6	Summary	116
6	Conclusions and Future Directions	117
6.1	Conclusions	117
6.1.1	Video Object Search	117
6.1.2	Video Object Discovery	120
6.2	Future Work	122
6.2.1	Future Work on Video Object Search	122

6.2.2	Future Work on Video Object Discovery	123
6.3	Systems to Integrate Discovery and Search	124
	Author's Publications	125
	References	127

Summary

In terms of volume, videos are becoming the largest big data. The sheer volume of video data demands powerful analytic tools to organize and make sense of them. This thesis proposes to tackle two fundamental problems in big video analytics, i.e., search and discovery, from an object-driven angle.

Objects that we consider are the fundamental components of a video, which are concise, visually meaningful and informational. The mere presence of certain objects in a video and their interactions can provide us rich information for video understanding. In addition, they can help establish a quick impression of the video by telling what are there, and provide a small footprint for video indexing, browsing and search.

For video object search, we aim to search for and locate a specific object spatio-temporally in the video volume. The main challenges are: 1) object appearance variations across video frames caused by pose and scale variations, partial occlusions, etc., 2) false positives introduced by background clutters, and 3) search efficiency. We propose to formulate video object search as a problem of finding the spatio-temporal object trajectories, where an object trajectory consists of a sequence of bounding boxes that locate the target object across frames. We also present a Max-Path search solution that can effectively reduce the complexity of trajectory search from exponential to linear to the video volume size. Furthermore, we present and evaluate the use of object proposals to speed up matching and trajectory search. Experimental results demonstrate three benefits of the proposed approaches. First, the formulation

as trajectory search can effectively improve matching accuracy by enforcing spatio-temporal coherency to overcome appearance variations and background clutters. In addition, the resulting trajectories offer an alternative to frames for measuring object occurrences and consequently the search performance. Second, the Max-Path based trajectory search is efficient and compatible with both dense confidence maps and coarsely sampled object proposals. Third, the object proposal based approach can significantly boost search efficiency without compromising accuracy.

For video object discovery, this thesis focuses on the discovery of representative objects from videos. We propose to address this problem by selecting representative object proposals generated from video frames. Although representative selection methods have been applied to video keyframe selection, directly applying them to object-level selection faces two major challenges. First, the key objects do not necessarily locate at the densest regions in the feature space due to the appearance variations of the same object across frames, hence, classic density based representative selection method may not work well. Second, the irrelevant and noisy proposals in the proposal pool may significantly affect representative selection methods based on sparse reconstruction. To address these challenges, we have devised a new formulation of sparse reconstruction based representative selection that can incorporate object proposal priors and locality prior in the feature space when selecting representatives. Consequently it can better locate key objects and suppress outlier proposals. Although complex constraints have been introduced, we show that the optimization can be converted into a proximal gradient problem and be solved by the fast iterative shrinkage thresholding algorithm (FISTA).

The proposed methods are compared against existing state-of-the-arts for object instance search and representative object discovery on challenging datasets. It shows that our methods can more accurately find relevant videos pertaining to an object of interest and discover key objects that capture the essence of a video.

List of Figures

1.1	Visual object search is a key enabler for applications such as contextual advertising (left), consumer profiling (middle) and video surveillance (right).	2
1.2	The sub-problems of video object search with representative approaches. Blocks in dashed box are our work.	3
1.3	The sub-problems of representative video object discovery with representative approaches. The block in dashed box is our work.	8
1.4	An example search system built on both video object discovery and video object search. Representative objects are first discovered from each video in the dataset off-line. When a query object comes in, it is matched against the representative objects of each video to quickly filter irrelevant videos and rank the rest. Finally, the target object can be localized in matched videos spatio-temporally using video object search. How each chapter in the thesis fit into the pipeline is also shown.	12

1.5	A toy example of how trajectory-based measure of search performance differs from frame-based measure. Two ground truth trajectories (T1 and T2) are in the dataset for a given query, consisting of 200 frames and 50 frames respectively. Method A is able to return 150 out of 200 frames from T1 and all 50 frames from T2, while method B is able to return all 200 frames of T1 but misses T2 entirely. The two methods are comparable in terms of frame-wise recall, whereas method A is clearly superior in terms of trajectory-wise recall.	15
2.1	Organization of topics covered in Section 2.1.	22
3.1	Overview of proposed approach to find object trajectories in videos. Video keyframes are first matched with the query to produce confidence maps with pixel-wise matching score (2 nd row, Section 3.2.2). Next low confidence videos (the gray cuboids in the 3 rd row) are filtered by ranking videos based on the keyframe scores (Section 3.2.3). Max-Path search then finds the globally optimal object trajectories in top ranked videos and re-rank them according to the trajectory scores (Section 3.2.1). Ground truth locations are highlighted in dashed blue boxes in the found trajectories (4 th row). Best viewed in color and magnification. © 2016 IEEE	44

3.2	Example videos from the 73-hour video dataset (only one keyframe is shown for each video). Videos with annotated object locations are from NTU-VOI. The remaining are distractors from Stanford I2V [1]. Three example queries are given on the left: KittyB, NTU and Pokka (Table 3.1). Their ground truth locations are annotated by magenta, blue and red bounding boxes, respectively. NTU-VOI dataset comprises diverse scenes, most of which are cluttered with small target objects. This makes NTU-VOI a challenging dataset for object instance search and localization. Best viewed in color and magnification. © 2016 IEEE	47
3.3	The impact of β on video mAP: our RVP-MP method consistently outperforms the RVP-Baseline [2] and SCFV-RANSAC [3] as β changes from -3 to -0.5 . © 2016 IEEE	52
3.4	Two example queries (left) and top trajectories returned by our method. Each trajectory is represented by one keyframe with magnified object region below or above the keyframe. The red boxes indicate our results, and the green boxes indicate the ground truth annotations. Given a single query example, our approach is able to find its occurrences in cluttered scenes regardless of appearance variations due to occlusions, motion blur, and changes in viewpoint, illumination and color. Best viewed in color and magnification. © 2016 IEEE	55

3.5	Example search results of "KittyG" (a) and "Plane" (b). Results of KCF tracker [4] are denoted in yellow bounding boxes in the top rows. Results of RVP-MP are in red in the bottom rows. Ground truth bounding boxes are in green. (a) KCF fails to find the correct start and end points of the trajectory when the object enters and exits the scene, while RVP-MP discover the start and end points automatically. (b) An inaccurate initial estimation of bounding box size will be carried on by KCF, while RVP-MP searches for the best bounding box size given the confidence maps. Best viewed in color and magnification. © 2016 IEEE	56
3.6	Trajectory mAP: our RVP-MP method consistently outperforms KCF tracker [4] in a wide range of overlap ratio δ_p . © 2016 IEEE	57
3.7	Example trajectories returned by our proposed RVP-MP in the presence of varying viewpoint (1^{st} and 2^{nd} rows) and scale (2^{nd} row), reflection (2^{nd} row) and occlusions (3^{rd} row). In addition, the start and end points of each trajectory are discovered automatically ($4^{th} - 6^{th}$ rows). RVP-MP results are marked in red, and the ground truth bounding boxes are in green. Best viewed in color and magnification. © 2016 IEEE	58
3.8	Trajectory mAP of RVP-MP for different temporal steps and overlap ratio δ_p . Best viewed in color and magnification. © 2016 IEEE	59

4.1	Overview of proposed approach to find object trajectories in videos. The query is only matched with candidate bounding boxes in each video keyframe (i.e., colored boxes in the 2 nd row, see Section 4.2.2). Then videos are ranked by the scores of their best matched bounding boxes (red boxes in the 3 rd row), while low-rank videos are filtered (the gray cuboids in the 3 rd row). At last the globally optimal object trajectories in top ranked videos are identified by the Max-Path search (Section 4.2.1) and videos are re-ranked according to the trajectory scores. Ground truth locations are highlighted in dashed blue boxes in the found trajectories (4 th row). Best viewed in color and magnification.	70
4.2	The impact of β on video mAP: Rand-MP consistently outperforms RVP-Baseline [2] and Rand-Max as β changes from -8 to -2 . It is comparable to RVP-MP (with best β) when $\beta \in [-8, -4]$.	74
4.3	The impact of β on video mAP: Prop-MP achieves comparable performance to RVP-MP in a wide range of β ; it also consistently outperforms Prop-Max and RVP-Baseline [2] as β changes from -2 to 0 .	77
4.4	Trajectory mAP: the proposed Rand-MP performs poorly compared with RVP-MP and the KCF tracker [4] (except for when $\delta_p = 0.1$) in a wide range of overlap ratio δ_p .	79
4.5	Trajectory mAP of the proposed Prop-MP w.r.t. the overlap ratio δ_p , as different number of proposals are sampled per frame.	80
4.6	Trajectory mAP: the proposed Prop-MP approach achieves similar performance compared with RVP-MP, while consistently outperforms KCF tracker [4] in a wide range of overlap ratio δ_p .	81

4.7	Example trajectories returned by our proposed Prop-MP in the presence of varying viewpoint (3^{rd} and 4^{th} rows) and scale (3^{rd} row), reflection (3^{rd} row) and occlusions (2^{nd} and 6^{th} rows). In addition, the start and end points of each trajectory are discovered automatically (1^{st} , 4^{th} – 6^{th} rows). Prop-MP results are marked in red, and the ground truth bounding boxes are in green. Best viewed in color and magnification.	82
4.8	(a) Average best IoU of ground truth videos and video mAP by Rand-Max, and (b) average best IoU of ground truth keyframes and video mAP by Rand-MP, with respect to different random template parameters $\mathcal{M} \times \mathcal{N}$	86
4.9	Average number of bounding boxes per ground truth keyframe whose $\text{IoU} > 0.5$ as a function of the number of candidate bounding boxes per keyframe. (a) random proposals using the object size prior, (b) Edge Boxes proposals using the objectness prior.	87
5.1	Representative video object discovery via object proposal selection. A pool of object proposals (middle row) are first generated from video keyframes (top). Representative object proposals (bottom) are selected from the pool to summarize the video. © 2016 IEEE	94
5.2	Data points in three clusters with background uniform noise (blue dots) and the representatives (red circles) found by (a) the proposed LLR-SDS method (b) SDS [5], (c) SMRS [6] and (d) SMRS with outlier removal [6]. © 2016 IEEE	103

5.3	Video summarization for topical object discovery: example representative objects selected by different methods. Keyframes are shown in the top row. Each following row shows the 8 representative object proposals selected by different methods in rank order (except for K-medoids and LDA-WCP). In the second row, regions in normal color indicate topical objects discovered by LDA-WCP, while those in green are background regions. © 2016 IEEE	109
5.4	Average recall with selection ratio $\eta \in [0.001, 0.2]$. © 2016 IEEE . . .	111
5.5	Evaluation of the relationship between the average recall and overlap ratio θ ($\eta = 0.01$). © 2016 IEEE	112
5.6	Average recall when the regularization parameter $\alpha_2 \in [-3, 1]$. A small α_2 (i.e., -3) leads to a small λ_2 , which produces a curve similar to that of SDS; while $\alpha_2 = 1$ leads to a solution similar to that of LLR (Section 5.2.3.4). © 2016 IEEE	113
5.7	Video summarization for object search: visual results of the 8 shots of Microphone from Groundhog Day ($\eta = 0.5\%$ of all object proposals are selected). For each shot, the object proposal with the best IoU with the ground truth (GT) is shown in the yellow bounding box and the GT is in red. Missed shots are highlighted by red rectangles. Overall our method produces summarizations that more accurately capture the GT. Note that given a shot, the best IoU object proposal selected by different methods may come from different keyframes. Best viewed in color and magnification. © 2016 IEEE	115

List of Tables

3.1	Statistics of Ground Truth (GT) Object Instance Trajectories in VOI Dataset © 2016 IEEE	49
3.2	Video mAP on Top 100 Retrieved Clips © 2016 IEEE	51
3.3	Video mAP on top 100 retrieved clips. RVP-MP consistently improves AP on all queries over RVP-Baseline, except for Ferrari (same AP). Overall RVP-MP improves RVP-Baseline by 16.71%. © 2016 IEEE .	53
4.1	Video mAP on top 100 retrieved clips using Rand-Max.	74
4.2	Video mAP on top 100 retrieved clips. Rand-MP achieves comparable performance compared with RVP-MP, and consistently improves AP on all queries over Rand-Max. Overall Rand-MP improves Rand-Max by 19.50%.	75
4.3	Video mAP on top 100 retrieved clips and search time using Prop-Max.	76
4.4	Video mAP and Max-Path search time on top 100 retrieved clips using Prop-MP (when $\beta = -1$).	76

4.5	Video mAP on top 100 retrieved clips. Prop-MP achieves comparable performance compared with RVP-MP, and consistently improves AP on all queries over Prop-Max. Overall Prop-MP improves PROP-Max by 17.38%.	78
4.6	Efficiency of Rand-MP and Prop-MP in comparison with RVP-MP	83
5.1	Average recall when selecting 8 object proposals. Methods are sorted by increasing average recall@8. Our method with objectness as prior weights (LLR-wSDS) achieves the best average recall among all. © 2016 IEEE	108
5.2	Average percentage of object proposals required to cover the ground truth object. © 2016 IEEE	114

Chapter 1

Introduction

1.1 Motivation

With videos becoming the largest big data, there has been increasing demand for analytic tools to organize and make sense of them. The main interest of this thesis lies in big visual data analytics, with a focus on the important *nouns* in videos, i.e., objects. Objects can be considered as the fundamental components of a video, which are concise, visually meaningful and informational. The mere presence of certain objects in a video and their relative spatio-temporal relationship convey rich information for video understanding. In addition, by analyzing the interactions between objects or between objects and the environment, more complicated semantics in videos can be recovered, such as to locate actions by tracking candidate objects across frames, or to construct a story based video summarization by analyzing the interactions among the key objects. Moreover, recent advances in computer hardware have made it feasible and economical to process videos at such a fine grain. Therefore, the research objective of this thesis is to develop principled methodologies and efficient systems to access, analyze and summarize big video data, from an object-driven perspective.

Figure 1.1: Visual object search is a key enabler for applications such as contextual advertising (left), consumer profiling (middle) and video surveillance (right).

1.2 Background

This thesis explores two fundamental problems in the video-object context: *search* and *discovery*.

1.2.1 Video Object Search

Search refers to methodologies that look for relevant information in a dataset by providing a query (i.e., what you are looking for). For text search, the goal is to retrieve relevant texts given certain keywords, whereas for video object search, given a query object, our goal is to not only retrieve relevant video clips that contain the target object, but also accurately locate instances of the target object spatio-temporally in these clips. Usually the query object is a specific object (e.g., the BMW logo) rather than an object class (e.g., logos, cars and dogs).

In recent years, video object search has attracted increasing interest from the research community and the industry because of its many applications such as contextual advertising, consumer profiling and video surveillance (Fig. 1.1). Taking contextual advertising and embedded marketing in online videos [7] for example, knowing exactly which frames in a video and locations in these frames where a particular product or logo appears not only enables close correlation between the content of an advertisement and the content of the video, but also allows for accurate measurement

Figure 1.2: The sub-problems of video object search with representative approaches. Blocks in dashed box are our work.

of the relevance of the product in specific contexts by tracking users' interactions with it (e.g., click-through rate). Consequently, better models for allocating and pricing advertising inventory can be built to maximize revenues for both the content providers and the sellers of the product.

Just like text search, video object search is essentially a matching problem constrained by time (especially when dealing with video data of large scale). Therefore we care about both matching accuracy and search efficiency. It can be decomposed into two sub-problems: object representation and the matching scheme. We discuss both sub-problems below. In this thesis, we focus on the latter to improve both matching accuracy and search efficiency.

1.2.1.1 Object Representation

Object representation aims to capture the visual appearance of an objects in terms of numerical feature(s) to facilitate comparison. Existing object features can be either hand crafted from low-level visual properties such as SIFT [8], SURF [9] and GIST [10], or mid-level features such as those learned from convolutional neural networks (CNN) [11]. Alternatively, we can categorize object representation by global, local and hybrid descriptors. The global descriptors summarize the appearance of the whole object as a single descriptor, and the local descriptors represent an object by a collection of local descriptors. The hybrid descriptors, such as the popular Bag-of-Words (BoW) [12] and its variants [13], produce a global representation (e.g., a single sparse vector of fixed dimension) from local descriptors. They also include approaches based on higher order statistics [14–17].

Because the same object may exhibit large appearance variations across video frames due to pose and scale variations, partial occlusions, motion blur, etc., and the background of a video scene can be cluttered, accurate matching given only one query example is challenging. To improve matching accuracy, the object representation has to be robust to appearance variations. However, generalizing the object representation can be a double-sided sword, as making the representation too generic may reduce its discriminative power. Consequently, the improvements in recall may come at the price of compromising precision due to increased number of false positives. As for search efficiency, certain object representations can significantly speed up matching, such as bit strings resulting from binary encoding [18–20]. However, we do not pursue the study of object representation in this thesis, but focus on the matching scheme.

1.2.1.2 Matching Scheme

Matching scheme refers to how matching is to be done. The choice of object representation directly affects the choice and performance of the matching scheme. For

instance, matching of global descriptors can be done by direct comparisons of these descriptors, whereas matching based on local descriptors typically depends on finding putative correspondences, which is followed by a check of geometric consistency.

The past decade has witnessed much progress in object instance search in image collections [2, 12, 21–28]. Naturally, one could directly apply the matching schemes tailored for image collections to video data by treating each frame independently. For instance, in *Video Google*, Sivic and Zisserman [12] first obtain the locations of objects in individual frames by identifying matched local regions that have K nearest neighbor support, then rank the keyframes or shots of a video that contain the object accordingly. The temporal consistency between the video frames is largely ignored except when rejecting unstable regions across frames in a shot. Similarly, the TRECVID object instance search challenge aims to find video shots that most likely contain a query topic (i.e., object) without concerning about its location in each frame or how many frames in the shot contain the topic. Consequently, most earlier systems first down-sample videos to keyframes, and consider the task as an image retrieval or instance search problem to find keyframes that most likely contain the query topics [29–32].

In this thesis, we propose an alternative matching scheme, which is based on object trajectories instead of individual frames. We refer to an object trajectory as a sequence of bounding boxes that locate the object of interest spatio-temporally across frames. The observation is that unlike image data, videos have an extra temporal dimension. Although the inherent spatio-temporal redundancy in videos presents great challenges for efficient storage and processing, it can also serve as an important context to localize objects and determine their importance in a video. Specifically, we formulate the problem of video object search as the spatio-temporal trajectory search problem (Chapter 3, Chapter 4). The goal is to find the top K object trajectories in a large video dataset. It is shown that compared with conventional frame-based matching schemes, the new formulation can effectively improve matching accuracy by

recovering missed detections due to object appearance variations and reducing false alarms caused by background clutters.

To efficiently solve the trajectory search problem, we propose to leverage the dynamic programming based Max-Path search [33] method, which reduces the complexity of trajectory search from exponential to linear to the video volume size. We also explore two approaches to achieve efficient matching. The first is a *dense* matching scheme, named Randomized Visual Phrases with Max-Path search (RVP-MP) (Chapter 3). It involves generating pixel-wise confidence maps from random spatial context [2] to enable fast scoring of any bounding box in the video using integral images built from these confidence maps [34]. The second is a *coarse* scheme, where only a few object proposals as candidates are matched against the query (Chapter 4), for which we evaluate and compare (1) random proposals based on object size priors (Rand-MP) and (2) proposals from principled Edge Boxes [35] (Prop-MP). Both the dense and coarse approach can be unified under the Max-Path search approach, where the difference is whether the spatio-temporal search is to be carried out on *dense* pixels or *coarse* object proposals. The efficiency of the former greatly depends on the confidence map generation and trajectory search based on *dense* nodes, while the latter largely depends on the number of candidate boxes to be checked. Experimentally, the latter object proposals based approach is shown to be advantageous to the former in search efficiency. However, its matching accuracy is sensitive to the quality of the object proposals. In addition, both schemes are orthogonal to the choice of object representation. Although we use off-the-shelf BoW object representation for studies in video object search, other global or hybrid object representations are also compatible with the proposed trajectory-based matching schemes.

Fig. 1.2 illustrates how Chapter 3 and Chapter 4 fits into the video object search structure and respective sub-problems they address.

1.2.2 Video Object Discovery

Discovery generally aims to discover patterns in a dataset without knowing beforehand what are there. The power of discovery lies in its ability to discover new, previously unknown information that would otherwise remain buried in a mass of data. Particularly, given the massive collections of data, discovering the *representatives* from them offers many benefits. Here we define the *representatives* as a subset of the data that appropriately represents the entire dataset. For instance, the representatives can help provide a brief summary of large datasets to facilitate data browsing and visualization. They also help remove outliers to mitigate data bias and reduce the size of the datasets to speed up data analysis. Take text for example, a representative subset or abstract of the text can be discovered for document summarization, or underlying topics can be discovered for text categorization. Similarly, discovering representative objects from videos offers a concise visual summary of the video content, which can in turn be used for search, or to produce annotations of the video to facilitate categorization. Moreover, further analysis (e.g., frequency, relationship) and processing of these objects may reveal high level semantics beyond these representative objects.

For video object discovery, in this thesis, we will focus on discovering *representative objects* from videos for video summarization and search, where the *representative objects* refer to a subset of objects that appropriately describes the entire set of objects from a video.

One strategy to tackle this problem is to take a two-step approach:

- (1) Locating all objects in the video.
- (2) Selecting representative objects from the pool of objects resulting from the previous step.

In this thesis, we use existing approaches to address the first problem and focus on the second problem.

Figure 1.3: The sub-problems of representative video object discovery with representative approaches. The block in dashed box is our work.

1.2.2.1 Locating Objects

To address the first problem, one needs to know: what is an object? Different from visual object search in the previous section, where a query object is provided as reference, here we have no idea what an object looks like and nothing to match against. Therefore, we need class-generic object detectors, which can be used to exhaustively check each possible region in a video and answer whether it is an object, regardless of which object it is. However, objects can be heterogeneous in colors and textures with widely varying geometries. Even if we simplify the geometry of objects in a video to rectangular windows in the frames, instead of more complex forms such as segmentations or contours, it is still challenging to come up with a model that can well capture all the possible variations in different objects. In addition, even for the same object, its appearance may change greatly depending on the lighting condition, partial occlusion or motion blur.

Fortunately, recently many researchers have started to investigate the problem

of generic object detection in images, which proposes windows or segments that are likely to localize objects in an image, regardless of their classes. The resulting windows or segments are commonly called object proposals. Object proposals have been shown to benefit many applications such as class-specific object detection and weakly supervised learning of object detectors [35–40].

Thanks to existing class-generic object detectors, we can sidestep the first problem of locating objects in the videos by directly taking the frame-wise object proposals resulting from these detectors. We do not focus on devising new class-generic object detectors in this thesis, but use off-the-shelf detectors to obtain the object proposals.

Since the major goal of object proposals for static images is to recall as many objects as possible, the proposed regions can be highly redundant with large overlaps, and some proposals can be noisy. Furthermore, as we are dealing with videos, by directly taking object proposals independently from individual frames, we have introduced additional redundancies from the temporal dimension, and the same object can exhibit great appearance variations across frames. All these present challenges for representative selection. Admittedly, one could consider the temporal consistency of the same object across frames to help filtering noisy object proposals and reducing redundancies, but we do not investigate the temporal information in the thesis, which can be our future work.

1.2.2.2 Selecting Representatives

As mentioned, this thesis focus on addressing this second problem for representative video object discovery. Since we choose to directly select representatives from the pool of frame-wise object proposals produced by existing class-generic object detectors, the selection approach needs to cope with large sample redundancies and noise, and object appearance variations.

But first, how do we define *representativeness*? We only have a vague idea that representative objects should well describe the entire set of all objects in a video, which in our case requires that the selected object proposals to appropriately represent all object proposals in the pool. In the literature, the definition of representativeness is closely coupled with the assumptions made on the data distribution. According to the different assumptions on the distribution of the data, existing representative selection approaches can be roughly categorized into two groups. The first group is clustering based, which assumes that there is a natural grouping of the data collection based on certain pair-wise similarity measure of data points. Therefore, algorithms in this group typically operate on the pair-wise similarities or dissimilarities. Examples in this category include the K-medoids algorithm [41], affinity propagation [42, 43] and density peak search [44]. The second group are subspace learning based methods, which assume that the data lie in one or multiple low-dimensional subspaces. Typically, these methods operate on the data vectors directly. Examples include dictionary learning in [45, 46] and dictionary selection in [5, 6, 47–49].

