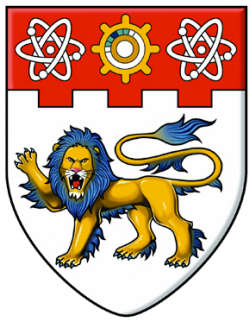


NANYANG TECHNOLOGICAL UNIVERSITY

DISCOVERING THEMATIC VISUAL OBJECTS IN UNCONSTRAINED VIDEOS

JIONG YANG

INTERDISCIPLINARY GRADUATE SCHOOL

NANYANG TECHNOLOGICAL UNIVERSITY

2018

DISCOVERING THEMATIC VISUAL OBJECTS IN UNCONSTRAINED VIDEOS

JIONG YANG

INTERDISCIPLINARY GRADUATE SCHOOL

NANYANG TECHNOLOGICAL UNIVERSITY

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

Doctor of Philosophy

2018

I would like to dedicate this thesis to my dear family.

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

Jiong Yang

2018

Acknowledgements

First of all, I would like to express my deepest gratitude to my Ph.D. advisor Professor Junsong Yuan for the insightful guidance, enthusiastic encouragement, continuous support, valuable research resources and opportunities provided during the Ph.D. study. His guidance helped me in both academic research as well as personal growth and future career outlook.

I would also like to thank my co-advisors, Professor Tsuhan Chen and Professor Dong Xu, and my mentor Professor Yap-Peng Tan, for their constructive feedbacks and supports during the Ph.D. study.

Besides my advisors and mentor, I would also like to express my particular gratitude to Dr. Xiaohui Shen, Dr. Zhe Lin, Dr. Brian Price and Dr. Jonathan Brandt from Adobe Research, as well as Dr. Gangqiang Zhao, Professor Yuan's former research staff, for their insightful guidance and patient directions at the beginning of my Ph.D. journey.

I would like to express my gratitude towards all the members in Professor Junsong Yuan's research group as well as my fellow lab mates in the ROSE Lab, for the enthusiastic research atmosphere, stimulating discussions, the sleepless nights before deadlines, and the fun we had in the last four years. My special gratitude also goes to the lab administrators of the ROSE Lab, for their continuous support and help during the Ph.D. study.

Last but not the least, I would like to thank my dear family. Their warm cares and continuous encouragement stayed with me during the tough times.

Abstract

Over the last decade, with the popularization of camera-equipped devices, there has been an explosive growth of video data. Despite the diverse visual contents, there are usually some thematic objects in these videos. As the key objects to be presented, thematic objects appear frequently and occupy highlighted positions in the video scenes, thus retain our impression after watching the videos, such as the bride and the groom in wedding ceremony videos, the birthday girl in birthday party videos, or product logo in commercial videos. Automatically discovering and localizing these thematic objects can benefit many real-world applications, such as video summarization, search, and labeling. However, this task is challenging as there is no prior information or initialization about the thematic objects. Moreover, there is usually background clutter, occlusions, or camera motions accompanying the targets. In this thesis, a systematic study is conducted on the automatic discovery and localization of thematic objects in videos.

We have studied this problem under various settings, including automatic discovery and localization of the thematic object in single videos, automatic discovery and segmentation of the thematic object in single videos, and automatic thematic action discovery and localization in collections of videos. In the absence of category-specific supervision and manual initialization, various category-independent cues have been explored to discover and localize the thematic objects. These include the spatiotemporal saliency to highlight

regions with salient appearance or motion with respect to the background, temporal smoothness of spatial locations and appearance variations along the object moving trajectory, and global appearance consistency of the object throughout its presence. When the discovery is performed in video collections instead of single videos, the semantic similarities in terms of appearance and/or motion patterns of the objects between different videos are also important. Novel techniques are proposed in this thesis to improve the reliability and efficiency of these cues as well as how they can be better explored to improve the discovery and localization performance. Extensive evaluations on both benchmarking as well as newly proposed datasets demonstrate the usefulness of these proposed methods as well as their superiority over existing approaches.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background, Motivations & Challenges

Over the last decade, there has been an explosive growth of video data with the popularization of camera-equipped mobile devices and the wide deployment of surveillance cameras in the cities. For example, there are nearly 400 hours of videos uploaded to YouTube every minute as reported in [40]. Despite the diverse visual contents in these videos, they usually contain some thematic objects [54, 139, 140, 142, 149, 156]. As the key objects to be presented, thematic objects appear frequently and occupy highlighted positions in the video scenes, thus retain our impression after watching the videos, such as the bride and the groom in wedding ceremony videos, the birthday girl in birthday party videos, or product logo in commercial videos. The thematic objects of a single video are the objects that saliently appear through the whole video sequence [139, 142, 149, 156], and the thematic objects of a collection of videos are the objects that saliently appear in most of these videos [54, 63, 140]. Different from salient object detection [9] which just focuses on locally salient regions in terms of appearance or motion, thematic video object discovery also imposes global semantic constraint over

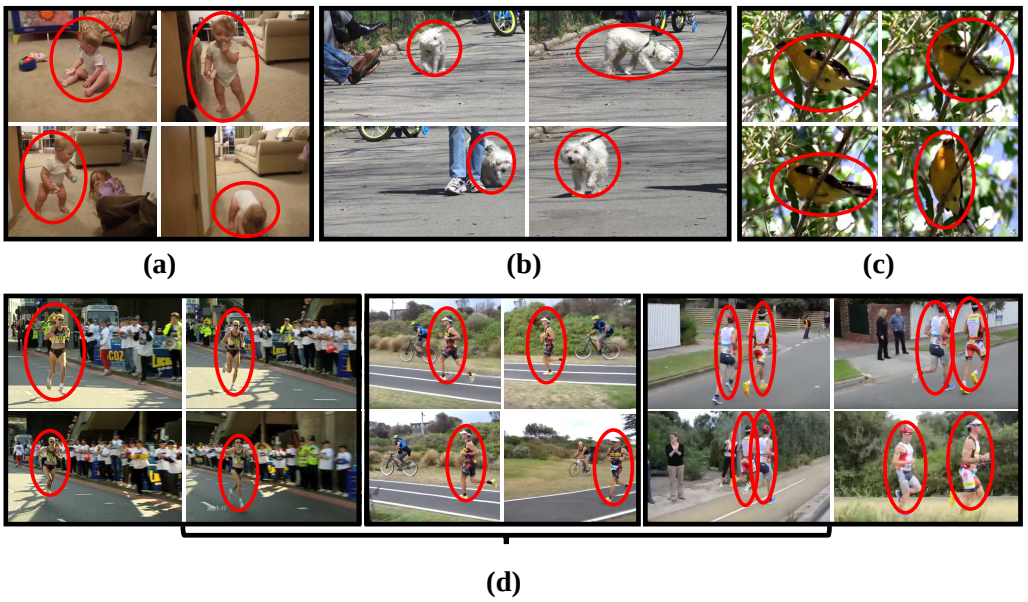


Fig. 1.1 Examples of thematic object/action discovery and localization. Each block of frames represents a video and the red circles denote the thematic objects to be discovered and localized. (a)-(c) are examples of thematic object discovery and localization in single videos. (d) is an example of thematic action discovery and localization in a collection of videos. It is worth noting that although the "biking" action appears in the second video of (d), it does not appear in other videos, and thus not captured as thematic action of the entire video collection.

the entire video sequence or video collection, *i.e.*, the same object needs to appear frequently in most of the frames in a single video or in most of the videos in a video collection. Some examples of thematic objects/actions in single videos as well as video collections are shown in Fig. 1.1. The thematic objects/actions in these videos are obvious to humans but can we discover and locate them automatically?

Automatically discovering and localizing these thematic objects are of great importance to many real-world applications. For example, it can be used to perform more efficient and effective video summarization and indexing by just focusing on the thematic part of the video instead of the cluttered background. It can also assist professional video editing by automatically segmenting the thematic part of the videos. More importantly, the automatic

processing requires no human intervention, and thus allows the efficient analysis of large-scale video dataset. In this thesis, a systematic study is conducted to tackle the problem of thematic object discovery and localization in videos. The input can be a single video or a collection of videos without any prior information or initialization, and the output is the spatiotemporal localizations, *e.g.*, sequences of bounding boxes or pixel-wise segmentation masks, of the thematic objects.

It is worth noting that, automatic thematic object discovery and localization is different from the widely studied category-specific object, action or event classifications and detections [58, 127, 144]. These methods are used to detect predefined categories of objects or actions, and thus, require the manual annotation of a large number of training videos. Furthermore, due to the diversity of video content, it is difficult to have a predefined set of categories that suit all scenarios. In addition, automatically discovering and localizing the thematic objects in videos can be used to assist the data annotation process of these category-specific detections, especially when the training datasets are huge and manual labeling is not practical [101].

Despite the recent studies on the automatic thematic object discovery and localization in videos, it remains to be an unsolved problem. Overall, there are three major challenges. First, there is no prior information about the category, shape, location, appearance or motion pattern about the thematic objects. Any types of objects can be thematic and they may appear at any spatiotemporal locations in the videos. Furthermore, the videos may be untrimmed, *i.e.*, the thematic object may not necessarily appear from the first frame to the last frame. As a consequence, the discovery and localization have to be performed simultaneously without any initialization. Secondly, the same object may appear or move differently in different videos, or in different frames of the same video, due to scale, viewpoint, speed, and illumination

variations, not to mention partial occlusion or non-rigid deformation. It is not a trivial task to associate the local regions belonging to the same object but with different appearance or motion patterns. Last but not the least, there is usually background clutter or camera motions accompanying the thematic objects, especially in consumer videos. Differentiating these distractions from the true target without any prior information is challenging.

It is worth noting that, the "thematic object" in this thesis is different from the concept of "foreground object" in photography. "Foreground object" usually means the object that is closest to the camera in a photography while "thematic object" is defined more on the semantic/attention level. For example, an object that is closest to the camera may or may not be the thematic object but should be a "foreground".

1.2 Content and Contributions of Thesis

In the purely uninitialized setting without any category-specific supervision or manual initialization, various category independent cues are investigated and proposed to perform the discovery and localization, such as spatiotemporal saliency, temporal smoothness of spatial location and appearance, and global appearance consistency. The spatiotemporal saliency estimation is built on the fact that the thematic objects typically look different from the background, *i.e.*, spatial saliency [43], or move differently from the background, *i.e.*, motion saliency [142]. The temporal smoothness of spatial location assumes that the targets always move smoothly from frame to frame, *e.g.*, it is unlikely that an object jumps from the top left corner to bottom right corner of the frame in a short period of time. The temporal smoothness of appearance assumes that the appearance variations of the object in nearby frames due to different scales, viewpoints, illumination conditions *etc*, will also be smooth. The

global appearance consistency assumes that the appearance of the thematic object is relatively consistent compared with the background. When the discovery and localization are performed in a collection of videos to find thematic objects, semantic similarities in terms of the appearance of the thematic objects across different videos should also be considered. Similarly, if the discovery and localization are performed in a collection of videos to find thematic actions, the semantic similarities in terms of the motion patterns across different videos should be considered.

In this thesis, we have explored the application of the above-mentioned category independent cues in different settings to discover and localize the thematic objects, *i.e.*, automatic thematic object discovery and localization in single videos, automatic thematic object discovery and segmentation in single videos, and automatic thematic action discovery and localization in collections of videos.

In the study of thematic object discovery and localization in single videos, we first propose to fuse spatial and motion saliency estimations to produce spatiotemporal saliency estimation. Most existing saliency models have either focused on just spatial saliency or motion saliency, while we observe that there is a strong complementation between them for video saliency estimation. For example, in some frames where the thematic object is relative still but its color contrast with the background is high, the spatial saliency is more useful, while in some frames where the background is cluttered but the thematic object undergoes large motion, the motion saliency is more useful. Hence, in this thesis, we propose an adaptive fusion scheme which automatically judges which saliency estimations are more useful for a particular frame. Then, to incorporate temporal smoothness of the object locations, we formulate the discovery and localization of thematic object as a salient tube search problem where each tube is a temporal sequence of spatially adjacent bounding

boxes within the video. A dynamic programming algorithm with linear time complexity is employed to efficiently find the most salient tube. To further constrain the global appearance consistency of the thematic object throughout the video, we propose an iterative appearance modeling technique to gradually refine the initial localization based on appearance.

We have also tried to apply similar temporal smoothness modeling technique to improve the per-frame image object proposal [42] in videos. Image object proposals are related to thematic object discovery and localization as it also produces localizations to capture the objects of arbitrary unknown categories. Image object proposals can also be used as the primitive input to both category independent thematic object discovery [55] as well as category-specific object detection [103, 144]. However, all previous image object proposal methods [3], or the methods that try to improve existing image object proposals [13, 62], have only used spatial information, *e.g.*, colors or edges. In this work, we propose to use temporal smoothness constraint to improve the per-frame image object proposals. An online version of the salient tube search algorithm in our first work is proposed to improve the per-frame image object proposals in videos. Besides temporal smoothness of spatial locations, we also incorporate the temporal smoothness of appearance variations along object moving trajectory in the search formulation.

In the study of thematic object discovery and segmentation in single videos, we use the spatiotemporal saliency estimation in our previous work as the initial prior on the thematic objects. To perform the pixel-wise discovery and localization, we first over segment the video frames into superpixels [2] and then formulate the segmentation as a binary node labeling problem in an undirected graph. In the graph, the nodes model superpixels and edges model the spatiotemporal proximities and appearance similarities between connected superpixels. Segmentation is achieved by assigning binary labels,

i.e., thematic object or background, to all the nodes in the graph. However, this segmentation formulation only enforces the local appearance consistency between spatiotemporally proximate superpixels. In order to constrain the global appearance consistency throughout the video, most existing works employ iterative approaches that separate the node labeling and appearance modeling as two alternative steps [96, 151]. In this work, we propose a framework to directly embed the global appearance constraint into the node labeling process by adding auxiliary nodes and edges to the original superpixel graph. The final segmentation can then be obtained by a single round of node labeling without further iterations or separate appearance modeling steps.

Besides thematic object discovery and localization in single videos, automatic discovery and localization of thematic actions in collections of videos is also studied. Besides being salient in each individual video, the thematic actions should also be common within the video collections, *i.e.*, the thematic actions should appear in many videos. Unlike most existing thematic object co-localization works which assume each video contains exactly one thematic object [54, 63], we tackle the problem in a more unconstrained scenario, *e.g.*, each video may contain zero, one or several thematic actions and there can be multiple types of thematic actions, *e.g.*, running, golfing *etc.* Since the focus of this work is to find human actions, we use human detections instead of the generic saliency estimation as the initial prior. Then the per-frame human detections are linked and tracked to produce a pool of spatiotemporal localizations in each video. However, it is inevitable that some of these initial candidate localizations capture noisy background or human actions that only appear in few videos. We thus propose to select the thematic actions from the initial localization corpus by enforcing the motion pattern similarities among the selected thematic actions. To perform the selection, we first build

a graph in which each node represents a candidate localization and each edge represents the motion pattern similarity between the connected localizations. The selection process is then formulated as a maximum density subgraph selection problem. A polynomial time optimization algorithm is proposed to efficiently perform the selection.

1.3 Organization of Thesis

The remaining part of this thesis is organized as follows. Chapter 2 reviews the related work in the literature. Chapter 3 presents the first work on automatic thematic video object discovery and localization in single videos. Chapter 4 presents the second work which proposes a non-iterative appearance modeling technique for the task of thematic video object segmentation in single videos. Chapter 5 presents the third work which proposes an online method to improve the per-frame image object proposals in videos by enforcing the temporal consistency of the proposals in different frames. Chapter 6 presents the fourth work which discovers and localizes the thematic actions in an unconstrained collection of videos, *i.e.*, each video may contain zero, one or several thematic action instances. The thesis is concluded in Chapter 7 and potential future work is discussed.

Chapter 2

Literature Review

In this chapter, we review the previous works that are related to the research work conducted in this thesis. In section 2.1, we review the appearance and motion-based saliency estimation models as well as the fusion of different saliency estimations. In section 2.2, we review the works that discover and localize the thematic objects in single videos. In section 2.3 and 2.4, we review the methods that discover and localize the thematic objects and actions, respectively, in collections of videos. A summarization of the discussed related works is shown in Fig. 2.1.

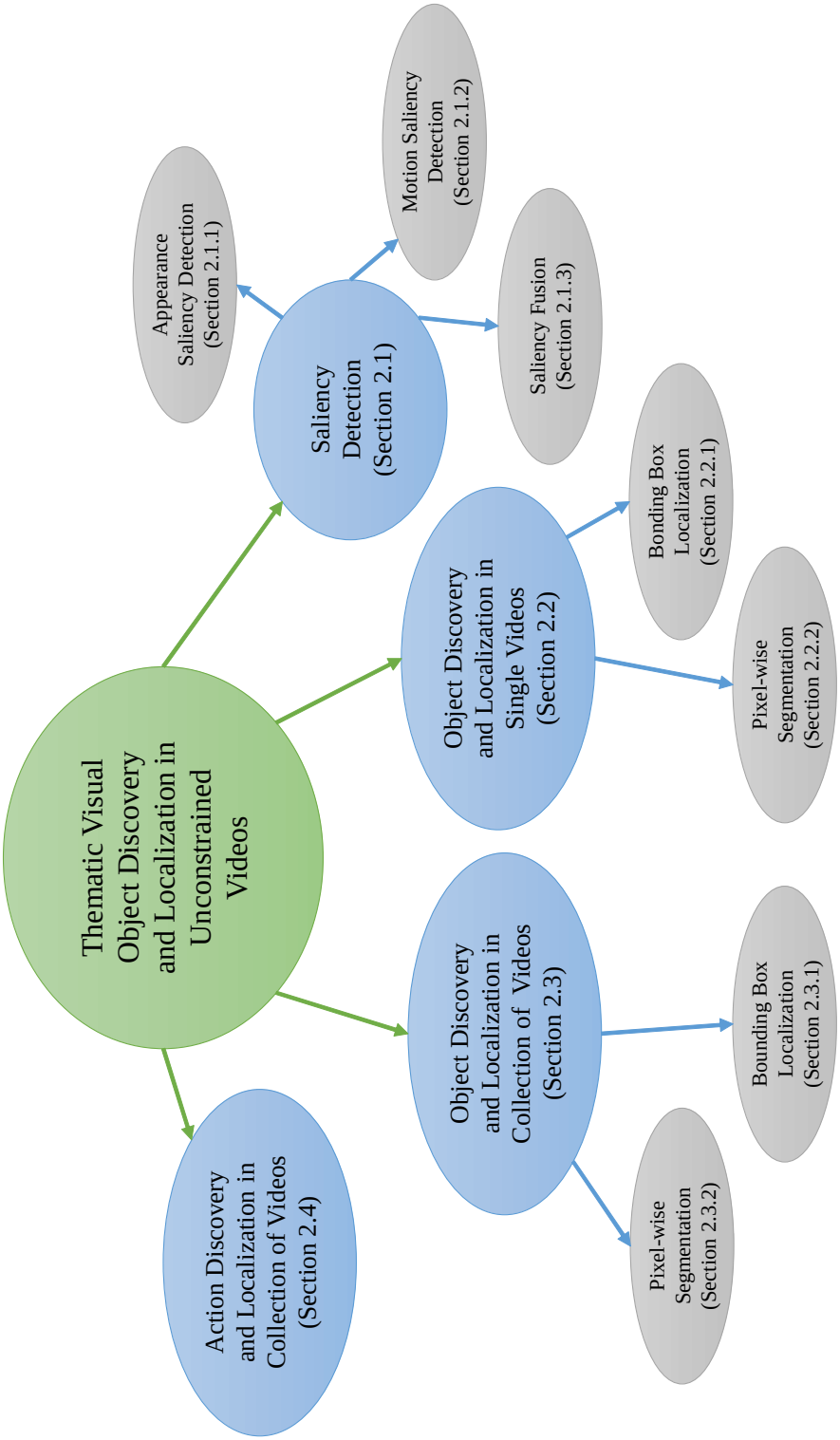


Fig. 2.1 An overview of the related works.

2.1 Appearance and Motion Saliency Detection

The task of saliency detection is to highlight the most attractive and informative pixels or regions in an image or video frame. The input is usually an image or video frame and the output is a score map indicating the saliency values of the pixels. In the task of automatic thematic video object discovery, the per-frame saliency detection is usually used as an important evidence to infer the thematic object in the absence of category-specific supervision. Existing saliency detection models usually resort to cues like the rarity, uniqueness, local or global contrast of pixels or image regions as well as high-level object priors such as shape or boundary.

2.1.1 Appearance Saliency Detection

Appearance-based saliency detection is to find the regions that look different from the rest parts of an image. Owing to its usefulness in many computer vision problems, it has gained much popularity in recent years and become a sub-area in the computer vision research community. The existing appearance saliency detection models can be broadly divided into two categories [32]: human fixation detection [43] and salient object detection [15, 154]. The former aims to predict sparse points where humans are likely to focus their attention on when viewing the image, while the latter tries to highlight the whole object that attracts humans' attention with well-defined object boundaries. Note that, We do not aim to give a complete review of all the saliency estimation models since we have mainly focused on the application of saliency estimations to the problem of thematic video object discovery. A more comprehensive study about this topic can be found in [9–11, 117].

The pioneering work of saliency estimation in [43] proposes a bottom-up method based on center-surround differences across multi-scale image features. The methods in [15, 99] propose to use the global or local region contrast to estimate the saliency values. The methods in [133, 138, 153, 163, 154] perform saliency estimation by assuming the background regions are usually connected to the image borders. Besides using these handcrafted cues, there are also many learning based saliency estimation methods [52, 68, 51, 131, 159, 132, 76, 67]. For example, the methods in [52, 68, 131] use supervised learning approach to map regional feature vectors to saliency scores. [67] proposes a method based on fully convolution neural network (FCN) with global input (whole image) and global output (whole saliency map). The method in [132] further proposes to append the saliency prior maps together with the original image as the input to the FCN network.

2.1.2 Motion Saliency Detection

Motion saliency detection is to find the regions that move differently from the rest parts of a video frame. Unlike appearance saliency estimation which has become a sub-area in the computer vision research, motion saliency estimation is more often used as an intermediate step to provide initial motion evidence. The input is usually two or several consecutive frames in order to extract motion information, and the target is to highlight the moving objects in the video frames. The method in [80] simply uses the difference values between two consecutive frames as motion saliency. [96] proposes a method to generate binary motion map for each frame labeling the pixels that belong to the moving object. It first extracts motion boundary based on optical flow fields and then labels the pixels inside a closed motion boundary polygon as the moving object. The method in [151] uses the average Frobenius norm of

optical flow gradient around an image region as the motion saliency scores of that region. The method in [63] proposes to compute the motion coherence score for an image patch using long-term point tracks. [128] proposes to stand out the moving objects from the background by eliminating camera motion from the optical flow fields. Inspired by this concept, the method in [142] directly uses the magnitude of the optical flow field after eliminating camera motion as the motion saliency scores. [142] also proposes a global contrast based approach to estimate the motion saliency scores. It builds a histogram of the optical flow fields and assumes the motion pattern of the true moving object will receive fewer votes than the background motion.

2.1.3 Saliency Fusion

Since different saliency estimation algorithms focus on different aspects of saliency, *e.g.*, color contrast, edges or motions, it is preferable to use different types of saliency measures under different scenarios. Hence, many works have proposed to fuse different types of saliency measures to improve the overall performance. These methods can be divided into three categories based on their adaptability. The first category of methods uses predefined fusion functions such as mean, multiplication, and maximization [86]. These methods can only marginally improve the performance of each individual saliency estimation as they use fixed fusion schemes regardless of the image content. The second category of methods use some empirical handcrafted measures [53, 26, 150] to assess the quality of each saliency estimation and combine the individual saliency measures based on their quality scores. The method in [26] proposes to use the spatial variance of the saliency value distribution across the 2D image. The method in [150] proposes a similar measure but uses motion variance. The method in [53] combines the

compactness measure proposed in [83] and the spatial distribution of saliency map to measure the quality of saliency estimation. These methods provide better performance compared with the ones using predefined fusion functions but the handcrafted quality measures may not be optimal given limited observations of humans on the saliency data. The last type of methods use learning-based approach to learn the fusion process [83, 84, 94, 142, 161, 162]. These methods extract handcrafted features to describe the saliency estimations and then learn from massive training data how these features can be combined to fuse different saliency maps.

2.2 Thematic Object Discovery and Localization in Single Videos

Thematic object detection in single videos is to discover and locate the thematic objects in an input video sequence. The localizations can be in the form of bounding box localizations [55, 77–81, 149, 156] or pixel-wise segmentations [46, 4, 102, 120, 96, 64, 82, 6, 65, 151, 85, 81, 156, 157, 71]. An important attribute of thematic video object detection is that the discovery and localization is category-independent and the types of the objects are unknown in advance. Hence, no category specific supervision is available about the thematic objects.

2.2.1 Bounding Box Localization

The method in [80] first extracts per-frame image saliency maps from the video and then locates the thematic object by finding salient regions with maximum saliency density with a branch and bound search algorithm. [78] also relies on per-frame saliency maps but locates the thematic object by

finding a temporally smooth trajectory of bounding boxes capturing the salient regions through dynamic programming. [149] proposes a bottom-up method to find the thematic object in short video clip. It gradually prunes uncommon local visual primitives and recovers the thematic objects within the video frames. Unlike [78] or [80] which rely on per-frame saliency as the primary cue to find the thematic object, [149] discovers the thematic object by assuming the thematic object appears in most of the video frames. The method in [55] takes the image object proposals as the primitive input and finds the thematic object by iteratively refining the object recurrence model, background model, and primary object model.

2.2.2 Pixel-wise Segmentation

Before reviewing the fully automatic thematic object segmentation works, we first discuss some works on interactive video object segmentation. Different from the fully automatic setting, these interactive methods require human intervention in the segmentation process. Some of these approaches require the user to provide pixel-wise segmentations on the first few frames for initialization [46, 4, 102, 120], while others require the user to continuously correct the segmentation errors [5, 69]. These methods generally require a considerable amount of human effort and, hence, are not scalable to large video collections.

In contrast, automatic video object segmentation does not require any human intervention and tries to automatically infer where the thematic object is [96, 64, 82, 6, 65, 151, 85, 81, 156, 157]. The approach in [96] relies on motion saliency as the initial cue and builds spatiotemporal superpixel graph to label the superpixels as either part of the thematic object or background. It also uses color GMMs to iteratively model the local appearance of thematic

object and background separately. Several works [64, 65, 82, 151, 6] use object segment proposals [24] as the primitive input which contributes significantly to the inefficiency of these methods. The method in [64] first uses spectral clustering to group proposals with coherent appearance and then trains color GMMs and location priors of the thematic object. Pixel-wise graph cut is used to produce the final segmentation mask for each individual frame. [82] adapts a similar pipeline with [64] but uses constrained maximum weighted cliques to group proposals. The method in [151] builds a spatiotemporal graph by connecting proposals and uses dynamic programming to find the most confident trajectory. It then uses pixel-wise graph cut to refine the final segmentation mask based on the initial proposal trajectory. The method in [6] produces multiple proposal chains by linking local segments using long-range temporal constraints. It then obtains the final segmentation by pixel-wise per-frame MRF smoothing using the appearance and location priors learned from the initial chains. The method in [65] tracks the proposals temporally using incremental regression and refines the final segmentations by composite statistic inferences. The method in [85] explores the segmentation problem in the MPEG2 compressed domain. On the P-Frames, it computes the motion saliency priors by compensating camera motion. On the I-Frames, it computes the color-based segmentation by the morphological approach. These two cues are then merged and followed by a spatiotemporal filtering using quadric surfaces to give the final segmentation result. The method in [81] first segments the selected keyframes into an over-complete set of segments using image segmentation algorithms like [107], and then employs the cohesive sub-graph mining technique to find the salient segments with similar appearance and strong mutual affinity. [156, 157] adopt a similar pipeline but use the topic model to discover the coherent segments. Both methods disregard the temporal smoothness of the object region and they

only aim at a rough localization instead of accurate segmentation. [71] proposes a method to discover primary objects in videos by integrating the topic models for appearance modeling and the Probabilistic Data Association (PDA) filter for motion modeling. Instead of providing an accurate pixel-wise segmentation, it only identifies the interest regions, *e.g.*, Maximally Stable Extremal Regions [89], that are likely to cover the thematic objects.

2.3 Thematic Object Discovery and Localization in Collections of Videos

Video object co-localization methods aim to localize the common objects in a video collection. The common objects refer to the objects that saliently appear in many videos. The localization can take the forms of either bounding box localization or pixel-wise segmentation. Similar to the methods reviewed in Section 2.2, many of these methods also formulate the discovery and localization process as the selection, linking or labeling of image object proposals or superpixels. The inter-video similarities are also incorporated in the formulation such that only common objects will be discovered. Compared with our thematic action co-localization work, most of these methods are for object co-localization instead of action co-localization. Moreover, most of these methods have implicitly assumed that each video contains exactly one common object, and thus, are rarely explored or evaluated in a fully unconstrained scenario like us.

2.3.1 Bounding Box Localization

The method in [113] performs image object co-localization by labeling image object proposals through quadratic programming. It incorporates the ob-

jectness priors, pair-wise similarities and global appearance discriminability of all the proposals in the formulation. [54] extends [113] to perform video object co-localization by also incorporating temporal consistency in the labeling process. The method in [17] performs image object co-localization by assigning a commonness score to each image object proposal using part based probabilistic Hough voting. [63] extends [17] to perform video object co-localization by combining the proposals' motion evidence scores with [63]'s commonness scores. Temporal consistency is enforced by linking the spatiotemporally proximate image regions through dynamic programming. The method in [48] first generates short proposal tubelets based on per-frame image object proposals, and then performs video object co-localization by assigning a co-saliency score to each image object proposal tubelet. A Viterbi like algorithm is used to link these tubelets.

2.3.2 Pixel-wise Segmentation

The method in [121] first generates object tracklets from initial image object segment proposals. It then performs semantic video object co-segmentation by tracklet co-selection via submodular optimization. Similar to [121], [152] also generates object tracklets. It then builds a completely connected graph on these tracklets, and perform the co-segmentation by finding the maximum-weighted clique. The methods in [30, 29] build a graph by modeling the object proposal segments in all the videos as nodes and the visual similarities between these proposals as edges. The co-segmentation is then performed by labeling these proposal segments in an energy minimization formulation. The methods in [130, 129] build a similar graph on superpixels but exclude inter-video connection to limit the graph size. Instead, they propose a multiple instance learning based technique to iteratively model the inter-

video appearance similarity. Different from the previous methods that try to segment the whole object, the method in [22] proposes a co-segmentation method to segment the parts of a particular object in a collection of videos by assuming every video contains an instance of this object. It performs the co-segmentation by labeling the superpixels in an energy minimization formulation.

2.4 Thematic Action Discovery and Localization in Collections of Videos

The existing action co-localization works have mainly focused on two scenarios, *i.e.*, co-localization in pairs of videos [18, 38, 97] and weakly supervised action co-localization with video level labels [20, 21, 92, 93, 108, 134, 143]. Few of these works have considered a fully unconstrained scenario like us, *i.e.*, the numbers and types of common actions are unknown in advance and each video may contain zero, one or several common actions.

The methods in [18] and [97] only generate temporal localizations instead of spatiotemporal localizations. Both methods formulate the common action discovery and localization as the search of subsequences with similar motion patterns between a pair of videos. The method in [18] proposes a branch and bound algorithm to perform the search, while the method in [97] uses a particle swarm optimization scheme to find the common actions. The method in [38] produces fine-grained spatiotemporal localizations in terms of pixel-wise segmentation. It discovers and localizes the common action between a pair of videos by labeling dense trajectories [127] in a Markov random field.

The approach in [93] tackles the weakly-supervised human activity recognition and localization by locating non-rectangular spatiotemporal discriminative regions that are inferred by clustering regions of similar texture and motion features. The method in [108] applies multiple instance learning to locate the labeled action by jointly optimizing inter- and intra-class distances. The approaches in [92] and [134] both extract spatiotemporal action localization proposals and then apply multiple instance learning to learn from these weakly labeled action proposals. Besides video level labels, the method in [92] also requires dot annotations on the video frames to better assist the multiple instance learning process. The method in [143] generates action localizations in terms of temporal intervals to locate the common action in a set of videos. It first generates short video subsequence by uniform sampling and then builds an affinity graph among these subsequences. The subsequences containing common actions are identified by their absorbing time in an absorbing Markov chain. The method in [20] discovers common motion patterns of articulated object classes, *e.g.*, tiger, horse, by clustering temporally partitioned video shots. It proposes a feature to describe pairs of trajectories in order to better characterize articulated object motion. [21] extends [20] to also recovers the spatial alignment among different instances of the same behavior using a Thin Plate Spline deformation model. [147] proposes a method to mine recurring events in a collection of videos. Although it operates in a less constrained scenario compared with the above weakly supervised setting, it only provides temporal localizations instead of spatiotemporal localizations. It first divides all videos into video primitives and then builds a matching trellis between these video primitives. The recurring events are discovered and localized by tracing all the continuous paths in the matching trellis through a forest growing procedure.

Chapter 3

Thematic Objects Discovery in Single Videos

In this work, we propose a new method for detecting thematic objects in unconstrained videos in a completely automatic setting. Here, we define the thematic object in a video as the object that presents saliently in most of the frames. Unlike previous works only considering local saliency detection or common pattern discovery, the proposed method integrates the local visual/motion saliency extracted from each frame, global appearance consistency throughout the video and spatiotemporal smoothness constraint on object trajectories. We first identify a temporal coherent salient region from the whole video and then explicitly learn a global appearance model to distinguish the thematic object against the background. In order to obtain high-quality saliency estimations from both appearance and motion cues, we propose a novel self-adaptive saliency map fusion method by learning the reliability of saliency maps from labeled data. As a whole, our method can robustly localize and track thematic objects in

diverse video contents, and handle the challenges such as fast object and camera motion, large-scale and appearance variation, background clutter and pose deformation. Moreover, compared with some existing approaches that assume the object is present in all the frames, our approach can naturally handle the case where the object is only present in part of the frames, e.g., the object enters the scene in the middle of the video or leaves the scene before the video ends. We also propose a new video dataset containing 51 videos for thematic object detection with per-frame ground-truth annotations. Quantitative experiments on several challenging video datasets demonstrate the superiority of our method compared with the recent state of the arts. This work has been published in the IEEE Transactions on Circuit and Systems for Video Technology [142].

3.1 Introduction

With the prevalence of online social video sharing, considerable amounts of videos are being created and processed every day. In many of those videos, there exists a thematic object that we want to focus our attention on, *e.g.*, a child or a pet in a "homemade" personal video. We define the thematic object in a video sequence as the object that presents saliently in most of the frames and some examples are shown in Fig. 3.1. In this work, we address the problem of automatically discovering the thematic objects in single videos, which is an essential step for many applications such as advertisement design [90] and video summarization [41, 115, 72]. Traditional video object detection and localization methods, however, are either too category specific (*e.g.*, face [125] and pedestrian detection [27]) or heavily rely on manual initialization

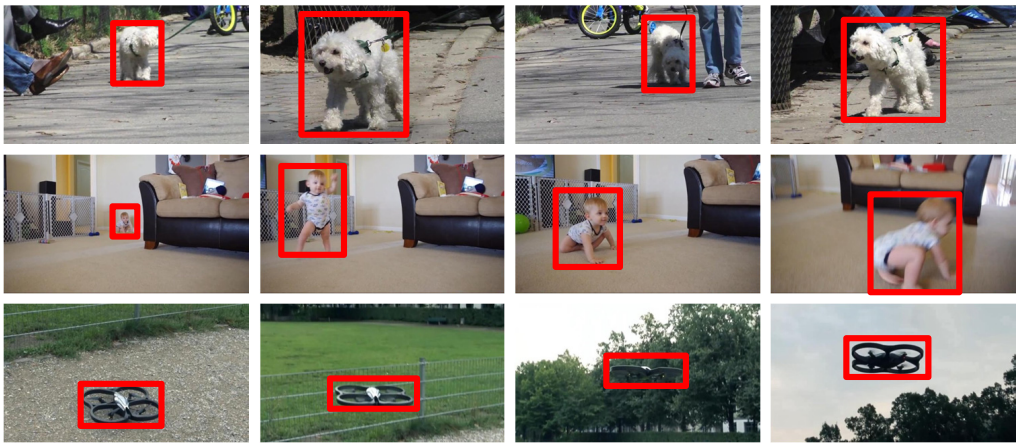


Fig. 3.1 Examples of thematic object discovery in single videos. Each row corresponds to one video and the red rectangle highlights the thematic object.

(*e.g.*, object tracking [39] and interactive object segmentation [37]). They are suitable for targeted object detection that is tailored to users' interests but are too limited for many multimedia applications that require automatically processing large volumes of video data with diverse content.

One way to automatically discover the thematic objects is by resorting to saliency detection [87, 50], as the thematic objects are usually distinctive either in appearance or in motion compared with the background. Regarding the three different saliency levels introduced in [32], we are referring to high-level salient object detection instead of visual attention modeling. Although, different image and motion saliency cues are explored and combined [77, 150, 86], these methods simply combine the appearance and motion saliency maps by weighted average where the weights are empirically determined. As a result, the final saliency map can be inevitably affected by the noise in every single map. Moreover, thematic object detection is not equivalent to salient object detection. Besides being locally salient, the thematic object also needs to be common throughout the video sequence, *i.e.*, present in most of the frames. Saliency is only one of the cues to determine where the thematic object is but there are more factors to consider, such as temporal

smoothness and appearance consistency across frames. In some pure saliency-based detection framework [77], little appearance information of the video object is captured, which may cause the detection to drift from the thematic object to other salient objects or background regions. On the other hand, in order to model the object appearance for automatic thematic object discovery, common visual pattern mining methods have been investigated [112, 73, 155, 149, 19, 137], and significant progress has been made. However, these unsupervised pattern mining methods require the object to appear frequently and have consistent appearances across the whole video sequence such that its visual pattern can be discovered. Their performance would degrade if the thematic object appears with large visual variation due to the illumination, scale and viewpoint variation, partial occlusion and deformation. Moreover, the static background with rich features may be more common than the thematic objects and treated as common object incorrectly.

In summary, pure saliency-based detection can easily drift among different salient objects or include salient background due to the lack of explicit appearance modeling. Hence, in order to tackle this problem, we propose to first use local saliency cues to automatically produce some weakly supervised information about the thematic object by considering the temporal consistency of the salient regions. This weak information is then used to explicitly learn an appearance model of the thematic object against the background regions in an iterative and discriminative manner. The evaluation results using a newly proposed dataset, *NTU-Adobe* thematic object discovery dataset, and two other challenging video datasets, *i.e.*, the *10-video-clip* dataset [31] and some selected categories in the UCF sports action dataset [104], demonstrate the efficacy of our method. We briefly introduce our method in the following.