However, directly applying existing methods to the pool of object proposals generated from video frames faces a few challenges. First, due to the appearance variations of the same object across frames, the representative objects do not necessary locate at the densest regions in the feature space. Consequently, classic density based representative selection method may not work well [44]. Second, there may be many irrelevant and noisy proposals in the proposal pool, which can affect the representative selection methods based on sparse reconstruction [5, 6]. In such a case, even a further filtering of outliers as post-processing [6] may be less effective if most representatives are outliers. Moreover, the definition of representativeness can be application dependent or subjective. For instance, in egocentric videos, objects that the subject interact with are usually more important than others [50], while in surveillance videos from a fixed camera, moving foreground objects likely carry more weights than static background objects. Therefore, a formulation that allows easy incorporation of prior knowledge is desirable, which can guide the selection of representatives.

To address the above challenges, we propose a novel formulation of sparse dictionary selection, which incorporates both object proposal priors and locality prior in the feature space (Chapter 5). The object proposal priors allows the incorporation of prior knowledge by favoring object proposals of high prior weights. And the locality prior prefers popular object proposals that appear more frequently while suppressing outlier proposals. Comparisons with existing representative selection approaches such as K-mediod [41], sparse dictionary selection [5, 6] and density based selection [44] validate that the proposed formulation can better capture the key video objects despite appearance variations, cluttered backgrounds and camera motions.

Figure 1.3 illustrates how Chapter 5 fits into the representative video object discovery structure and respective sub-problem it address.

1.2.3 Video Object Search vs. Video Object Discovery

The commonness between video object search and video object discovery is that both aim to extract objects from the videos. In addition, video object discovery may benefit video object search by reducing the search space. Figure 1.4 illustrate an example search system that is build on both video object discovery and video object search. Given a video dataset, we first discover a small number of representative objects from each video. Then when a query object comes in, we can quickly filter irrelevant videos and rank relevant videos by only matching these few representative objects. If the goal is to only retrieve relevant videos, we are done. Otherwise if we also want to locate the target object in each frame of the videos that contain it, we can further perform video object search on the returned relevant videos. We highlight places in the pipeline that can be addressed by corresponding chapters in this thesis.

However, video object search and video object discovery differ in two aspects:

- (1) Video object search is supervised in the sense that one example query object is provided. On the contrary, video object discovery is unsupervised, where

Figure 1.4: An example search system built on both video object discovery and video object search. Representative objects are first discovered from each video in the dataset off-line. When a query object comes in, it is matched against the representative objects of each video to quickly filter irrelevant videos and rank the rest. Finally, the target object can be localized in matched videos spatio-temporally using video object search. How each chapter in the thesis fit into the pipeline is also shown.

one tries to discover objects from the videos without prior knowledge of what the objects can be (although certain formulations such as ours can allow the incorporation of priors if available).

- (2) In terms of performance evaluation, video object search cares about both efficiency and accuracy, while video object discovery cares about the recall of different objects. Although the accuracy of video object search is also related to the recall (i.e., usually evaluated by the average precision (AP), which is the area under the precision-recall curve), video object search aims to recall all occurrences of the query object in all frames that it appears, whereas video object discovery needs to recall one occurrence of each object. In other words, discovery cares about the recall of different objects rather than that of all occurrences of the same object.

1.3 Thesis Overview

In what follows, we summarize the research challenges and contributions of this thesis on video object search and discovery.

1.3.1 Object Instance Search in Large-scale Videos

1.3.1.1 Challenges

As mentioned in Section 1.2.1, in essence, object search is a matching problem with time constraint. Different from detection, where models need to be learned to capture variations in object categories, in most search problems, only one example instance is given as the query. We often bypass the training step, but match the query directly against a dataset. Although this alleviates the burden of collecting proper training samples, which can be labor intensive, directly matching a query object against large video datasets faces three challenges:

- (1) Even the same object may present large appearance variations across video frames due to pose and scale variations, partial occlusions, etc., which can decrease the matching accuracy.
- (2) The background clutters in videos can easily introduce false positives.
- (3) How to ensure efficiency in large video datasets?

1.3.1.2 Contributions

- Our first contribution is to formulate the problem of object instance search in videos as a problem of spatio-temporal trajectory search, where each trajectory is a temporal series of bounding boxes that locate the object of interest across frames (Chapter 3). The goal is to find the top- K trajectories in the video dataset. Compared with conventional frame-based methods, the proposed trajectory-based formulation is shown to effectively improve matching accuracy by enforcing the spatio-temporal coherency across video frames. It helps recover missed detections due to object appearance variations and reduces false alarms caused by background clutters. Moreover, the resulting trajectories offer an alternative to frames for measuring object occurrences and consequently the search performance. We elaborate on this point below.

Note that finding object trajectories differs from finding video frames that contain the object. For the latter, we can treat video frames as a collection of independent images and apply existing object instance search methods for image collections. The recall is evaluated based on the number of ground truth frames retrieved. Contrarily, an object trajectory is a spatio-temporally coherent tube in the video where the object appears, which consists of multiple frames. Here we care about the number of ground truth trajectories that are recalled.

Ground Truth Trajectories		Frame-wise Recall	Trajectory-wise Recall
Method A		0.8	1
Method B		0.8	0.5

Figure 1.5: A toy example of how trajectory-based measure of search performance differs from frame-based measure. Two ground truth trajectories (T1 and T2) are in the dataset for a given query, consisting of 200 frames and 50 frames respectively. Method A is able to return 150 out of 200 frames from T1 and all 50 frames from T2, while method B is able to return all 200 frames of T1 but misses T2 entirely. The two methods are comparable in terms of frame-wise recall, whereas method A is clearly superior in terms of trajectory-wise recall.

Figure 1.5 shows an example of how the difference would affect performance evaluation. Assume the dataset contains two ground truth trajectories for a query: one consists of 200 frames (T1), and the other consists of 50 frames (T2). Search method A is able to return 150 out of the 200 frames in T1, and all 50 frames in T2. Search method B is able to find all 200 frames in T1, but none of the 50 frames in T2. If evaluated by frame-wise recall, both methods achieve the same recall of 0.8, as 200 out of 250 ground truth frames are retrieved, although method B missed 50% of the ground truth trajectories. On the contrary, trajectory-wise recall is able to capture the difference, which is 1.0 for method A and 0.5 for method B.

Depending on the applications, finding trajectories may be more critical than finding individual frames. Thinking of searching for a suspicious bag in hundreds of surveillance cameras covering Singapore, one probably could afford missing a few frames in some trajectories, as long as these trajectories are found, rather

than missing the entire bag trajectory in the surveillance footage from one of the MRT station cameras.

- For search efficiency, we present an efficient solution to spatio-temporal search using the Max-Path search [33], which reduces the exponential search complexity to be linear to the video volume size. Furthermore, we present two efficient matching schemes that are compatible with the Max-Path search: one based on *dense* confidence maps (i.e., Randomized Visual Phrases with Max-Path search (RVP-MP) proposed in Chapter 3) and the other based on *coarse* object proposals (i.e., random proposals based on object size priors (Rand-MP) and proposals from principled Edge Boxes [35] (Prop-MP) proposed in Chapter 4).
- Systematic evaluation on a 73-hour video dataset validates the effectiveness of the object proposal based matching scheme in achieving competitive accuracy with a 3 to 60-time speedup compared to RVP-MP in Chapter 3 (Table 4.6). It validates that good object proposals can produce a much smaller but relevant search space with significant gain in speed and memory efficiency.

1.3.2 Discovering Representative Objects from Videos

1.3.2.1 Challenges

In light of the recent successes of category independent object proposals for detection and weakly supervised learning [35–40, 51, 52], we propose to discover representative objects from frame-wise object proposals. It has been shown that generating multiple overlapping rectangular object proposals in general provides more accurate object regions than segmentation by sidestepping the harder problem of full segmentation. Then start with a pool of frame-level object proposals produced by off-the-shelf object proposal methods, the problem can be formulated as a representative selection problem: selecting a few exemplar object proposals to reconstruct the whole pool.

Although representative selection methods have been applied to video keyframe selection [6], directly applying them to object-level selection faces two major challenges.

- (1) The appearance of the same object may change significantly across frames due to pose and scale variations, partial occlusions, etc., hence the key objects do not necessary locate at the densest regions in the feature space. Consequently, classic density based representative selection method may not work well [44].
- (2) Since object proposals are just candidates of key objects, there may be many irrelevant and noisy proposals in the proposal pool. These outliers may significantly affect the representative selection methods based on sparse reconstruction [5, 6]. In such a case, even a further filtering of outliers as post-processing [6] may be less effective if most representatives are outliers. Without prior knowledge of the object, it is difficult to locate the key object accurately.

1.3.3 Contributions

We formulate the representative object discovery problem as the sparse dictionary selection problem. Our contributions include:

- A new formulation of sparse reconstruction based representative selection, which can incorporate object proposal priors and locality prior in the feature space when selecting representatives. Therefore, object proposals of high prior weights, such as high objectness scores, are more likely to be selected as key objects, while background clutters of low weights can be suppressed. In addition, by introducing a locality prior matrix that considers the local affinity structure of the data to regularize the selection matrix, the selection would prefer popular object proposals that appear more frequently while outlier proposals are likely to be suppressed.

- Although complex constraints are introduced, we show how to convert the optimization into a proximal gradient problem and solve it by the fast iterative shrinkage thresholding algorithm (FISTA) [53], which has a fast convergence rate of $O(1/m^2)$ in m iterations.
- Our proposed formulation is generic. The F-norm regularization in (5.8) can also incorporate other prior weight matrix $\mathbf{W}$ to the selection matrix $\mathbf{S}$, without being limited to the LLR-induced locality prior we used in Chapter 5.

Comparisons with existing representative selection approaches such as K -mediod, sparse dictionary selection [5, 54] and density based selection [44] validate that this formulation can better capture the key video objects in videos despite appearance variations, cluttered backgrounds and camera motions.

1.4 Road Map

In the next chapter, a more detailed overview of the problem setting is presented and important related works are discussed. Chapter 3 and Chapter 4 are devoted to object instance search in videos. Chapter 3 proposes to formulate the problem as spatio-temporal trajectory search to improve the matching accuracy by enforcing the spatio-temporal coherency across video frames. In addition, we develop an efficient solution to find the top K object trajectories in videos by combining randomized spatial context with dynamic programming. Then, build upon the work in Chapter 3, Chapter 4 proposes a more efficient alternative based on object proposals, where two different priors for generating object proposals are presented and evaluated. The object proposal based solution is shown to significantly improve the speed of matching and spatio-temporal search without compromising accuracy. Chapter 5 is devoted to discovering representative objects from videos for summarization and search, instead

of following keyframe based paradigms. Here a new formulation for sparse reconstruction based representative selection is proposed and validated on both synthetic and real datasets. In each chapter, direct comparisons against state-of-the-art methods are presented. Finally, Chapter 6 concludes the thesis with a discussion on future directions.

Chapter 2

Background and Related Work

In this chapter, we set the stage for the rest of this thesis by providing further background and related work in video object search and discovery.

2.1 Object Instance Search in Large-scale Videos

Figure 2.1 shows the organization of the topics related to object instance search in large-scale videos.

2.1.1 Object Representation and Matching

The first question to be addressed is the object representation. Since different object representations directly affect the choice and performance of the matching scheme, we discuss them together below.

For object instance search, usually the target object only occupies a small region in a video frame. Hence we need to evaluate the similarity between the query object and each sub-region in each video frame. Without loss of generality, we focus on

Figure 2.1: Organization of topics covered in Section 2.1.

bounding box size sub-images instead of other region representations, such as super pixels and object contours.

2.1.1.1 Global vs. Local Descriptors and Matching

Global features are window-based, which summarize the appearance of the whole object or sub-image as a single descriptor. Examples include GIST [10] and features from the top layers of deep convolutional neural networks (CNNs) [11]. Recently CNN features have been shown to achieve state-of-the-art performance for many computer vision tasks such as object detection [51], image retrieval [55] and object instance search [21]. Global representations naturally lead to matching schemes based on direct comparisons of these descriptors. However, partial occlusions or large viewpoint changes can easily corrupt the global representation taken from the window.

Alternatively, an object or a sub-image can be represented based on local features [8, 9], which are popular for matching object instances [21–23]. Local features have been shown to be robust to transformations and appearance variations. The locality makes them more robust in partial occlusions and clutters compared to global

features. Typically matching based on local features depends on finding putative correspondences, which is followed by a check of geometric consistency. However, matching sets of local features based on a linear scan approach of finding correspondences is computationally more expensive than matching pairs of global features.

In fact, even when a single global feature is considered per window, there are $m^2 \times n^2$ possible windows in an $m \times n$ frame, which is further scaled by the number of frames in a large video dataset. Various approximate nearest neighbor search methods can be used to speed up the matching, such as FLANN [56], Product Quantization [57], Locality Sensitive Hashing (LSH) [58], and unsupervised similarity-preserving binary encoding approaches including Iterative Quantization (ITQ) [19], Spectral Hashing (SH) [20], Anchor Graph Hashing (AGH) [18], etc.

2.1.1.2 Hybrid Descriptors and Matching: Bag-of-Words (BoW) and High-Order Representations

The popular Bag-of-Words (BoW) model [12] can be seen as a hybrid approach, which produces a global representation (i.e., a single sparse vector of fixed dimension) from local descriptors. It first build a visual vocabulary by finding a few prototypical features. The high dimensional local descriptors are then quantized into discrete *visual words*. Therefore, instead of matching local descriptors in the raw vector space, we can now achieve fast matching easily by looking up an inverted file. The global representation is computed as a sparse vector denoting the frequency of each visual word. The key advantage of using BoW to representing a window over other global descriptors is that once the local features of an image are obtained and quantized into *visual words*, the BoW representation of any given window in the image can be easily derived from these visual words. On the contrary, other global descriptors of one window usually cannot be *reused* by another window (even if they partially overlap), but has to be calculated from scratch. When the number of potential windows to be evaluated is large, this can be costly. Although weak global features such as a

first-order channel feature computed as a sum of pixels in a fixed rectangular region in a single channel can be translation invariant (i.e., can be evaluated only once on the entire image), they may not offer good performance without the help of classifiers.

Recently, approaches based on higher order statistics [14–17] have shown to achieve better search performance than the classical BoW model and its variants [13]. Furthermore, by decomposing the global appearance model and similarity measure of Vector of Locally Aggregated Descriptors (VLAD) and Fisher vector, the method proposed in [21] permits the application of much larger vocabularies in these high order descriptors, which were earlier prohibited by the amount of processing memory it requires. This results in a substantial improvement of performance in generic object instance search in images with a single example. While the research on search in this thesis employs the simple quantized SIFT descriptors [8], the framework does not forbid the use of decomposed VLAD and Fisher vectors [21].

2.1.1.3 Spatial Context for BoW Matching

Although the bag-of-visual-words (BoW) scheme [12, 25, 59–62] has been widely adopted for object matching, quantizing high-dimensional descriptors into visual words can cause information loss; besides, the optimal formation of a visual vocabulary is unclear. One effective approach to mitigate the quantization error is to consider spatial context instead of an individual point, which is less restrictive than geometric verification such as random sample consensus (RANSAC) [25, 63]. For instance, to incorporate the spatial context to enhance the discriminative power of visual words, co-occurring visual words within a constrained spatial distance can be bundled into a visual phrase [64–66] or feature group [67] as the basic unit for object matching. In [26], each local feature is combined with its k spatial nearest neighbors to generate a feature group. In [68], each image is partitioned into non-overlapping grid cells which bundle the local features into grid features. However, unlike the whole-image retrieval problem, target object instances may appear at any scale, while such feature

groups are not scale invariant and not capable of handling the various objects without a *priori* knowledge. Also it is not a trivial problem to select the optimal k or grid size. Moreover, it is inconvenient if the user wants to change the scale of feature group when the dataset needs to be re-indexed. To address this problem, the Randomized Visual Phrases (RVPs) [22] is proposed to consider spatial context in varying shapes and sizes, and thereby provides a robust partial matching.

2.1.1.4 Object Matching in Images vs. in Videos

Object instance search in images has received much interest in recent years [2, 12, 21, 23] [24–26][27, 28]. The seminal work [12] first recasts object search as text retrieval. It introduces inverted file index on quantized local descriptors and the Bag-of-Words (BoW) model to make fast matching possible on large image datasets. Since then, it has spawn much work on instance search in large image datasets as well as in video shots. For instance, TRECVID instance search challenge aims to locate video shots that most likely contain a query topic. Most earlier systems first down-sample videos to keyframes, and consider the task as an image retrieval or instance search problem to find keyframes that most likely contain the query topics [29–32]. Recently, aggregation over several keyframes per shot into a single global signature has been shown to boost the search performance, such as representing each shot by the average of the BoW vectors of its multiple keyframes [69–71]. However, global signatures may not be reliable when objects are partially occluded or the scene is cluttered.

In comparison, objects in videos are spatio-temporally coherent. Therefore, the main drawback of directly applying instance search approaches for images to videos is the loss of spatio-temporal coherency across video frames, which may result in sub-optimal performance, as will be shown in the experiments later in Chapter 3 and Chapter 4. Existing approaches focus on the use of temporal continuity to improve the quality of feature descriptors [12, 72, 73]. The regions detected in each frame within a shot are first tracked and then aggregated to describe this scene region

throughout the track. In comparison, our work enforces spatio-temporal consistency at the trajectory level instead of feature descriptor level. Also we do not rely on tracking to find correspondences (between bounding boxes in our case) across frames.

Nevertheless, Chapter 3 and Chapter 4 in this thesis do benefit from the design of existing image-based instance search systems in that we adopt the feature descriptor quantization and inverted file index to enable efficient frame-wise filtering and matching [12].

2.1.2 Object Localization

2.1.2.1 Object Localization in Images

For instance search in images, locations of object instances are usually obtained by post-processing through geometric verification, such as random sample consensus (RANSAC) [25]. However RANSAC requires sufficient number of matched points to reliably estimate a transformation. Spatial random partition is proposed in [64] to discover and locate visual common objects, which is later extended to search [2]. Recently, localized search has been shown to lead to better search performance compared to global descriptor matching, especially when target objects are small or scenes are cluttered [21].

For object detection in images, the classical sliding window approach requires evaluation in a large number of windows, which is prohibitively expensive. Because the size of the target object is unknown, different sizes and aspect ratios have to be checked at each possible location. For instance, even an image of a moderate size 360×480 would result in over 29 billion possible bounding boxes. This makes the sliding window approach unsuitable for large-scale applications.

One strategy to reduce the search time in windows is to use context to prioritize search and quickly eliminate those windows that appear unlikely. For instance,

Fleuret and Geman [74] design a sequential testing which is coarse-to-fine in both in the exploration of poses and the representation of objects. Viola and Jones [75] propose a cascade structure which allows background regions to be quickly discarded while spending more time on promising face-like regions. In addition, branch-and-bound based efficient subimage retrieval (ESR) [24] and efficient subwindow search (ESS) [76] have been proposed to find the subimage with maximal similarity to the query. The key is to derive a bound on the classifier output, which allows the use of a priority queue to dramatically reduce the search time on any image. However, branch-and-bound does not consider the geometric constraints of the object model, thus can decrease localization accuracy when multiple instances of the same object are close by. It may also mistakenly detect locations where the object does not appear. Global scene context has also been used to obtain localization priors. For instance, Torralba *et al.* [77] show that global scene context can assist saliency computation to guide eye movements and attention toward locations likely to contain the target objects. Similar works that leverage global context include [78, 79]. The intuition is that whether an object is present in an image is biased by the global scene (e.g., a car is more likely to be on the street than in a kitchen); besides, the global structures largely determine the location and scale of a present object.

Recently, category-independent object proposals have become a popular alternative to sliding windows, which enhance the efficiency of object detection by only evaluating bounding box locations that are likely to contain an object, which is usually a few thousand per image [35–40]. The use of object proposals is also explored in Chapter 4 for object matching and localization in videos, and in Chapter 5 for object localization before representative selection. To further reduce the number of proposal windows to be evaluated, in [80], Gonzalez-Garcia *et al.* propose an active search strategy to sequentially choose the next window based on previously observed windows. They explore the use of context learned from the training set (i.e., the statistical relationship between the appearance of a window and its location relative to the object) to allow the search to jump to distant windows in the image. For instance,

if a window appears like part of the sky, it suggests that cars or motorcycles might be far below. The context is extracted efficiently using Random Forests. In addition, according to the score of the classifier, search can be steered towards promising areas surrounding a high-score window, and away from areas around low-score windows. Consequently, the proposed method can find the objects by evaluating only a few hundred windows, which is much fewer than the number produced by object proposal generators.

2.1.2.2 Spatio-Temporal Object Localization in Videos

The common limitation of the above approaches is that the localization is performed independently on individual images or frames. Therefore, when directly applied to videos, they do not enforce spatio-temporal consistency across video frames. Inspired by 2D object proposals in static images, several approaches have been proposed that extend recent 2D object proposal methods to generate spatio-temporal video tube proposals for action detection and localization [81–83].

Bergh *et al.* [81] extend the objectness approach [36] to video. Jain *et al.* [83] extend the selective search [37] and relies on the estimation of the dominant motion and an independent motion evidence map to compute initial supervoxels, which led to tubelets from motion. Oneata *et al.* [82] generalize randomized Prim [84] to spatio-temporal proposals, using optical flow as a cue to derive similarity for both supervoxel and proposal generation. These approaches are more effective when the object is in motion with respect to the background. However, most of these 3D generalizations are based on a super-pixel segmentation pre-processing step [36, 37, 81–85], which is computationally demanding for proposal generation, especially for large scale video processing.

To alleviate the computational expensive segmentation step of existing proposals, Gemert *et al.* [86] propose to bypass the segmentations completely by generating

proposals directly from the dense trajectories used to represent videos during classification. The proposed Action localization Proposals from dense Trajectories (APT) approach uses an efficient clustering algorithm to deal with the high number of trajectories in a video.

Besides the above video proposal methods, Max-Path search [33, 87] has also been proposed for spatio-temporal event detection. It uses dynamic programming to search for the optimal path over the 3D trellis connecting bounding boxes that are scored prior to the spatio-temporal search. The key challenge to extend Max-Path to the problem of object instance search in videos, however, is the computational cost of scoring all considered bounding boxes in order to construct such a trellis. A sliding-window based scoring of bounding boxes is viable for detection problems, as real-time processing is usually good enough [33]. But it is infeasible for many practical search systems.

In Chapter 3, we tackle this challenge by leveraging a randomized approach to quickly obtain pixel-wise confidence scores, which in combination with integral images [34] permits the application of Max-Path search to large video datasets. Some preliminary results on a 5.5-hour consumer video dataset can be found in [88]. Our previous work [89] combines Hough Voting and Max-Path search to locate object centers in a video sequence. However, it can only produce trajectories of the object center rather than the object itself. Moreover, it is not efficient for large scale videos. Later in Chapter 4, we present a more efficient alternative based on object proposals to further speed up spatio-temporal search and localization of object instances with competitive performance.

2.1.3 Object Instance Search vs. Object Detection

Although both require object localization, object instance search in large video volumes should be differentiated from the problem of large scale object detection [51, 90,

91]. The fundamental differences between object detection and object instance search are that object detection usually deals with object categories rather than a particular object; therefore it requires training of classifiers. When talking about large scale, large scale search refers to the scalability with respect to the database size, while large scale detection refers to the scalability with regard to the number of object categories. In other words, object instance search concerns about fast localization of a given object instance in a large number of database images or frames, while large scale object detection requires fast localization of a large number of object categories per image/frame, whereas the dataset size is usually not a concern. Due to the different aims, object instance search can benefit from efficient indexing of the database for fast matching [12], while large scale detection can be sped up by efficient locality sensitive-hashing to replace linear convolution with ordinal convolution [91], or by sharing “deep features” across categories and limiting the regions to be evaluated to those selected region proposals [51, 90]. Second, accurate localization of object detection heavily relies on the availability of sufficient training images with bounding box annotations [51], while object instance search needs only one example. Although [90] proposes to address this challenge by learning the difference between classification and detection, then using this knowledge to transfer classifiers without bounding box annotations to detectors, they still require training data for these classifiers. On the contrary, with only one example of an object instance, the proposed object instance search approaches in this thesis are able to locate its occurrences in videos by direct local interest point matching in combination with spatio-temporal context.