Firstly, in order to obtain more accurate saliency estimations, we fuse different types of saliency cues including appearance based image saliency,

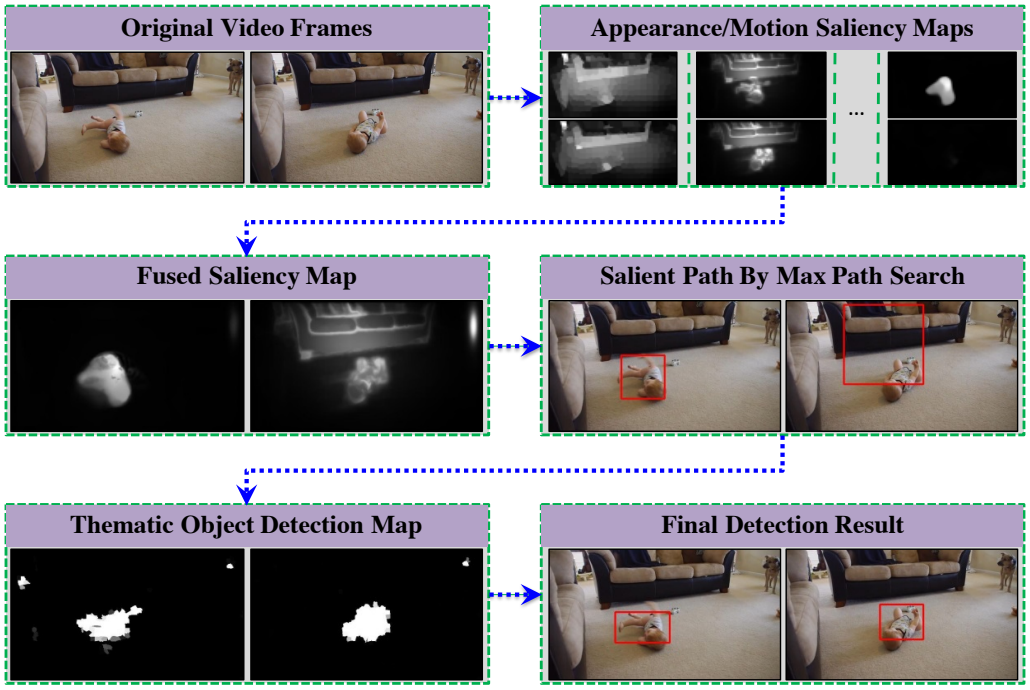


Fig. 3.2 Work-flow of the proposed detection framework by an example.

motion-based saliency, and semantic saliency. Instead of simply combining these different saliency cues empirically, we propose a novel learning-based saliency fusion technique, *SVM (support vector machine) Fusion* that can judge the quality of each saliency map and determine its combination weight in an adaptive manner. Then in order to find spatiotemporally coherent salient regions, we formulate the problem into the framework of max path search [119].

Secondly, in order to ensure appearance consistency in the detection process and better distinguish the thematic object against the background, we use the pure saliency-based detections as weakly supervised information to explicitly learn the appearance of the thematic object in an iterative manner. Then the learned model is used to produce a much cleaner and appearance-consistent thematic object detection map, based on which much better detection results can be obtained.

In summary, the target of our work is to automatically discover and localize the thematic object in a given video sequence without any human interaction. The overall workflow of the proposed method is shown in Fig. 3.2. The major contributions of this work are:

1. We propose a novel learning-based saliency map fusion technique which can adaptively fuse appearance and motion saliency maps to make use of their strong complementation.

2. We propose an effective appearance modeling scheme to explicitly model the appearance of the thematic object and background in a discriminative manner.

3. Besides the intermediate steps, we propose a unified end-to-end framework for automatic thematic video object discovery and localization by exploring the local saliency cues, spatiotemporal smoothness constraint, and discriminative appearance modeling.

4. We propose a new multi-category video object dataset for automatic thematic video object discovery. Per-frame ground truth bounding box annotations are provided for each video. It will be shared with the research community.

3.2 Video Saliency Detection

As discussed in section 2.1, visual saliency has been extensively studied in the literature. For the thematic object discovery problem, however, any single saliency cue cannot provide robust detections due to the diversity of video content. For example, motion saliency cues are more suitable for the case where the thematic object moves differently from the background,

while image saliency cues are more suitable when the thematic object is visually very different from the background. Moreover, current techniques in visual saliency estimation are not always perfect and may produce noisy and incorrect saliency maps. As each type of saliency map only works for specific cases in identifying the thematic object, the incorporation of different saliency cues is essential for robust object discovery. In this section, we first introduce the individual saliency cues employed in our work, *i.e.*, static image saliency, motion saliency and semantic saliency. The fusion of saliency maps is subsequently described.

3.2.1 Saliency Cue Estimation

Image Saliency

Static image saliency is computed individually for each video frame. Image saliency generally emphasizes local regions with distinct textures and colors compared with the rest of the image. In this work, we use two state-of-the-art image saliency measures: the *PCA (Principle Component Analysis) Image Saliency* proposed in [87] and the *AMC (Absorbed Markov Chain) Image Saliency* proposed in [50]. The *PCA Image Saliency* detects the saliency of each image patch in a sliding window manner by considering the global contrast of the local color and pattern, while the *AMC Image Saliency* detects the saliency of each superpixel as its absorbing time in an Absorbed Markov Chain. These two methods can complement each other as they use different underlying techniques. Other image saliency measures such as [15] and [35] can certainly be applied as well. Examples of these two types of saliency maps are shown in the second and third row of Fig. 3.7, respectively.

Motion Saliency

Similar to the static image saliency, we measure the motion saliency as distinct motion patterns based on dense optical flow. Two types of motion saliency are used in this work. The first one is computed as the magnitude of the *"ω-flow"* [44], and we name it *ω motion saliency*. *"ω-flow"* emphasizes local motion by removing the global motion from the original dense flow field. Similar to [44], we estimate the global motion as a 6-parameter affine model using the Motion2D software[1] which is robust to complicated global motions like camera zooming or rotation. However, if the global motion is wrongly estimated, the obtained saliency map will be totally corrupted (an example is shown in the fifth row and the fifth column of Fig. 3.7). Hence, we use another type of motion saliency based on global motion contrast, namely *GC (Global Contrast) motion saliency*. We use a simple but effective voting based approach to estimate this global motion contrast: we use each pixel's flow vector to vote in the quantized x-y parameter space of the flow and then take the logarithm of the reciprocal of each cell's voting score as the global motion contrast score of those pixels voting in that cell. Mathematically, the global motion contrast saliency map of an image can be computed as:

$$\begin{aligned} GC(x,y) &= V(f(x,y)) \\ V(u,v) &= \log(\frac{1}{|P(u,v)|}) \\ P(u,v) &= \{(x,y) \mid f(x,y) = (u,v)\} \end{aligned} \tag{3.1}$$

where $GC(x,y)$ is the global motion contrast saliency score at pixel location (x,y), $f(x,y)$ gives the quantized bin of pixel (x,y)'s optical flow field, $V(u,v)$ is the saliency score of bin (u,v) on the quantized optical flow space, $P(u,v)$ is the collection of pixels whose optical flow values are quantized to bin

[1] http://www.irisa.fr/vista/Motion2D

(u, v) and $|P|$ is the cardinality of set P. Median and Gaussian filterings are applied afterward to both types of motion saliency maps for abrupt noise rejection and smoothing. Examples of these two types of saliency maps are shown in the fourth and fifth row of Fig. 3.7, respectively.

Semantic Saliency

In order to model object-level saliency, we use two types of higher level semantically meaningful priors: face and human body. Other semantical priors can also be added. Each prior will produce a separate saliency map for each frame. Viola-Jones face detector [125] is used to detect faces and the Latent SVM object detector [27] with the human body model trained on the *VOC 2007* dataset[2] is used to detect human bodies. Since both detectors give bounding boxes as detection results, we use a Butterworth filter like smoothing function to re-weight the boxes to obtain a smoother saliency map. We also perform a median filter like approach to filter the bounding box detections along the temporal axis to suppress the false detections without neighboring support and recover missed detections with strong neighboring support. Examples of these two types of saliency maps are shown in the sixth and seventh row of Fig. 3.7, respectively.

Finally, all saliency maps are normalized linearly to the range between 0 and 1.

3.2.2 Saliency Fusion

Although we have obtained 6 types of saliency maps per video frame, using more than a single map does not necessarily produce better results unless we have a proper fusion technique. For example, if averaging is used, the

[2]http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

overall map quality can be significantly affected by even a single corrupted map. On the other hand, if we can selectively reject or assign a lower weight to those corrupted maps and only use or mainly focus on the good ones, we will then have a higher chance to obtain more robust saliency estimation. This is most useful when the input maps can complement each other such as the appearance and motion saliency maps. Hence, we propose a novel learning-based $SVM - Fusion$ technique which can automatically judge the quality of each saliency map and predict the optimal fusion weight. Unlike traditional methods which use ground truths on the thematic objects to evaluate the quality of a saliency map, we only use the saliency map itself to perform the evaluation in the test phase because ground truth data is not available in real scenarios. In the following, we first discuss an experiment in Section 3.2.2 to demonstrate the potential performance gain achievable through adaptive maps fusion. This is also our initial motivation to propose the *SVM-Fusion* technique. We then elaborate the fusion technique and two post-processing steps in detail in Section (3.2.2) and (3.2.2), respectively.

Best Fusion Weight

We have explicitly conducted an experiment to explore the potential performance improvement achievable by linearly fusing the maps using the "best" possible weights. The "best" weights are computed as follows.

Let's first denote our 6 different types of saliency maps as $\{S_i : 1 \leq i \leq 6\}$. The ground truth saliency map, G, is computed by filling the labeled bounding box with 1 and the rest with 0. We then formulate the computation of the best weights of each saliency map as the following least square optimization problem with linear constraints:

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \quad \|A\mathbf{w} - \mathbf{b}\|_2 \\ s.t. &\sum_i w_i = 1, w_i > 0, \end{aligned} \tag{3.2}$$

where A is a matrix with 6 columns and the i^{th} column of A is the vectorized version of S_i, $\mathbf{w}$ is a 6×1 column vector where w_i is the combination weight of map S_i, and $\mathbf{b}$ is the vectorized version of G. The objective function requires the combined map to be as close as possible to the ground truth map and the two linear constraints require the weights to be non-negative and sum up to 1. Certainly such "best" weights can be hardly achieved without knowing the ground truth, but its superior performance compared with the individual saliency maps before fusion, as shown in Table 3.1 and 3.3, motivates us to seek a good fusion approach that does not require the ground truth to estimate the fusion weights.

SVM-Fusion

To automatically measure the saliency map quality, a Support Vector Machine is trained to predict the quality of each saliency map. We design 13 features to represent each saliency map and collect training samples from an independent video dataset based on the bounding box annotations on the thematic object.

Based on our observation, a good saliency map will have the majority of its saliency scores concentrated on the thematic object region of the image and thus exhibits a compact distribution, while a bad saliency map will have most of its saliency scores spread all over the frame. Hence, in order to reflect the quality of a saliency map, we extract the following features from each saliency map: (1) *Distribution Measure of Saliency Value (4 features)*: this includes the mean, variance, skewness and kurtosis of the saliency scores on each map. (2) *Spatial Pyramid Entropy (4 features)*: we first partition the

saliency map into N regular grids, *e.g.*, $N = 256$ for a 16×16 partition. In the following, we use the term "saliency energy" to denote the summation of the saliency scores inside a grid/region. The set of saliency energy $\{s_i\}$ of all the grids in each partition is normalized to be a discrete probability distribution and the entropy is computed as $E = -\sum_{k=1}^{N} \frac{s_k}{\sum_{p=1}^{N} s_p} \log \frac{s_k}{\sum_{p=1}^{N} s_p}$. Essentially, this value will be low when most of the saliency scores are concentrated at a few grids and vice versa. In our experiment, we use four different partition levels to form the spatial pyramid, *i.e.*, 8×8, 16×16, 24×24 and 32×32, and each level contributes one feature. (3) *Spatial Variance (2 features)*: we use the concept of spatial variance from [16, 26]. It measures the variance of a distribution in which the random variable is the spatial location of each pixel and the probability is in proportional to its saliency score. We measure the spatial variance along the vertical and horizontal directions separately as two features. A small spatial variance implies that most of the saliency scores are concentrated in a compact region on the map. (4) *Inter-Map Coherence (3 features)*: this set of features aims to measure the coherence among different saliency maps. For each map, we first threshold the saliency scores to obtain a binary map in which '1' represents salient region and '0' represents background region. We then compute the percentage of the salient region on a map that are also salient on each of the other maps and take the maximum value as its inter-map support. We use three different values from high to low to threshold the saliency map and obtain three inter-map support features for each map. In total, a feature vector of 13 dimensions is used to characterize each saliency map.

In order to have enough training samples and avoid over-fitting, we have collected a separate training dataset composed of 24 video clips with manually labeled ground truth bounding boxes on the thematic objects. This dataset is completely independent of those used in the experiments. We simply use

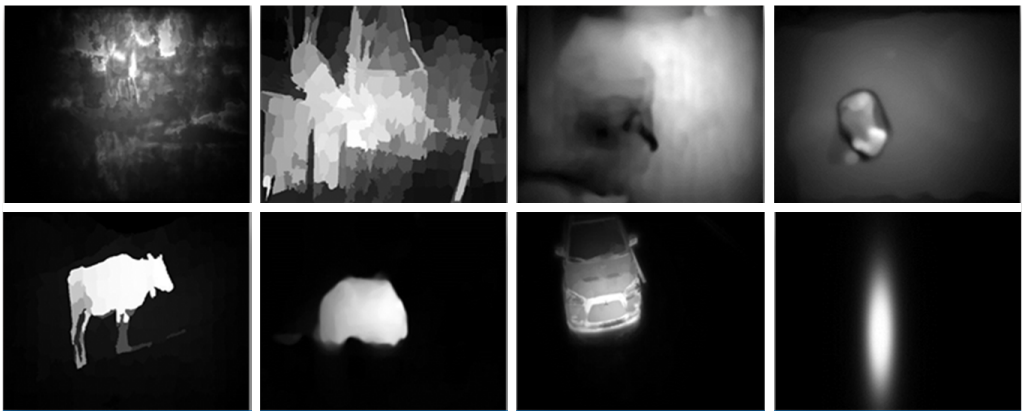


Fig. 3.3 Some selected saliency map training samples: the first row shows some negative training samples (saliency maps with low quality) and the second row shows some positive training samples (saliency maps with high quality).

it to train the fusion model and the trained model will be fixed throughout all the experiments. Each training sample $\{\mathbf{x}_i, y_i\}$ corresponds to an actual saliency map S_i where $\mathbf{x}_i$ is the 13 dimensional feature vector extracted from S_i and y_i is the binary label indicating the quality of S_i. The value of y_i is determined by the percentage of S_i's saliency energy inside the ground truth bounding box with respect to the saliency energy of the whole map, *i.e.*, $z_i = \frac{Trace(S_i{}^T G_i)}{\mathbf{1}^T S_i \mathbf{1}}$ where G_i is the ground truth saliency map corresponding to S_i which is obtained by filling its ground truth bounding box with 1 and the rest with 0. We then set $y_i = +1$ if $z_i \geq 0.8$, $y_i = -1$ if $z_i \leq 0.2$ and discard the rest. In total, we have collected 2078 positive samples and 1982 negative samples. See Fig. 3.3 for some examples.

To predict the quality of a given saliency map, a support vector machine with RBF (Radial Basis Function) kernel is trained on the training samples and the parameters are selected by cross-validation. We observe that our cross-validation accuracy is about 97% which implies that the 13 dimensional features can well reflect the quality of the saliency map. We then use the learned SVM model to predict the quality of a given saliency map, and the

probability estimates [6] are used as the combination weights, *e.g.*, a saliency map with 90% probability of being a good map will have weight 0.9. Note that we only use the SVM model to predict the combination weights of the four images and motion saliency maps. The weights of the two semantic saliency maps are always set to 1 because they are indeed smoothed bounding boxes and always exhibit very compact distributions. Some examples of the combined maps using this learned weight are shown in the last row of Fig. 3.7, which is significantly better than average. Another advantage of the proposed method is that it is very fast and convenient to use as only the trained SVM model is required during testing and the trained model can be used to any new saliency estimation techniques without retraining.

Post processing

In general, high quality motion saliency cues are more reliable than image saliency cues in videos as it is more robust to cluttered background [150]. This is because, in an automatic setting without initialization, regions moving together is more likely to correspond to an object than regions with uniform appearance. Hence, we empirically emphasize motion cue by suppressing the weights of the two image saliency cues when the former is of good quality. More specifically, if we use w_p, w_a, w_g and w_ω to denote the weights of *PCA Image Saliency*, *AMC Image saliency*, *GC Motion Saliency* and ω *Motion Saliency*, respectively, we perform the following nonlinear weight adjustment:

$$(w_p, w_a, w_g, w_\omega) := \begin{cases} (0, 0, w_g, w_\omega), & w_g > \eta \ \textit{or}\ w_\omega > \eta \\ (w_p, w_a, w_g, w_\omega), & \textit{otherwise} \end{cases} \tag{3.3}$$

where η is a threshold (set to 0.8 in the experiment) to determine whether the quality of motion saliency is good enough. The saliency maps are then fused

linearly using the adjusted weights and normalized to the range between 0 and 1.

Temporal warping is also applied to the fused saliency maps based on the optical flow directions to further enforce the temporal consistency of the fused saliency maps. Similar to [96], we apply both forward and backward warping. The forward warping is formulated as:

$$S_f := w_f S_f + \tau w_{f-1} S_{f,f-1} \tag{3.4}$$

where S_f is the saliency map of frame f and $S_{f,f-1}$ is the warped saliency map from frame $f-1$ to frame f, w_f is the *SVM-Fusion* quality measure on S_f and τ is a positive decay weight smaller than one. Note that the warping is done sequentially from the first frame to the last frame and the warped version of S_{f-1} is used to update S_f. This means that we have implicitly used all the previous frames before S_f to update it. Similarly, the backward warping is formulated as

$$S_f := w_f S_f + \tau w_{f+1} S_{f+1,f} \tag{3.5}$$

and is performed from the last frame to the first frame. These two warping processes are performed separately and the resultant maps are averaged to give the final warped saliency map.

This temporal warping process is very useful in our experiment. It can effectively reject noise and recover missed detections by considering its neighboring frames. Fig. 3.4 shows an example in which the warping process successfully recovers the missed saliency detections. However, in rare cases where many adjacent frames are corrupted by consistent noise, this warping process will also propagate this noise to nearby frames.

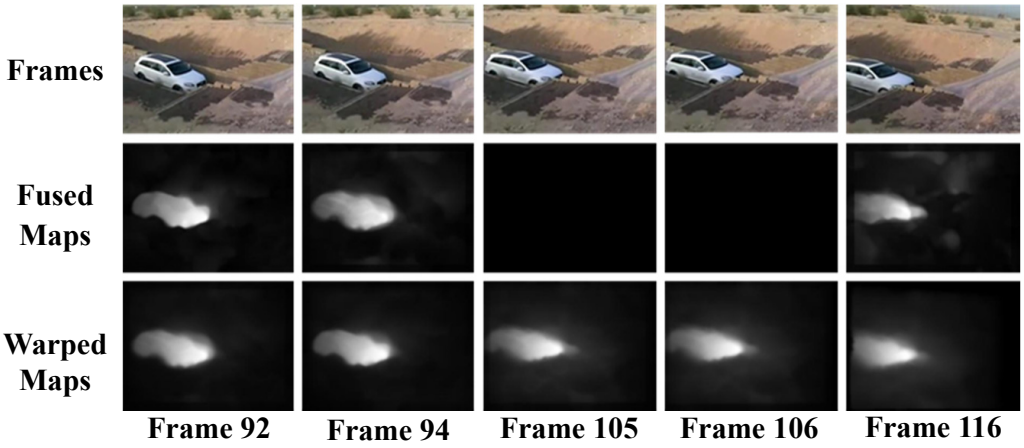


Fig. 3.4 An example showing how map warping recovers missed saliency detections. The top row is the original frames, the second row is the fused saliency maps and the third row is the warped saliency maps.

3.3 Thematic Object Discovery by Max Path Search

After we have obtained the fused saliency maps for all the frames, each video is now represented as a collection of saliency maps, $V = \{S_i\}$ where S_i denotes the saliency map of the i^{th} frame. In the following sections, we will use $S(t, x, y)$ to denote the saliency score at pixel location (x, y) on frame t. The fused saliency maps, even when correctly highlighting the thematic object, may still contain other salient objects or salient background regions. Therefore, we employ the max path search algorithm [119] to detect temporally consistent salient regions for our thematic object detection. The detection result is in the form of a spatiotemporal path where each node corresponds to a bounding box on a frame. In the following sections, we will first give an overview of the max path search algorithm and then discuss how we formulate our detection problem in the max path search framework.

3.3.1 Overview of Max Path Search

Max path search algorithm [119] is an optimal path discovery technique that searches for a global optimal spatio-temporal path in the 3D volume or trellis. Suppose we have a 3D spatio-temporal volume $\mathbb{G}$ composed of a set of nodes, $n_i \in \mathbb{G}$, indexed by its location (x_i, y_i) and time, t_i. A path p in $\mathbb{G}$ is defined as a temporal sequence of nodes, $p = \{n_1, n_2, ..., n_m\}$, which satisfies the path connectivity constraints between adjacent nodes. For example, a temporal adjacent connectivity constraint can be expressed as $t_{i+1} = t_i + 1$ and a spatial 8-neighbor connectivity constraints can be expressed as $x_i - 1 \leq x_{i+1} \leq x_i + 1$, $y_i - 1 \leq y_{i+1} \leq y_i + 1$. Each n_i has an associated score s_i and we define the overall score of a path p, $M(p)$, as the accumulated scores of all its nodes:

$$M(p) = \sum_{i=1}^{N(p)} s_i, \tag{3.6}$$

where $N(p)$ is the length of path p. The max path search algorithm can then be used to find the global optimal path p^* (with highest path score) in linear time complexity, *i.e.*, $O(w \times h \times n)$ where w, h and n denote the width, height and length of the spatio-temporal volume, respectively:

$$p^* = \arg\max_{p \in path(\mathbb{G})} M(p), \tag{3.7}$$

where path($\mathbb{G}$) denotes the set of all possible paths in $\mathbb{G}$. The detailed description of the algorithm and proof of the global optimality and time complexity can be found in [119].

3.3.2 Salient Path Discovery via Max Path Search

Since our saliency map assigns per pixel saliency score, it is natural to treat each pixel as a node. However, our detection requires finding a path

representing a spatio-temporal volume instead of a spatio-temporal pixel trajectory. Hence we want each node on the path to correspond to a bounding box instead of a pixel. We adapt a similar approach to [119] where each node still corresponds to a pixel location but the score of the node is the saliency energy of a window centered at that pixel. We now use $\Omega_{p,n}$ to denote the n^{th} node of path p and its score can be expressed as

$$s_{p,n} = \sum_{i=t_{p,n}}^{i=b_{p,n}} \sum_{j=l_{p,n}}^{j=r_{p,n}} S(k_{p,n}, i, j), \tag{3.8}$$

where $b_{p,n}$, $t_{p,n}$, $l_{p,n}$ and $r_{p,n}$ denotes the bottom, top, left and right coordinates of the bounding box corresponding to node $\Omega_{p,n}$ and $k_{p,n}$ denotes the frame in which node $\Omega_{p,n}$ resides. In order to support scale variations in our detection, each pixel location can correspond to more than one node differed by the window size. The window size can be represented as two extra parameters, scale and aspect ratio, which can be embedded into the original 3D spatio-temporal trellis. For example, if we allow s different scales and a different aspect ratios in the detection, the original $w{\times}h{\times}t$ 3D trellis will become a $w{\times}h{\times}t{\times}s{\times}a$ 5D trellis. Similarly, we can add connectivity constraints to these new dimensions, *e.g.*, a connectivity constraint requiring the two immediately connected nodes to have the same or adjacent scale and aspect ratio levels can be expressed as $s_i - 1 \leq s_{i+1} \leq s_i + 1$ and $a_i - 1 \leq a_{i+1} \leq a_i + 1$, where s_i and a_i denote the scale and aspect ratio levels of node n_i, respectively. Mathematically, our object discovery problem can be formulated as the following optimization problem which can be solved by the max path search:

$$p^* = \arg\max_{p \in path(\mathbb{G})} \sum_{n=1}^{N(p)} s_{p,n}. \tag{3.9}$$

In addition, due to the score summation operation in the node and path score computation, the saliency score must be discriminative, *e.g.*, positive score means salient region and negative score means non-salient region. Otherwise the optimal path will always span from the first frame to the last frame and each node will correspond to the maximum possible bounding box. Hence, we subtract a small positive offset, γ, from the original saliency score such that both the path and bounding box can exclude non-salient regions. We will evaluate the selection of this small positive number in Section 3.5.

3.4 Iterative Appearance Modelling

Although our max path search over fused saliency maps can apparently improve several baseline approaches as shown in the experiment, it still lacks an explicit appearance model among the nodes on a path. As a result, this can cause the detected path to drift from the thematic object to other salient objects or salient background regions. In other words, the pure saliency-based detection framework can only identify if region A and region B are salient but is not able to identify whether salient region A and salient region B correspond to the same salient object. An appearance model to enforce the inter-node consistency would largely alleviate this problem. In this section, we will discuss how we explicitly model the appearance of the thematic object using the initially discovered salient path. In short, we use the initial thematic object detections based on the discovered salient path as weakly supervised information to iteratively learn the appearance model of the thematic objects as well as the background.

We first represent each video sequence as a collection of superpixels, $\mathcal{T} = \{p_i\}$ and the saliency score of the i^{th} superpixel, s_i, is defined as the average saliency value of its enclosing pixels. Three types of features are

used to describe each superpixel, *i.e.*, dilated dense SIFT (Scale Invariant Feature Transform) [75] histogram, dilated texton histogram and mean color in the RGB space. These features have also been shown to be very useful in image parsing [116]. The total feature vector dimension is 203 as 100 visual words are used for both the dilated dense SIFT histogram and dilated texton histogram. Note that we don't use the color histogram and color thumbnail features as in [116] because the size of our superpixel is quite small and its color is very uniform. In the following, we use f_i to denote the feature vector of the i^{th} superpixel.

To explicitly model the appearance of the thematic object, we iteratively train a linear SVM classifier which treats the superpixels in the background as negative samples and the superpixels in the thematic object as positive samples. At the first iteration, we select the training samples based on the fused saliency map and the discovered salient path. A superpixel will be selected as a positive training sample if it is completely inside the salient path and its saliency score is high enough while a superpixel will be selected as negative training sample if it is completely outside the salient path. Then a linear SVM is trained and used to assign each superpixel, p_i, a probability of being part of the thematic object, q_i. Subsequently, the positive and negative training samples are reselected for the next iteration based on this probability. The iteration will stop when the training samples in adjacent iterations do not change much, *i.e.*, more than 99.5% of the training samples are the same. This whole process is summarized in Algorithm 1 and we will discuss the selection of the involved parameters in Section 3.5. The final probability estimate, q_i of each superpixel, s_i is then used to vote a pixel-wise detection confidence map on the thematic object.

Although some related appearance modeling techniques have been explored in videos and images (collections) such as [14, 96, 37], our method

Algorithm 1 Iterative Appearance Modelling

Input: salient path P, the collection of superpixles $\mathcal{T} = \{p_i\}$ and their corresponding saliency score $\{s_i\}$ and feature vector $\{f_i\}$,
Parameters: θ_s is the threshold to select the positive training superpixels before the first iteration, θ_u and θ_l are the thresholds to select the positive and negative training superpixels in the following iterations
Output: probability estimate, $\{q_i\}$, of each superpixel being part of the thematic object.

$\mathcal{F} = \mathcal{B} = \emptyset$
for each $p_i \in \mathcal{T}$ **do**
    if $p_i \notin P$ **then**
        $\mathcal{B} = \mathcal{B} \cup f_i$
    else if $s_i > \theta_s$ **then**
        $\mathcal{F} = \mathcal{F} \cup f_i$
    end if
end for

while true **do**
    $\mathbf{M} = TrainLinearSvm(\mathcal{F}, \mathcal{B})$
    for each $p_i \in \mathcal{T}$ **do**
        $q_i = PredictLinearSvm(\mathbf{M}, f_i)$
    end for

    $\mathcal{F} = \mathcal{B} = \emptyset$
    for each $p_i \in \mathcal{T}$ **do**
        if $q_i > \theta_u$ **then**
            $\mathcal{F} = \mathcal{F} \cup f_i$
        else if $q_i < \theta_l$ **then**
            $\mathcal{B} = \mathcal{B} \cup f_i$
        end if
    end for
    if $\mathcal{B}$ and $\mathcal{F}$ do not change **then**
        break
    end if
end while

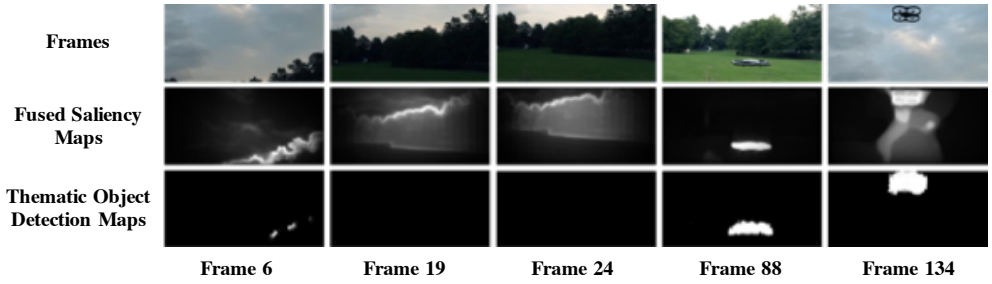


Fig. 3.5 Comparison between the saliency maps and the thematic object detection maps after appearance modeling for several frames of an example video clip. In this video clip, the thematic object is the aerial vehicle which enters the scene at frame 31.

has some unique properties: (1) it is fully automatic and does not require any human interventions; (2) it is a global model for the entire video; (3) it integrates the appearance of thematic object and background in a single unified discriminative model. A global model allows information sharing between distant frames and a discriminative model can better differentiate the thematic object against background. This is especially helpful for videos because many frames share common background and the relative position of the thematic object usually changes in the background across frames, *i.e.*, cover different portions of the background. An example is shown in Figure 5 where the thematic object only enters the scene after frame 31. The pure saliency-based detection will also include the salient regions around the trees in the beginning frames as shown in the second row. However, these tree regions can be successfully suppressed in our appearance modeling process because of the negative (background) training samples selected around the tree regions in the later frames where the detections are correct. The thematic object detection maps in the later frames after the object enters the scene also look cleaner compared with the saliency map. Last but not least, thanks to the efficient implementation of linear SVM with bag of words like sparse

features, the modeling process is very efficient as shown in Table 3.6 even with large training sizes, *i.e.*, thousands of superpixels per frame.

After obtaining the detection maps F, the max path search is run to produce the final detection result. In addition, we also convolve the detected path by a median and mean filter along the temporal axis to get a smoother detection.

3.5 Experiments

We have evaluated the performance of the proposed detection framework on the *NTU-Adobe* dataset and some existing benchmark datasets, *i.e.*, the *10-video-clip* dataset and some selected categories from the *UCF Sports Action* dataset. We first introduce the employed evaluation metrics in Section (3.5.1) and the new *NTU-Adobe* dataset in Section (3.5.2). Then we evaluate the proposed *SVM-Fusion* and iterative appearance modeling in Section (3.5.3) and (3.5.4), respectively. Finally we compare the proposed technique with two state-of-the-art object discovery methods, the methods in [156] and [96], and one state-of-the-art object tracking method, the method in [39], using the *NTU-Adobe* dataset in Section 3.5.5. We also compare with another video salient object detection method [77] using the *Ten-Video-Clip* dataset [31] and some selected categories of the *UCF Sports Action* dataset [104]. The computational cost of the proposed method is discussed in Section (3.5.6).

3.5.1 Evaluation Metrics and Experimental Setup

Three metrics are used in the evaluation process: (1) *CDR (correct detection ratio)*: This metric measures the quality of the detected path for each video. A frame is considered to be correctly detected if the *overlap over union* ratio between the detected bounding box and the ground truth bounding box is

greater than 0.5. The frames that are neither on the ground truth path nor the detected path will not be considered; (2)*FMS (f-measure of saliency map)*: This metric directly measures the quality of the saliency map compared with the ground truth saliency map. We follow the standard definition of *f-measure* in terms of *precision* and *recall*: $f\text{-}measure = \frac{2 \times precision \times recall}{precision + recall}$. Let S_g and S_d denote the ground truth saliency map and the estimated saliency map, respectively, the precision and recall are computed as $precision = \frac{Trace({S_g}^T S_d)}{\mathbf{1}^T S_d \mathbf{1}}$ and $recall = \frac{Trace({S_g}^T S_d)}{\mathbf{1}^T S_g \mathbf{1}}$, respectively; (3) *FMP (f-measure of path)*: This is the metric used in [77] and it measures the quality of a detected path. It is defined as:

$$FMP = \frac{(1+\alpha) \times \text{precision} \times \text{recall}}{\alpha \times \text{precision} + \text{recall}} \tag{3.10}$$

where $\text{precision} = \frac{|M_g \cap M_d|}{|M_d|}$ and $\text{recall} = \frac{|M_g \cap M_d|}{|M_g|}$. M_g and M_d are the mask on the ground truth and detected thematic object region, respectively. It is computed for each frame and averaged for each video. We use it to compare with the results reported in [77].

After obtaining the combined saliency maps, we subtract a small positive number, 0.2, from the original saliency scores to get discriminative values. We apply the immediate neighbor connectivity constraint for all the dimensions, *e.g.*, $t_{i+1} = t_i + 1$, $x_i - 1 \leq x_{i+1} \leq x_i + 1$, $y_i - 1 \leq y_{i+1} \leq y_i + 1$, $s_i - 1 \leq s_{i+1} \leq s_i + 1$ and $a_i - 1 \leq a_{i+1} \leq a_i + 1$ and, hence, each node will have $3^4 - 1 = 80$ neighbors. To support a wide range of scale variations, we set the allowed bounding box scale (width) as $40, 60, ..., 240$ and the aspect ratio (*width/height*) as 0.4, 0.8, 1.0, 1.4, 1.8, 2.0. In addition, we choose a step size of 10 pixels vertically and horizontally in the two spatial dimensions while scanning the 5-D trellis in the max path search algorithm for efficiency. All the following experiments will use the same parameter configurations. In addition, in order to evaluate the effect of the empirically chosen small offset

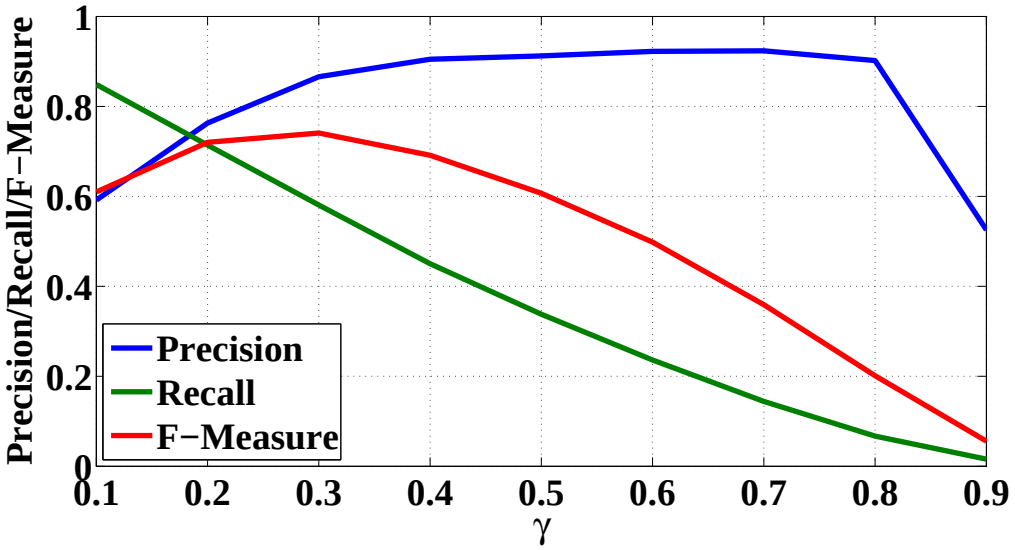


Fig. 3.6 The precision, recall and f-measure of the detected salient path while different offset values, γ, are subtracted from the saliency maps.

γ subtracted from the saliency value, we run the saliency-based detection on the NTU-Adobe dataset without appearance modeling using different γ values and the results in terms of the path precision, recall and f-measure (computed based on the definition of FMP with $\alpha = 1$) are shown in Fig. 3.6. As expected, the larger the γ, the lower the recall and the higher the precision. The highest detection accuracy in terms of f-measure is achieved when γ is around 0.2 to 0.3 and the detection result is quite stable around this range.

3.5.2 NTU-Adobe dataset

Most of the existing video object detection benchmark datasets have focused on specific categories of objects such as the *UCF Sports Action* dataset [104] for human action detection and the *UIUC-NTU Youtube Walking* dataset [119] for walking pedestrian detection. The huge *Youtube Object* dataset [3] is a multi-category dataset but only weakly annotated. A densely-annotated

[3] http://people.ee.ethz.ch/ presta/youtube-objects/website/

multi-category dataset for automatic thematic video object discovery is desirable. Hence we collect a new multi-category dataset containing 51 video clips including animals (11 videos), babies (19 videos), walking or standing pedestrians (7 videos), cars (5 videos), motorcycles (3 videos), helicopters (2 videos), toy cars (2 videos), boat (1 video) and parachute (1 video). Example frames can be found in Fig. 3.1, 3.2, 3.4, 3.5, 3.7 and 3.9. This new dataset contains 18834 frames in total and the resolution ranges from 320×240 to 640×360. In this new dataset, 9 videos are borrowed from the *Youtube Object* Dataset, 3 videos are borrowed from the *SegTrack* dataset[4], 3 videos are borrowed from the *UIUC-NTU Youtube walking* dataset and the other 36 videos are downloaded from YouTube. Most of the videos are "home-made" videos without advanced video editing because we are mainly targeting personal videos instead of professional ones like films or commercial advertisement. As a result, there are few shot changes and the objects always move smoothly during its presence. The ground truths are manually labeled in the form of bounding boxes on each frame containing the thematic object. Note that each video only has one thematic object. In addition, this dataset can also be used for thematic object discovery in a group of videos because some videos share the same thematic object. The dataset can be downloaded from our project website[5].

3.5.3 Saliency Fusion

In this section, we compare the performance of each individual saliency map and the different saliency fusion techniques using *CDR* which measures the quality of the detected salient path. We first compare our proposed *SVM-Fusion* technique (without nonlinear weight adjustment and map warping)

[4]http://cpl.cc.gatech.edu/projects/SegTrack/
[5]http://jiongsresearch.weebly.com/primary-video-object-discovery.html

with the individual saliency maps, the "best" fusion weight based on Eq.(3.2) and some other existing map fusion techniques in the literature. These methods include *Max* [86], *Multiplication* [86], *Mean* [86], *Spatial Variance* [26] *Motion Variance* [150] and *Pixel-wise Aggregation (PW)* [84]. Please refer to the respective papers for the technical details. The results are summarized in Table 3.1. As expected, the "best weight" has the highest detection accuracy and can be regarded as an upper bound of the best results achievable using weighted combination. Note that *PW* is not using weighted combination and, hence, its performance is not bounded by this "best weight" theoretically. It can be seen that our proposed *SVM-Fusion* technique outperforms the other techniques. We also show some qualitative results in Fig. 3.7 comparing the *Mean* and *SVM-Fusion* technique to demonstrate the effectiveness of the proposed learning based fusion approach. From the result we can see that the proposed *SVM-Fusion* technique can adaptively assign lower weights to the corrupted saliency maps and emphasize the good ones.

We have also evaluated the effectiveness of the two post-processing steps, *i.e.*, *nonlinear weight adjustment* and *map warping*, and the results are shown in Table 3.2. The results show that the map warping process can apparently improve the performance while the *nonlinear weight adjustment* seems to degrade the performance when used alone. However, when the *nonlinear weight adjustment* is used together with *map warping*, it improves the performance.

Table 3.1 Evaluation results of different saliency map fusion techniques on *NTU-Adobe* dataset.

	CDR
PCA Saliency	35.93%
AMC Saliency	33.92%
GC Saliency	47.72%
W Saliency	36.26%
Max[86]	32.34%
Mean[86]	59.82%
Multiplication[86]	17.98%
Spatial Variance[26]	63.23%
Motion Variance[150]	63.71%
PW[84]	65.00%
SVM-Fusion(ours)	68.81%
Best (upper bound)	79.00%

Table 3.2 Evaluation results of the nonlinear fusion weight adjustment and map warping.