In addition, for real-time detection in videos, slow feature extractors can be a killer, since features need to be extracted for each video frame or keyframe on the fly. Luckily, weak features can somehow be compensated by a well-trained classifier, which can be elaborate since the training is done off-line. For instance, the boosted object detection system by Viola and Jones uses simple and efficient Haar features [75]. On the contrary, object instance search in videos can afford more computationally costly yet higher quality features, as the video dataset are usually preprocessed and indexed,

therefore only the feature of the query needs to be extracted on the fly. However, the matching has to be even more efficient than real-time detection if the dataset size is large.

2.2 Discovering Representative Objects from Videos

Discovering the important *nouns* (i.e., representative objects) from videos can spawn a wide range of applications, and the most relevant are works in object-driven video summarization. Besides, since we propose to discover representative objects from frame-wise object proposals, and formulate the problem of representative object discovery as a sparse dictionary selection problem with a locality prior matrix that considers the local affinity structure of the data, we will also discuss below related works in representative selection and local structure learning.

2.2.1 Object-driven Video Summarization and Object Discovery

Visual summaries of a video can take many forms such as keyframes [50, 92, 93], skims [94–97], montages [98] and dynamic synopses [99, 100]. Recently there has been increasing interest in object-driven approaches to produce the above forms of summaries [50, 95, 99, 101–106], which is closely related to our work. Some object-driven video summarization methods require prior knowledge. For instance, in [101, 107], frame-level labels are required to help identify the object of interest. By learning to predict important object regions in egocentric videos using egocentric and saliency cues, concise visual summaries for egocentric videos can be produced driven by those regions [50]. In [95] object-like windows (i.e., object proposals) are taken from each frame as the initial pool of objects, based on which relationships between sub-events are discovered. However, these objects only act as an intermediate to help select sub-shots that construct a story. The ultimate goal is not to summarize videos into key

objects and it is not fully unsupervised. Although methods in [108, 109] can discover objects via topical models in an unsupervised way, it relies on image segments instead of object proposals. Also it targets at grouping the segments instead of selecting representative ones. Meanwhile, although Cho *et al.* [102] utilize object proposals to address unsupervised discovery and localization of primary objects with multiple object classes, it targets for noisy image collections instead of a single video clip. Moreover, it only discovers a single object instance per image.

2.2.2 Representative Selection

Representative selection can be roughly categorized into clustering based methods and subspace learning based methods. As for clustering based methods, K-medoids algorithm is a representative one [41], which selects K clustering centroids as representatives to minimize within cluster distances. Instead of using one centroid to represent each sample, Elhamifar *et al.* improve K-medoids clustering, so that each sample is able to be represented by multiple centroids [54]. Representatives are also centroid-like in the methods of affinity propagation [42, 43] and density peak search [44]. In [44], cluster centers (i.e., representatives) are identified based on the assumptions that cluster centers have a higher density than their neighbors and that they are at a relatively large distance from points with higher densities.

There has also been recent interest in applying linear subspace learning to find representatives from data, e.g., dictionary learning in [45, 46] and dictionary selection in [5, 6, 47–49]. These methods usually require that each sample can be linearly expressed by representatives at a low reconstruction error. To filter outliers, Elhamifar *et al.* [6] rank the representatives based on the norms of the rows in the coefficient matrix and only pick the top ones as final representatives. Another recent work [110] can find a subset of the source set to efficiently describe the target set, given pairwise dissimilarities between two sets. However, it does not consider the prior weight of

the data samples in their applications. Relying on dictionary selection, our method also involves data reconstruction. But unlike existing schemes, we introduce a locality prior when constrain the reconstruction, which can be learned by the k -nearest neighbors of each sample. Therefore, our method prefers a local reconstruction of each sample, which shows robustness to noise.

2.2.3 Local Structure Learning

Local geometric structure of data plays a vital role in data understanding and analysis. Most approaches are based on the analysis of k -nearest neighborhoods. To model the connectivity structure of data, building neighborhood graph may enjoy great popularity [111, 112]. However, the method of local structure learning relevant to ours is local linear reconstruction. For example, when mapping high-dimensional data into a low-dimensional space (LLE), locally linear embedding (LLE) leverages local linear reconstruction within neighborhoods to build the projection matrix [113]. The idea of locally linear reconstruction is also employed in locality-constrained linear coding (LLC) [114] and locality-constrained affine subspace coding (LASC) [115], where the locally linear constraints can guarantee similar images having similar codes while recoding image features.

Chapter 3

Spatio-temporal Search of Object Instances in Videos

3.1 Overview

Object instance search concerns about searching for and locating instances of a particular object (e.g., the Starbucks logo, instead of object categories such as cars and dogs) in a large video corpus. Inspired by the bounding box search that locates object instances in images [21, 24], in this chapter, we propose to formulate object instance search in videos as a problem of spatio-temporal trajectory discovery, where each trajectory is a temporal series of bounding boxes that encompass the object of interest across frames¹. The goal is therefore to find the top- K spatio-temporal trajectories that locate and track the object instances in videos. As mentioned in Section 1.3.1.2, formulating the problem of object instance search in videos as a problem of spatio-temporal trajectory search offers two benefits. First, compared with conventional frame-based methods, the proposed trajectory-based formulation can explore the spatio-temporal coherency across video frames to improve matching accuracy by recovering missed detections due to object appearance variations and reducing false

¹A substantial portion of this chapter has appeared in [116] © 2016 IEEE.

alarms caused by background clutters. Second, trajectories offer an alternative to frames for measuring object occurrences and the search performance, which can be critical for applications where the recall of object trajectories is more important than the recall of frames that contain the object (refer to Figure 1.5).

As an object instance can appear in any frame and at any location, the number of trajectory candidates is huge. To efficiently find the top- K trajectories, we build upon the recent dynamic programming approach of Max-Path search [33] to locate the top- K trajectories efficiently in large video corpus, with a complexity linear to the video volume size. However, Max-Path search requires a 3D trellis structure, which connects per frame bounding boxes that are scored prior to the spatio-temporal search. Such a trellis is usually built per frame by a sliding-window based scoring of all considered bounding boxes across different scales and aspect ratios, which is computationally expensive. While this may be less of a concern for detection problems, as real-time processing is usually good enough, it is prohibitive when searching tens of hours of videos. This bottleneck is the key challenge in adapting Max-Path search for detection problems to object instance search in large video datasets. We address this challenge by leveraging a randomized scheme to efficiently generate pixel-wise confidence scores. It is achieved by averaging the matching score of each pixel over a pool of randomly generated image patches, thus avoiding a sliding-window scanning of all pixels, which is time consuming. Consequently, the score of any bounding box can be calculated as the summation of pixel scores inside the bounding box, which is only $O(1)$ using integral images [34].

In addition, we have produced a new video dataset to facilitate future research on object instance search in videos, called NTU Video-Object-Instance (NTU-VOI)². It consists of 146 clips captured by mobile cameras or downloaded from YouTube, with frame-wise bounding box annotations of object instances. In total 33,018 frames are annotated. More information on this dataset can be found in Section 3.4.1.

²<https://sites.google.com/site/jingjingmengsite/research/ntu-voi/data>

To our best knowledge, this is the first video dataset available that provides per frame bounding box annotations of object instances. The YouTube-Objects dataset is another dataset of annotated object videos from YouTube [117]. However it annotates object classes instead of specific object instances. Also, for each class, the bounding box annotations are only provided for one frame per shot on 100-290 different shots, rather than for all ground truth frames. Recent Stanford I2V dataset [1] is a large-scale video dataset with annotated ground-truth video clips and precise temporal segments, but the bounding box locations in ground truth frames are not provided.

The proposed approach is evaluated on a 73-hour video dataset (Figure 3.2). It is shown that by enforcing spatio-temporal consistency via trajectory search, the proposed approach improves the performance of object instance search and localization when compared with existing search methods that treat frames independently [2, 3]. It also achieves better localization accuracy when compared with a state-of-the-art Kernelized Correlation Filters (KCF) tracker [4]. Its effectiveness in incorporating spatio-temporal cues into search is also demonstrated by its ability to find object occurrences in cluttered scenes regardless of large appearance variations due to motion blur, occlusions, and changes in viewpoint, illumination and color. Its efficiency is demonstrated by its ability to find the top 100 trajectories of an object instance in the 73-hour dataset in about 200 seconds.

3.2 Problem Formulation

Consider the entire video database as a long video sequence $\mathcal{V} = \{\mathcal{F}_1, \mathcal{F}_2 \dots \mathcal{F}_n\}$, where $\mathcal{F}_i \in \mathcal{V}$ is the i^{th} frame with a size of $w \times h$. Given a query object Q , our goal is to find in $\mathcal{V}$ a number of spatio-temporal trajectories of bounding boxes such that each trajectory captures one instance of Q in $\mathcal{V}$.

Denote a trajectory $T = \{B_i\}_{i=i_1}^{i_k}$ as a temporal sequence of spatial bounding boxes that locate the object instance in each frame $B_i = \{x_i, y_i, s_i, a_i, t_i\}$, where x_i

and y_i are the center coordinates, s_i and a_i are the scale and aspect ratio of bounding box B_i , and t_i is its temporal index. Then each trajectory $T = \{B_i\}_{i=i_1}^{i_k}$ should satisfy the following constraints:

1) the smoothness constraint of the trajectory:

$$x_i - k \leq x_{i+1} \leq x_i + k, y_i - k \leq y_{i+1} \leq y_i + k, t_{i+1} = t_i + 1 \quad (3.1)$$

where $(2k+1) \times (2k+1)$ is the valid neighborhood size that the center of one bounding box can move to in the next frame;

2) the smoothness constraint of the bounding box scale:

$$\frac{s_i}{s_{i+1}} \in \{1 - r, 1, 1 + r\} \quad (3.2)$$

where $0 < r < 1$ and a bounding box is only allowed to change to one of these three relative scales in the next frame.

These two constraints ensure the spatio-temporal trajectory can track and accurately locate the object in the video, despite the camera motion, object motion and scale variations. It is also worth noting that for each trajectory of bounding boxes, it can start and end at any temporal or spatial location within the 3D video space, as long as the temporal index of the start point is before that of the end point.

Assume we can score any bounding box B_i in a given video frame, and denote $s(B_i)$ as its discriminative score, i.e., a positive score of $s(B_i)$ shows positive evidence of the object instance's presence while a negative score of $s(B_i)$ shows negative evidence. Given a trajectory T , its confidence score $s(T)$ is calculated as the sum of the discriminative scores of the bounding boxes along T , i.e.

$$s(T) = \sum_{B_i \in T} s(B_i) \quad (3.3)$$

Similarly, a large positive score $s(T)$ indicates that the object is highly likely to appear along T , and a negative score indicates otherwise. Given a video database $\mathcal{V}$

and a query object Q , our objective is to find the top K trajectories that capture K instances of the object query. Our top- K trajectory search is then formulated as the following optimization problem:

$$\begin{aligned} \mathbf{T}^* &= \arg \max_{\mathbf{T} \subseteq \mathcal{T}} \sum_{T \in \mathbf{T}} s(T) \\ s.t. & \forall T_i, T_j \in \mathbf{T}, T_i \cap T_j = \emptyset \\ & |\mathbf{T}| = K. \end{aligned} \tag{3.4}$$

Here $\mathcal{T}$ indicates the set of all possible trajectories in $\mathcal{V}$, while $\mathbf{T}$ is a subset of K trajectories we are searching for. Let us assume that trajectories do not overlap with each other, which simplifies the search and is also a reasonable assumption for object instance search. As the trajectories do not overlap with each other, the top- K search will boil down to the search of the individual best trajectories:

$$T^* = \arg \max_{T \in \mathcal{T}} s(T) \tag{3.5}$$

The search is repeated K times by removing the current best path from $\mathcal{T}$ once discovered.

However, the search of optimal spatio-temporal trajectory is still a non-trivial problem given the huge number of trajectory candidates in $\mathcal{T}$, which can start and end at any spatio-temporal location in the video and also carry bounding boxes of arbitrary sizes. The search space of all possible trajectories is $O(whn(2k+1)^{2n})$, where whn is the size of the 3D video volume, and k is the smoothness constraints of T , as defined in the previous section. In the following section, we will explain how to perform the fast spatio-temporal trajectory search.

3.2.1 Fast Spatio-Temporal Trajectory Search

Exhaustively searching for the globally optimal trajectory $T^* \in \mathcal{V}$ (Eq.3.4) is infeasible due to the exponential complexity of the problem. To reduce the search complexity,

we propose to employ the Max-Path search approach proposed in [33, 87], which uses dynamic programming to obtain the globally optimal trajectory.

Here we briefly explain the procedure of Max-Path search. Max-Path search runs on a 3D trellis structure that connects per frame bounding boxes that are scored prior to the spatio-temporal search. The Max-Path search starts from all possible bounding boxes in the first frame, where each bounding box initiates a trajectory and tries to propagate it from the current frame to the next. Given the smoothness constraint of the trajectory, each bounding box will only search for bounding boxes in its neighborhood in the next frame. Whether the trajectory can continue or cannot depends on the accumulated score of the trajectory $s(T)$. If $s(T)$ is positive, then the trajectory continues to grow; otherwise, a new path will be initiated from the current bounding box. During the search, each bounding box $\mathcal{B}$ carries the positive score $s(T)$, which is the best score of all possible trajectories ended at $\mathcal{B}$, and passes it to the next bounding box in the trajectory. Once the search reaches the final frame, we can find the best trajectory by the highest score $s(T)$. Multiple trajectories can be found by removing the current best trajectory at the end of each round before searching for the next best trajectory. As proven in [33, 87], such a dynamic programming strategy can guarantee to find the best trajectory among all candidates, with complexity linear to the video volume size ($O(whn(2k + 1)^2)$).

Despite the successes of applying the Max-Path search to the action detection problem, its extension to object instance search is however not straightforward. As we are searching a large video corpus, it will be computationally expensive to use a sliding-window approach to obtain the bounding box score $s(\mathcal{B})$ at all locations and scales [87]. This is less of a concern for detection problems, where real-time processing is usually good enough. Particularly, note that with a single example, we do not train a discriminative classifier, but directly match the query against the dataset. Exhaustively matching the query at $w \times h$ locations and varying scales in each frame is prohibitive for large video datasets.

In the following section, we will explain how to efficiently obtain the confidence scores of any considered bounding boxes in order to extend Max-Path search to video instance search.

3.2.2 Efficient Confidence Map Generation

We observe that efficient calculation of any bounding box score requires an additive scoring scheme. In other words, if the score of any bounding box can be calculated as the sum of the scores of its containing pixels or interest points [27], we can quickly obtain the score of any bounding box using integral images [34]. Note that image search methods based on global descriptors cannot serve this purpose, as they do not provide matching scores at such a fine level. Hence, we propose to use Randomized Visual Phrases (RVP) to efficiently obtain pixel-wise confidence scores in each frame [22]. Moreover, scoring on the resulting confidence maps from RVP satisfies the additive property.

Specifically, a given query is matched with a collection of overlapping random patches in each database image, generated by partitioning the database image using a set of random templates. Each random patch bundles a collection of visual words and is called a visual phrase. We independently calculate the matching score between each RVP and the query object, and treat it as the voting weight of the corresponding patch. The confidence score of a pixel p can then be calculated as the average of the voting scores of the RVPs that cover this pixel:

$$\bar{s}(p) = \frac{1}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} s(P_{p,k}) \quad (3.6)$$

where $\mathcal{K}$ is the total number of RVPs (i.e. patches) covering pixel p , which is equal to the partition rounds, and $s(P_{p,k})$ is the matching score of a patch. In our implementation, histogram intersection ($HI(\cdot)$) is used as the similarity measure to compute $s(P_{p,k})$. Given the histogram representation of the query, h_Q , and that of a patch,

h_P , which describe the respective visual word frequency, the histogram intersection $HI(.) = \sum_v \min(h_Q^v, h_P^v)$.

This randomized approach offers two benefits. First, it results in more accurate matching, as spatial context of varying sizes has been taken into consideration. Second, it generates pixel-wise confidence scores efficiently, as only a few patches in each image are evaluated and aggregated to produce the confidence map. However, the original RVP relies on a heuristic segmentation coefficient α to rank images and locate target objects in each image [22]. On the contrary, in this work we use Max-Path search to jointly evaluate confidence maps across video frames (Section 3.2.1), instead of segmenting individual frames independently. Our approach not only removes the dependency of search and localization on the segmentation coefficient α , but also boosts the search performance by leveraging the spatio-temporal cues (Section 3.5.1.3).

As the resulting confidence maps are not discriminative for Max-Path search [87]. Similar to [87], we add a negative threshold to each confidence map to introduce negative values. To accommodate object appearance variations in different video, instead of a fixed threshold [87], we set the threshold adaptively to be proportional to the average pixel-wise confidence score of each video (excluding zero confidence maps). If we denote the total number of non-zero confidence maps in video V_i as $\mathcal{N}_{NZ}$, the threshold is calculated as:

$$MP_{thres} = \beta \frac{\sum_{\mathcal{F} \in V_i} \overline{s_{\mathcal{F}}}}{\mathcal{N}_{NZ}} \quad (3.7)$$

and $\overline{s_{\mathcal{F}}}$ is the average pixel-wise confidence score of frame $\mathcal{F}$, defined as

$$\overline{s_{\mathcal{F}}} = \frac{\sum_{p \in \mathcal{F}} \bar{s}(p)}{|\mathcal{F}|} \quad (3.8)$$

where $|\mathcal{F}|$ is the total number of non-zero pixels in frame $\mathcal{F}$, and $\bar{s}(p)$ indicates the confidence score of pixel $p \in \mathcal{F}$ calculated by Eq. 3.6. The final confidence score of pixel $p \in \mathcal{F}$ is computed as:

$$s(p) = \bar{s}(p) + MP_{thres} \quad (3.9)$$

How the negative coefficient β affects the search performance will be examined in Section 3.5.1.3.

Once the discriminative confidence map with pixel-wise confidence score is obtained, given any bounding box $\mathcal{B}$, its confidence score $s(\mathcal{B})$ can be calculated as the summation of the confidence scores of pixels inside $\mathcal{B}$. This is an $O(1)$ operation with the help of integral images.

3.2.3 Coarse-to-Fine Search

To further improve efficiency, the search is performed in two steps. First, a given query is matched against coarsely sampled keyframes to fast rank videos based on the best matched keyframes. Only for those top ranked videos, the search and localization will be refined using Max-Path to find the globally optimal object trajectories, which is more computationally expensive. In fact, any image search methods besides RVP can be used to generate the initial ranking of all videos, such as Scalable compressed Fisher Vectors (SCFV) tested in our experiments (Section 3.5.1.1). We choose to use RVP because it can both produce the initial video ranking and generate the confidence maps for Max-Path in a single run.

Figure 3.1 summarizes our propose approach for spatio-temporal trajectory discovery.

3.3 Evaluation Metrics

We use the widely adopted average precision (AP) and mean average precision (mAP) measures to evaluate the search performance. Given a ranked list of R retrieved

Figure 3.1: Overview of proposed approach to find object trajectories in videos. Video keyframes are first matched with the query to produce confidence maps with pixel-wise matching score (2nd row, Section 3.2.2). Next low confidence videos (the gray cuboids in the 3rd row) are filtered by ranking videos based on the keyframe scores (Section 3.2.3). Max-Path search then finds the globally optimal object trajectories in top ranked videos and re-rank them according to the trajectory scores (Section 3.2.1). Ground truth locations are highlighted in dashed blue boxes in the found trajectories (4th row). Best viewed in color and magnification. © 2016 IEEE

results, the AP is calculated as the area under the precision-recall curve:

$$AP = \frac{\sum_{r=1}^R (Prec(r) \times rel(r))}{\#Ground\ Truth} \quad (3.10)$$

where $Prec(r)$ is the precision at cut-off r in the ranked list, and $rel(r)$ is an indicator function which equals 1 if the r^{th} result is relevant (positive), 0 if otherwise. mAP is the mean average precision over all queries.

3.3.1 Video mAP

Video mAP evaluates how effective a search system finds the ground truth clips that contain the query object, without considering the exact locations of the object within each retrieved clip. Similar to image retrieval systems [61], the video average precision (video AP) is calculated as the area under the precision-recall curve given a ranked list of videos, and video mAP is defined as the mean of video APs over all queries. Precision is the number of returned positive videos relative to the total number of returned videos. Recall is the number of returned positive videos relative to the total number of positives in the dataset. A video is considered positive if it indeed contains the query object, and negative if otherwise.

3.3.2 Trajectory mAP

To evaluate the search quality more precisely, the trajectory mAP metric is also introduced. Different from video mAP, trajectory AP and mAP are calculated based on a ranked list of object trajectories instead of videos. Precision is defined as the number of returned positive trajectories relative to the total number of returned trajectories. Recall is the number of returned positive trajectories relative to the total number of positives in the dataset. As we assume that the trajectories are

non-overlapping (Section 3.2), only one best trajectory for each video is found and ranked.

However, to obtain the *trajectory-wise* precision-recall curves, we need a criterion to judge whether a returned trajectory is *positive*. It is straightforward for images and videos, but not so obvious for trajectories. Therefore, we introduce Path IoU [33] to measure this relevance, which is the overlapped volume of two paths divided by the union volume of the two. Formally,

$$PIoU(T_i, T_j) = \frac{\sum_{t=1}^N \mathcal{B}_i^{(t)} \cap \mathcal{B}_j^{(t)}}{\sum_{t=1}^N \mathcal{B}_i^{(t)} \cup \mathcal{B}_j^{(t)}} \quad (3.11)$$

where N is the total number of frames in the sequence, $\mathcal{B}_i^{(t)}$ is the bounding box of trajectory T_i in frame t , and $\mathcal{B}_j^{(t)}$ is the bounding box of trajectory T_j at the same temporal location. Denote the ground truth trajectory as G , a retrieved trajectory T^* is positive when $PIoU(T^*, G) > \delta_p$, and negative otherwise. δ_p is the threshold.

We note that the trajectory-based metric is more challenging and more precise for evaluating the relevance of retrieved trajectories to the ground truth trajectory. This score approaches 1, i.e. 100%, when a returned trajectory fully overlaps with the ground-truth trajectory, and will be 0 when the two have no overlaps.

3.4 Experimental Setup

3.4.1 Dataset

We evaluate the proposed approach on a 73-hour video dataset, comprising 26-minute ground truth videos and a subset of Stanford I2V dataset [1] as distractors.

The ground truth set, called NTU Video-Object-Instance dataset (NTU-VOI), consists of 146 video clips captured by mobile cameras or downloaded from YouTube,

Figure 3.2: Example videos from the 73-hour video dataset (only one keyframe is shown for each video). Videos with annotated object locations are from NTU-VOI. The remaining are distractors from Stanford I2V [1]. Three example queries are given on the left: KittyB, NTU and Pokka (Table 3.1). Their ground truth locations are annotated by magenta, blue and red bounding boxes, respectively. NTU-VOI dataset comprises diverse scenes, most of which are cluttered with small target objects. This makes NTU-VOI a challenging dataset for object instance search and localization. Best viewed in color and magnification. © 2016 IEEE

with per-frame bounding box annotations of target object locations (i.e. the 10 queries in Table 3.1). Each clip consists of a single shot and contains up to two ground truth trajectories. The average duration of these clips is 10.54 seconds. The annotations are obtained by four trained student helpers using the LabelMe Video tool [118]. In total 151 trajectories with a total of 33,018 frames are annotated. As can be seen in Figure 3.2, the VOI dataset covers diverse scenes, most of which are cluttered with the target objects occupying only a small portion of the frame. This makes VOI a challenging dataset for object instance search and localization.

The distractor set consists of all I2V October 2012 newscast videos that have a resolution greater than 800×450 pixels. The average duration of distractor clips is 2.5 minutes.

All 73-hour videos are first resized to a spatial resolution of 800×450 pixels, then sampled uniformly at 1 fps, resulting in 263,180 keyframes. Although advanced

keyframe detection methods [119, 120] can be applied instead to obtain the keyframes, uniform sampling works reasonably well for this dataset.

3.4.2 Queries

Table 3.1 summarizes the statistics of the 10 query objects used in our experiments. Among them, KittyB, KittyG and Plane are 3D objects, while the remaining 7 are 2D objects. For a fair evaluation, external images are used as queries instead of objects cropped out from the testing video frames. Specifically, for each 3D object, we take one picture from a single view as the query. For the 2D objects, we use Google search with the text object name to obtain the query image. The query images are shown in Table 3.1 as well. In all our experiments, only a single query image is used to search an object.

3.5 Experimental Results

3.5.1 Video mAP

As video mAP does not concern about the exact object locations within retrieved clips, image search methods can be directly applied to rank videos and calculate video mAP. To this end, two existing visual search systems are evaluated in comparison with the proposed trajectory discovery approach, which is called Randomized Visual Phrases with Max-Path search (RVP-MP). The two baseline methods to be compared with are: (1) Scalable compressed Fisher vector with RANSAC (SCFV-RANSAC), which is based on global descriptor matching followed by RANSAC geometric verification [3], and (2) the baseline RVP (RVP-Baseline) [2], which is based on bundled local descriptors. It is worth mentioning that both baseline systems rank videos based on the score of the best matched keyframe, while the proposed RVP-MP

Table 3.1: Statistics of Ground Truth (GT) Object Instance Trajectories in VOI Dataset © 2016 IEEE

Query Name	Illustration	No. of GT Trajectories	Ave. GT Size (Width x Height)
100Plus		17	98 x 105
Ferrari		14	61 x 71
KittyB		21	198 x 206
KittyG		12	233 x 243
Maggi		16	112 x 72
NTU		15	71 x 87
PKU		8	97 x 102
Plane		9	260 x 157
Pokka		12	91 x 51
Starbucks		27	92 x 97

ranks videos by trajectory scores. For each query, video AP is computed based on the top 100 retrieved videos. And the APs are averaged over the 10 queries to get the video mAP. In the following, we first explain the implementation of the two baseline systems and the proposed RVP-MP, then the results are presented.

3.5.1.1 SCFV-RANSAC

Scalable compressed Fisher vector (SCFV) is a state-of-the-art global image descriptor [3], which has been adopted by the MPEG standardization of compact descriptor for visual search (CDVS). In our experiments, the performance of SCFV followed by RANSAC geometric verification is tested using the API from the authors. The dimension of local descriptors is reduced to 32 using PCA, and the number of Gaussian mixture components is set to 512. As in [3], a separate set of images (also from the authors) is used to train the Gaussian mixture model, PCA and correlation weights.

For each video, first the global descriptor SCFVs are generated for all keyframes, which are matched against the SCFV signature of the query to obtain a ranked keyframe list. Then the top 200 keyframes of each video are re-ranked using RANSAC. The best matched keyframe after geometric verification is picked to score and rank database videos.

3.5.1.2 RVP-Baseline

As mentioned before, RVP is a state-of-the-art visual object search method for image datasets. Its good search quality can be attributed to its ability to provide robust matching under varying spatial contexts, thanks to a randomized partition scheme.

In the experiments, interest points are first extracted using Hessian-Affine detectors [121] and represented as SIFT local descriptors [8]. SIFTs are sampled every 10th keyframe to build a vocabulary of 1M words using FLANN [56]. Once all SIFTs

Table 3.2: Video mAP on Top 100 Retrieved Clips © 2016 IEEE

	SCFV-RANSAC [3]	RVP-Baseline [2]				
		α				
		1.0	2.0	3.0	4.0	5.0
mAP	0.379	0.357	0.366	0.364	0.375	0.396

are quantized and indexed into an inverted file for each video, RVP is applied to rank keyframes of each video. We use $HI(\cdot)$ as the matching kernel for RVP and the partition parameters is set to 200 rounds of partitions with 8×4 -sized random templates.

As RVP ranks images based on the scores of segmented salient regions, we test RVP under varied segmentation threshold $\alpha \in \{1.0, 2.0, 3.0, 4.0, 5.0\}$. Table 3.2 summarizes the video mAP of SCFV-RANSAC and that of RVP-Baseline under different α . It is shown that even without geometric verification, RVP-Baseline achieves comparable video mAP as SCFV-RANSAC.

3.5.1.3 RVP-MP

Different from the above two baseline approaches, RVP-MP uses the best trajectory score instead of the best keyframe score as video score to rank database videos. The best trajectory score is calculated as the Max-Path score of each video (i.e. $s(T^*)$, Eq. 3.5).

As mentioned in Section 3.2.2, Max-Path search is only applied on top ranked videos for efficiency. Therefore we first follow the baseline RVP (Section 3.5.1.2, with $\alpha = 5.0$) to obtain a ranked list of database videos based on the best keyframe score. Then for each of the top 100 clips, we run Max-Path search on the resulting confidence maps from RVP to find the best trajectory T^* , and use $s(T^*)$ to re-rank the top 100 clips accordingly. We use the Max-Path algorithm with multi-scale extension and

Figure 3.3: The impact of β on video mAP: our RVP-MP method consistently outperforms the RVP-Baseline [2] and SCFV-RANSAC [3] as β changes from -3 to -0.5 . © 2016 IEEE

a starting height of 60 pixels. The aspect ratio is fixed to be the same as that of the query. The spatial step is set to 10 pixels and the temporal step is synchronized with the keyframe sample rate (i.e. 1 fps). We compute the summations of multi-scale bounding boxes using integral images [34].

The video mAP of RVP-MP is evaluated under varying negative coefficient β (Eq. 3.7). Figure 3.3 illustrates the video mAP of RVP-MP in comparison with RVP-Baseline ($\alpha = 5.0$) and SCFV-RANSAC, as β increases from -3.0 to -0.5 , at an interval of 0.5. It is observed that our RVP-MP consistently outperforms the other two methods regardless of changing β . When $\beta = -2.0$, RVP-MP improves RVP-Baseline by 16.71% and SCFV-RANSAC by 22.17%. It validates that spatio-temporal consistency is beneficial to object instance search in videos.

Table 3.3: Video mAP on top 100 retrieved clips. RVP-MP consistently improves AP on all queries over RVP-Baseline, except for Ferrari (same AP). Overall RVP-MP improves RVP-Baseline by 16.71%. © 2016 IEEE

	RVP-Baseline [2]	RVP-MP
100Plus	0.069	0.088
Ferrari	0.214	0.214
KittyB	0.090	0.323
KittyG	0.586	0.667
Maggi	0.589	0.625
NTU	0.758	0.800
PKU	0.771	0.830
Plane	0.329	0.475
Pokka	0.425	0.428
Starbucks	0.132	0.174
Video mAP	0.396	0.462

Furthermore, Table 3.3 demonstrates the effectiveness of ranking video using object trajectory scores instead of best matched keyframe scores. It is shown that RVP-MP ($\beta = -2.0$) consistently improves the video AP across all queries over RVP-Baseline, except for one query (i.e. Ferrari) that achieves the same video AP. The average improvement on video AP is 39.78%. Because RVP-MP is not sensitive to β , we fix β at -2.0 in all the following experiments.

3.5.2 Trajectory mAP

3.5.2.1 RVP-MP vs. KCF

Besides Max-Path, an alternative approach to obtain object trajectories in videos is tracking. Therefore, we adapt a state-of-the-art tracking method to the problem of object instance search in videos, and compare its performance with RVP-MP in terms of trajectory mAP.

We use Kernelized Correlation Filters (KCF) [4] with the source code provided

by the authors. The chosen kernel is Gaussian. To make a fair comparison with RVP-MP, we start with the same top 100 videos obtained by the baseline RVP, with a segmentation threshold $\alpha = 5.0$ (Section 3.5.1.2). To initialize the tracker in each of the top 100 clips, we fit a bounding box to the segmented region from the best matched keyframe in this clip. We consider the resulting bounding box as valid if it has a minimum height of 60 pixels. Otherwise we check the box from the second best keyframe, and so on. Empirically it is observed that this unsupervised approach produces reasonable initialization results. Once the initial bounding box of a clip is obtained, the KCF tracker is run from the initialization frame forward and backward, and concatenate the two trajectories as the final trajectory. Note that although SCFV-RANSAC (Section 3.5.1.1) can also provide object bounding box locations to initialize the tracker, it is observed that it produces much fewer reliable bounding boxes compared to RVP. This is because the target objects are usually small; hence the number of matched points are insufficient to reliably estimate a transformation using RANSAC.

As the KCF tracker runs on every frame instead of just the keyframes, for RVP-MP, we also run the Max-Path algorithm across all frames. The other Max-Path parameters remain the same as in previous section 3.5.1.3. Following Eq. 3.11, for the computation of the precision-recall curves and consequently the trajectory mAP, a trajectory T^* is considered positive if $PIoU(T^*, G) > \delta_p$, where G is the annotated ground truth trajectory and δ_p is the overlap threshold.

Figure 3.4 demonstrates that even on the highly cluttered NTU-VOI dataset, the proposed RVP-MP is able to find object trajectories with large appearance variations due to viewpoint changes, illumination changes, color changes, motion blur and occlusions.

Figure 3.5 shows visual comparisons of example trajectory search results from RVP-MP and KCF. We can see that a typical tracker like KCF fails to terminate the trajectory when the object exits the scene (1^{st} row, Figure 3.5 (a)). On the contrary,

Figure 3.4: Two example queries (left) and top trajectories returned by our method. Each trajectory is represented by one keyframe with magnified object region below or above the keyframe. The red boxes indicate our results, and the green boxes indicate the ground truth annotations. Given a single query example, our approach is able to find its occurrences in cluttered scenes regardless of appearance variations due to occlusions, motion blur, and changes in viewpoint, illumination and color. Best viewed in color and magnification. © 2016 IEEE

Figure 3.5: Example search results of "KittyG" (a) and "Plane" (b). Results of KCF tracker [4] are denoted in yellow bounding boxes in the top rows. Results of RVP-MP are in red in the bottom rows. Ground truth bounding boxes are in green. (a) KCF fails to find the correct start and end points of the trajectory when the object enters and exits the scene, while RVP-MP discover the start and end points automatically. (b) An inaccurate initial estimation of bounding box size will be carried on by KCF, while RVP-MP searches for the best bounding box size given the confidence maps. Best viewed in color and magnification. © 2016 IEEE

our RVP-MP approach can discover the start and end points of each trajectory automatically (2^{nd} row, Figure 3.5 (a)). In addition, as the initial object location for KCF is given by the best segmented bounding box generated by the baseline RVP, an inaccurate initial estimation of the object size (i.e. size of initial bounding box) will be carried on by the tracker, thus affecting the accuracy of the resulting trajectory (1^{st} row, Figure 3.5 (b)). On the other hand, using multi-scale Max-Path search, RVP-MP can search for and adjust the sizes of bounding boxes along the trajectory (2^{nd} row, Figure 3.5 (b)).

Figure 3.6 compares the trajectory mAP of RVP-MP and KCF with different overlap ratio δ_p . It shows that RVP-MP consistently outperforms KCF on trajectory mAP as δ_p increases from 0.1 to 0.7. The improvement is between 7 and 197%

Figure 3.6: Trajectory mAP: our RVP-MP method consistently outperforms KCF tracker [4] in a wide range of overlap ratio δ_p . © 2016 IEEE

(39.05%, 14.63%, 20.76%, 7.13%, 70.46%, 197.10%, 163.08%, respectively).

Figure 3.7 presents more example trajectories resulting from RVP-MP. Our approach accurately locates the objects regardless of varying viewpoint (1st and 2nd row) and scale (2nd rows), reflection (2nd row) and occlusions (3rd row). In addition, it can automatically discover the start and end points of each trajectory (Figure 3.7, 4th – 6th rows).

3.5.2.2 Parameter Sensitivity

We test the Max-Path search at different temporal granularities: 1 fps, 2 fps, 3 fps, 5 fps, 10 fps, up to every frame. Figure 3.8 plots the trajectory mAP of RVP-MP with

Figure 3.7: Example trajectories returned by our proposed RVP-MP in the presence of varying viewpoint (1st and 2nd rows) and scale (2nd row), reflection (2nd row) and occlusions (3rd row). In addition, the start and end points of each trajectory are discovered automatically (4th – 6th rows). RVP-MP results are marked in red, and the ground truth bounding boxes are in green. Best viewed in color and magnification. © 2016 IEEE

different temporal sample intervals, with respect to the overlap ratio δ_p . As can be seen, the performance of RVP-MP degrades gracefully as the temporal step increases.

3.5.3 Efficiency

All experiments were conducted on a quadcore dual-processor machine with 2.30 GHz CPU and 32 GB RAM, without GPU. We parallelized both RVP and Max-Path search in 8 threads and implemented the algorithms in C++. Excluding I/O, the average time to obtain the top 100 trajectories for a query object instance is 200 seconds on the 73-hour dataset. Although less accurate, KCF tracker using RVP-Baseline for initialization is faster than RVP-MP. It takes only 142 seconds on average to obtain the top 100 trajectories.

Figure 3.8: Trajectory mAP of RVP-MP for different temporal steps and overlap ratio δ_p . Best viewed in color and magnification. © 2016 IEEE

3.6 Discussions

3.6.1 Measuring Object-Video Relevance by Trajectories

Experimental evaluation shows that object trajectories are a better indicator of video relevance compared with individual frames. This validates that ensuring spatio-temporal consistency is beneficial to object instance search in videos because it helps filter false alarms and reduce missed detections caused by appearance variations or cluttered backgrounds. This is because the false positives usually appear randomly at inconsistent spatial locations, in the long run, the accumulated trajectory confidence score of a false positive path cannot be greater than that of the true object trajectory. On the other hand, occasional missed detections can be recovered as long

as the accumulated confidence score of the trajectory is sufficiently high.

When spatio-temporal localization is considered for a more precise evaluation of search quality, in terms of trajectory mAP, the proposed RVP-MP is shown superior to the state-of-the-art KCF tracker (with the initial bounding box being produced by RVP). The main advantage of Max-Path search over tracking is its ability to automatically discover the starting and ending frames of the object trajectory. Meanwhile, it does not require initialization of the target object.

3.6.2 Object Instance Search vs. Detection

A key component to ensure scalability of search is indexing. Generally indexing is suited for situations where the database will be searched frequently for different queries, therefore the computational cost and storage overhead of indexes are justifiable. However, in situations where the video archive is unlikely to be revisited in the future, e.g., post-crime investigation in surveillance videos, indexing may not be necessary. Then how to efficiently search the video repository for small target without indexing the whole dataset presents new challenges. In this case, detection may be more appropriate than search.

In fact, the boundary between object instance search and object instance detection can be blurry. Which approach should be taken to find small visual objects in the large repository is largely application dependent. Among existing works on search and detection, we can roughly summarize the two differences between the two approaches, which consequently determine which is more appropriate for certain applications. Query wise, search only requires a single example and no training. However, for detection to achieve a reasonable accuracy, sufficient training data of the query needs to be provided. Dataset wise, typically the dataset is pre-processed to build indexes to reduce the storage cost and enable fast on-line matching, although binary codes from hashing can alternatively be utilized to reduce the footprint and

also achieve fast on-line matching. Fast on-line matching is critical for large scale search applications, because the dataset will be visited many times for searching different queries. However, this is usually not the case for detection, where usually the dataset just needs to be scanned once, but possibly for multiple targets. Therefore, detection is often done without indexing the dataset, but may require finding multiple targets at the same time for efficiency.

3.7 Summary

In this chapter, we explore the use of spatio-temporal cues to boost the performance of object instance search in videos, and propose to formulate the problem as finding the top- K spatio-temporal object trajectories. Consequently, the relevance of a video to a given query object is innovatively measured by the trajectory-wise similarity instead of keyframe-wise. By utilizing RVP to quickly obtain per pixel confidence scores, thus enabling fast scoring of any bounding boxes in a video, we extend the dynamic programming based Max-Path search from detection to search. Experiments on a 73-hour video dataset validate that the proposed approach is effective in improving search quality compared with state-of-the-art methods that treat frames independently. The effectiveness of incorporating spatio-temporal coherency into search is also demonstrated by its ability to find object occurrences in cluttered scenes regardless of large appearance variations due to motion blur, occlusions, and changes in viewpoint, illumination and color. Compared with individual frames, the resulting spatio-temporal trajectories turn out to be better indicators of video relevance because of their ability to better handle false alarms and missed detections across frames caused by appearance variations or cluttered backgrounds. This approach also improves the spatio-temporal localization of objects compared with the state-of-the-art KCF tracker. In addition, a new video dataset, NTU-VOI, is made available to facilitate future research on object instance search in videos.

Chapter 4

Object Proposals to Speed Up Spatio-Temporal Search

4.1 Overview

This chapter presents how to further improve the efficiency of object instance search in videos upon RVP-MP (Chapter 3) without compromising the accuracy. Recall that the efficiency of Max-Path search depends on a 3D trellis structure that connects bounding boxes in each frame, which are scored prior to the spatio-temporal search. To score all possible bounding boxes, Randomized Visual Phrases (RVP) [22] is utilized to efficiently obtain pixel-wise confidence scores for each frame by estimating only a set of randomly sampled patches (windows). Consequently, the score of any bounding box that may result from the exhaustive sliding-window search in a given frame can be calculated as the sum of the scores of its containing pixels, which is only $O(1)$ with the help of the integral image built from the confidence map. In what follows, we will first explain the two bottleneck of RVP-MP before presenting our solutions.

First, although the pixel-wise confidence maps are necessary to build integral images for Max-Path search, calculating the confidence scores down to the pixel level

is not fast even with RVP. Recall that we need integral images built from the pixel-wise confidence maps, so we can quickly score *any* bounding box that may result from the exhaustive sliding-window search. One natural question to ask is whether it is really necessary to score and check all possible bounding boxes. If only a small number of bounding boxes may contain the object in each frame, we should be able to evaluate them fast enough without the help of integral images, hence without producing pixel-wise confidence maps. Then the problem boils down to how to determine which subset of bounding boxes should be checked and how to perform spatio-temporal search accordingly.

The second bottleneck concerns with the coarse-to-fine search schedule. It turns out that the speed of the coarse matching greatly affects the final speed of the system. As can be seen in Table 4.6, without GPU, it takes on average 134.8 seconds to match a query object against keyframes in the 73-hour dataset (Section 3.4.1). Recall that the initial ranking of all videos are based on the RVP-Baseline [2], which consists of random patch matching, voting and segmentation. Voting is to mitigate ambiguity introduced by random spatial context by averaging. The purpose for segmentation is to more accurately evaluate similarity within a local region. As has been shown, locality can benefit object instance search in image datasets [2, 21]. However, voting and segmentation are costly. Avoiding segmentation alone can cut the time to half for matching. So, can we incorporate locality without voting and segmentation?

Recently, category-independent object proposals have become a popular alternative to sliding windows for object detection. They enhance the efficiency of detection by only evaluating bounding box locations that are likely to contain an object, which is usually a few thousand per image [35–38]. In light of the successes of object proposals for detection and weakly supervised learning [51, 52], we propose to address both bottlenecks of RVP-MP by matching the query with only a few candidate bounding boxes instead of all bounding boxes in the video. To this end, we present and compare two different approaches of generating the candidates: a random approach based on

the object size prior and a principled approach based on the objectness score from Edge Boxes proposals [35]. We refer to the former as Rand-MP and the latter as Prop-MP, although the former can also be seen as random object proposals. We also describe how to perform spatio-temporal search based on object proposals while bypassing the generation of confidence maps. The proposed solution is systematically evaluate on the same 73-hour dataset as in the previous chapter in comparison with RVP-MP. It shows that both approaches achieve competitive performance compared to RVP-MP in terms of video mAP, but with a 3 to 60-time speedup (Table 4.6). Prop-MP also achieves comparable trajectory mAP compared with RVP-MP, while Rand-MP performs worse. Section 4.5.1 provides further analysis on the cause of the difference.

4.2 Problem Formulation

The goal is the same as in Section 3.2. Given a query object Q and the entire video database as $\mathcal{V} = \{\mathcal{F}_1, \mathcal{F}_2 \dots \mathcal{F}_n\}$, where $\mathcal{F}_i \in \mathcal{V}$ is the i^{th} frame with a size of $w \times h$, we aim to find in $\mathcal{V}$ a number of spatio-temporal trajectories of bounding boxes such that each trajectory captures one instance of Q in $\mathcal{V}$.

Denote a trajectory $T = \{B_i\}_{i=i_1}^{i_k}$ as a temporal sequence of spatial bounding boxes that locate the object instance in each frame $B_i = \{x_i, y_i, s_i, a_i, t_i\}$, where x_i and y_i are the center coordinates, s_i and a_i are the scale and aspect ratio of bounding box B_i , and t_i is its temporal index. Assume instead of the set of all possible bounding boxes $\mathcal{B}$, only a candidate set $\mathcal{B}_S \subset \mathcal{B}$ is considered, where $|\mathcal{B}_S| \ll |\mathcal{B}|$. Because $|\mathcal{B}_S|$ is a rather coarse sampling of $\mathcal{B}$, it is difficult to reinforce the smoothness constraint of the bounding box scale as in previous chapter (Eq. (3.2)). Instead, we supplement the smoothness constraint of the trajectory with an overlap constraint on consecutive bounding boxes. Formally, each trajectory $T = \{B_i\}_{i=i_1}^{i_k}$ should satisfy the following constraints:

1) The smoothness constraint of the trajectory:

$$x_i - k \leq x_{i+1} \leq x_i + k, y_i - k \leq y_{i+1} \leq y_i + k, t_{i+1} = t_i + 1 \quad (4.1)$$

where $(2k+1) \times (2k+1)$ is the valid neighborhood size that the center of one bounding box can move to in the next frame;

2) The overlap constraint of consecutive bounding boxes:

$$\frac{B_i \cap B_{i+1}}{B_i \cup B_{i+1}} > \eta \quad (4.2)$$

where the intersection area divided by the union area (IoU) of two consecutive bounding boxes on the trajectory must be above a threshold η .

Note that the above two constraints may not ensure the smoothness of the resulting spatio-temporal trajectory, because the change in scale and aspect ratio from B_i to B_{i+1} could be noticeable. Nevertheless, experimentally a simple post-processing on the trajectory using either median or mean filter can effectively mitigate the visual artifacts. Again, each trajectory of bounding boxes can start and end at any temporal location within the 3D video volume, as long as the temporal index of the start point is before that of the end point. However, the spatial locations are constrained by the sample set $\mathcal{B}_S$.

Denote the discriminative score of a bounding box $B_i \in \mathcal{B}_S$ in a given video as $s(B_i)$, where $s(B_i) > 0$ shows positive evidence of the object instance's presence while $s(B_i) < 0$ shows negative evidence. Similar to Eq. (3.4), the confidence score $s(T)$ of trajectory T is calculated as the sum of the discriminative scores of the bounding boxes along T , i.e.

$$s(T) = \sum_{B_i \in T} s(B_i), s.t. B_i \in \mathcal{B}_S \quad (4.3)$$

where $s(T)$ indicates the likelihood that the object appears along T .

We can therefore rewrite the objective of finding the top K trajectories that capture K instances of the object query in the video database $\mathcal{V}$ as the following optimization problem:

$$\begin{aligned} \mathbf{T}^* &= \arg \max_{\mathbf{T} \subseteq \mathcal{T}_S} \sum_{T \in \mathbf{T}} s(T) \\ s.t. \forall T_i, T_j \in \mathbf{T}, T_i \cap T_j &= \emptyset \\ |\mathbf{T}| &= K. \end{aligned} \tag{4.4}$$

where $\mathbf{T}$ is a subset of K trajectories we are searching for. Different from Eq. (3.4), instead of searching for $\mathbf{T} \subseteq \mathcal{T}$, where $\mathcal{T}$ is the set of all possible trajectories in $\mathcal{V}$, here we are limited to the subset of trajectories $\mathcal{T}_S \subset \mathcal{T}$ that can only be produced by the candidate bounding boxes $B_i \in \mathcal{B}_S$. Because of the assumption that $T_i \cap T_j = \emptyset$, the top- K search can be rewritten as the search of the individual best trajectories:

$$T^* = \arg \max_{T \in \mathcal{T}_S} s(T) \tag{4.5}$$

To find the top K trajectories, we can remove the current best path from $\mathcal{T}_S$ once discovered and repeat the search K times.

4.2.1 Spatio-Temporal Search with Candidate Bounding Boxes

Given $B_i \in \mathcal{B}_S$, where $|\mathcal{B}_S| \ll |\mathcal{B}|$, the number of candidate trajectories $|\mathcal{T}_S|$ becomes much smaller than before, i.e., $|\mathcal{T}|$. Assume at most b bounding boxes are sampled per frame, then an exhaustively search for the optimal spatio-temporal trajectory in $\mathcal{T}_S$ requires $O(bnm^n)$, where $m \leq b$ is the number of bounding boxes in a frame (and $\in \mathcal{B}_S$) that satisfy the smoothness constraint of the trajectory (Eq. (4.1)). Although it is already a much smaller search space compared with $O(whn(2k+1)^{2n})$ for all possible trajectories (as $b \ll wh$), the search complexity can be further reduced to $O(bnm)$ using the Max-Path search.