SVM-Fusion	*Nonlinear Adjustment*	*Warping*	*CDR*
✓			68.81%
✓		✓	70.58%
✓	✓		67.72%
✓	✓	✓	72.92%

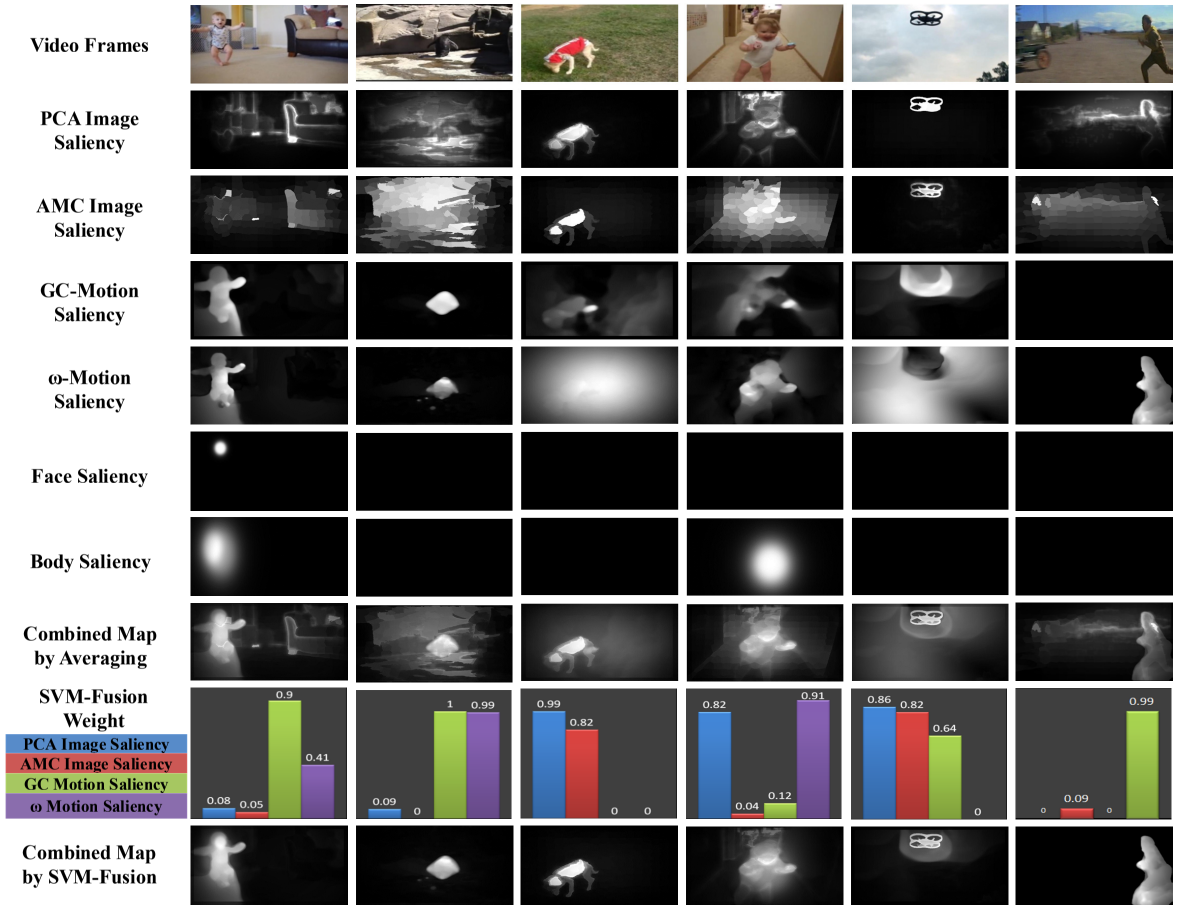


Fig. 3.7 Examples of the various types of saliency maps and the map fusion results by averaging and our proposed *SVM-Fusion* method without *nonlinear weight adjustment* and *warping*. The first five examples are from the *NTU-Adobe* dataset and the last example is from the *Camo and Hollywood2* dataset [94].

Table 3.3 Evaluation results of different saliency map fusion techniques on the *FT* dataset.

	FMS
PCA Saliency	0.63
AMC Saliency	0.73
Global Contrast Saliency	0.71
GBMR Saliency	0.77
Max[86]	0.71
Mean[86]	0.75
Multiplication[86]	0.63
PW[84]	0.72
Spatial Variance[26]	0.76
SVM-Fusion(ours)	0.79
Best (upper bound)	0.80

Although our fusion method is proposed to fuse appearance and motion cues, we conduct one more experiment to explore its potential to fuse only image saliency cues. The *FT* dataset [1] is used for evaluation. Four image saliency estimation algorithms are used, *i.e.*, PCA Saliency [87], AMC Saliency [50], GBMR(Graph-Based Manifold Ranking) Saliency [138] and Global Contrast Saliency [15]. We use the previously trained SVM-Fusion model in this experiment since it is independent of the saliency estimation methods. We sampled 1K images from the *MSRA10K* [15] dataset, *i.e.*, rank all the 10K images in descending order and use the first 1K images, to train the PW model since it requires the training to use the same set of saliency estimation techniques as the testing process. The saliency estimation accuracy is measured by *FMS*. The experiment result is shown in Table 3.3.

It can be seen that our fusion method can improve the saliency accuracy from 0.77(best accuracy of individual map) to 0.79 and all the other fusion method drop the performance. Note that the best possible fusion accuracy according to Eq.(3.2) is only 0.80. This implies that the potential performance gain of fusing different image saliency cues is not very significant. This is expected because if one image saliency estimation algorithm performs badly

on a particular image, other image saliency algorithms are also likely to perform badly on that image since all of them rely on appearance cue at the first place. This further confirms that it is more meaningful to fuse appearance and motion cues.

3.5.4 Appearance Modelling

In this experiment, we present the evaluation results on the appearance modeling. We use the SLIC [2] algorithm to segment each video frame into roughly 1500 superpixels. The parameters for training samples selection are set as: $\theta_s = 0.3$ and $(\theta_l, \theta_u) = (0.5, 0.5)$, respectively. The *CDR* on the *NTU-Adobe* dataset without and with the appearance modeling are 72.92% and 81.19%, respectively. Note that our detection framework without appearance modeling has already done a good job in many videos. But there are still cases where the saliency detection is distracted by the background even after *SVM-fusion*, or the thematic object only appears in part of the video. The adoption of the appearance modeling can alleviate the distraction of the background and significantly boost the performance in these cases. Some quantitative and qualitative results are shown in Fig. 3.9. The first row shows an example in which the thematic object only enters the video after frame 31 and the second row shows an example in which the thematic object leaves the scene before the video ends. In both cases, the explicit appearance modeling can successfully exclude those irrelevant frames. The third and fourth rows show examples where saliency maps are noisy and the pure saliency-based detections include many background regions or cannot cover the entire object. The appearance modeling significantly improves the detections in these cases. In addition, we have also evaluated the sensitivity of our appearance modeling technique with respect to the choice of θ_s, θ_l and

θ_u. We first evaluate θ_s by fixing θ_l and θ_u to be 0.3 and 0.7, respectively. The evaluation result is shown in the left column of Fig. 3.8. From the result, we can see that 0.3 is a reasonable choice as the detection performance is very stable around 0.0 to 0.3 and starts to drop apparently from 0.4 onwards. This is expected as the purpose of θ_s is to reject the background regions around the boundary of the bounding box and a relatively small value should be appropriate. For θ_l and θ_u, we fix θ_s to be 0.3 and evaluate them in pair, *e.g.*, (0.1, 0.9), (0.2, 0.8). The evaluation result is shown in the right column of Fig. 3.8. It can be seen that the detection accuracy is the highest at (0.5, 0.5) and remains very stable from $(0.3, 0.7)$ to $(0.7, 0.3)$. At first glance, it may look unreasonable that the performance is still high at $\theta_l = 0.7$ and $\theta_u = 0.3$ since this seems to include many background superpixels into the positive training set and vice versa. However, we have observed that during the iterations, most of the superpixels are assigned near extreme values, *i.e.*, approaching 0 or 1. Hence, adjusting these parameters within the range from (0.3, 0.7) to (0.7, 0.3) will have a relatively small impact on the sample selection during each iteration. This also implies that our method is not sensitive to these two parameters and (0.5, 0.5) is a reasonably good choice. Note that the detection accuracy is relatively low around $(0.1, 0.9)$ and $(0.2, 0.8)$ because it may be too strict in selecting training samples with this setting.

3.5.5 Comparison with state of the arts

We first compare the performance of the proposed framework with [156], [39] and [96] on the NTU-Adobe dataset using *CDR*. [156] employs the LDA model to discover the thematic video objects. In the experiment, we use the same setting as [156] but do not incorporate the word co-occurrence prior as it is not reliable in the employed videos. The output of this method is a

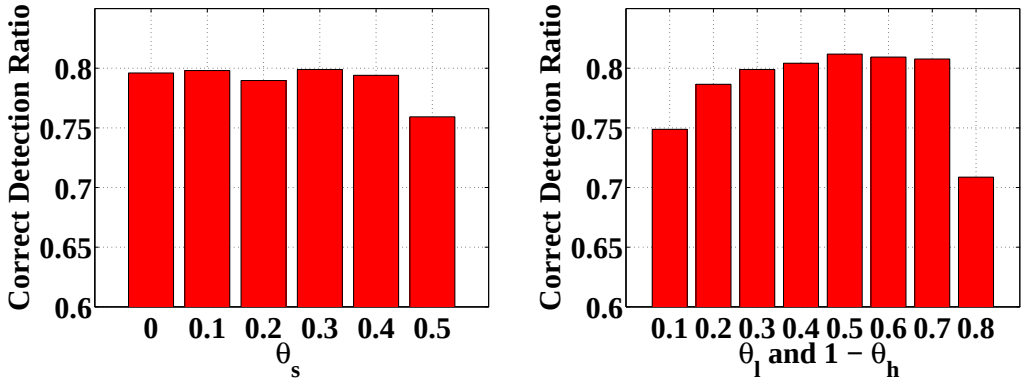


Fig. 3.8 Correct detection ratio with different θ_s and (θ_l, θ_u)values; the left curve shows the effect of θ_s while fixing $(\theta_l, \theta_u) = (0.3, 0.7)$ and the right curve shows the effect of (θ_l, θ_u) while fixing $\theta_s = 0.3$.

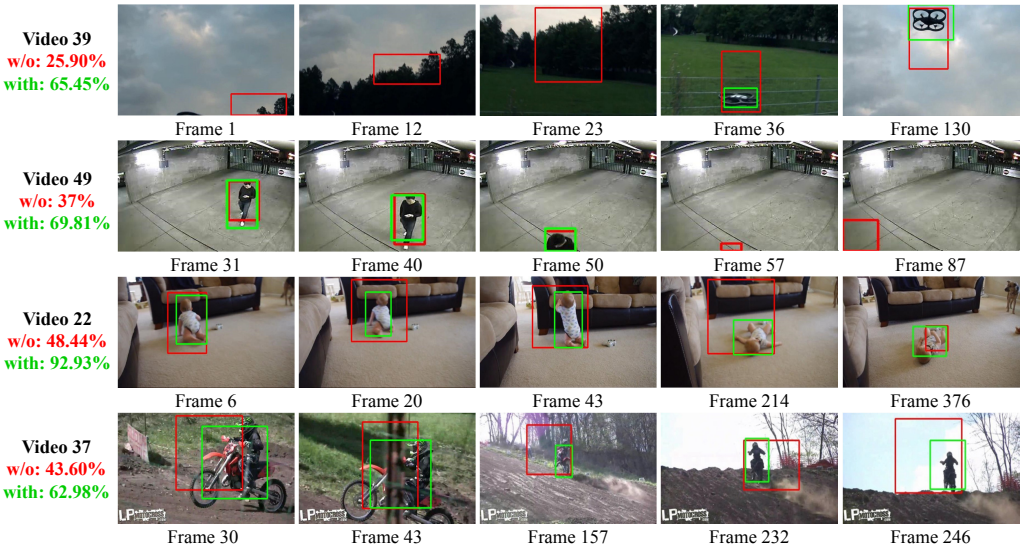


Fig. 3.9 Comparisons of the detection results with and without appearance modeling for some cases where the pure saliency-based detection fails. Each row refers to one video and the first column shows the overall detection accuracy in *CDR*. In the subsequent columns, the red and green box indicate the detection results before and after appearance modeling, respectively.

set of bounding boxes which localize the objects in frames. [39] is one of the best video object tracking methods evaluated in [136]. In the experiment, we use the ground truth bounding box on the first frame (or the nearest frame if the first frame does not contain the thematic object) of each video to manually initialize the tracking. The output of this technique is a set of tracked bounding boxes on the subsequent frames. The method in [96] is the most recent state-of-the-art automatic thematic video object segmentation method. The output of this technique is per-frame segmentation masks. For comparison, we fit a minimum bounding box on the largest connected regions on the segmentation mask of each frame as the detection result. We have summarized the *CDR* of these methods as well as our proposed detection framework in Table 3.4. Besides the overall comparison results, we also list the results for each category in the dataset. Our method performs better than the others on the "*animals*", "*babies*", "*cars*", "*motorcycles*" and "*people walking*" categories, and worse than [96] on the "*others*" category. In the "*others*" category, there is one video sequence in which the thematic object occupies the whole width of the frame throughout the video and the max path search space does not cover that large bounding box size for efficiency. In another video sequence, the thematic object becomes very small for a long duration and both the two image saliency cues incorrectly focus on a very compact background region which corrupts the fused saliency map. [96] correctly identifies the thematic object because it does not rely on the image saliency cues. However, we cannot simply abandon the image saliency cues because they are very useful in many other videos as shown in the comparison in Table 3.1. On average, our method outperforms all the others, *e.g.*, around 14% improvement compared with [96]. Note that the tracking method [39] needs manual initialization and our method is completely automatic in terms of user interaction.

Table 3.4 Comparison with state of the arts on *NTU-Adobe* dataset using the correct detection ratio.

	[156]	[39]	[96]	Ours
animals (11 videos)	26.54%	35.11%	71.11%	**76.46%**
babies (19 videos)	16.50%	47.05%	61.01%	**84.31%**
cars (5 videos)	52.00%	44.38%	86.79%	**90.20%**
motorcycles (3 videos)	34.00%	39.35%	63.50%	**83.59%**
people walking (7 videos)	32.39%	49.55%	57.29%	**85.73%**
others (6 videos)	20.33%	59.55%	**83.08%**	64.93%
all (51 videos)	25.79%	44.99%	67.95%	**81.19%**

We also compare with [77] using our detection framework on the datasets used in their paper. [77] is a pure saliency-based detection approach. It fuses two saliency maps, *i.e.*, image saliency and motion saliency, by average and uses the max path search to find the thematic object. In addition, it uses the optical flow connectivity to model the edge score between two temporally adjacent nodes in the trellis. In our approach, we don't model the edge score because, in the max path search framework, each edge score can only depend on its two immediately connected nodes which limits its effectiveness, while the extra computational cost is significant. We run our detection framework on the *10-video-clip* dataset [31] and the *skate boarding* (12 videos), *swing side angle* (13 videos) and *run side* (13 videos) categories of the *UCF Sports Action* dataset [104]. In order to compare with their reported result, we use the same metric as in [77] to evaluate the detection accuracy. Note that the *horse riding* category is not chosen because the ground truth labeling of many sequences are not suitable for thematic object detection, *i.e.*, both the horse and person should be the thematic object but only the person is labeled. The results are summarized in Table 3.5. From the results, we can see that our detection framework outperforms [77] especially for the skate and swing categories. The relatively small improvement, *i.e.*, 0.02, in the *10-video-clip* dataset and the *run* category of the *UCF Sports Action* dataset

Table 3.5 Comparison with [77] using the fmp on the *10-video-clip* dataset and three categories of the *UCF Sports Action* dataset.

	10-video-clip	*skate*	*swing*	*run*
[77]	0.72	0.43	0.50	0.55
Ours	**0.74**	**0.59**	**0.60**	**0.57**

is because there are several video clips where all our four saliency maps miss the correct thematic object and we are unable to recover the detections by saliency fusion and appearance modeling. However, the performance on most of the other videos is superior. For example, in the *10-video-clip* dataset, our method outperforms [77] for 8 out of the 10 clips. We don't have the per-video statistic for the *run* category of the *UCF Sports Action* dataset as they only provide the final score in their paper.

3.5.6 Computational Cost

We summarize the averaged per frame computational time on the NTU-Adobe dataset in Table 3.6. We exclude the computational time of the various saliency maps, optical flow, SLIC superpixel, SIFT/Texton feature extraction as these are not our main contributions. Note that the proposed framework needs to extract the *SVM-Fusion* features from 5 maps (the 4 saliency maps plus the fused saliency map) and run the max path search twice (once on the warped saliency map and once on the detection map after appearance modeling). The max path search algorithm is implemented in C++ and the rest is implemented in Matlab. The experiments were conducted on a normal desktop computer with a quad-core i5 processor and 8GB of RAM.

If we consider all the saliency map computation and feature extraction steps, the end-to-end computational time of the proposed method would be around 13 seconds per frame. The tracking method in [39] reports a real-time

Table 3.6 The averaged (mean ± standard deviation) per frame computational time for the various modules.

	Time (ms)
SVM-Fusion Feature Extraction	47 ± 11
Fusion Weight Computation	0.019 ± 0.002
Fused Saliency Map Warping	97 ± 28
Iterative Appearance Modelling	219 ± 127
Max Path Search	58 ± 16

performance but an initialization is required at the first frame. The method in [96] runs at around 15 seconds per frame as reported in Table 4.3. The method in [156] does not report their end-to-end computational time, but it uses the normalized cut algorithm to segment each video frame under multiple different configurations, *i.e.*., different scales and segment numbers, and each configuration takes around 3 seconds. The method in [77] does not report the computational time at all but they use the method in [35] and optical flows as the input. [35] reports a computational time of 5 seconds per frame using GPU. Although our method is not fast among the compared ones, their computational times are within similar scales.

3.6 Conclusion and Future Work

In this work, we propose a novel approach for fully automatic thematic video object discovery. We first find a smooth spatiotemporal salient path in the video and then explicitly model the appearance of the thematic object and background in a global and discriminative manner. To make use of the strong complementation between appearance and motion saliency cues, we propose an effective fusion technique to adaptively fuse these two types of cues. The proposed fusion method is not only effective but also very fast and easy to use compared with similar methods in the literature. In addition, a new dataset containing 51 videos with per-frame bounding box labeling is proposed to

better suit the performance evaluation purpose of automatic thematic video object discovery. Experimental evaluations validate the superior performance of the proposed method compared with state-of-the-art approaches on both the new dataset and some existing benchmark datasets.

However, despite the promising performance, the proposed method only discovers and localizes the most thematic object in each video. It cannot be directly used to find multiple thematic objects. Of course, following [119], we can just run our method multiple times to discover multiple thematic objects by removing the previously discovered ones but we believe this is not the correct way to approach this problem. It requires the number of thematic objects to be known in advance and thus violates the fully automatic constraint of the proposed method. We plan to study the problem of discovering multiple thematic objects simultaneously in a fully automatic setting in our future work.

Chapter 4

Thematic Video Object Segmentation by Non-iterative Appearance Modeling

Automatic segmentation of the thematic object in a video clip is a challenging problem as there is no prior knowledge of the thematic object. Most existing techniques thus adopt an iterative approach for appearance modeling, i.e., fix the appearance model while optimizing the segmentation and fix the segmentation while optimizing the appearance model. However, these approaches may rely on good initialization and can be easily trapped in local optima. Also, they are usually time-consuming for analyzing videos. To address these limitations, we propose a novel and efficient appearance modeling technique for automatic thematic video object segmentation in the Markov Random Field (MRF) framework. It embeds the appearance constraint as auxiliary nodes and edges in the MRF structure and can optimize both the segmentation and appearance model parameters simultaneously

in one graph cut. Extensive experimental evaluations validate the superiority of the proposed approach over state-of-the-art methods, in both efficiency and effectiveness. This work has been published in the IEEE Transactions on Image Processing [139].