In this case, a 3D trellis can be built using the set of candidate bounding boxes $\mathcal{B}_S$. Denote the set of candidate bounding boxes in a given frame t as $\mathcal{B}_S(t)$, each

node on the trellis at time t corresponds to a bounding box $B(t) \in \mathcal{B}_S(t)$. Given the discriminative matching score of any candidate bounding box $s(B)$, we can use dynamic programming to find the best trajectory ending at any node $B \in \mathcal{B}_S(i)$ in frame i by checking the best trajectory score among its neighbors $B_p \in \mathcal{B}_S(i-1)$ in the previous frame $i-1$. The neighborhood is defined by the smoothness constraint (Eq. (4.1)) and overlap constraint (Eq. (4.2)). The pseudo code of the proposed spatio-temporal search with sampled bounding boxes is given in Algorithm 1.

It is worth noting that since the number of nodes (i.e., bounding boxes) in each frame on the trellis is much smaller than before (i.e., $b \ll wh$), we can now afford direct calculation of $s(B)$ for each node rather than resorting to integral images. Therefore we can bypass the generation of pixel-wise confidence maps, which is time-consuming. Experimentally it validates that the Max-Path search on the candidates indeed runs faster than before without the help of integral images (Table 4.6).

4.2.2 Bounding Box Scoring

To obtain the discriminative score $s(B)$ for each candidate bounding box $B \in \mathcal{B}_S$, we first match each B against the query to get the matching score $\bar{s}(B)$. Recall that to produce a pixel-wise confidence map, RVP requires matching $\mathcal{K} \times \mathcal{M} \times \mathcal{N}$ random patches, where $\mathcal{K}$ is the partition rounds, and $\mathcal{M} \times \mathcal{N}$ is the number of random patches per round. Specifically, a random template of size $\mathcal{M} \times \mathcal{N}$ is used to produce these non-overlapping rectangular patches (i.e., a frame is cut into $\mathcal{M}$ pieces horizontally and $\mathcal{N}$ pieces vertically). When the number of sampled bounding boxes per frame $b \ll \mathcal{K} \times \mathcal{M} \times \mathcal{N}$, scoring all candidate boxes in $\mathcal{B}_S$ is much faster than generating the confidence map via RVP, given the same matching kernel.

Although the matching score $\bar{s}(B)$ is not discriminative, similar to how we introduce negative scores to the confidence maps in Section 3.2.2, we can convert $\bar{s}(B)$ to

Algorithm 1: Spatio-Temporal Search with Sampled Bounding Boxes

Input :
 $s(B)$: the discriminative score of bounding box $B \in \mathcal{B}_S$;
 $\mathcal{B}_S(t)$: the set of sampled bounding boxes in frame t .
Output:
 $S(B, t)$: the accumulated score of the best trajectory ending at B in frame t ;
 T^* : the best trajectory;
 S^* : score of the best trajectory.

```

1   $S(B, 1) = s(B), \forall B \in \mathcal{B}_S(1)$ ;
2  for  $i \leftarrow 2$  to  $n$  do
3      for  $B \in \mathcal{B}_S(i)$  do
4           $B_p^* \leftarrow \arg \max_{\substack{B_p \in \mathcal{B}_S(i-1) \\ B_p \in N(B)}} S(B_p, i-1)$ ;
5           $\triangleleft B_p \in N(B)$  satisfies Eq. (4.1) (4.2);
6          if  $S(B_p^*, i-1) > 0$  then
7               $S(B, i) \leftarrow S(B_p^*, i-1) + s(B)$ ;
8          else
9               $S(B, i) \leftarrow s(B)$ ;
10         end
11         if  $S(B, i) > S^*$  then
12              $S^* \leftarrow S(B, i)$ ;
13              $T^*$  push back  $B$ ;
14         end
15     end
16 end
    
```

a discriminative score by adding a negative threshold. Again, the threshold is adaptively set for each video to accommodate intra-video object appearance variations, instead of using a fixed value [87]. However, different from Section 3.2.2, here the threshold is set to be proportional to the average matching score of $B \in \mathcal{B}_S$ in a given video clip. Formally, the threshold MP_{thres} is defined as follows:

$$MP_{thres} = \beta \frac{\sum_{B \in \mathcal{B}_S} \bar{s}(B)}{|\mathcal{B}_S|} \quad (4.6)$$

Above, $\mathcal{B}_S$ is the set of sampled bounding boxes in the video, while $|\mathcal{B}_S|$ indicates the total number of samples. The final confidence score of $B \in \mathcal{B}_S$ is computed as:

$$s(B) = \bar{s}(B) + MP_{thres} \quad (4.7)$$

We shall examine how the negative coefficient β affects the search performance in Section 3.5.1.3.

Figure 4.1: Overview of proposed approach to find object trajectories in videos. The query is only matched with candidate bounding boxes in each video keyframe (i.e., colored boxes in the 2nd row, see Section 4.2.2). Then videos are ranked by the scores of their best matched bounding boxes (red boxes in the 3rd row), while low-rank videos are filtered (the gray cuboids in the 3rd row). At last the globally optimal object trajectories in top ranked videos are identified by the Max-Path search (Section 4.2.1) and videos are re-ranked according to the trajectory scores. Ground truth locations are highlighted in dashed blue boxes in the found trajectories (4th row). Best viewed in color and magnification.

4.2.3 Coarse-to-Fine Search

As in Section 3.2.3, search is performed in two steps for efficiency. First, a given query is matched against $B \in \mathcal{B}_S$ from coarsely sampled keyframes. Then each video is ranked by the score of its best matched bounding box. For those top ranked videos, we perform the adapted Max-Path search as described in Section 4.2.1 to find the globally optimal object trajectories, and videos are re-ranked accordingly.

The entire workflow is illustrated in Figure 4.1.

4.3 Experimental Setup

Dataset and Queries. The same 73-hour dataset as in Section 3.4.1 is used to evaluate the proposed sampling methods, consisting of 26-minute ground truth videos from NTU-VOI and a subset of Stanford I2V dataset [1] as distractors. Same as before, all 73-hour videos are resized to a spatial resolution of 800×450 pixels and sampled uniformly at 1 fps, resulting in 263,180 keyframes. The same 10 queries for RVP-MP are tested (Table 3.1).

Evaluation Metrics. Video mAP and trajectory mAP as defined in Section 3.3 are used.

Features and Matching. For all following experiments, the same set of SIFT local features as for RVP-MP in Chapter 3 are used, which are quantized to visual words using the same 1M vocabulary built from SIFTs in every 10th keyframe. Each sampled bounding box $B \in \mathcal{B}_S$ is then represented as a sparse 1M dimensional histogram recording the word frequency of B . Different from experiments in the previous chapter, the normalized histogram intersection, i.e., $NHI(\cdot)$, is used instead of histogram intersection ($HI(\cdot)$) as the matching kernel to score B . It is because $HI(\cdot)$ does not penalize mismatched points in B , which tends to return bigger boxes that

do not locate the object accurately. RVP can mitigate this problem by averaging over multiple rounds of partitions, while here each bounding box is matched independently. To make a fair comparison, we have also re-run RVP and RVP-MP with the $NHI(\cdot)$ kernel, but the performance is slightly worse than that with the $HI(\cdot)$ kernel. Therefore, we report the proposed approach using $NHI(\cdot)$ kernel while still compare with the results of RVP and RVP-MP with the $HI(\cdot)$ kernel.

4.4 Experimental Results

4.4.1 Candidate Selection Methods

In this section, we experimentally evaluate the proposed approach with two different approaches of generating the candidates: a random approach based on the object size prior and a principled approach based on the objectness score from Edge Boxes proposals [35]. Specifically, for the former, with the prior knowledge that the target objects are generally small in the dataset, we use 8×4 -sized random templates as for RVP (Section 3.5.1.2) to partition each frame multiple rounds, and use the resulting random patches as candidates. For the latter, frame-level object proposals are produced by the high recall Edge Boxes method [35] and proposals with top objectness scores are selected as candidates. We refer to the former as **Rand-MP** and the latter as **Prop-MP**. The former can also be seen as random object proposals with the size prior. Later in Section 4.5.1, we will present analysis of how the size prior affects the search performance in comparison with the objectness prior.

For both proposal generating approaches, a coarse ranking is first obtained based on the score of the best matched proposal for each video (Section 4.2.2). We refer to the coarse video ranking according to the score of best matched random proposal as **Rand-Max**, and that according to the score of best matched Edge Boxes proposal as **Prop-Max** (Section 4.2.3).

4.4.2 Video mAP

For video mAP, both Rand-MP and Prop-MP are compared against RVP-MP (Section 3.5.1). And Max-Path is carried out on the top 100 videos for spatio-temporal trajectory search and re-ranking. Empirically the neighborhood size k is set to 100 pixels (Eq. (4.1)) and η is set to 0 (Eq. (4.2)). The overlap constraint is relaxed because the keyframes are sample coarsely at 1 fps and target objects are generally small. Note that RVP-Baseline ranks videos based on the score of the best matched keyframe (to be exact, the best segmented region in a keyframe), while the proposed Rand-MP and Prop-MP rank videos based on trajectory scores. For each query, video AP is computed based on the top 100 retrieved videos. And the APs are averaged over the 10 queries to get the video mAP.

4.4.2.1 Rand-MP

We first test how the number of partition rounds affect the coarse ranking accuracy and time using Rand-Max. Table 4.1 shows that increasing the number of partition rounds above 50 will not further improve video mAP, which means that 1,600 random patches per frame can provide a good coverage of objects. In fact, even with as few as 480 random patches (i.e., 15 rounds of partitions), we can achieve a video mAP of 0.388 using Rand-Max, which is comparable to that of RVP-Baseline with 200 rounds of partitions (i.e., video mAP = 0.396, Table 3.3). As can be seen in Table 4.1, the fewer random patches to be matched per keyframe, the faster the coarse round search. Therefore we sample 480 random patches per keyframe in all following experiments to provide a trade-off between accuracy and speed.

Once all random patches have been matched against a given query and a coarse ranking is produced by Rand-Max, the Max-Path search is applied to top ranked videos to find the spatio-temporal trajectories for video re-ranking. The video mAP of Rand-MP is evaluated under varying negative coefficient β (Eq. (4.6)). Figure 4.2

Table 4.1: Video mAP on top 100 retrieved clips using Rand-Max.

#Rds of Partition	200	150	100	50	20	15	10
#Rand Patches	6400	4800	3200	1600	640	480	320
Video mAP	0.395	0.393	0.396	0.400	0.387	0.388	0.364
Time (s)	64.58	51.46	37.57	28.84	24.84	24.76	24.51

Figure 4.2: The impact of β on video mAP: Rand-MP consistently outperforms RVP-Baseline [2] and Rand-Max as β changes from -8 to -2 . It is comparable to RVP-MP (with best β) when $\beta \in [-8, -4]$.

illustrates the video mAP of Rand-MP in comparison with RVP-Baseline and Rand-Max as β increases from -8 to -2 , at an interval of 1. The best video mAP of RVP-MP is also plotted as reference (when $\beta = -1.5$, Figure 3.3). It is observed that Rand-MP is not sensitive to $\beta \in [-8, -4]$ and consistently outperforms Rand-Max and RVP-Baseline in this range. However, its performance starts to decrease as β increases above -4 . When $\beta = -5$, Rand-MP improves RVP-Baseline by 17.11% and Rand-Max by 19.50%. It is comparable to the best video mAP achieved by RVP-MP.

Table 4.2: Video mAP on top 100 retrieved clips. Rand-MP achieves comparable performance compared with RVP-MP, and consistently improves AP on all queries over Rand-Max. Overall Rand-MP improves Rand-Max by 19.50%.

	Rand-Max	RVP-MP	Rand-MP
100Plus	0.062	0.088	0.118
Ferrari	0.170	0.214	0.196
KittyB	0.185	0.323	0.230
KittyG	0.558	0.667	0.583
Maggi	0.614	0.625	0.625
NTU	0.741	0.800	0.781
PKU	0.794	0.830	0.859
Plane	0.262	0.475	0.556
Pokka	0.427	0.428	0.472
Starbucks	0.071	0.174	0.220
Video mAP	0.388	0.462	0.464

Table 4.2 details the video AP of each of the 10 queries for Rand-MP ($\beta = -5$), in comparison with Rand-Max and RVP-MP. It shows that Rand-MP consistently improves the video AP over Rand-Max across all queries, with an average improvement of 19.50% in video mAP.

The speed of Rand-MP is summarized in Table 4.6. It validates that the proposed Rand-MP improves efficiency without compromising search accuracy of object instances in videos. More discussions on efficiency are in Section 4.4.6.

4.4.2.2 Prop-MP

We first compare the performance of ranking videos by the best matched proposal in comparison with RVP-Baseline [2], as the number of proposals per keyframe decreases from 500 to 50 (Table 4.3). It is shown that with as few as 200 proposals per keyframe, we can achieve a video mAP of 0.402 by simply matching the proposals to the query, which is even slightly better than the RVP-Baseline (i.e., 0.396). Using more proposals than 300 per keyframe decreases the mAP. It is likely because the

Table 4.3: Video mAP on top 100 retrieved clips and search time using Prop-Max.

#Proposals	500	400	300	200	100	50	RVP-Baseline [2]
Video mAP	0.397	0.400	0.402	0.402	0.397	0.381	0.396
Time (s)	32.88	32.50	30.86	28.41	27.72	26.79	138.80

Table 4.4: Video mAP and Max-Path search time on top 100 retrieved clips using Prop-MP (when $\beta = -1$).

#Proposals	500	400	300	200	100	50	RVP-MP	Rand-MP (480 patches)
Video mAP	0.466	0.473	0.471	0.469	0.466	0.452	0.462	0.464
MP Time (s)	28.31	18.73	9.92	4.25	1.28	0.38	65.67	20.43

ground truth objects are generally covered by the top ranked proposals, therefore, adding more lower ranked proposals only introduces more distractors. Also the video mAP achieved by Prop-Max using only 200 proposals is comparable to that of Rand-Max (i.e., 0.400, Table 4.1) using 1,600 random patches (i.e., 50 rounds of random partitions with 8×4 random templates). However, the time for coarse ranking using the two approaches are comparable, although 8 times more random patches are used (i.e., 28.41 seconds for Prop-Max vs. 28.84 seconds for Rand-Max). It is likely because most random patches do not hit the ground truth or match any visual words in the query, therefore when using the inverted index file to find matched interest points, those random patches are never hit or evaluated. On the other hand, object proposals from Edge Boxes may have a much higher probability to intersect with the ground truth objects.

Nevertheless, fewer number of object proposals to be evaluated results in a significant speedup of spatio-temporal trajectory search. As shown in Table 4.4, it only takes on average 1.28 seconds to obtain the top 100 trajectories for a query when using 100 object proposals per keyframe, while achieving comparable video mAP to that of RVP-MP and Rand-MP (when using 480 random patches per keyframe). In comparison, RVP-MP takes 65.67 seconds and Rand-MP takes 20.43 seconds. Also

Figure 4.3: The impact of β on video mAP: Prop-MP achieves comparable performance to RVP-MP in a wide range of β ; it also consistently outperforms Prop-Max and RVP-Baseline [2] as β changes from -2 to 0 .

when re-ranking videos by spatio-temporal trajectories, Prop-MP consistently improves Prop-Max in video mAP. The improvement is 17.38% when 100 object proposals is evaluated per keyframe.

As in the previous section, the video mAP of Prop-MP is evaluated under varying negative coefficient β (Eq. (4.6)) when the top 100 proposals are taken per keyframe. Figure 4.3 illustrates the video mAP of Prop-MP in comparison with Prop-Max, and RVP-MP and RVP-Baseline presented in the previous chapter ($\alpha = 5.0$, same as in Section 3.5.1), as β increases from -2.0 to 0 . It is observed that Prop-MP performs competitively in comparison with RVP-MP in video mAP, and is not sensitive to $\beta \in [-2, 0]$. Moreover, Prop-MP consistently outperforms RVP-Baseline and Prop-Max when $\beta \in [-2, 0]$. For all following experiments, β is fixed at -1 for Prop-MP.

Table 4.5: Video mAP on top 100 retrieved clips. Prop-MP achieves comparable performance compared with RVP-MP, and consistently improves AP on all queries over Prop-Max. Overall Prop-MP improves PROP-Max by 17.38%.

	Prop-Max	RVP-MP	Prop-MP
100Plus	0.074	0.088	0.118
Ferrari	0.151	0.214	0.157
KittyB	0.139	0.323	0.371
KittyG	0.596	0.667	0.650
Maggi	0.556	0.625	0.563
NTU	0.758	0.800	0.778
PKU	0.805	0.830	0.847
Plane	0.382	0.475	0.524
Pokka	0.417	0.428	0.417
Starbucks	0.095	0.174	0.239
Video mAP	0.397	0.462	0.466

Table 4.5 details the video AP of each of the 10 queries for Prop-MP (top 100 proposals per keyframe), in comparison with Prop-Max and RVP-MP. It shows that Prop-MP consistently improves the video AP over Prop-Max across all queries, except for Pokka (same AP), with an average improvement of 17.38%.

4.4.3 Trajectory mAP

Same as in Chapter 3, the proposed Rand-MP and Prop-MP are also compared with the KCF tracker [4], while running the Max-Path algorithm across all frames. Since now we consider every frame, the neighborhood size k is empirically set to 30 pixels (Eq. (4.1)) and η is set to 0.7 (Eq. (4.2)). The negative coefficient β is fixed at -5 as before for Rand-MP, and -1 for Prop-MP. Different from RVP-MP (Section 3.5.1.3), here we initialize KCF with the best matched random patch or Edge Boxes proposal in a given video sequence resulting from Rand-Max or Prop-Max, instead of the bounding box fitted to the segmented region of the best matched keyframe resulting from RVP. Following Eq. (3.11), for the computation of the precision-recall

Figure 4.4: Trajectory mAP: the proposed Rand-MP performs poorly compared with RVP-MP and the KCF tracker [4] (except for when $\delta_p = 0.1$) in a wide range of overlap ratio δ_p .

curves and consequently the trajectory mAP, a trajectory T^* is considered positive if $PIoU(T^*, G) > \delta_p$, where G is the annotated ground truth trajectory and δ_p is the overlap threshold.

4.4.4 Rand-MP

As mentioned in Section 4.4.2.1, 480 random patches produced by 15 random templates of size 8×4 are used to evaluate Rand-MP. Figure 4.4 shows the trajectory mAP of Rand-MP in comparison with RVP-MP and KCF under different overlap ratio δ_p . Contrary to its competitive performance in video mAP, Rand-MP performs significantly worse than RVP-MP in terms of trajectory mAP. It also performs worse

Figure 4.5: Trajectory mAP of the proposed Prop-MP w.r.t. the overlap ratio δ_p , as different number of proposals are sampled per frame.

than the KCF tracker initialized by the best matched random patch from Rand-Max, when $\delta_p \geq 0.2$. It shows that random proposals using the object size prior are not effective in providing accurate localization of the target objects with spatio-temporal search. This will be discussed further in Section 4.5.1.3.

4.4.5 Prop-MP

Figure 4.5 plots the trajectory mAP of Prop-MP with respect to the overlap ratio δ_p , when selecting between 50 and 500 proposals per frame. As can be seen, except for when the overlap ratio is low (i.e., $\delta_p = 0.1$), using more proposals per frame does not necessarily improves the trajectory mAP. And when more accurate localization is desirable (e.g., $\delta_p \geq 0.4$), 100 proposals turns out to work better than 500 proposals

Figure 4.6: Trajectory mAP: the proposed Prop-MP approach achieves similar performance compared with RVP-MP, while consistently outperforms KCF tracker [4] in a wide range of overlap ratio δ_p .

per frame. This again can be attributed to the distractors introduced by low-rank proposals.

Figure 4.6 compares the proposed Prop-MP with RVP-MP (Section 3.5.1.3) and the KCF tracker with varying overlap ratio δ_p , in terms of trajectory mAP. For a fair comparison with Rand-MP, which is set to sample 480 random patches per frame, we use the top 500 proposals generated by Edge Boxes for each frame for Prop-MP. Contrary to Rand-MP, Prop-MP is comparable to or better than RVP-MP in terms of trajectory mAP. It also consistently outperforms the KCF tracker initialized by Prop-Max as δ_p increases from 0.1 to 0.7. It is also interesting to note that when accurate trajectory localization of the target object is desirable (i.e., $\delta_p \geq 0.5$), the KCF initialized by the best matched proposal of each video (i.e., Prop-Max) performs

Figure 4.7: Example trajectories returned by our proposed Prop-MP in the presence of varying viewpoint (3^{rd} and 4^{th} rows) and scale (3^{rd} row), reflection (3^{rd} row) and occlusions (2^{nd} and 6^{th} rows). In addition, the start and end points of each trajectory are discovered automatically (1^{st} , 4^{th} – 6^{th} rows). Prop-MP results are marked in red, and the ground truth bounding boxes are in green. Best viewed in color and magnification.

comparably to RVP-MP.

Figure 4.7 shows visualization of example trajectories resulting from Prop-MP. Our approach accurately locates the objects regardless of varying viewpoint (3^{rd} and 4^{th} rows) and scale (3^{rd} row), reflection (3^{rd} row) and occlusions (2^{nd} and 6^{th} rows). In addition, it can automatically discover the start and end points of each trajectory (Figure 4.7, 1^{st} , 4^{th} – 6^{th} rows).

4.4.6 Efficiency

As in Chapter 3, all experiments were conducted on a quadcore dual-processor machine with 2.30 GHz CPU and 32 GB RAM, without GPU. Both Rand-MP and Prop-MP are parallelized in 8 threads and implemented in C++. The search time of

Table 4.6: Efficiency of Rand-MP and Prop-MP in comparison with RVP-MP

	RVP-MP	Rand-MP	Prop-MP
Matching	134.80	24.76	27.72
Max-Path Search	65.67	20.43	1.28
Time (s)	200.57	45.19	29.00

Rand-MP with 480 random patches and that of Prop-MP with 100 Edge Boxes proposals per frame are reported in Table 4.6, as they produce comparable video mAP (i.e., 0.464 for Rand-MP and 0.466 for Prop-MP) to that of RVP-MP (i.e., 0.462). Excluding I/O, the average time to obtain the top 100 trajectories for a query object instance on the 73-hour dataset are 45.19 seconds and 29 seconds respectively, which are about 4.4 and 6.9 times faster than that of RVP-MP. Specifically, using either prior, matching proposals with the query takes roughly 1/5 of the time required by RVP matching. This is because a much fewer number of bounding boxes need to be matched per frame, compared with 6,400 ($200 \times 8 \times 4$) random patches per frame for RVP. This shows the effectiveness of both approaches in speeding up the coarse round matching. In addition, the Max-Path search also becomes faster since the 3D trellis is now built at a coarser scale (where nodes are object proposals) than that of RVP-MP (where nodes are down-sampled pixels). When using only 100 proposals per frame, it takes only 1.28 seconds to grow the trajectories, which is over 50 times faster. It validates that even without building the integral images (e.g., as for RVP-MP), we can achieve efficient spatio-temporal search using object proposals.

4.5 Discussions

4.5.1 Object Size Prior vs. Objectness Prior

Recall that the random templates used to produce random patches for Rand-MP imposes a strong assumption on the target object size. We assume that the target

objects are small, therefore random template of size 8×4 are used throughout the experiments. Essentially, object size prior is used to guide the selection of bounding boxes for Rand-MP, while objectness prior is used to guide the selection of bounding boxes for Prop-MP. Experimentally it shows that both priors are effective in achieve comparable video mAP to that of RVP-MP, with a fraction of the time. However, the object size prior seems not to be as effective as the objectness prior in obtaining accurate object trajectories. Below we will analyze the coverage of ground truth objects using the two proposal generation approaches and explain the reasons for the differences in performance.

4.5.1.1 Metrics

We first introduce the metrics used to measure the coverage of the ground truth objects.

- **Ave. Best IoU of GT Videos:** Denote the set of ground truth bounding boxes in a ground truth video $\mathcal{V}_i$ for a given query as $\mathcal{G}_{v_i}$, the set of sampled bounding boxes in $\mathcal{V}_i$ as $\mathcal{B}_{S_{v_i}}$, then the *best* intersection over union (IoU) score of $\mathcal{V}_i$ can be defined as:

$$\max_{\substack{B \in \mathcal{B}_{S_{v_i}} \\ G \in \mathcal{G}_{v_i}}} \frac{B \cap G}{B \cup G} \quad (4.8)$$

The average best IoU of GT videos is obtained by averaging over all ground truth videos for all queries.