4.1 Introduction

The thematic object in a video sequence can be defined as the object that is locally salient and present in most of the frames [142, 151]. The target of automatic thematic video object segmentation is to segment out the thematic object in a video sequence without any human intervention. It has a wide range of applications including video object recognition, action recognition, and video summarization. Some examples are shown in Fig. 4.1. The existing works on video object segmentation can be divided into two groups based on the amount of human intervention required: interactive segmentation [46, 5] and fully automatic segmentation [64, 151, 96, 66]. Our method belongs to the latter and does not assume the object is present in all the frames.

Following the outstanding performance of Markov Random Field (MRF) based methods in image object segmentation [105, 114, 15], many of the existing video object segmentation approaches also build spatiotemporal MRF graphs and show promising results [96, 46, 151]. These approaches build a spatiotemporal graph by connecting spatially or temporally connected regions, *e.g.*, pixels [114] or superpixels [96], and cast the segmentation problem into a node labeling problem in a Markov Random Field. This process is illustrated graphically in Fig. 4.2. Such automatic thematic video object segmentation methods usually have three major steps: initial visual or motion saliency estimation, spatiotemporal graph connection, and thematic object appearance modeling. Automatic thematic object appearance modeling is important as

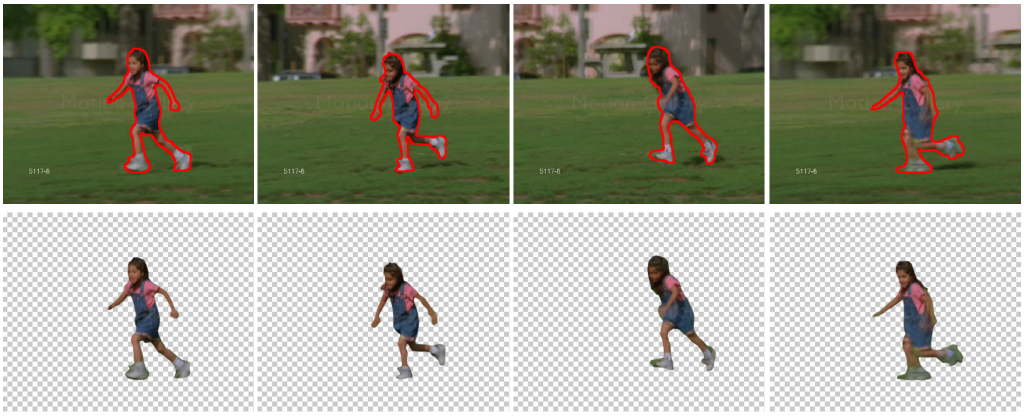


Fig. 4.1 Illustration of thematic object segmentation in videos. The top row is the original video frames with the expected segmentation results rendered as red contours. The bottom row is the same segmentation results after removing the background.

the saliency estimation is usually noisy especially along object boundaries due to the cluttered background or background motions. However, it is challenging because there is no prior knowledge about thematic object and background regions. Formally, with the presence of appearance constraints, there are two groups of parameters in the optimization process, *i.e.*, segmentation labels $\mathbf{x}$ and appearance model $\boldsymbol{\Theta}$. For many commonly used appearance models such as Gaussian Mixture Models (GMM) [96] or Multiple Instance Learning [130], it is intractable to solve both parameters simultaneously. Hence, many existing methods adopt an iterative approach. They use the segmentation result of the previous iteration to train the appearance models of the thematic object and background, which are then used to refine the segmentation in the next iteration. However, these methods can be easily trapped in local optima and are time-consuming especially for video data.

Recently, [114] proposed an appearance modeling technique in the graph based interactive image segmentation framework which can solve both the segmentation labels and appearance model parameters simultaneously without iteration. In their approach, they model each pixel as a node and quantize

it into a bin in the RGB histogram space. It shows that when the appearance of the thematic object and background are represented non-parametrically in the RGB histogram space, the appearance constraint is equivalent to adding auxiliary nodes and edges to the original MRF structure. However, due to the fundamental difference between image data and video data, the original approach in [114] is not practically applicable to video because it requires each node to be described by a single bin in the histogram space. For video object segmentation, superpixels are generally used due to the large data volume and more robust features like SIFT [75] or Textons are beneficial to better capture the viewpoint and lighting variations between different frames. As a result, each pixel will now have multiple features and each node will correspond to multiple pixels. Hence, in this work, we extend the efficient appearance modeling technique in [114] to thematic video object segmentation by addressing these challenges. The proposed appearance modeling technique is more general than [114] and can handle all the above-mentioned difficulties. The resultant auxiliary connections are also different from [114] because in [114] each pixel node is connected to one auxiliary node while in our approach each superpixel node can be connected to multiple auxiliary nodes. In summary, in [114], each node can only be described by a one-hot histogram feature, *i.e.*, only one bin of a histogram feature can have a nonzero value, while our method is applicable to the general histogram features or even the combination of multiple features. Experimental evaluations also validate the superiority of the proposed approach compared to directly applying [114] for automatic thematic video object segmentation.

In summary, the major contribution of this work is that we propose an efficient and effective appearance modeling technique in the MRF based segmentation framework for thematic video object segmentation. It embeds the appearance constraint directly into the graph by adding auxiliary

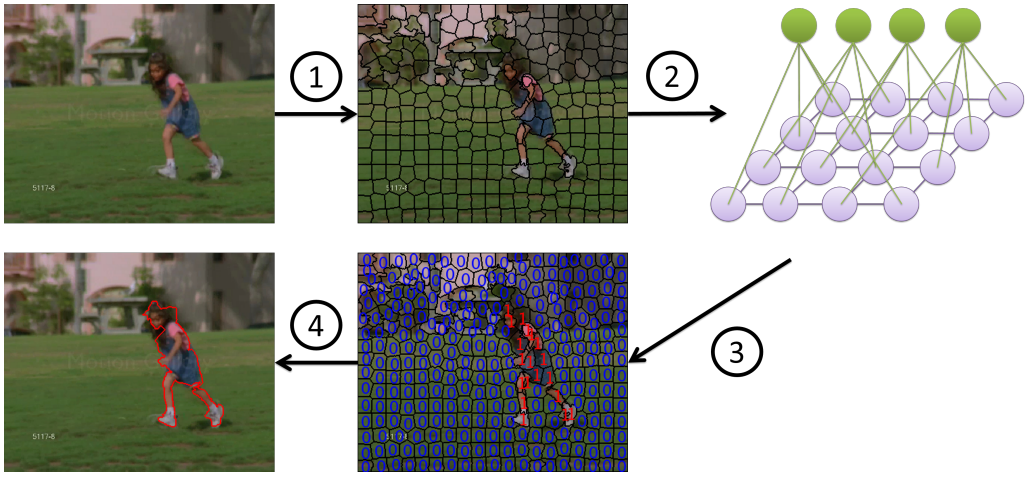


Fig. 4.2 The overall workflow of the proposed segmentation framework. 1. Superpixel segmentation; 2. Graph construction: the purple nodes and edges represent the superpixels and the spatiotemporal neighborhood connections between them. They are used to encourage the spatiotemporal smoothness of the segmentation. The green nodes and edges represent the auxiliary nodes and connections for appearance modeling. They are used to encourage the appearance coherence and disparity within and between the thematic object and background regions, respectively; 3: Node labeling by MRF inference; 4: Final segmentation result.

nodes/connections, and the resultant graph-partition problem can be solved efficiently by one graph cut. Although inspired by the idea of [114], we have made the non-trivial extension from static images to videos, and we generalize the framework in more complicated cases.

In the following sections of this chapter, we will present, in Section 4.2, the entire graph structure for thematic video object segmentation and emphasize how we formulate and optimize both the label and appearance model parameters simultaneously. The proposed method is evaluated in Section 4.3 on two benchmark datasets and compared with the recent state of the art. The entire chapter is concluded at Section 4.4.

4.2 Proposed Approach

In this section, we introduce the proposed approach for automatic thematic video object segmentation. The input is a plain video clip without any annotations and the output is a pixel-wise spatiotemporal segmentation of the thematic object in all the frames. Similar to many existing image and video object segmentation approaches, we cast the segmentation to a two-class node labeling problem in a Markov Random Field. Within the MRF graph, each node is modeled as a superpixel and will be labeled as either thematic object or background in the segmentation process. The overall workflow is shown in Fig. 4.2. In this work, we first segment each video frame into a set of superpixels using the SLIC algorithm [2] and then represent each node in the MRF as a superpixel. We typically have around 2500 superpixels per video frame. We choose not to use pixels because the computational and memory cost will be high for video data in our framework. Meanwhile, superpixels produced by SLIC [2] can preserve most of the boundaries, and over-segmentation is not a critical concern.

In the following, we use s_i^j to denote the j^{th} superpixel of the i^{th} frame, N to denote the total number of frames and M_i to denote the number of superpixels in the i^{th} frame. The segmentation target is to assign each superpixel s_i^j a label x_i^j indicating if it is part of the thematic object, $x_i^j = 1$, or background, $x_i^j = 0$. The overall optimization formulation in terms of the graph energy minimization is expressed as

$$\mathbf{x}^* = \arg\min_{\mathbf{x},\mathbf{\Theta}} E(\mathbf{s}, \mathbf{x}, \mathbf{\Theta}) \tag{4.1}$$

where $E(\mathbf{s}, \mathbf{x}, \mathbf{\Theta})$ is defined as

$$E(\mathbf{s}, \mathbf{x}, \mathbf{\Theta}) = \Phi_u(\mathbf{s}, \mathbf{x}) + \alpha_p \times \Phi_p(\mathbf{s}, \mathbf{x}) + \alpha_a \times \Phi_a(\mathbf{s}, \mathbf{x}, \mathbf{\Theta}). \tag{4.2}$$

The vector $\mathbf{x}$ and $\mathbf{\Theta}$ denote the $\{0, 1\}$ labeling of all the superpixels and the appearance model parameters, respectively, $\mathbf{s}$ denotes the collection of all the superpixels and Φ_u, Φ_p and Φ_a denote the unary potential, pairwise potential and appearance constraint potential, respectively. α_p and α_a are two weight parameters for linear combination.

4.2.1 Unary Potentials

Since saliency has been proven to be effective in highlighting the thematic object in a completely automatic setting by simulating where human looks [3, 56, 79, 96, 85, 149], we use it to model the unary potential of each node. In order to capture different aspects of saliency, four saliency estimations are employed including both appearance and motion saliency, *i.e.*, AMC image saliency [111], GBMR image saliency, [138], GC motion saliency [142] and W motion saliency [142]. To produce a single saliency estimation for each frame, we combine these saliency maps by weighted linear combination where the weight of each saliency map is determined by the SVM-Fusion technique proposed in [142]. The SVM-Fusion technique can adaptively predict the quality of each saliency map without using ground truth, and thus the weighted combination can adaptively reject noise and emphasize the most proper saliency cues for each individual frame. We also warp the saliency estimations along the optical flow direction to encourage temporal smoothness. The saliency value of a superpixel is then computed as the average saliency value of the pixels. An alternative is to use the peak saliency value instead of the average. However, we did not find these two approaches are statistically different under the paired t-test with a significance level of 0.05. Let $A(s_i^j)$ denote the saliency value of superpixel s_i^j, its unary potential

is given by:

$$\phi_u(s_i^j) = \begin{cases} -\log(A(s_i^j)) & \text{if } x_i^j = 1 \\ -\log(1 - A(s_i^j)) & \text{if } x_i^j = 0 \end{cases}. \tag{4.3}$$

The total unary term in Eq.(4.2) can be computed as

$$\Phi_u(\mathbf{s}, \mathbf{x}) = \sum_{i}^{N} \sum_{j}^{M_i} \phi_u(s_i^j). \tag{4.4}$$

This definition implies that it is costly to label a highly salient superpixel as background and vice versa.

4.2.2 Pairwise Potentials

There are two types of neighborhood relationships between superpixels in videos, *i.e.*, spatial neighborhoods and temporal neighborhoods. Two superpixels are spatially connected if they share a common edge and temporally connected if they have pixels linked by optical flow. In the MRF graph, only neighboring superpixels will have nonzero edge and the edge weight represents the cost induced by assigning different labels to the connected superpixels. Hence, the edge weight is usually measured as the inverse likelihood of the existence of a real edge between two superpixels. Apart from using local similarity, we also use the high level edge detections on both the appearance and motion domains to determine the edge weight. More specifically, we use color and optical flow orientation histogram to compute the local similarity and the structural forest edge detector [23] to compute the edge strengths. Note that, to detect motion boundaries for each frame, we first convert the XY dense flow vector of each pixel to a color representation using the method proposed in [70] and then apply the edge detection in the color domain. The appearance and motion edge maps are then combined by the maximum

operation. Overall, the spatial and temporal pairwise potentials between neighboring superpixels are computed as

$$\begin{aligned}\phi_s(s_i^j, s_p^q) =&(1 - e(s_i^j, s_p^q)) \times (1 - \delta(x_i^j, x_p^q))\times \\ &\exp(-\beta_s^{-1}\|\mathbf{F_i^j} - \mathbf{F_p^q}\|^2) \\ \phi_t(s_i^j, s_p^q) =&c(s_i^j, s_p^q) \times (1 - \delta(x_i^j, x_p^q))\times \\ &\exp(-\beta_t^{-1}\|\mathbf{H_i^j} - \mathbf{H_p^q}\|^2).\end{aligned} \tag{4.5}$$

Here, $e(s_i^j, s_p^q)$ denotes the average edge strength between superpixel s_i^j and s_p^q, $c(s_i^j, s_p^q)$ denotes the percentage of pixels in s_p^q that are linked to s_i^j by optical flow, and δ is the standard Kronecker delta function, *i.e.*, $\delta(u, v) = 1$ if $u = v$ and $\delta(u, v) = 0$ if $u \neq v$. $\mathbf{F_i^j}$ is the concatenation of color and optical flow orientation histogram and $\mathbf{H_i^j}$ is the color histogram. The motion feature is only included in the spatial pairwise potentials because temporal pairs correspond to superpixels in different frames. The overall pairwise potential is then computed as the weighted summation of all the spatial and temporal pairwise terms:

$$\begin{aligned}\Phi_p(\mathbf{s}, \mathbf{x}) =&\alpha_s \times \sum_{\{s_i^j, s_p^q\}\in\mathcal{N}_s} \phi_s(s_i^j, s_p^q)+ \\ &\alpha_t \times \sum_{\{s_i^j, s_p^q\}\in\mathcal{N}_t} \phi_t(s_i^j, s_p^q)\end{aligned} \tag{4.6}$$

where $\mathcal{N}_s$ and $\mathcal{N}_t$ denote the collections of all the spatial and temporal neighborhood pairs, respectively. α_s and α_t are two weight parameters for linear combination.

4.2.3 Appearance Auxiliary Potential

In general, the appearance constraint $\Phi_a(\mathbf{s}, \mathbf{x}, \mathbf{\Theta})$ in Eq.(4.2) can be written as $\Phi_a(\mathbf{s}, \mathbf{x}, \mathbf{\Theta}) = f(\mathbf{s}, \mathbf{x}, g(\mathbf{s}, \mathbf{x}))$ where f measures how consistent the current labeling $\mathbf{x}$ is with the appearance model, and g computes the appearance model parameters given the current labeling $\mathbf{x}$. However it is impossible to have an analytical expression to $\Phi_a(\mathbf{s}, \mathbf{x}, \mathbf{\Theta})$ for many popular appearance models because the appearance model training usually involves complicate optimization process, *e.g.*, EM optimization in GMM, so for such methods Eq.(4.1) cannot be solved analytically. Hence, an alternative optimization scheme is usually employed to solve Eq.(4.1), *i.e.*, fix the appearance model while solving $\mathbf{x}$ and fix $\mathbf{x}$ while optimizing the appearance model. Inspired by [114], in this work we propose an appearance model for video object segmentation in which $\Phi_a(\mathbf{s}, \mathbf{x}, \mathbf{\Theta})$ can be expressed analytically in terms of $\mathbf{x}$, and Eq.(4.1) can be solved efficiently by one graph cut. In the following, we first review the method of [114] on static image segmentation and then discuss the challenges in adapting the idea to videos and how we overcome them.

The method in [114] models each pixel as a node and represents each node as a single bin in the RGB histogram space for appearance modeling. Let p_i and x_i denote the i^{th} pixel and its label, respectively, b_i denote the assigned bin of pixel p_i, H denote the dimensionality of the histogram space and P denote the total number of pixels. Furthermore we use Ω_F^k and Ω_B^k to denote the number of pixels assigned to the k^{th} bin in the thematic object and background regions, respectively, and Ω^k to denote the number of pixels assigned to the k^{th} bin in the entire image, *i.e.*, $\Omega_F^k = |\{p_i | x_i = 1\}|$, $\Omega_B^k = |\{p_i | x_i = 0\}|$ and $\Omega^k = \Omega_F^k + \Omega_B^k$ where $|.|$ denotes the Cardinality of a set. Then the probability of the k^{th} histogram bin being thematic and

background is given by $p(F|k) = \frac{\Omega_F^k}{\Omega^k}$ and $p(B|k) = \frac{\Omega_B^k}{\Omega^k}$, respectively. Finally, the appearance constraint potential of each pixel p_i can be computed as

$$\phi_a(p_i) = \begin{cases} -\ln p(F|b_i) & \text{if } x_i = 1 \\ -\ln p(B|b_i) & \text{if } x_i = 0 \end{cases}. \tag{4.7}$$

Then the total appearance constraint potential of all the pixels, *i.e.*, the last term in Eq.(4.2), can be computed as

$$\begin{aligned} \Phi_a &= \sum_{i=1}^{P} \phi_a(p_i) \\ &= \sum_{i=1}^{P} -\delta(x_i, 1) \times \ln p(F|b_i) - \delta(x_i, 0) \times \ln p(B|b_i) \\ &= -\sum_{i=1}^{P} (\delta(x_i, 1) \times \ln \frac{\Omega_F^{b_i}}{\Omega^{b_i}} + \delta(x_i, 0) \times \ln \frac{\Omega_B^{b_i}}{\Omega^{b_i}}) \\ &= -(\sum_{k=1}^{H} \Omega_F^k \times \ln \frac{\Omega_F^k}{\Omega^k} + \sum_{k=1}^{H} \Omega_B^k \times \ln \frac{\Omega_B^k}{\Omega^k}) \\ &= -\sum_{k=1}^{H} (\Omega_F^k \times \ln \frac{\Omega_F^k}{\Omega^k} + \Omega_B^k \times \ln \frac{\Omega_B^k}{\Omega^k}). \end{aligned} \tag{4.8}$$

The inner part of the summation in Eq.(4.8) can be approximated by $\left|\Omega_F^k - \Omega_B^k\right|$ since $\Omega_F^k + \Omega_B^k = \Omega^k$. Hence, $\Phi_a(\mathbf{x}, \mathbf{\Theta}) \approx -\sum_{k=1}^{H} \left|\Omega_F^k - \Omega_B^k\right| = \sum_{k=1}^{H} 2\min(\Omega_F^k, \Omega_F^k) - \Omega^k$. As we are only interested in minimizing $\Phi_a(\mathbf{x}, \mathbf{\Theta})$ instead of its absolute value, we can drop the constant term Ω^k and the multiplier 2. Eventually, the appearance model is reduced to

$$\Phi_a(\mathbf{x}, \mathbf{\Theta}) = \sum_{k=1}^{H} \min(\Omega_F^k, \Omega_B^k), \tag{4.9}$$

and the inner part of this summation is the number of pixels that are assigned to the k^{th} bin taking the minority label. Interestingly, this appearance term turns out to be equivalent to adding some auxiliary nodes and edges to

the MRF graph. The addition procedure is simple: 1) add H auxiliary nodes in which each node corresponds to a bin of the histogram, and the unary potential of these newly added nodes are set to $-\log(0.5)$; 2) Connect each pixel to the auxiliary node that corresponds to its assigned bin. The rationality of this equivalence is that the auxiliary nodes are guaranteed to be labeled as the majority label of its connected pixels when the graph energy is minimized and, hence, the cost incurred by each auxiliary node is equal to the number of connected pixels taking the minority label.

A naive extension of [114] to our superpixel based video object segmentation is to take the mean RGB color of each superpixel and assign it to one of the bins in the color histogram space. However raw color features alone may not be robust enough to accurately capture the viewpoint and lighting variations between frames. Hence, we propose to use more advanced features, *i.e.*, SIFT and Texton, to measure the similarity between image regions. The fusion of these features has shown promising results in many vision problems such as [116, 142, 160]. It is worth noting that these features are not computed only on the pixels within a superpixel, but rather are computed in a larger window around a pixel. For example, all SIFT features computed on all videos in our experiment cover more than a single superpixel. This means that important context information around the superpixels is contained in these features. However, for video object segmentation, both the original appearance modeling approach in [114] and its naive extension are not readily applicable to these multi-feature situations. This is because both SIFT and Texton are key point based features and are not confined to any arbitrarily shaped superpixel. Hence, we adopt a different approach to extract and fuse these features. We first extract these features around a set of key points defined by a dense grid, *e.g.*, sample a key point every 4 pixels horizontally and vertically. We then use the bag of words approach to

quantize each type of feature to a particular bin and assign each key point to a single bin by taking the Cartesian Product of the different types of features. However, unlike the case of single pixels, each node will now contain more than a single feature point, and the original approach in [114] cannot handle this situation. Hence, we propose a variation of the original technique to handle the cases where each node is described by a set of bins, *i.e.*, a full histogram, instead of a single bin in the histogram. In the following, we will introduce this new method and prove that it can also be equated by adding auxiliary nodes and edges.

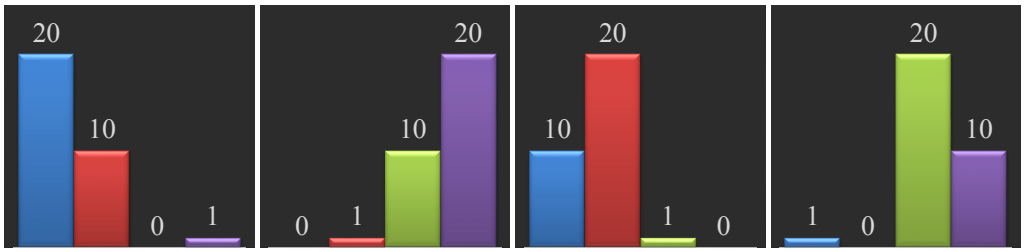
For consistency, we first redefine some of the terms used in the description of the pixel wise approach in [114]. Let $b_i^{j,k}$ denote the number of votes in the k^{th} bin of superpixel s_i^j's histogram, *i.e.*, the number of feature points in superpixel s_i^j that are assigned to the k^{th} bin, H denote the total number of bins in the histogram feature space, Ω_F^k and Ω_B^k denote the total number of votes in the k^{th} bin from the superpixels in the thematic object or background regions, respectively and Ω^k denote the total number of votes in the k^{th} bin in all the superpixels, *i.e.*, $\Omega_F^k = \sum_i^N \sum_j^{M_i} \delta(x_i^j, 1) b_i^{j,k}$, $\Omega_B^k = \sum_i^N \sum_j^{M_i} \delta(x_i^j, 0) b_i^{j,k}$ and $\Omega^k = \Omega_F^k + \Omega_B^k$. We can then compute the probability of the k^{th} bin being thematic and background as $p(F|k) = \frac{\Omega_F^k}{\Omega^k}$ and $p(B|k) = \frac{\Omega_B^k}{\Omega^k}$, respectively. With the Naive Bayes assumption on the feature points in a superpixel, we can compute the probability of superpixel s_i^j being thematic and background as $p(F|s_i^j) = \prod_{k=1}^H p(F|k)^{b_i^{j,k}}$ and $p(B|s_i^j) = \prod_{k=1}^H p(B|k)^{b_i^{j,k}}$, respectively. Then the last term in Eq.(4.2) is computed as

$$\Phi_a(\mathbf{s}, \mathbf{x}, \mathbf{\Theta}) = \sum_i^N \sum_j^{M_i} \phi_a(s_i^j), \tag{4.10}$$

where

$$\phi_a(s_i^j) = \begin{cases} -\ln p(F|s_i^j) & \text{if } x_i^j = 1 \\ -\ln p(B|s_i^j) & \text{if } x_i^j = 0 \end{cases} = \begin{cases} -\sum_{k=1}^{H} b_i^{j,k} \times \ln p(F|k) & \text{if } x_i^j = 1 \\ -\sum_{k=1}^{H} b_i^{j,k} \times \ln p(B|k) & \text{if } x_i^j = 0 \end{cases}. \quad (4.11)$$

An example is shown in Fig. 4.3 to illustrate how this term can enforce the appearance constraints. From this figure, it can be seen that minimizing the appearance term encourages the appearance coherence and disparity within and between the thematic object and background regions, respectively. Note that, the minimum value of the appearance term is achieved when all the nodes are labeled as thematic or background, *i.e.*, $[x_A, x_B, x_c, x_D] = [0, 0, 0, 0]$ or $[1, 1, 1, 1]$. However, this rarely occurs in practice because this will cause a very high unary potential, while the overall objective is to minimize the summation of all the three potential terms. The second best labeling in Fig. 4.3, *i.e.*, $[0, 1, 0, 1]$ (or $[1, 0, 1, 0]$), implies that the first two bins mainly correspond to the background (or thematic object) regions, and the last two bins mainly correspond to the thematic (or background) regions.



$[x_A, x_B, x_C, x_D]$	$\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$
[0, 0, 0, 0]	0.00
[0, 0, 0, 1]	20.56
[0, 0, 1, 0]	20.56

$[x_A, x_B, x_C, x_D]$	$\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$
[0, 0, 1, 1]	79.31
[0, 1 0, 0]	20.56
[0, 1, 0, 1]	17.67

$[x_A, x_B, x_C, x_D]$	$\Phi_a(\mathbf{s}, \mathbf{x}, \Theta)$
[0, 1, 1, 0]	79.31
[0, 1 1, 1]	20.56

Fig. 4.3 An example illustrating how the potential term defined in Eq.(4.10) and Eq.(4.11) enforces the appearance constraints. The bar plots show the histogram features of four example superpixel nodes, and the tables show the cost incurred by labeling the four nodes differently.

By substituting Eq.(4.11) into Eq.(4.10) we have

$$
\begin{aligned}
&\Phi_a(\mathbf{s}, \mathbf{x}, \boldsymbol{\Theta}) \\
&= -\sum_{i}^{N}\sum_{j}^{M_i}\sum_{k=1}^{H} \delta(x_i^j, 1) \times b_i^{j,k} \times \ln p(F|k) + \\
&\delta(x_i^j, 0) \times b_i^{j,k} \times \ln p(B|k) \\
&= -\sum_{k=1}^{H}\sum_{i}^{N}\sum_{j}^{M_i} \delta(x_i^j, 1) \times b_i^{j,k} \times \ln p(F|k) + \\
&\delta(x_i^j, 0) \times b_i^{j,k} \times \ln p(B|k) \\
&= -\sum_{k=1}^{H} [\ln p(F|k) \times \sum_{i}^{N}\sum_{j}^{M_i} \delta(x_i^j, 1) \times b_i^{j,k} + \\
&\ln p(B|k) \times \sum_{i}^{N}\sum_{j}^{M_i} \delta(x_i^j, 0) \times b_i^{j,k}] \\
&= -\sum_{k=1}^{H} \left(\Omega_F^k \times \ln \frac{\Omega_F^k}{\Omega_k} + \Omega_B^k \times \ln \frac{\Omega_B^k}{\Omega_k} \right). \qquad (4.12)
\end{aligned}
$$

It can be seen that we arrive at a similar conclusion as Eq.(4.8), and the new appearance term can also be equated by adding auxiliary nodes and edges to the original MRF structure. The difference is that we now add edges

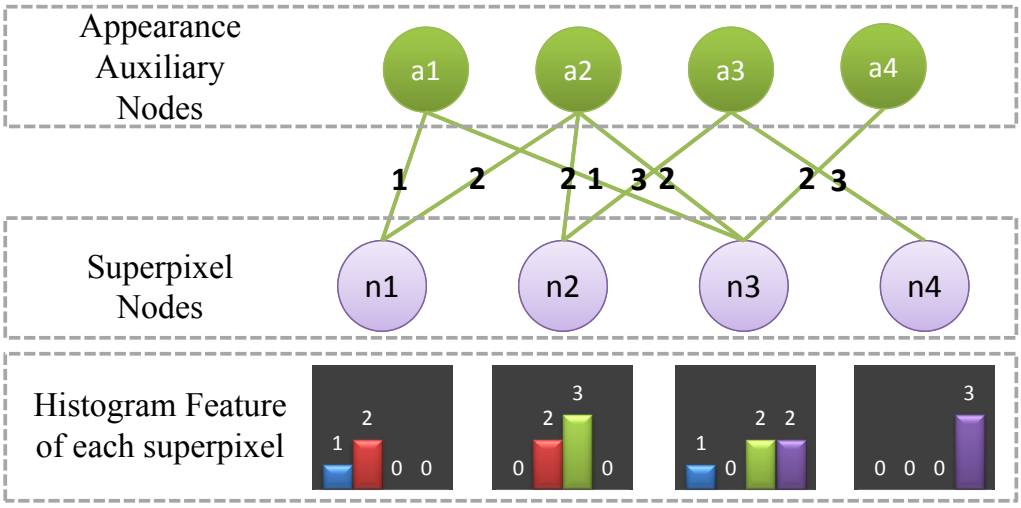


Fig. 4.4 A toy example illustrating how the appearance auxiliary nodes are connected to the superpixel nodes. In this example, there are only four superpixel nodes indicated by the purple discs. Each superpixel node is described by a 4-bin histogram which corresponds to the four green discs on the top. The numbers on the green edges indicate the weights of the auxiliary connections between the superpixel nodes and auxiliary nodes. Note that the edges between the superpixel nodes are omitted for simplicity.

to connect every pair of superpixel and appearance auxiliary node, and the edge weight is set to the corresponding bin's vote. For example, the weight of the auxiliary edge connecting superpixel node s_i^j and the k^{th} auxiliary node is the number of feature points in s_i^j that are assigned to the k^{th} bin. This process is illustrated in Fig. 4.4. Compared with the original pixel-wise approach, the proposed method is applicable to more complicated features besides color and can handle the cases where each node is described by a full histogram instead of a single bin.

A potential concern of the proposed framework is that the dimensionality of the histogram feature, *i.e.*, the number of auxiliary nodes need to be added, is extremely large due to the effect of Cartesian Product. For example, if we use 64 bins for each RGB channel, 100 words for both the dense SIFT and Texton bag of words features, there will be $64^3 \times 100 \times 100 \approx 2.6 \times 10^9$ bins in total. However, in practice, a superpixel node will be connected to

an appearance auxiliary node only if the corresponding bin is not empty and an appearance auxiliary node will be added to the graph only when it is connected to at least two different superpixels. Hence, the actual number of auxiliary nodes and connections added to the graph is much less than the theoretical upper bound due to the sparsity of the histograms. For example, in a 98 frame video sequence, there are 221,559 superpixel nodes, 142,384 appearance auxiliary nodes and 1,105,807 connections between them. To show that the auxiliary connections are meaningfully distributed among the nodes, the statistics on the number of auxiliary connections linked to each superpixel and auxiliary nodes are shown in Fig. 4.5 for the 98 frame video sequence. It can be seen that the auxiliary connections distribute stably among the superpixel nodes while highly unbalanced among the auxiliary nodes, *e.g.*, the most connected auxiliary node has around 21,570 connections while the least connected auxiliary node has only 2 connections. However, the most connected auxiliary node is far from dominating the auxiliary connections as it only contributes around 2% of the entire auxiliary connections.

4.2.4 Optimization

We use the max flow algorithm proposed in [12] to solve for the optimal labels. With the benefit of the proposed appearance modeling technique, the optimization is a single round process and it only takes seconds to optimize a video with hundreds of frames. As also shown in the experiment, the addition of the auxiliary nodes and edges only introduces negligible extra computation cost.

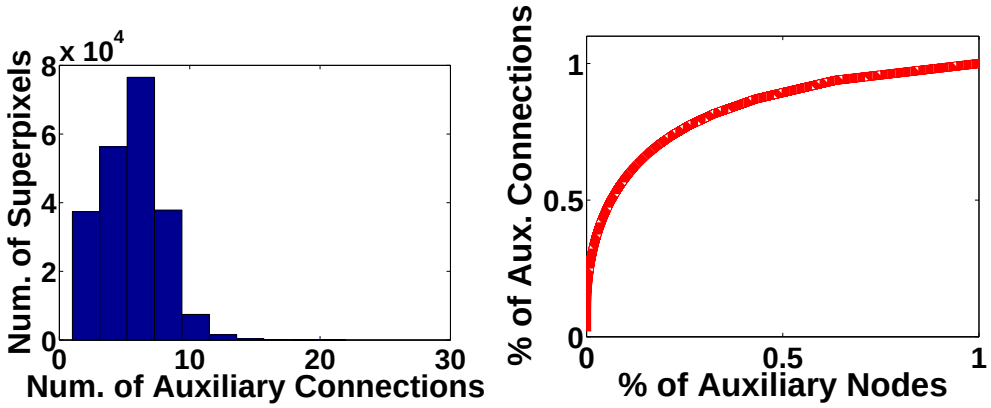


Fig. 4.5 The statistics on the number of auxiliary connections linked to each superpixel node (left) and auxiliary node (right) on the bird_of_paradise sequence in SegTrack v2. The left plot shows the histogram of the number of auxiliary connections linked to each superpixel node. On the right plot, the horizontal axis is the percentage of auxiliary nodes and the vertical axis is the percentage of the total amount of auxiliary connections, *e.g.*, point (0.3, 0.8) means 30% of the auxiliary nodes with the highest connectivity contribute 80% of the auxiliary connections. Note that we choose to not simply plot the histogram on the number of auxiliary connections linked to each auxiliary node because the distribution is highly unbalanced.

4.3 Experiment

4.3.1 Dataset and Experimental Setup

In order to evaluate the effectiveness of the proposed appearance modelling technique, we run experiments on several benchmark datasets including the SegTrack v2[1] and 10-video-clip dataset[2] [31]. The videos in these two datasets are quite challenging. Many of the videos contain cluttered background and dynamic scenes due to camera motion or moving background objects. Some videos even contain fast motions such as the girl, monkey and monkeydog sequences in the SegTrack v2 dataset and the VWC102T, DO02_001 and DO01_055 sequences in ten video clip dataset. Some videos also contain cluttered background motions such as the swaying tree leaves and grass in the

[1]http://www.cc.gatech.edu/ fli/SegTrack2/dataset.html
[2]http://www.brl.ntt.co.jp/people/akisato/saliency3.html

BR128T, BR130T and DO01_030 sequences in the ten video clip dataset. In some videos, the thematic objects are visually very similar to the background, *i.e.*, low contrast along object boundaries, such as the birdfall, frog and worm sequences in the SegTrack v2 dataset. We evaluate the proposed approach against several state-of-the-art methods [96, 151, 65, 64, 47]. We also compare with several baseline methods in order to separate the contributions of the different components. Pixel-wise Jaccard similarity coefficient, *i.e.*, intersection over union ratio, is used to evaluate the segmentation accuracy of each video.

The major parameters involved in the proposed method are the weights associated with each potential term in Eq.(4.2) and Eq.(4.6). In the experiment, we empirically set $\alpha_p\alpha_s = 240$, $\alpha_p\alpha_t = 160$, and $\alpha_a = 18$, and these parameters are kept fixed throughout all the experiments and videos unless otherwise specified. The β_s and β_t in Eq.(4.5) are set to the double average of the L2 feature distance between all the spatial and temporal pairs in a particular video, respectively, *i.e.*, $\beta_s = 2\langle\|\mathbf{F_i^j} - \mathbf{F_p^q}\|^2\rangle$ and $\beta_t = 2\langle\|\mathbf{H_i^j} - \mathbf{H_p^q}\|^2\rangle$ where $\langle.\rangle$ denotes averaging over all pairs. In the appearance modeling, we use 64 bins for each color channel and 100 words for both the dense SIFT and Texton histograms.

4.3.2 Experimental Results

Table 4.1 Comparison results on SegTrack v2 dataset

video	video dimension	ours	ours w/o App.	[96]	[151]	[64]	[65]	[47]
bird_of_paradise	$640 \times 360 \times 98$	**94.49%**	76.12%	94.43%	-	92.20%	94.00%	93.92%
birdfall2	$259 \times 327 \times 30$	66.23%	54.46%	57.78%	71.00%	49.00%	62.50%	**73.29%**
frog	$480 \times 264 \times 279$	80.68%	47.16%	69.34%	74.00%	0.00%	65.80%	**81.58%**
girl	$400 \times 320 \times 21$	81.63%	68.40%	74.94%	82.00%	87.70%	**89.20%**	86.75%
monkey	$480 \times 270 \times 31$	68.51%	27.21%	64.02%	62.00%	79.00%	**84.80%**	63.96%
monkeydog	$320 \times 240 \times 71$	**78.71%**	60.99%	78.19%	75.00%	-	58.80%	76.12%
parachute	$414 \times 352 \times 51$	89.91%	60.36%	91.46%	94.00%	**96.30%**	93.40%	94.68%
soldier	$528 \times 224 \times 32$	83.44%	64.78%	69.89%	60.00%	66.60%	**83.80%**	36.84%
worm	$480 \times 364 \times 243$	81.57%	75.11%	74.19%	60.00%	**84.4%**	82.80%	61.79%
average	-	**80.57%**	59.40%	74.92%	72.25%	69.40%	79.46%	74.33%
runtime (seconds per frame)	-	6.84s	6.81s	14.90s	>82s	>82s	>82s	-

The video dimension is in the format of width×height×frame number.

Table 4.2 Comparison results on ten-video-clip dataset

video	video dimension	ours	ours w/o App.	[96]	[151]	[47]
AN119T	352 × 288 × 100	**95.68%**	94.99%	94.50%	91.48%	93.53%
BR128T	352 × 288 × 118	**70.74%**	32.66%	34.82%	33.88%	8.39%
BR130T	352 × 288 × 84	**80.27%**	57.44%	29.17%	66.42%	78.83%
DO01_013	352 × 288 × 89	**93.84%**	79.74%	91.80%	91.66%	38.67%
DO01_014	352 × 288 × 101	93.62%	82.33%	**94.54%**	89.58%	92.41%
DO01_030	352 × 288 × 101	55.59%	18.24%	**77.91%**	43.92%	57.28%
DO01_055	352 × 288 × 63	52.53%	51.33%	68.40%	49.05%	**77.68%**
DO02_001	352 × 288 × 83	**93.22%**	39.15%	78.74%	77.29%	59.76%
M07058	352 × 288 × 72	81.16%	**82.95%**	77.71%	74.72%	73.18%
VWC102T	352 × 288 × 107	**83.72%**	78.09%	83.70%	82.57%	77.80%
average	-	**80.04%**	61.69%	73.13%	70.06%	65.75%

The video dimension is in the format of width×height×frame number.

The comparison results with some state-of-the-art methods for both datasets are shown in Table 4.1 and 4.2. Some qualitative comparisons are also shown in Fig. 4.6 and 4.7. From the numerical comparisons, it can be seen that the proposed method is not only faster but also more accurate than the existing state-of-the-art approaches for both datasets. The efficiency of the proposed method is because of its simplicity, *i.e.*, one graph cut on a sparsely connected graph in which the unary, pairwise and appearance potentials can be computed efficiently. The importance of appearance modeling is also revealed by comparing to our baseline approach without appearance constraint (the columns under "ours w/o App." in Table 4.1 and 4.2). From the qualitative examples in Fig. 4.6, it can be seen that our initial saliency estimation is usually noisy and can only highlight the rough location of the thematic object without detailed shape and boundary. As a consequence, our baseline approach without appearance constraint can only improve the segmentation performance by smoothing around the local edges. It is not able to correct those large regions corrupted by saliency. Moreover, the two examples shown in Fig. 4.7 imply that our method can handle the cases where the thematic object is absent in some frames. The method in [96] applies appearance constraint by training color GMMs in the local frames iteratively. It has shown better performance over our baseline approach but still fails when there is color overlap between thematic object and background or the saliency estimation is consistently corrupted in a sequence of frames. Compared with [96], our appearance model is a global model across all the frames and employs more powerful features besides color. It consistently outperforms [96] in the shown examples. Furthermore, the addition of the appearance constraint only introduces negligible extra computation cost due to its efficiency.