- **Ave. Best IoU of GT Keyframes:** Denote the set of ground truth bounding boxes in a ground truth keyframe $\mathcal{F}_i$ for a given query as $\mathcal{G}_{F_i}$, the set of sampled bounding boxes in $\mathcal{F}_i$ as $\mathcal{B}_{S_{F_i}}$, then the *best* intersection over union (IoU) score of $\mathcal{F}_i$ can be defined as:

$$\max_{\substack{B \in \mathcal{B}_{S_{F_i}} \\ G \in \mathcal{G}_{F_i}}} \frac{B \cap G}{B \cup G} \quad (4.9)$$

The average best IoU of GT keyframes is obtained by averaging over all ground truth keyframes for all queries. Specifically for the 73-hour dataset, since each target object appears at most once in a keyframe, $|\mathcal{G}_{F_i}| = 1$.

- **Ave. #Proposals/Keyframe with IoU $> \theta$:** Given the set of proposals $\mathcal{P}_{F_i}$ in a ground truth keyframe $\mathcal{F}_i$ for a given query, the number of proposals that have an IoU $> \theta$ with the ground truth $\mathcal{G}_{F_i}$ can be calculated as:

$$\sum_{B \in \mathcal{P}_{F_i}} \mathbb{I}\left(\frac{B \cap G}{B \cup G} > \theta\right) \quad (4.10)$$

Here θ is the IoU threshold and $\mathbb{I}(\cdot)$ is the indicator function. We then average over all ground truth keyframes for all queries.

- **Ave. #Patches/Keyframe with IoU $> \theta$:** Similar to above, but $B \in \mathcal{P}_{F_i}$ is replaced with $B \in \mathcal{R}_{F_i}$, where $\mathcal{R}_{F_i}$ is the set of random patches in $\mathcal{F}_i$.

4.5.1.2 Size Prior for Video mAP

Figure 4.8 illustrates how the object size prior affects the video mAP. The total number of resulting random patches is fixed to 480, but the random template size parameters $\mathcal{M} \times \mathcal{N}$ change from 24×20 to 2×1 , resulting in increasing average size of the random patches. Here $\mathcal{M}$ is the number of cuts performed horizontally and $\mathcal{N}$ is the number of cuts performed vertically to partition each frame into random patches.

As shown in Figure 4.8 (a), for the initial coarse round video ranking using Rand-Max, as the average size of the random patches increases (i.e., moving towards the right along the x-axis), the average best IoU of ground truth videos first increases slightly and peaks at 8×4 , then starts to decrease as the average size of patches increases further. In addition, the video mAP seems to be positively correlated with the average best IoU of ground truth videos. This is not surprising as Rand-Max uses

Figure 4.8: (a) Average best IoU of ground truth videos and video mAP by Rand-Max, and (b) average best IoU of ground truth keyframes and video mAP by Rand-MP, with respect to different random template parameters $\mathcal{M} \times \mathcal{N}$.

the score of the best matched random patch to rank each video, and a higher IoU usually leads to more accurate matching.

Similar trend is also observed in Figure 4.8 (b), where the average best IoU of ground truth keyframes and video mAP from Rand-MP peak at 8×4 template size. These two metrics are also positively correlated. This is also intuitive since the trajectory score is affected by the keyframe-wise matching score, and higher keyframe-wise best IoUs generally lead to a higher matching score.

It turns out that coincidentally the 8×4 -sized random templates used to generate the random patches in previous experiments is optimal in size to best cover the ground truth objects in this dataset. This validates that the object size is a strong prior that directly affects the coverage of the ground truth, and consequently the performance of search. When set properly, random patches performs competitively in comparison with carefully designed object proposals (e.g., 8×4 on this dataset) in terms of video mAP.

Figure 4.9: Average number of bounding boxes per ground truth keyframe whose IoU > 0.5 as a function of the number of candidate bounding boxes per keyframe. (a) random proposals using the object size prior, (b) Edge Boxes proposals using the objectness prior.

4.5.1.3 Size Prior vs. Objectness Prior for Trajectory mAP

Different from video mAP, trajectory mAP concerns about accurate localization of the object in each keyframe. Although random proposals can effectively rank videos, the spatio-temporal localization is quite off compared with RVP-MP and Prop-MP (Figure 4.4). To understand what has caused the poor performance of random patches in trajectory mAP, let us look at the number of candidate bounding boxes in each keyframe that have sufficient overlap with the ground truth, resulting from both approaches. The common practice is to measure the overlap with the ground truth in terms of IoU. Because an object trajectory has to satisfy the smoothness constraints in Eq.(4.1) and Eq. (4.2), the more bounding boxes with sufficient overlap with the ground truth per keyframe, the more likely we will find a smooth trajectory with accurate localization.

Figure 4.9 (a) plots the average number of random patches per ground truth keyframe that have an IoU > 0.5, as a function of the number of random patches sampled per keyframe. Here the optimal-size random template of 8×4 are used as

in Section 4.4.3. For comparison, Figure 4.9 (b) shows the average number of object proposals per ground truth keyframe that have an $\text{IoU} > 0.5$, as a function of the number of Edge Boxes object proposals sampled per keyframe. It is clear that object proposals using the objectness prior provide a much better coverage of the ground truth with much fewer number of proposals. With 480 random patches per keyframe (which is reported in Figure 4.4), on average less than 1 patch per keyframe (0.373 to be exact) has an $\text{IoU} > 0.5$. Even with as many as 6,400 random patches per keyframe (i.e., using 200 rounds of partitions using 8×4 templates), there are on average only 5.9 random patches with an $\text{IoU} > 0.5$. On the other hand, 13 out of the top 500 object proposals have an $\text{IoU} > 0.5$. In other words, compared with the objectness prior, the object size prior cannot effectively guide the selection towards the ground truth, while given the same budgeted number of bounding boxes per keyframe, therefore the localization errors caused by these sparse samples cannot be remedied by spatial-temporal filtering alone (i.e., Max-Path search). This explains the inferior performance of Rand-MP in terms of trajectory mAP.

4.5.2 Efficiency of Spatio-Temporal Search

As shown in Table 4.6, the proposed object proposal based approaches, especially when using the objectness prior, leads to a significant speedup of the spatio-temporal search of object instances without compromising the accuracy. The efficiency can be attributed to the effectiveness of good priors in pruning the trajectory search space. So instead of considering all possible bounding boxes that could possibly form a trajectory, search is steered towards a small number of bounding boxes that are likely to contain the object of interest. Specifically, we can now afford the Max-Path search for spatio-temporal search without the help of integral images, and consequently avoiding the time-consuming process of producing pixel-wise confidence maps. Instead, scoring of a few candidate bounding boxes can be done via direct matching. And experimentally it shows that when the number of bounding boxes is reduced to 100

per keyframe, spatio-temporal search becomes over 50 times faster than before, i.e., no sampling but with integral images (Table 4.6). It suggests that good object proposals can produce a much smaller but relevant search space, whose gain in speed may outweigh the loss of not using an integral image.

As a side note, when a small vocabulary is used, integral histogram can be utilized to speed up matching of individual bounding boxes. However, since a large vocabulary is crucial for achieving a good mAP on large datasets (e.g., 1 M on the 73-hour video dataset), integral histograms would be too memory prohibitive to use.

4.5.3 3D Object Proposals and Spatio-Temporal Search

As discussed in Section 2.1.2.2 of related work, as extensions of 2D object proposals, spatio-temporal video tube proposals have been proposed in recent years for action detection and localization [81–83]. Although the output of these 3D object proposal generation methods is also trajectories, these trajectories are query independent. On the contrary, the output trajectories from spatio-temporal search presented in this and the previous chapters are query-dependent. However, the 3D object proposals can be seen as an alternative to produce a candidate object space, which is a set of candidate trajectories instead of bounding boxes discussed in this chapter. Then a given query may be matched against these candidate trajectories to produce a query-dependent trajectory ranking.

4.6 Summary

In this chapter, we have described the use of object proposals to speed up spatio-temporal object instance search in large scale videos without compromising the accuracy. It has been shown that good proposal priors are helpful in selecting candidate bounding boxes that are likely to contain the target, thus creating a reduced search

space for object trajectory search in videos. We evaluate different priors for generating the object proposals as candidates: the object size prior and the objectness prior. In terms of the video mAP, both priors are effective in produce competitive video-wise ranking compared with RVP-MP in the previous chapter, while matching only a few hundred bounding boxes per frame. However, in terms of trajectory mAP, the objectness prior turns out to be more effective than the object size prior in producing an accurate spatio-temporal localization of the target object. Experiments reveal that it is because the average number of candidate bounding boxes per keyframe that have sufficient overlap with the ground truth resulting from the object size prior is less than that from the objectness prior (Figure 4.9). However, when accurate localization is less of a concern, using the object size prior to produce random proposals for spatio-temporal search can be a good alternative to carefully designed object proposals such as the Edge Boxes. Its video mAP is comparable to that of RVP-MP and Prop-MP, and it is faster than RVP-MP (Table 4.6).

Although the candidate bounding boxes are scored by direct matching without the help of integral images, experimentally it shows that when the number of bounding boxes per keyframe is small, spatio-temporal search can be much faster than before, i.e., no sampling but with integral images. For instance, with 100 object proposals per keyframe, Max-Path search becomes over 50 times faster (Table 4.6). It suggests that good object proposals can produce a much smaller but relevant search space, whose gain in speed may outweigh the loss of not using an integral image. In addition, since pixel-wise confidence maps are not required, the proposed solution is also more memory efficient.

An interesting observation is that simply picking the best matched object proposal for each video could achieve comparable video mAP to that of more complicated matching scheme, such as RVP [2]. It shows that for a video that contains the object of interest to be retrieved, just providing one bounding box in any keyframe that has sufficient overlap with the ground truth region would be sufficient. Another observation is that window-based matching of object instances suffers from the scalability

issue when a large number of bounding boxes need to be checked per frame. When we only want to find videos that contain the query object, while accurate localization is less of a concern, can we select better than directly using the size prior or objectness prior? In this case, we do not need to ensure coverage of all instances of the target across frames, but as long as one occurrence of the object is covered, the video can be recalled. This motivates the research work in the next chapter on representative object selection from videos. We will show that given the large spatio-temporal redundancy in videos, we can further reduce the search space by discovering a few key objects that are representative of the video.

Chapter 5

Representative Object Discovery for Video Summarization and Search

5.1 Overview

With videos becoming the largest big data, it raises increasing needs to summarize, index and browse the large corpus of video content. As a common practice, videos are often described by a set of representative video frames, i.e., keyframes [50, 92, 93]. Although these keyframes can capture the important scenes, they often do not pick out the key objects from less informative backgrounds in a video¹.

Contrary to keyframes, discovering representative objects from videos can be attractive to many applications. For example, these key objects can serve as icons to establish a quick impression of the video by telling what are there. Directly related to previous chapters, they also provide a small footprint for indexing, browsing and search, so we can quickly find relevant videos to a query by matching the key objects (Figure 1.4). Besides object-level summarization and search, as these key objects are

¹A substantial portion of this chapter has appeared in [122] © 2016 IEEE.

Figure 5.1: Representative video object discovery via object proposal selection. A pool of object proposals (middle row) are first generated from video keyframes (top). Representative object proposals (bottom) are selected from the pool to summarize the video. © 2016 IEEE

essential components of higher level semantics in videos, once identified, they can also be used to recover or help understand more complicated semantics of videos, e.g., semantic segmentation after region classification [51], constructing story based video summarization by analyzing the interactions among the key objects [95], or egocentric activity recognition [123] and tracking [124].

Motivated by the recent successes of category independent object proposals for detection and weakly supervised learning [51, 52], we propose to discover representative video objects from object proposals for video summarization and search (Figure 5.1). It has been shown that generating multiple overlapping object proposals in general provides more accurate object regions than segmentation by sidestepping the harder problem of full segmentation [35, 125]. Start with a pool of frame-level object proposals produced by a high recall object proposal method [35], we formulate video object

discovery as a representative selection problem: selecting a few exemplar object proposals to reconstruct the whole pool. Although representative selection methods have been applied to video keyframe selection [6], directly applying them to objects faces many challenges. First of all, the appearance of the same object may change significantly across frames due to pose and scale variations, partial occlusions, etc. Therefore the key objects do not necessarily locate at the densest regions in the feature space. Consequently, classic density based representative selection method may not work well [44]. Moreover, since object proposals are just candidates of key objects, there may be many irrelevant and noisy proposals in the proposal pool. These outliers may significantly affect the representative selection methods based on sparse reconstruction [5, 6]. In such a case, even a further filtering of outliers as post-processing [6] may be less effective if most representatives are outliers. Without prior knowledge of the object, it is difficult to locate the key object accurately.

To address the above challenges, we propose a new formulation of sparse reconstruction based representative selection, which has the following advantages. First, it can incorporate object proposal priors when selecting the representatives. Therefore, object proposals of high prior weights, e.g., high objectness scores, are more likely to be selected as key objects, while background clutters of low weights can be suppressed. Second, our new formulation also considers the local affinity structure of the data samples by introducing a locality prior matrix to regularize the selection matrix. As an outcome, it prefers popular object proposals that appear more frequently while outlier proposals are likely to be suppressed. Third, although complex constraints are introduced, we convert our optimization into a proximal gradient problem and solve it by the fast iterative shrinkage thresholding algorithm (FISTA). As is well known, FISTA has a fast convergence rate, which is $O(1/m^2)$ in m iterations [53].

We evaluate our proposed method on both synthetic data and two benchmark video datasets in comparison with existing representative selection approaches such as K-mediod [41], sparse dictionary selection [5, 6], and density based selection [44].

The favourable results validate that our formulation fits better to the key video object summarization problem, and the selected proposals can better capture key video objects despite object appearance variations, background clutters and camera motions.

5.2 Problem Formulation

We formulate the representative video object discovery problem using object proposals as the representative selection problem. Given n object proposals extracted from a video sequence, each of the object proposal can be represented by a feature vector $\in \mathbb{R}^d$. These feature vectors are arranged as the columns of the data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$. Our goal is to find a compact subset of the n data points that are representative of $\mathbf{X} \in \mathbb{R}^{d \times n}$.

5.2.1 Preliminaries: sparse dictionary selection

In [5], a sparse dictionary selection method is proposed for abnormal event detection, which is also applicable to representative selection [6]. Instead of learning a dictionary of arbitrary atoms, [5] requires that all atoms of the dictionary must come from the actual data points. In other words, the dictionary is a compact subset of the data matrix $\mathbf{X}$. Denote the selection matrix by $\mathbf{S} \in \mathbb{R}^{n \times n}$, they solve

$$\min_{\mathbf{S} \in \mathbb{R}^{n \times n}} \frac{1}{2} \|\mathbf{X} - \mathbf{XS}\|_F^2 + \lambda_1 \|\mathbf{S}\|_{1,2}, \quad (5.1)$$

where $\|\mathbf{S}\|_{1,2} = \sum_{i=1}^n \|\mathbf{S}_{i,\cdot}\|_2$, associated with the parameter λ_1 as a regularization, and $\|\mathbf{S}_{i,\cdot}\|_2$ is the l_2 norm of the i^{th} row of $\mathbf{S}$. Once (5.1) is solved by the proximal gradient method [53], the dictionary is constituted by selecting data points whose corresponding $\|\mathbf{S}_{i,\cdot}\|_2 \neq 0$. These data points can be considered as representatives of the dataset. Consequently, the video sequence can be summarized by the object proposals corresponding to these data points.

Similar to [6], [5] can be adapted to selecting any number of representatives by measuring and ranking the selection confidence of the i^{th} data point $\mathbf{x}_i$ according to $\|\mathbf{S}_{i,\cdot}\|_2$.

5.2.2 Weighted sparse dictionary selection

Note that in (5.1), all data points are treated equally. However, a good video summarization can certainly benefit from prior knowledge from application domain or user specifications. For instance, when summarizing egocentric videos, objects that the subject interact with are usually more important than others [50], while in surveillance videos from a fixed camera, moving foreground objects likely carry more weights than static background objects. To better leverage priors, we propose a simple extension to [5] for representative selection

$$\min_{\mathbf{S} \in \mathbb{R}^{n \times n}} \frac{1}{2} \|\mathbf{X} - \mathbf{XS}\|_F^2 + \lambda_1 \sum_{i=1}^n \frac{1}{\rho_i + \epsilon} \|\mathbf{S}_{i,\cdot}\|_2, \quad (5.2)$$

where ρ_i is the prior selection weight for the i^{th} sample, and ϵ is a tiny number to avoid dividing by zero.

Similar to [5], the problem of weighted sparse dictionary selection can also be optimized by the proximal gradient iteration [53], but needs a proximal decomposition [126].

5.2.3 Locally linear reconstruction (LLR) induced sparse dictionary selection

As indicated in [6], sparse dictionary selection prefers keeping the vertices of convex hull spanned by input data to make sure each sample can be reconstructed at a low cost. It is thus extremely sensitive to noise. To mitigate this issue, we encourage local reconstruction for each sample to improve the robustness of representative selection.

5.2.3.1 Locality prior of linear reconstruction

Inspired by locally linear embedding [113], we build a locality prior matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ of representative selection based on that each sample $\mathbf{x}_i$ is only allowed to be locally linear reconstructed by its k -NNs, $\mathcal{N}(\mathbf{x}_i) \setminus \mathbf{x}_i$:

$$\begin{aligned} \min_{\mathbf{W} \in \mathbb{R}^{n \times n}} \quad & \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \sum_{j: \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \setminus \mathbf{x}_i} w_{ji} \mathbf{x}_j\|_2^2, \\ \text{s.t.} \quad & \sum_{j: \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \setminus \mathbf{x}_i} w_{ji} = 1, \\ & w_{ji} = 0, \forall \mathbf{x}_j \notin \mathcal{N}(\mathbf{x}_i) \setminus \mathbf{x}_i. \end{aligned} \quad (5.3)$$

This above objective can also be rewritten as $\frac{1}{2} \|\mathbf{X} - \mathbf{XW}\|_F^2$. Problem (5.3) can be solved by a constrained least squares optimization [113].

Specifically, let $i(\mathcal{N}) = \{j_1(i), j_2(i), \dots, j_k(i)\}$ index the k -NNs of $\mathbf{x}_i$, then $\mathbf{1}^T \mathbf{w}_{i(\mathcal{N})} = 1$, as $\mathbf{w}_{i(\mathcal{N})}$ collects all the reconstruction weights from the k -NNs of $\mathbf{x}_i$. We need to solve that $\forall i = 1, 2, \dots, n$,

$$\min_{\mathbf{w}_{i(\mathcal{N})} \in \mathbb{R}^k} \frac{1}{2} \|\mathbf{x}_i \mathbf{1}^T - \mathbf{X}_{i(\mathcal{N})} \mathbf{w}_{i(\mathcal{N})}\|_2^2, \text{ s.t. } \mathbf{1}^T \mathbf{w}_{i(\mathcal{N})} = 1, \quad (5.4)$$

which is a constrained least squares problem, and has a closed-form solution [127]:

$$\mathbf{w}_{i(\mathcal{N})} = \frac{\mathbf{G}_i^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{G}_i^{-1} \mathbf{1}}, \quad (5.5)$$

where $\mathbf{G}_i = \|\mathbf{x}_i \mathbf{1}^T - \mathbf{X}_{i(\mathcal{N})}\|_2^2 = (\mathbf{x}_i \mathbf{1}^T - \mathbf{X}_{i(\mathcal{N})})^T (\mathbf{x}_i \mathbf{1}^T - \mathbf{X}_{i(\mathcal{N})})$ is the local Gram (covariance) matrix of $\mathbf{x}_i$.

More efficiently, we can equivalently solve the linear system of equations (which can be derived from the method of Lagrange multipliers) in lieu of explicit matrix inversion,

$$\mathbf{G}_i \mathbf{w}_{i(\mathcal{N})} = \mathbf{1}, \quad (5.6)$$

and then normalize $\mathbf{w}_{i(\mathcal{N})}$ to hold the affine constraint.

To avoid an ill conditioned $\mathbf{G}_i$ when $k > d$, we let

$$\mathbf{G}_i = \mathbf{G}_i + \frac{\Delta^2}{k} \cdot \text{tr}(\mathbf{G}_i) \cdot \mathbf{I}, \quad (5.7)$$

where $\Delta^2 \ll 1$ [127].

5.2.3.2 LLR-induced sparse dictionary selection

To introduce locality information of data for representative selection, we propose a new optimization problem combined with the locality prior matrix $\mathbf{W}$ in the following:

$$\min_{\mathbf{S} \in \mathbb{R}^{n \times n}} \frac{1}{2} \|\mathbf{X} - \mathbf{XS}\|_{\text{F}}^2 + \lambda_1 \sum_{i=1}^n \frac{1}{\rho_i + \epsilon} \|\mathbf{S}_{i,\cdot}\|_2 + \frac{1}{2} \lambda_2 \|\mathbf{S} - \mathbf{W}\|_{\text{F}}^2, \quad (5.8)$$

where the third term regularizes the selection matrix $\mathbf{S}$ by $\mathbf{W}$, and λ_2 is a locality regularization parameter. Hence, data samples are preferable to be reconstructed by nearby representatives. Moreover, dense samples are more likely to be selected as representatives than sparse noise, as the former can contribute more to the reconstruction of surrounding samples in comparison with the latter.

Problem (5.8) is complex due to three optimization terms. But we will show that it can be converted into a proximal gradient optimization and solved by the FISTA method [53] through a proximal decomposition [126], which converges fast with rate $O(1/m^2)$ in m iterations.

5.2.3.3 Optimization

To solve our optimization problem (5.8), we first expand the objective function ($\mathcal{O}$ for short) and rewrite it as

$$\begin{aligned} \mathcal{O} = & \frac{1}{2} \text{tr}\{\mathbf{X}^T \mathbf{X} - 2(\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{W}^T) \mathbf{S} \\ & + \mathbf{S}^T (\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{I}) \mathbf{S} + \lambda_2 \mathbf{W}^T \mathbf{W}\} \\ & + \lambda_1 \sum_{i=1}^n \frac{1}{\rho_i + \epsilon} \|\mathbf{S}_{i,\cdot}\|_2. \end{aligned} \quad (5.9)$$

We then let

$$f(\mathbf{S}) = \frac{1}{2} \text{tr} \{ \mathbf{X}^T \mathbf{X} - 2 (\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{W}) \mathbf{S} + \mathbf{S}^T (\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{I}) \mathbf{S} + \lambda_2 \mathbf{W}^T \mathbf{W} \}, \quad (5.10)$$

and

$$g(\mathbf{S}) = \lambda_1 \sum_{i=1}^n \frac{1}{\rho_i + \epsilon} \|\mathbf{S}_{i,\cdot}\|_2. \quad (5.11)$$

Thus, we decompose the objective function $\mathcal{O}$ into two convex functions, with f smooth and g nonsmooth, i.e.,

$$\mathcal{O} = f(\mathbf{S}) + g(\mathbf{S}), \quad (5.12)$$

for which, we can apply the proximal gradient method, FISTA [53]. It then becomes iteratively solving

$$\text{prox}_{\mathcal{R}}(\mathbf{Z}) = \arg \min_{\mathbf{S} \in \mathbb{R}^{n \times n}} \frac{1}{2} \|\mathbf{S} - \mathbf{Z}\|_F^2 + \frac{1}{L} g(\mathbf{S}), \quad (5.13)$$

where

$$\begin{aligned} \mathbf{Z} &= \mathbf{S} - \frac{1}{L} \frac{\partial}{\partial \mathbf{S}} f(\mathbf{S}) \\ &= \mathbf{S} + \frac{1}{L} \{ -(\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{W}) + (\lambda_2 \mathbf{I} + \mathbf{X}^T \mathbf{X}) \mathbf{S} \}, \end{aligned} \quad (5.14)$$

and L is the smallest Lipschitz constant, which equals to the spectral radius ($r(\cdot)$) of $\lambda_2 \mathbf{I} + \mathbf{X}^T \mathbf{X}$, i.e.,

$$L = r(\lambda_2 \mathbf{I} + \mathbf{X}^T \mathbf{X}) = \lambda_2 + r(\mathbf{X}^T \mathbf{X}). \quad (5.15)$$

We next follow the decomposition tactic in [126], then Problem (5.13) is solvable, and for $i = 1, 2, \dots, n$,

$$\mathbf{S}_{i,\cdot} = \arg \min_{\mathbf{s} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{s} - \mathbf{Z}_{i,\cdot}\|_2^2 + \frac{\lambda_1}{L(\rho_i + \epsilon)} \|\mathbf{S}_{i,\cdot}\|_2, \quad (5.16)$$

After applying soft-thresholding [128] to the above n group lasso signal approximators, we have, for $i = 1, 2, \dots, n$,

$$\mathbf{S}_{i,\cdot} = \mathbf{Z}_{i,\cdot} \max \left\{ \left(1 - \frac{\lambda_1}{L(\rho_i + \epsilon)} \right), 0 \right\}. \quad (5.17)$$

Algorithm 2: LLR-induced Weighted Sparse Dictionary Selection (5.8).

Require: $\mathbf{X}$, $\{\rho_i\}_{i=1}^n$, k , λ_1 , λ_2
Ensure: $\mathbf{S}$
 1: $L \leftarrow \lambda_2 + r(\mathbf{X}^T \mathbf{X})$
 $\triangleright$ Lipschitz constant (Equation (5.15))
 2: $\mathbf{S} \leftarrow \mathbf{0}$, $\mathbf{V} \leftarrow \mathbf{S}$, $t \leftarrow 1$
 $\triangleright$ Initialization
 3: **repeat**
 4: $\mathbf{Z} \leftarrow \mathbf{V} + \frac{1}{L} \left\{ -(\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{W}) + (\lambda_2 \mathbf{I} + \mathbf{X}^T \mathbf{X}) \mathbf{V} \right\}$
 $\triangleright$ Equation (5.14)
 5: $\mathbf{U} \leftarrow \mathbf{S}$, $\mathbf{S}_{i,\cdot} \leftarrow \mathbf{Z}_{i,\cdot} \max\{(1 - \frac{\lambda_1}{\|\mathbf{Z}_{i,\cdot}\|_2}), 0\}$, $i = 1, 2, \dots, n$
 $\triangleright$ Equation (5.17)
 6: $\tau = t - 1$, $t \leftarrow (1 + \sqrt{1 + 4t^2})/2$
 7: $\mathbf{V} \leftarrow \mathbf{S} + \tau(\mathbf{S} - \mathbf{U})/t$
 8: **until** convergence

We show the representative selection procedure in Algorithm 2, where we integrate a decomposed soft-thresholding strategy into an accelerated proximal gradient procedure, which is known to have a fast convergence rate $O(1/m^2)$ in m iterations.

5.2.3.4 Parameter setting

Sparsity regularization parameter λ_1 . As in Algorithm 2, we initialize $\mathbf{S}$ by a zero matrix. Then according to (5.14), after the first iteration, we have

$$\mathbf{Z} = \frac{1}{L} \left\{ -(\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{W}) \right\}. \quad (5.18)$$

As indicated by the thresholding of $\mathbf{Z}$ in (5.17), when λ_1 is large enough, e.g., $\lambda \geq \lambda_1^{\max}$, we obtain $\mathbf{S} = \mathbf{0}$, which means we select nothing. Therefore, to avoid an empty selection, we let $\lambda_1 \leq \lambda_1^{\max}$ and solve $\lambda_1^{\max}$ by substituting $\mathbf{S} = \mathbf{0}$ into (5.17) as follows:

$$\begin{aligned}
 \lambda_1^{\max} &= L \max_{0 \leq i \leq n} \{(\rho_i + \epsilon) \|\mathbf{Z}_{i,\cdot}\|_2\} \\
 &= \max_{0 \leq i \leq n} \{(\rho_i + \epsilon) \|\mathbf{x}_i^T \mathbf{X} + \lambda_2 \mathbf{W}_{i,\cdot}\|_2\}.
 \end{aligned} \quad (5.19)$$

In our experiments, we let $\lambda_1 = \frac{\lambda_1^{\max}}{\alpha_1}$ and tune α_1 between the interval $[2, 30]$. Given λ_2 , a smaller α_1 indicates a larger λ_1 , which implies a sparser selection.

Locality regularization parameter λ_2 . Let us consider the LLR-induced sparse selection in (5.8). When $\lambda_2 = 0$, the problem becomes a weighted sparse dictionary selection as in (5.2). When $\lambda_2 = +\infty$, it sparsely selects representatives based on the rows of $\mathbf{W}$. Furthermore, as shown in (5.14), λ_2 balances the contributions of $\mathbf{W}$ and $\mathbf{X}\mathbf{X}^T$ to the proximal operation in (5.13). For ease of tuning λ_2 , we let

$$\lambda_2 \leftarrow \kappa \times \frac{r(\mathbf{X}^T \mathbf{X})}{r(\mathbf{W})}, \quad (5.20)$$

where we set $\kappa = 0.02 \times 5^{\alpha_2}$ and α_2 between $[-2, 2]$ in our experiments.

5.3 Experiments on Synthetic Data

We first evaluate the effectiveness of the locality prior in handling outliers on synthetic data, in comparison with the Sparse Dictionary Selection (SDS) [5] and Sparse Modeling Representative Selection (SMRS) [6]. We refer to our proposed method as Locally Linear Reconstruction induced Sparse Dictionary Selection (LLR-SDS).

We consider the noisy data shown in Figure 5.2, which consists of data points in three clusters and uniform background noise. The top 30 representatives found from the 2,018 points by each compared method are shown in Figs. 5.2 (a)-(c). As can be seen, both SDS and SMRS select the outlier points at the border of the convex hull, showing these two methods are sensitive to noise. This is because those points contribute a lower linear reconstruction cost to the dataset than others, which meets the requirement of dictionary selection. As SMRS has a post-processing to filter outliers [6], we also run its outlier removal for comparison in Figure 5.2 (d). Since most selection of SMRS are outliers as shown in Figure 5.2 (c), it is difficult to improve the results by post-processing. In contrast, our proposed LLR-SDS method considers a

Figure 5.2: Data points in three clusters with background uniform noise (blue dots) and the representatives (red circles) found by (a) the proposed LLR-SDS method (b) SDS [5], (c) SMRS [6] and (d) SMRS with outlier removal [6]. © 2016 IEEE

locality constraint in addition to the linear reconstruction cost for dictionary selection. As a result, it can reject most noisy outliers and select the points in the clusters. For our method, we use $k = 5$ nearest neighbors to build the locality prior matrix $\mathbf{W}$, and set the sparsity regularization parameter $\alpha_1 = 5$ for λ_1 , and the locality regularization parameter $\alpha_2 = 1$ for λ_2 . The same sparsity regularization parameter is used for SDS and SMRS.

5.4 Experiments on Real Datasets

Next, we test our approach on two real datasets to evaluate its applicability to topical object discovery and object search in videos.

5.4.1 Settings

Although our approach is agnostic to the object proposal method, we use Edge Boxes [35] with default settings to generate category independent object proposals for a controlled comparison with existing methods on all datasets. In light of the good performance reported using features from the top layers of deep convolutional neural networks (CNNs) for many computer vision tasks such as object detection [51], image retrieval [55] and object instance search [21], we choose to represent each object proposal by the CNN feature after dimension reduction. Specifically, we take the 4096-dimensional output from the fully connected layer 6 of Caffe [11], using the pre-trained model on ILSVRC 2012 without fine-tuning. The CNN features are then reduced to 256-dimension by PCA and whitening [129]. The same set of CNN-PCA features are used for comparison with existing work unless noted otherwise.

5.4.2 Baseline algorithms

The proposed LLR-SDS is compared with a variety of different methods for representative selection, including Objectness [35], K-medoids [41], Density [44], Sparse Modeling Representative Selection (SMRS) [54], Sparse Dictionary Selection (SDS) [5], Locally Linear Reconstruction (LLR) and Latent Dirichlet Allocation with Word Co-occurrence prior (LDA-WCP) [108]. Each baseline approach is briefly introduced below.

Objectness refers to directly ranking object proposals based on the objectness scores from Edge Boxes.

K-medoids is a standard clustering algorithm for selecting representatives, where K is the number of clusters (or representatives). Similar to K-means, K-medoids aims to find a partitioning of the dataset into K clusters, which minimizes the sum of within-cluster distances. The l_2 distance is used in the experiments.

Density peak search (Density) [44] is a recently proposed density-based clustering method. It is chosen as a baseline because it has the advantage of discovering clusters of non-spherical shape over K-medoids and can automatically find the correct number of clusters regardless of the dimensionality of the manifold in which data are embedded. For Density, representatives are ranked and selected according to the parameter γ , which is the product of local density and minimum distance between the point and any other point with higher density [44]. Experimentally, the cut-off kernel to estimate local density performs better than the Gaussian kernel on the two datasets, therefore it is used for comparison.

Sparse Modeling Representative Selection (SMRS) [54] is a state-of-the-art sparse dictionary selection method that finds representatives via linear subspace learning. It is a popular sparse dictionary selection method via linear subspace learning, which learns sparse self-representation of dataset to find representatives. We

follow the authors to tune its parameter α , which controls the sparsity regularization, similar to α_1 of our method. To make a fair comparison, results reported in this section are without the outlier detection post-processing.

Sparse Dictionary Selection (SDS) [5] refers to ranking and selecting object proposals according to the l_2 norms of the rows of the selection matrix $\mathbf{S}$ obtained from the algorithm proposed in [5] (Section 5.2.1).

Locally Linear Reconstruction (LLR) selects object proposals according to the l_2 norms of the rows of the LLR prior matrix $\mathbf{W}$ (Section 5.2.3.1). Therefore, it favors data points with sufficient density over the outliers. Given k , which is the number of nearest neighbors that are allowed to linearly reconstruct each sample, $\mathbf{W}$ is found by solving a constrained least squares problem [113].

Latent Dirichelet Allocation with Word Co-occurrence prior (LDA-WCP) [108] is a topical model for discovery topical video objects. We adapt LDA-WCP to object proposal selection by selecting the highest score segment in the entire video for each topic to summarize the video.

Among the above, Density, SMRS, Objectness and LDA-WCP are tested using the source code from respective authors. We use the MATLAB implementation of K-medoids and our own implementations of SDS and LLR.

5.4.3 Evaluation Metric

Denote the set of selected object proposals from a video as $\mathbf{P}$ and suppose a video contains t different key objects. For the i_{th} key object, suppose the set of ground truth bounding boxes in all keyframe is $\mathbf{G}_i$, and the *best* intersection over union (IoU) score $\mathbf{S}$ with the i_{th} key object is defined as

$$\mathbf{S}(\mathbf{P}, \mathbf{G}_i) = \max_{\substack{p \in \mathbf{P} \\ g \in \mathbf{G}_i}} \mathbf{S}(p, g) = \max_{\substack{p \in \mathbf{P} \\ g \in \mathbf{G}_i}} \frac{p \cap g}{p \cup g} \quad (5.21)$$

The recall of a video is determined by the number of key objects that are recalled, i.e.,

$$\mathbf{R} = \frac{\sum_{i=1}^t \mathbb{I}(\mathbf{S}(\mathbf{P}, \mathbf{G}_i) \geq \theta)}{t} \quad (5.22)$$

Here θ is the overlap threshold and $\mathbb{I}(\cdot)$ is the indicator function. Average recall is the mean of the recall of all videos.

5.4.4 Topical Video Object Discovery

We first demonstrate the effectiveness of the proposed LLR-SDS in discovering multiple topical objects in videos. We run our experiments on the "multiple" object subset of the topical video object discovery dataset [108], which consists of 10 commercial video sequences from YouTube. Each video contains multiple well-defined topical objects such as the product logo and has multiple shots. As in [108], keyframes from each video are sampled at two frames per second. For each keyframe, we take the top 100 object proposals according to the objectness score [35] and summarize from them. The overlap threshold θ is set to 0.5.

Note that LDA-WCP requires a predefined number of topics for each video, which is set to 8 for all videos in [108]. Hence, for a fair comparison, we also select 8 object proposals by each of the other methods to summarize a video. SMRS is tested on a range of $\alpha \in \{2, 3, 5, 10, 20, 30\}$ and we report the best average recall obtained. For our proposed LLR-SDS, we fix $k = 3$ nearest neighbors to construct the LLR prior matrix $\mathbf{W}$ (Section 5.2.3.1), $\alpha_1 = 2$ for λ_1 and $\alpha_2 = -1$ for λ_2 for all videos (Section 5.2.3.4). The same $k = 3$ is used for LLR and the same $\alpha_1 = 2$ is used for SDS. We also evaluate the effectiveness of prior weights by simply using the objectness score from Edge Boxes [35] as ρ_i (Eq. 5.8) for each object proposal. We refer to our method with objectness weights as LLR-wSDS.

Table 5.1 compares our approach with other methods in terms of the average recall of all videos. Without objectness prior weights, our proposed LLR-SDS and SMRS

	Average Recall@8
Objectness [35]	0.173
LLR	0.243
Density [44]	0.360
LDA-WCP [108]	0.360
K-medoids	0.410
SDS [5]	0.417
SMRS [6]	0.430
LLR-SDS (Ours)	0.430
LLR-wSDS (Ours)	0.547

Table 5.1: Average recall when selecting 8 object proposals. Methods are sorted by increasing average recall@8. Our method with objectness as prior weights (LLR-wSDS) achieves the best average recall among all. © 2016 IEEE

achieve the same highest average recall of 0.43. Although using the objectness score directly for summarization performs poorest among all, integrating it into our LLR-SDS formulation as prior weights further improves LLR-SDS and SMRS by 27.2%. A close examination of the Objectness results reveals that since object proposals are highly redundant across frames, an object proposal that is scored highest in one frame usually scores the highest in other frames as well. Therefore when selecting few (i.e., top 8) object proposals purely based on the objectness scores, multiple snapshots of one or two dominant object(s) are often picked out but the other topical objects are missed. Note that LDA-WCP is based on quantized SIFT features, while the others except for Objectness are based on CNN features. This could account for the mediocre performance of LDA-WCP.

Figure 5.3 shows an example video with representative object proposals selected by different methods. The 8 object proposals selected by each method are displayed in rank order in each row, except for LDA-WCP and K-medoids, which produce no ranks for the selection. There are 5 topical objects in this commercial: the blue juice drink, the green juicy juice, the boy, and 2 toy trucks. Both LLR-SDS and LLR-wSDS capture 4 out of 5 (missing one of the toy trucks) using only 8 proposals. With

Figure 5.3: Video summarization for topical object discovery: example representative objects selected by different methods. Keyframes are shown in the top row. Each following row shows the 8 representative object proposals selected by different methods in rank order (except for K-medoids and LDA-WCP). In the second row, regions in normal color indicate topical objects discovered by LDA-WCP, while those in green are background regions. © 2016 IEEE

objectness weights, LLR-wSDS seems to improve the ranking of object proposals with more accurate coverage of the ground truth over LLR-SDS. For instance, the 1_{st} object proposal selected by LLR-wSDS provides a more accurate coverage of the blue juice drink than the 1_{st} object selected by LLR-SDS.

5.4.5 Video Summarization for Object Search

Next we evaluate the applicability of LLR-SDS to object instance search by summarizing shots of a feature movie by object proposals. Groundhog Day [26] is a benchmark dataset for evaluating the object retrieval performance over the entire movie on a ground truth test set of six object queries. It consists of 5,640 keyframes (752 shots)

from the full length feature film "Groundhog Day". Contrary to the Topical Video Object Discovery dataset, the target objects in this dataset are usually small and the scenes are much cluttered.

For each keyframe, the top 200 object proposals with an aspect ratio $\in [0.25, 4]$, and a minimal size of 30×30 pixels are extracted. Because object locations in the keyframes are not provided, bounding box locations in all ground truth keyframes are manually annotated by a student helper. The algorithms are run on all ground truth shots that have ≥ 8 keyframes (Table 5.2). SDS is tested in a range of $\alpha \in \{2, 3, 5, 10, 15, 20, 30\}$ and $\alpha = 10$ is selected for comparison because it produces the best average recall. For the proposed LLR-SDS, k is fixed at 5 nearest neighbors to construct the LLR prior matrix $\mathbf{W}$ and $\alpha_1 = 15$ for calculating λ_1 . The objectness score as prior weights (as in Section 5.4.4) has also been tested, which, however, does not boost the performance on the Groundhog Day dataset. It is likely due to the scene clutters and generally low resolutions of the target objects in this dataset. Therefore, we evaluate LLR-SDS using equal weights on this dataset. It is worth noting though that other priors could be effective on this dataset such as object size and scene context. Unless noted otherwise, the overlap threshold θ is set to 0.7 in all following experiments (Eq. 5.21), as a greater overlap with the ground truth generally leads to a higher matching score and increases the chance for an object to be retrieved.

5.4.5.1 Results

Figure 5.4 illustrates the average recall on all ground truth shots of the six objects with the selection ratio $\eta \in [0.001, 0.2]$. It is shown that the proposed LLR-SDS outperforms the other methods and achieves an average recall of 0.74 with as few as 10% of the object proposals. With 14% of the object proposals, it achieves an average recall of 0.78, while the average recall obtained using all object proposals is 0.80. It also improves upon selection by SDS or LLR alone.

 Figure 5.4: Average recall with selection ratio $\eta \in [0.001, 0.2]$. © 2016 IEEE

In addition, Figure 5.5 shows the relationship between the average recall and overlap ratio θ of our proposed LLR-SDS, when $\eta = 0.01$.

To evaluate the sensitivity of α_2 for λ_2 (Section 5.2.3.4), we test LLR-SDS with $\alpha_2 \in [-3, 1]$. Figure 5.6 plots the average recall curves obtained with respect to the two extreme cases discussed in Section 5.2.3.4: i.e., (1) $\lambda_2 = 0$ and (2) $\lambda_2 = +\infty$. The former is equivalent to the sparse dictionary selection (SDS) (Eq. 5.2), while the latter produces a selection matrix $\mathbf{S}$ similar to the LLR prior matrix $\mathbf{W}$, where we rank and select object proposals by the l_2 norms of the rows of $\mathbf{W}$ (LLR). We fix $\alpha_1 = 15$ for LLR-SDS and SDS, and $k = 5$ for LLR-SDS and LLR. It is observed

Figure 5.5: Evaluation of the relationship between the average recall and overlap ratio θ ($\eta = 0.01$). © 2016 IEEE

Figure 5.6: Average recall when the regularization parameter $\alpha_2 \in [-3, 1]$. A small α_2 (i.e., -3) leads to a small λ_2 , which produces a curve similar to that of SDS; while $\alpha_2 = 1$ leads to a solution similar to that of LLR (Section 5.2.3.4). © 2016 IEEE

Object	#shots	SDS	LLR	LLR-SDS
Red clock	9	23.31 %	38.64 %	12.48%
Black clock	8	15.86%	16.68%	12.99%
Frames sign	6	1.65%	47.88%	2.25%
Digital clock	14	7.92%	9.20%	5.49%
Phil sign	9	6.51%	23.18%	5.43%
Microphone	8	7.13%	4.77%	0.97%
Average	9	10.40%	23.39%	6.60%

Table 5.2: Average percentage of object proposals required to cover the ground truth object. © 2016 IEEE

that when $\alpha_2 \leq -3$, the recall curve of LLR-SDS converges to that of SDS. On the other hand, when $\alpha_2 \geq 0$, the recall curve of LLR-SDS almost entirely overlaps with that of LLR. Experimentally, on the Groundhog dataset, LLR-SDS achieves higher average recall than either SDS or LLR when $\alpha_2 \in [-2, -0.5]$.

We further evaluate the effectiveness of LLR-SDS in terms of the percentage of proposals required to provide accurate localization of the ground truth object, in comparison with SDS and LLR. An object proposal is considered to accurately locate the ground truth if its IoU with the ground truth ≥ 0.7 or it is the best IoU of all object proposals in a video. Table 5.2 shows that on average, with as few as 6.60% of all object proposals, LLR-SDS is able to cover the object of interest when summarizing a shot. Except for the Frames Sign, LLR-SDS requires fewer object proposals than both SDS and LLR to ensure accurate localization of the ground truth, while LLR requires the most.

Figure 5.7 shows visual comparisons of our results with others on all shots of Microphone, when selecting $\eta = 0.5\%$ of all object proposals. For each shot, we visualize the object proposal that has the highest IoU with the ground truth among all selected. In general, our method produces summarizations that more accurately locate the ground truth than others in all 8 shots of Microphone.

Figure 5.7: Video summarization for object search: visual results of the 8 shots of Microphone from Groundhog Day ($\eta = 0.5\%$ of all object proposals are selected). For each shot, the object proposal with the best IoU with the ground truth (GT) is shown in the yellow bounding box and the GT is in red. Missed shots are highlighted by red rectangles. Overall our method produces summarizations that more accurately capture the GT. Note that given a shot, the best IoU object proposal selected by different methods may come from different keyframes. Best viewed in color and magnification. © 2016 IEEE

5.5 Discussion

5.5.1 Object Discovery for Video Search

As experimentally shown in the previous chapter, when searching objects in videos, the large visual redundancies in videos calls for efficient sampling of the search space (which may also be the case when dealing with image collections taken within the same scene). Traditionally videos are sampled into a collection of key frames, then methods for object search in image datasets can be applied. Alternatively, we can treat videos as a collection of candidate object windows (e.g., object proposals), the goal is thus to reduce the redundancy in the candidate pool. To this end, one can either prioritize search by pre-selection (e.g., sampling bounding boxes with priors as in Chapter 4, and discovering representative object proposals as in this chapter), or prune regions in the run-time based on the query. Both Chapter 4 and this chapter

focus on the former. Although a few works have explored object proposals for object search in images or videos, the scalability of these approaches either depends on the objectness score from the proposal generation method (i.e., only picking the top few proposals according to the objectness score), or hashing each proposal’s feature into a binary string. In comparison, by formulating the problem as representative object proposal selection via sparse reconstruction, we can better capture the key objects and scale easily with a given budget.

5.6 Summary

In this chapter we discover key objects from videos by selecting representative object proposals from a pool of frame-wise object proposals. These key objects can serve as video icons and establish a brief impression of the video. By telling what objects appear in the video, we can use these key objects to help search, browse, and index the video clips. To select key objects, we have proposed a new formulate of sparse dictionary selection to select representative object proposals, i.e., locally linear reconstruction induced sparse dictionary selection (LLR-SDS). The new formulation considers both object proposal priors and locality priors in the feature space thus can better handle outlier proposals when identifying the key objects. Results on synthetic data and two benchmark real datasets validate the advantages of the proposed approach when compared with existing representative selection methods.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

As the basic comprehensible components of videos, the presence of certain objects are important for searching and understanding video content. In this thesis, we study how an object-driven approach can benefit video data analysis. We focus on two fundamental problems in the video-object context, video object search and discovery.

6.1.1 Video Object Search

Chapter 3 and Chapter 4 are devoted to video object search, where we address the following three challenges:

- (1) The appearance variations of the same object across video frames.
- (2) The false positives introduced by background clutters in videos.
- (3) Search efficiency in large video datasets.

To address the first two challenges, in Chapter 3, we propose to formulate the problem of object instance search in videos as a problem of spatio-temporal trajectory search to enforce the spatio-temporal coherency across video frames. We also extend the dynamic programming based Max-Path search [33] for detection to locating the top- K trajectories in large video repositories, which reduces the complexity of trajectory search from exponential to linear to the video volume size. For efficient matching, we present two matching schemes, both of which are compatible with the Max-Path search. The first is based on *dense* confidence maps (i.e., Randomized Visual Phrases with Max-Path search (RVP-MP) proposed in Chapter 3), and the second is based on *coarse* object proposals (i.e., random proposals based on object size priors (Rand-MP) and proposals from principled Edge Boxes [35] (Prop-MP) proposed in Chapter 4). Experimental results have been presented to demonstrate the effectiveness of the proposed approaches.

Our main contributions in video object search are as follows:

- Compared with conventional frame-based methods, the proposed trajectory-based formulation can effectively improve matching accuracy by enforcing the spatio-temporal coherency across video frames. It helps recover missed detections due to object appearance variations (i.e., challenge (1)) and filter false alarms caused by background clutters (i.e., challenge (2)). Moreover, the resulting trajectories can be used as an alternative to frames for measuring object occurrences and the search performance, which is especially important for applications where finding trajectories is more critical than finding individual frames (Figure 1.5).
- The Max-Path based trajectory search is efficient with a complexity linear to the video volume size. It is also compatible with both dense confidence maps and coarsely sampled object proposals .

- Systematic evaluation on a 73-hour video dataset demonstrates that the object proposal based matching scheme can achieve comparable accuracy to RVP-MP (Chapter 3) with a 3 to 60-time speedup (Table 4.6).

Both Chapter 3 and Chapter 4) validates that utilizing the spatio-temporal coherency is beneficial to object instance search in videos. Yet practically, for the sake of efficiency, we only use object trajectories to re-rank top videos after the coarse round search by frame-wise segmented regions (Chapter 3) or object proposals (Chapter 4). Further speed-up of trajectory search is likely to be necessary to allow for direct spatio-temporal search on all database videos (i.e., skipping the coarse round).

The search efficiency boils down to the number of possible bounding boxes that may contain a target object in each video. As the object can be of any size and at any location in any frame, a naive search would be costly. Although Chapter 3 allows for fast scoring of *any* possible bounding box that may contain the target object, with the help of integral images generated from pixel-wise confidence maps, producing these confidence maps in the first place is time-consuming. Object proposals, on the other hand, help prune the search space to a much smaller number of bounding boxes likely to contain the target object (Chapter 4). Since a query only needs to be matched against these few bounding boxes, we can do so fast enough without using integral images, hence save the time of calculating confidence maps. And the use of object proposals is shown to be surprisingly effective in improving speed and memory efficiency upon the RVP-MP method in Chapter 3.

However, using object proposals can be a double-sided sword. When the proposals are not sufficiently reliable, the resulting candidate bounding boxes may not provide a good coverage of the target objects, hence decrease the accuracy. As shown in Table 4.3, increasing the number of object proposals beyond 300 per frame (i.e., adding more lower ranked proposals) would introduce more distractors and cause the mAP to drop. Future approaches could benefit from deriving better object proposals

to improve the recall of objects for search. Possible directions including exploring the spatio-temporal context, or query-dependent object proposal selection. It is worth noting that we index the videos by quantized local features of each frame instead of the global feature of each bounding box. Therefore, although query-dependent object proposal selection would result in a different set of sampled bounding boxes for each query, we do not need to re-index the dataset between runs.

6.1.2 Video Object Discovery

Chapter 5 is devoted to video object discovery. Different from keyframes, which capture the important scenes, we would like to pick out the representative objects in a video from less informative backgrounds. To avoid the hard problem of locating objects in videos, we propose to discover the key objects in a video from off-the-shelf frame-wise object proposals. The main challenges are:

- (1) Since the appearance of the same object may change significantly across frames due to pose and scale variations, partial occlusions, etc., the key objects do not necessarily locate at the densest regions in the feature space. Consequently, classic density based representative selection method may not work well [44].
- (2) The proposal pool may contain many irrelevant or noisy object proposals, which may significantly affect the representative selection methods based on sparse reconstruction [5, 6]. In such a case, even a further filtering of outliers as post-processing [6] may be less effective if most representatives are outliers. Without prior knowledge of the object, it is difficult to locate the key object accurately.

To overcome the object appearance variations across frames and noisy proposals from the background, we present a new formulation of sparse dictionary selection, named locally linear reconstruction induced sparse dictionary selection (LLR-SDS),

which considers both object proposal priors and locality priors in the feature space. Our contributions include:

- A new formulation of sparse reconstruction based representative selection is proposed, which incorporates object proposal priors and locality prior in the feature space when selecting representatives. Consequently, object proposals of high prior weights (e.g., high objectness scores) are more likely to be selected as key objects, while background clutters of low weights can be suppressed. In addition, by introducing a locality prior matrix that considers the local affinity structure of the data to regularize the selection matrix, the selection would prefer popular object proposals that appear more frequently while outlier proposals are likely to be suppressed. Comparisons with existing representative selection approaches such as K -mediod, sparse dictionary selection [5, 54] and density based selection [44] validate that this formulation can better capture the key video objects despite appearance variations, cluttered backgrounds and camera motions.
- Despite complex constraints, the proposed optimization problem can be efficiently solved by the fast iterative shrinkage thresholding algorithm (FISTA) [53] once converted into a proximal gradient problem.
- Our proposed formulation is generic. The F-norm regularization in (5.8) can also incorporate other prior weight matrix $\mathbf{W}$ to the selection matrix $\mathbf{S}$, without being limited to the LLR-induced locality prior we used in Chapter 5.

The discovered representative objects can be used to visualize and browse videos by showing the important *nouns* in the video, be indexed for search, or derive high level semantics. Although object discover is usually off-line and not as time-critical as object search, observing that the optimization can be slow when the data matrix $\mathbf{X}$ is large, we may improve the efficiency of the proposed approach by reducing the

size of $\mathbf{X}$. For instance, we can cut long videos into shots, or pre-filter frame-wise object proposals by eliminating noisy object proposals before selection using spatio-temporal coherency and context. As mentioned above, due to the generic nature of our formulation, the spatio-temporal coherency or context can also be formulated into the prior weight matrix $\mathbf{W}$ to benefit selection.

6.2 Future Work

The work presented in this thesis can be used as a stepping stone to build more useful tools for big video data analysis. Specifically, the following are a few directions I would like to explore in the future.

6.2.1 Future Work on Video Object Search

For video object search, a direct extension is to investigate local appearance consistency between neighboring frames. In the work presented in this thesis (Chapter 3 and Chapter 4), although the objective of spatio-temporal search requires the found trajectory to have optimal accumulated similarity to a given query, it does not enforce the coherency of the object appearance along the found trajectory; therefore it may be susceptible to local appearance ambiguity. In future, we would like to incorporate the local appearance constraint for spatio-temporal search while ensuring system scalability and efficiency. In addition, we would like to extend previous approaches to finding videos containing multiple objects of interest.

6.2.2 Future Work on Video Object Discovery

6.2.2.1 Common Object Mining from Videos

One limitation of the proposed approach in Chapter 5 is that it handles each video independently. For many applications such as internet security and trend detection in social videos, it is of great interest to discover common objects among multiple videos, especially in a large video corpus. One way to tackle this problem is to first discover the representative objects from each video using the approach presented in Chapter 5. We can then sift through the pool of representative objects to find the common objects, which are present in a good number across different videos. Alternatively, the commonness can be directly formulated into the object discovery algorithm. However, given a large video dataset instead of a single video sequence, the challenge is to come up with a formulation that can be solved efficiently.

6.2.2.2 From Nouns to Verbs

Moving forward, we also plan to go beyond the *presence* of important *nouns* (i.e., objects) in videos, to the analysis of important *verbs*. In other words, we would like to further the research in the direction of deriving high level semantic through the analysis of interactions between important objects or between objects and their surroundings in videos. As the first step, we can extend the proposed solution to human actions by taking as input a pool of 3D action proposals, instead of 2D object proposals. As to interactions, the question is how to infer them from important objects and their configurations and context over time.

6.3 Systems to Integrate Discovery and Search

Once each video can be well-represented by a few key objects, big video data can be browsed, indexed and searched efficiently. Besides searching videos by key objects, the object-based video representation also allows us to measure object-based video similarity to find similar videos given a query video. The video matching problem can then be converted to a set matching problem. Build upon the proposed object discovery and search methods, I would like to develop applications and build large-scale systems to efficiently manage large video repositories.

Author's Publications

- [1] Jingjing Meng, Hongxing Wang, Junsong Yuan, and Yap-Peng Tan, “From keyframes to key objects: video summarization by representative object proposal selection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [2] Jingjing Meng, Junsong Yuan, Jiong Yang, Gang Wang, and Yap-Peng Tan, “Object instance search in videos via spatio-temporal trajectory discovery”, IEEE Transactions on Multimedia (T-MM), vol. 18, no. 1, pp. 116-127, Jan. 2016.
- [3] Jingjing Meng, Junsong Yuan, Yap-Peng Tan, and Gang Wang, “Fast object instance search in videos from one example”, IEEE International Conference on Image Processing (ICIP), 2015.
- [4] Yuning Jiang, Jingjing Meng, Junsong Yuan, and Jiebo Luo, “Randomized spatial context for object search”, IEEE Transactions on Image Processing (T-IP), vol. 24, no. 6, pp. 1748 1762, June 2015.
- [5] Jingjing Meng, Junsong Yuan, Gang Wang, Yap-Peng Tan, and Jianbo Xu, “Object instance search in videos”, International Conference on Information, Communications and Signal Processing (ICICS), 2013.
- [6] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang, “Robust part-based hand gesture recognition using kinect sensor”, IEEE Transactions on Multimedia (T-MM), vol. 15, no. 5, pp. 1110-1120, 2013.

- [7] Yuning Jiang, Jingjing Meng, and Junsong Yuan, “Randomized visual phrases for object search, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

References

- [1] A. Araujo, J. Chaves, D. Chen, R. Angst, and B. Girod. Stanford I2V: A News Video Dataset for Query-by-Image Experiments. In *Proc. ACM Multimedia Systems*, pages 237–242, 2015.
- [2] Yuning Jiang, Jingjing Meng, Junsong Yuan, and Jiebo Luo. Randomized spatial context for object search. *IEEE Transactions on Image Processing (T-IP)*, 24(6):1748–1762, June 2015.
- [3] Ling-Yu Duan, Jie Lin, Jie Chen, Tiejun Huang, and Wen Gao. Compact descriptors for visual search. *IEEE MultiMedia*, 21(3):30–40, 2014.
- [4] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.
- [5] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [6] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [7] Xian-Sheng Hua, Tao Mei, and Alan Hanjalic. *Online Multimedia Advertising: Techniques and Technologies: Techniques and Technologies*. IGI Global, 2010.

- [8] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.
- [9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.
- [10] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision (IJCV)*, 42(3):145–175, 2001.
- [11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [12] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.
- [13] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [14] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.
- [15] F. Perronnin, Yan Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391, June 2010.
- [16] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on*

REFERENCES

- Pattern Analysis and Machine Intelligence (TPAMI)*, 34(9):1704–1716, Sept 2012.
- [17] R. Arandjelovic and A. Zisserman. All about vlad. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1578–1585, June 2013.
- [18] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *International Conference on Machine Learning (ICML)*, pages 1–8, 2011.
- [19] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 817–824. IEEE, 2011.
- [20] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in Neural Information Processing System (NIPS)*, pages 1753–1760, 2009.
- [21] Ran Tao, E. Gavves, C.G.M. Snoek, and A.W.M. Smeulders. Locality in generic instance search from one example. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2099–2106, June 2014.
- [22] Yuning Jiang, Jingjing Meng, and Junsong Yuan. Randomized visual phrases for object search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [23] Xiaohui Shen, Zhe Lin, Jon Brandt, Shai Avidan, and Ying Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn reranking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [24] Christoph H. Lampert. Detecting objects in large image collections and videos by efficient subimage retrieval. In *Proc. IEEE Intl. Conf. on Computer Vision (ICCV)*, 2009.

REFERENCES

- [25] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [26] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2009.
- [27] Jingjing Meng, Junsong Yuan, Yuning Jiang, Nitya Narasimhan, Venu Vasudevan, and Ying Wu. Interactive visual object search through mutual information maximization. In *Proc. ACM Multimedia (MM)*, pages 1147–1150, 2010.
- [28] Jerome Revaud, Matthijs Douze, and Cordelia Schmid. Correlation-based burstiness for logo retrieval. In *ACM International Conference on Multimedia (MM)*, MM '12, pages 965–968, New York, NY, USA, 2012. ACM.
- [29] Takahito Kawanishi, Akisato Kimura, Kunio Kashino, Shin'ichi Satoh, Duy-Dinh Le, Xiaomeng Wu, and Sbastien Poullot. Ntt communication science laboratories and nii in trecvid 2010 instance search task. In *TRECVID'10*, pages –1–1, 2010.
- [30] Duy-Dinh Le, Cai-Zhi Zhu, Sebastien Poullot, Vu Q Lam, Vu H Nguyen, Nhan C Duong, Thanh D Ngo, Duc A Duong, and Shinichi Satoh. National institute of informatics, japan at trecvid 2012.
- [31] Cai-Zhi Zhu and Shin'ichi Satoh. Large vocabulary quantization for searching instances from videos. In *ACM International Conference on Multimedia Retrieval (ICMR)*, pages 52:1–52:8, New York, NY, USA, 2012. ACM.
- [32] Yan Yang and Shinichi Satoh. Efficient instance search from large video database via sparse filters in subspaces. In *IEEE International Conference on Image Processing (ICIP)*, pages 3972–3976, 2013.

REFERENCES

- [33] Du Tran, Junsong Yuan, and David Forsyth. Video event detection: From subvolume localization to spatiotemporal path search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(2):404–416, 2014.
- [34] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511–I–518 vol.1, 2001.
- [35] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision (ECCV)*, 2014.
- [36] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.
- [37] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171, 2013.
- [38] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really? *arXiv preprint arXiv:1406.6962*, 2014.
- [39] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(7):1312–1328, 2012.
- [40] Ian Endres and Derek Hoiem. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(2):222–234, 2014.
- [41] Leonard Kaufman and Peter Rousseeuw. Clustering by means of medoids. In *Statistical Data Analysis Based on the L1 Norm and Related Methods*, pages 405–416. North-Holland, 1987.

REFERENCES

- [42] Brendan J Frey and Delbert Dueck. Mixture modeling by affinity propagation. In *Advances in Neural Information Processing System (NIPS)*, pages 379–386, 2005.
- [43] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.
- [44] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 2014.
- [45] Chunlei Yang, Jinye Peng, and Jianping Fan. Image collection summarization via dictionary learning for sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1122–1129, 2012.
- [46] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.
- [47] F Dornaika and I Kamal Aldine. Decremental sparse modeling representative selection for prototype selection. *Pattern Recognition (PR)*, 48:3714–3727, 2015.
- [48] Huaping Liu, Yunhui Liu, Yuanlong Yu, and Fuchun Sun. Diversified key-frame selection using structured optimization. *IEEE Transactions on Industrial Informatics*, 10(3):1736–1745, 2014.
- [49] Luming Zhang, Yingjie Xia, Kuang Mao, He Ma, and Zhenyu Shan. An effective video summarization framework toward handheld devices. *IEEE Transactions on Industrial Electronics*, 2015.
- [50] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [51] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

- [52] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *Proc. European Conf. on Computer Vision (ECCV)*, 2010.
- [53] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- [54] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery. In *Advances in Neural Information Processing System (NIPS)*, pages 19–27, 2012.
- [55] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *Proc. European Conf. on Computer Vision (ECCV)*. 2014.
- [56] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340. INSTICC Press, 2009.
- [57] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1):117–128, 2011.
- [58] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pages 253–262. ACM, 2004.
- [59] Jia Deng, Alexander C Berg, and Li Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 785–792. IEEE, 2011.

- [60] Yin-Hsi Kuo, Hsuan-Tien Lin, Wen-Huang Cheng, Yi-Hsuan Yang, and W.H. Hsu. Unsupervised auxiliary visual words discovery for large-scale image object retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 905–912, 2011.
- [61] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.
- [62] Xiaogang Wang, Ke Liu, and Xiaoou Tang. Query-specific visual semantic spaces for web image re-ranking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 857–864. IEEE, 2011.
- [63] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. European Conf. on Computer Vision (ECCV)*, 2008.
- [64] Junsong Yuan, Ying Wu, and Ming Yang. Discovery of collocation patterns: from visual words to visual phrases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [65] Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, and Shipeng Li. Descriptive visual words and visual phrases for image applications. In *Proc. ACM Multimedia (MM)*, pages 75–84. ACM, 2009.
- [66] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry preserving visual phrases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [67] Shiliang Zhang, Qingming Huang, Gang Hua, Shuqiang Jiang, Wen Gao, and Qi Tian. Building contextual visual vocabulary for large-scale image applications. In *Proc. ACM Multimedia (MM)*, pages 501–510. ACM, 2010.

REFERENCES

- [68] Yuning Jiang, Jingjing Meng, and Junsong Yuan. Grid-based local feature bundling for efficient object search and localization. In *IEEE International Conference on Image Processing (ICIP)*, pages 113–116, 2011.
- [69] Cai-Zhi Zhu, Yu-Hui Huang, and S. Satoh. Multi-image aggregation for better visual object retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4304–4308, May 2014.
- [70] Nicolas Ballas, Benjamin Labbé, Hervé Le Borgne, Philippe Gosselin, David Picard, Miriam Redi, Bernard Merialdo, Boris Mansencal, Jenny Benois-Pineau, Stéphane Ayache, et al. Irim at trecvid 2014: Semantic indexing and instance search. In *Proceedings of TRECVID*, 2014.
- [71] Herv Jgou Cai-Zhi Zhu and Shin’ichi Satoh. Nii team: Query-adaptive asymmetrical dissimilarities for instance search. In *Proceedings of TRECVID*, 2013.
- [72] A. Araujo, M. Makar, V. Chandrasekhar, D. Chen, S. Tsai, H. Chen, R. Angst, and B. Girod. Efficient video search using image queries. In *IEEE International Conference on Image Processing (ICIP)*, pages 3082–3086, Oct 2014.
- [73] A. Araujo, J. Chaves, R. Angst, and B. Girod. Temporal aggregation for large-scale query-by-image video retrieval. In *IEEE International Conference on Image Processing (ICIP)*, 2015.
- [74] Francois Fleuret and Donald Geman. Coarse-to-fine face detection. *International Journal of Computer Vision (IJCV)*, 41(1-2):85–107, 2001.
- [75] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision (IJCV)*, 57(2):137–154, 2004.
- [76] Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

REFERENCES

- [77] Antonio Torralba, Aude Oliva, Monica S Castelhana, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological Review*, 113(4):766, 2006.
- [78] Antonio Torralba. Contextual priming for object detection. *International Journal of Computer Vision (IJCV)*, 53(2):169–191, 2003.
- [79] Kevin Murphy, Antonio Torralba, William Freeman, et al. Using the forest to see the trees: a graphical model relating features, objects and scenes. *Advances in Neural Information Processing Systems (NIPS)*, 16:1499–1506, 2003.
- [80] Abel Gonzalez-Garcia, Alexander Vezhnevets, and Vittorio Ferrari. An active search strategy for efficient object class detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3022–3031. IEEE, 2015.
- [81] Michael Bergh, Gemma Roig, Xavier Boix, Santiago Manen, and Luc Gool. Online video seeds for temporal window objectness. In *IEEE International Conference on Computer Vision (ICCV)*, pages 377–384, 2013.
- [82] Dan Oneata, Jerome Revaud, Jakob Verbeek, and Cordelia Schmid. Spatio-temporal object detection proposals. In *European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, September 2014. Springer.
- [83] Mihir Jain, Jan van Gemert, Herve Jegou, Patrick Bouthemy, and Cees G.M. Snoek. Action localization with tubelets from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 740–747, Los Alamitos, CA, USA, June 2014.
- [84] Santiago Manén, Matthieu Guillaumin, and Luc Gool. Prime object proposals with randomized prim’s algorithm. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2536–2543, 2013.

REFERENCES

- [85] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 725–739. Springer, 2014.
- [86] Jan C van Gemert, Mihir Jain, Ella Gati, and Cees GM Snoek. Apt: Action localization proposals from dense trajectories. In *British Machine Vision Conference (BMVC)*, volume 2, page 4, 2015.
- [87] Du Tran and Junsong Yuan. Optimal spatio-temporal path discovery for video event detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [88] Jingjing Meng, Junsong Yuan, Yap-Peng Tan, and Gang Wang. Fast object instance search in videos from one example. In *IEEE International Conference on Image Processing (ICIP)*, 2015.
- [89] Jingjing Meng, Junsong Yuan, Gang Wang, Yap-Peng Tan, and Jianbo Xu. Object instance search in videos. In *International Conf. on Information, Communications and Signal Processing (ICICS)*, pages 1–4, 2013.
- [90] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 3536–3544. Curran Associates, Inc., 2014.
- [91] Thomas Dean, Mark Ruzon, Mark Segal, Jon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [92] Aditya Khosla, Rosyati Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

REFERENCES

- [93] Gunhee Kim, Leonid Sigal, and Eric P Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [94] Michael Gygli and Helmut Grabner¹ Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3090–3098, 2015.
- [95] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [96] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Proc. European Conf. on Computer Vision (ECCV)*. 2014.
- [97] Wen-Sheng Chu, Yale Song, and Alejandro Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [98] Min Sun, Ali Farhadi, Ben Taskar, and Steve Seitz. Salient montages from unconstrained videos. In *Proc. European Conf. on Computer Vision (ECCV)*. 2014.
- [99] Yael Pritch, Alex Rav-Acha, Avital Gutman, and Shmuel Peleg. Webcam synopsis: Peeking around the world. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.
- [100] Alex Rav-Acha, Yael Pritch, and Shmuel Peleg. Making a long video short: Dynamic video synopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

REFERENCES

- [101] David Liu, Gang Hua, and Tsuhan Chen. A hierarchical visual model for video object summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.
- [102] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [103] C. Lu, R. Liao, and J. Jia. Personal object discovery in first-person videos. *IEEE Transactions on Image Processing (T-IP)*, 2015.
- [104] Junsong Yuan, Gangqiang Zhao, Yun Fu, Zhu Li, Aggelos K Katsaggelos, and Ying Wu. Discovering thematic objects in image collections and videos. *IEEE Transactions on Image Processing (T-IP)*, 21(4):2207–2219, 2012.
- [105] J. Yang, G. Zhao, J. Yuan, X. Shen, Z. Lin, B. Price, and J. Brandt. Discovering primary objects in videos by saliency fusion and iterative appearance estimation. *IEEE Trans. on Circuits and Systems for Video Technology (TCSVT)*, 2015.
- [106] Gangqiang Zhao and Junsong Yuan. Discovering thematic patterns in videos via cohesive sub-graph mining. In *IEEE International Conference on Data Mining series (ICDM)*, pages 1260–1265. IEEE, 2011.
- [107] Le Wang, Gang Hua, Rahul Sukthankar, Jianru Xue, and Nanning Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *Proc. European Conf. on Computer Vision (ECCV)*. 2014.
- [108] Gangqiang Zhao, Junsong Yuan, and Gang Hua. Topical video object discovery from key frames by modeling word co-occurrence prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

REFERENCES

- [109] Gangqiang Zhao, Junsong Yuan, Gang Hua, and Jiong Yang. Topical video object discovery from key frames by modeling word co-occurrence prior. *IEEE Transactions on Image Processing (T-IP)*, 24(12):5739–5752, Dec 2015.
- [110] Ehsan Elhamifar, Guillermo Sapiro, and S Shankar Sastry. Dissimilarity-based sparse subset selection. *arXiv preprint arXiv:1407.6810*, 2014.
- [111] Jianbo Shi. Jitendra malik ”normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):888–905, 2000.
- [112] Partha Niyogi. Locality preserving projections. *Advances in Neural Information Processing System (NIPS)*, 30(3):55–59, 2002.
- [113] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [114] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, T. Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3367, 2010.
- [115] Peihua Li, Xiaoxiao Lu, and Qilong Wang. From dictionary of visual words to subspaces: Locality-constrained affine subspace coding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2348–2357, 2015.
- [116] Jingjing Meng, Junsong Yuan, Jiong Yang, Gang Wang, and Yap-Peng Tan. Object instance search in videos via spatio-temporal trajectory discovery. *IEEE Transactions on Multimedia (T-MM)*, 18(1):116–127, Jan 2016.
- [117] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3282–3289, June 2012.

REFERENCES

- [118] Jenny Yuen, Bryan Russell, Ce Liu, and Antonio Torralba. Labelme video: Building a video database with human annotations. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1451–1458. IEEE, 2009.
- [119] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1):3, February 2007.
- [120] Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Transactions on Multimedia (T-MM)*, 14(1):66–75, Feb 2012.
- [121] Michal Perd’och, Ondrej Chum, and Jiri Matas. Efficient representation of local geometry for large scale object retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 9–16. IEEE, 2009.
- [122] Jingjing Meng, Hongxing Wang, Junsong Yuan, and Yap-Peng Tan. From keyframes to key objects: Video summarization by representative object proposal selection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [123] Tomas McCandless and Kristen Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *British Machine Vision Conference (BMVC)*, 2013.
- [124] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [125] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.

REFERENCES

- [126] Jiayu Zhou, Jun Liu, Vaibhav A Narayan, and Jieping Ye. Modeling disease progression via fused sparse group lasso. In *ACM SIGKDD conference on Knowledge Discovery and Data Mining (KDD)*, pages 1095–1103, 2012.
- [127] Lawrence K Saul and Sam T Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research (JMLR)*, 4:119–155, 2003.
- [128] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.
- [129] Hervé Jégou and Ondřej Chum. Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening. In *European Conference on Computer Vision (ECCV)*. 2012.