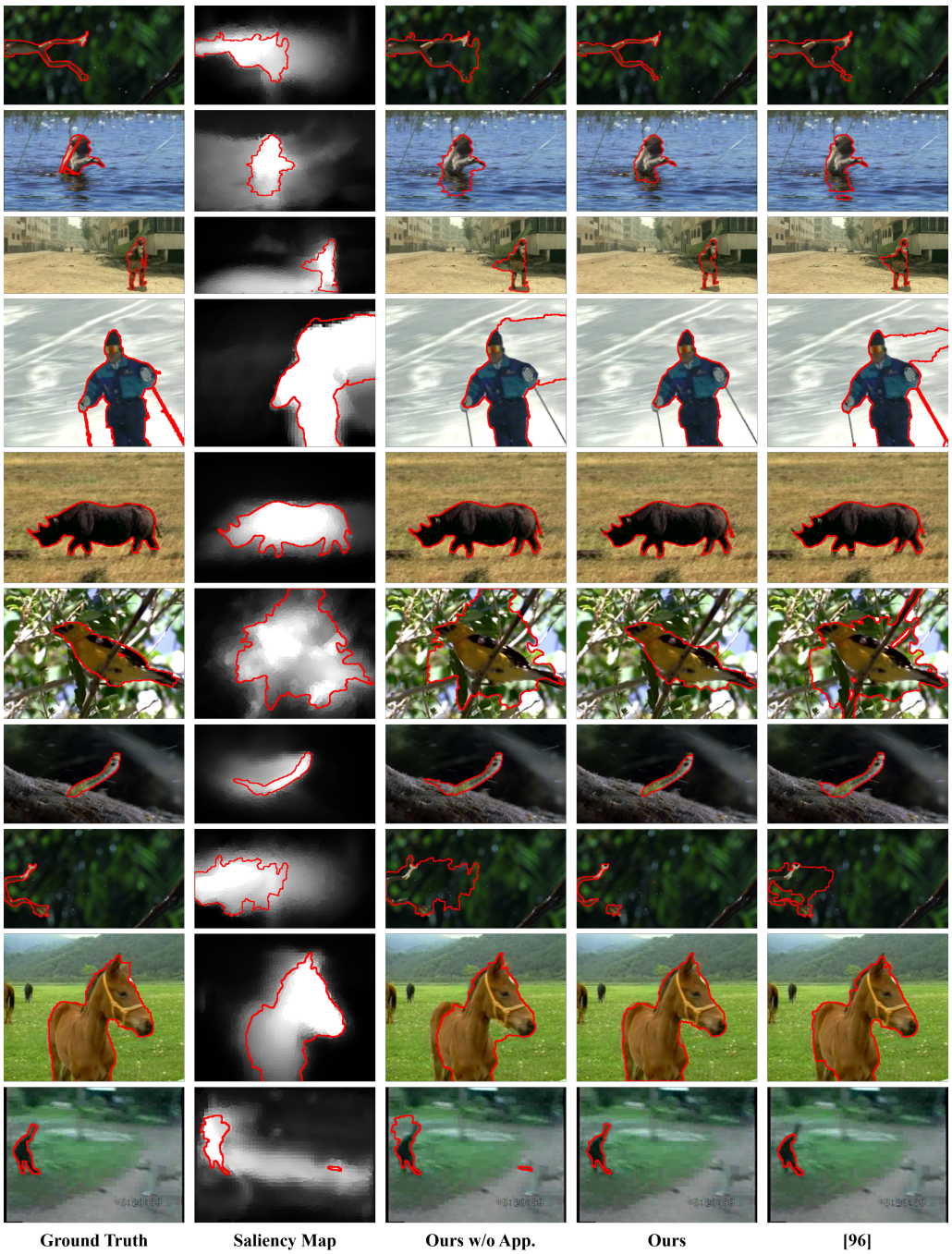


Fig. 4.6 Some qualitative results and comparisons.

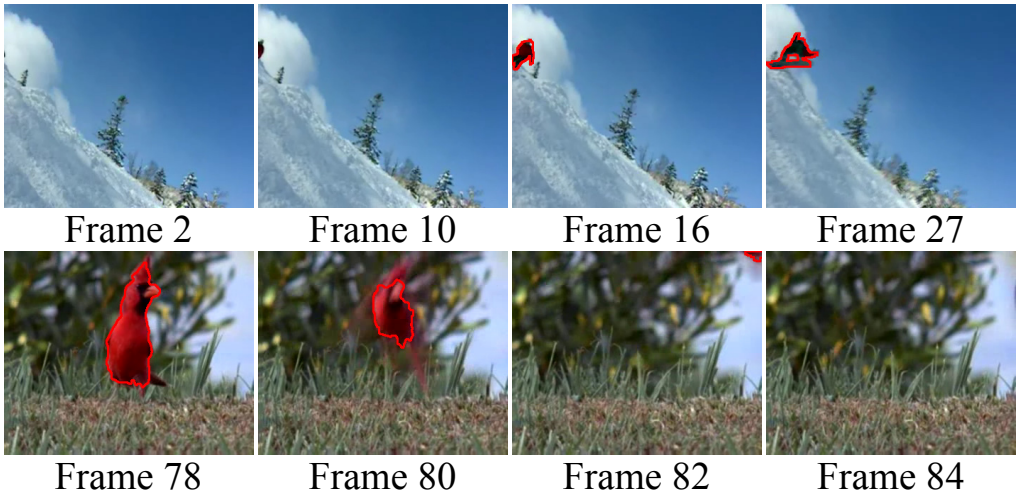


Fig. 4.7 Two examples in which the object is absent in the beginning (top example) or end (bottom example).

Besides comparing with the state of the art, we also compare with several baseline methods in order to show the importance of the various components of the proposed method. The compared baseline methods are:

1. Segmentation by unary potential. In this approach, we exclude the pairwise and appearance terms. It directly measures the quality of the initial saliency estimation.

2. Naive extension of [114] (1). This approach applies the image based pixel-wise segmentation method proposed in [114] to each individual frame. In this method, we compute the unary potential as in Section 4.2.1, formulate the spatial pairwise potential based on the description in [105] and add the appearance constraint following [114]. The weights on the pairwise and appearance terms are set to 4 by grid search to accommodate the changes of the potential definitions.

3. Naive extension of [114] (2). In this approach, we use the average RGB value of each superpixel to describe each node. It directly applies the technique proposed in [114] since each node only corresponds to one

bin in the color histogram space. The weight of the appearance term is reset to 7 by grid search to accommodate this change.

4. Our method without SIFT/Texton features, *i.e.*, ours with only color features. This baseline approach removes the dense SIFT and Texton features in the appearance modeling process. The difference to baseline (3) is that we still extract color features from sampled key points instead of computing the average. The weight on the appearance term is set to 2.7 by grid search to accommodate this change.

The comparison results in terms of the average Jaccard similarity coefficient for all the videos are shown in Fig. 4.8(a). Note that we have also used HSV color space in place of RGB in the settings of baseline (3), (4) and the proposed full method, and the weights on the appearance terms are re-tuned for a fair comparison. The paired t-tests with a significance level of 0.05 have also been conducted to show the statistical meaningfulness of these comparisons. The p-values of these tests are shown in Fig. 4.8(b). From Fig. 4.8(b), it can be seen that the comparisons between baseline 1 and all the other methods and the comparisons between the proposed methods and all the baseline methods are statistically meaningful. The comparisons among all the rest baseline methods are not statistically meaningful. From the poor performance of baseline (1), it can be seen that the initial saliency estimation is far from a good segmentation. The comparison with baseline (4) shows the benefits of adding the dense SIFT and Texton feature by Cartesian Product. The comparisons with baseline (2) and (3) show that it is not trivial to extend the method proposed in [114] to videos and validate the necessity of our superpixel based approach with rich features.

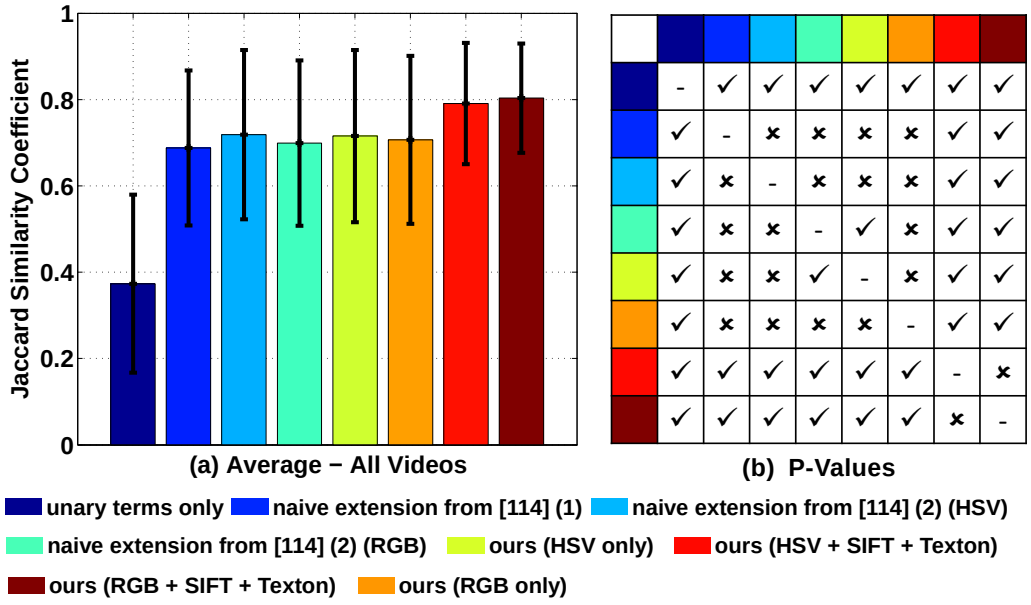


Fig. 4.8 Comparisons with several baseline methods. (a) shows the average segmentation accuracy of the proposed methods and several baseline settings. (b) shows the p-values of the paired t-tests conducted on the pairwise comparisons among the 8 methods listed in (a). A tick symbol means the p-value is smaller than 0.05 and a cross symbol means the p-value is greater than 0.05. It can be seen that, the comparisons between the proposed methods (last two bars in (a)) and the baseline methods (first 6 bars in (a)) are statistically meaningful.

4.3.3 Parameter Analysis

The major parameters involved in this framework are the three weights associated with the unary term, pairwise term and appearance term, respectively. Since the unary and pairwise terms have been explored in most of the MRF segmentation formulations, we evaluate the weight on the newly proposed appearance term in this section. In order to do this, we conduct an experiment to compare the segmentation accuracy by varying this weight and the results for both datasets are shown in Fig. 4.9(a) and (b), respectively. It can be seen that, although each video sequence has its own preferred optimal weight, their trends are roughly consistent, *i.e.*, segmentation accuracy improves rapidly with increasing weights at the beginning, gradually saturates around 50 to 100 and some videos start to drop after 100. This implies that, within a wide range, the framework is not very sensitive to the weight of this newly proposed appearance term. We have also compared the segmentation accuracy between using a universal weight for all the videos as described in Section 4.3.1 and individually selecting the best weight for each video. The result is shown in Fig. 4.9(c) and it can be seen that tunning the weight for each individual video can produce more accurate segmentation. However, the improvement is not very significant due to the stableness of the proposed technique on different videos, and a universal weight setting is generally more meaningful in practice.

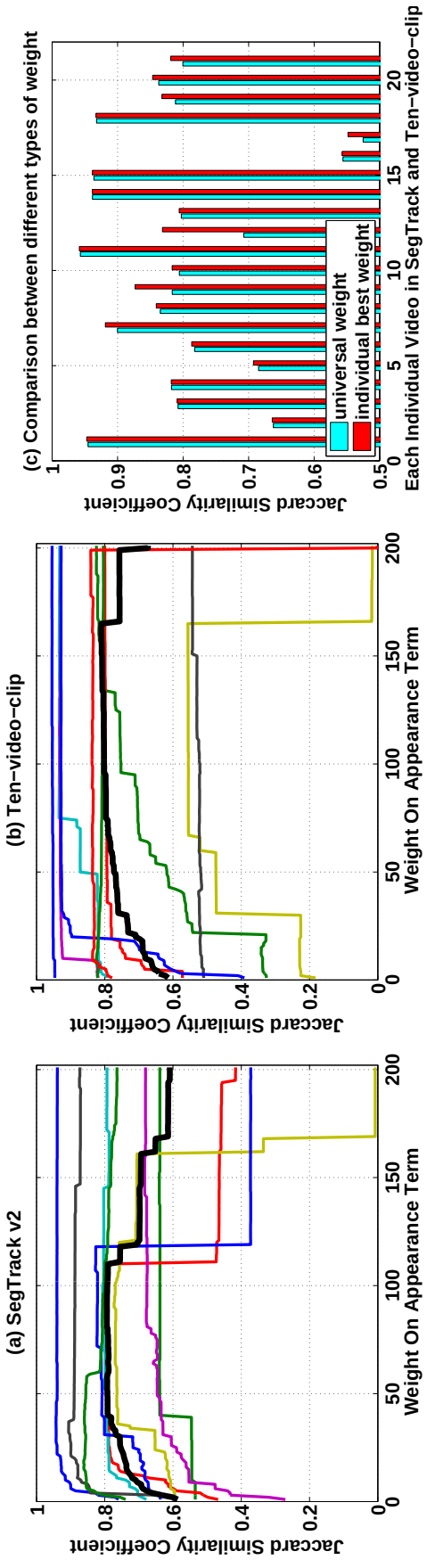


Fig. 4.9 Evaluation results regarding the weight of the appearance term. The first two curves show the segmentation accuracy of the SegTrack v2 and ten-video-clip dataset, respectively, by varying the weight from 0 to 200. The colorful thin lines indicate each individual video and the black thick lines indicate the average of each dataset. The rightmost bar plot shows the comparison of segmentation accuracy between using a universal weight for all the videos as described in Section 4.3.1 and individually selecting the best weight for each video according to the first two curves. In the bar plot, horizontal label 1-9 indicate the 9 videos in the SegTrack v2 dataset, 10 indicates the average of SegTrack v2 dataset, 11-20 indicate the 10 videos in the ten-video-clip dataset and 21 indicates the average of the ten-video-clip dataset. Note that the vertical axis of the bar plot starts from 0.5 instead of 0.

4.3.4 Error Analysis

Despite the good performance of the proposed approach, segmentation errors are always inevitable and some typical examples are shown in Fig. 4.10. The most common error is the inclusion of background or exclusion of thematic object regions along low contrast object boundaries, such as the left leg of the frog in the first column of Fig. 4.10, the right arm of the monkey in the second column of Fig. 4.10 and the reflection of the monkey on the water surface in the third column of Fig. 4.10. This is the built-in difficulty of thematic object segmentation as we do not have prior knowledge of the object of interest and it is challenging to generate an accurate boundary in these low contrast regions. The second type of error is the inclusion of background regions in the gap between the object parts such as the grass between the two legs of the monkey in the second column of Fig. 4.10. These regions are labeled as part of the thematic object because they are blurred with high saliency value by the saliency warping/smoothing process along imperfect optical flows. The third type of error is the loss of thin structures attached to the main body of the object such as the legs of the bird in the last column of Fig. 4.10. These thin parts are either missed by the initial saliency estimation or smoothed away by the MRF smoothing. A common solution in the static image segmentation literature is to employ higher order potentials, such as Robust P(n) [61], to enforce the high-level structure of the object, and we leave this as our future work. In addition, the segmentation error in the fourth column of Fig. 4.10 is caused by severely corrupted saliency estimations. The saliency consistently fails to highlight the lower part of the flower due to the cluttered motion background, *e.g.*, both the flower and the leaves are swaying in the wind. Moreover, there happens to be a strong edge

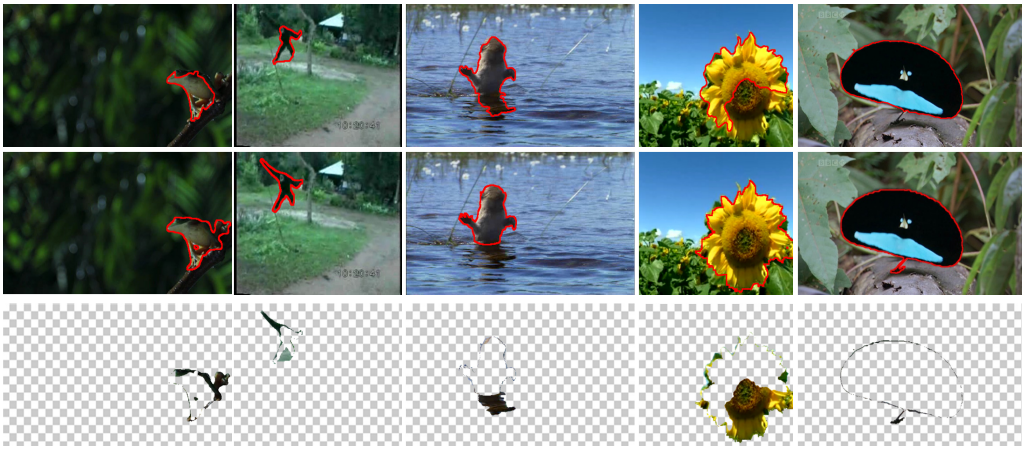


Fig. 4.10 Some typical segmentation errors. The first row is the segmentation result, the second row is the ground truth segmentation and the last row is the segmentation errors.

between the heart and the upper part of the flower and the MRF smoothing fails to prevent the separation.

4.3.5 Limitations

In addition to the common segmentation errors discussed in Section 4.3.4, another limitation of our method is that each node must be described by a histogram feature. Although histogram feature is one of the most commonly used types of feature for visual description, non-histogram features, such as CNN feature, are also useful in many vision tasks. We plan to further generalize the proposed method to arbitrary types of feature in the future work. Another limitation of our method is that, we process the video in batch mode, *i.e.*, the whole video is segmented at once regardless of its length. This may cause memory and computational issues when the video is long. To overcome this limitation, we plan to introduce a progressive version of the proposed segmentation method to process long videos part by part.

4.3.6 Computation Speed

As shown in Table 4.1, our method is efficient compared with the other approaches and the detailed time usage of the various components is shown in Table 4.3. All the experiments are conducted on a Dual-Core i5 PC with 8GB of RAM, and the time statistics in Table 4.3 are based on the bird_of_paradise sequence in SegTrack v2 because it has the highest per-frame resolution. For the AMC [111][3] and GBMR [138][4] image saliency detection, SLIC [2][5] superpixel segmentation and structured forests edge detection [23][6], we use the code provided by the authors. For optical flow computation, Texton feature extraction and MRF inference, we use the code provided with [70][7], [116][8] and [12][9], respectively. For SIFT feature extraction, we use the VLFeat implementation[10]. All the other components are implemented by ourselves in Matlab. The detailed parameter settings of the various components can be found in Section 4.3.1. Due to the good architecture of our method, we are able to parallelize many of the components. For example, we could run the two image saliency detections, optical flow computation, SLIC superpixel segmentation, and SIFT/Texton feature extraction concurrently in multiple threads since they do not depend on each other. Similarly, we can also compute the two motion saliency maps concurrently in two threads after obtaining optical flows. In Table 4.3, we highlight the components that can run concurrently using the same color. Overall, we can achieve 6.84 seconds[11] per frame with these two parallelization schemes.

[3] http://202.118.75.4/lu/Project/saliency_MC_iccv13/absorb_MC.html
[4] http://faculty.ucmerced.edu/mhyang/project/cvpr13_saliency/cvprsaliency.htm
[5] http://ivrl.epfl.ch/research/superpixels
[6] https://github.com/pdollar/edges
[7] https://people.csail.mit.edu/celiu/OpticalFlow/
[8] http://www.cs.unc.edu/ jtighe/Papers/ECCV10/
[9] http://pub.ist.ac.at/ vnk/software.html
[10] http://www.vlfeat.org/overview/dsift.html
[11] $6.84 = \max\{0.22, 1.25, 4.82, 0.80, 1.15, 0.15\} + \max\{1.08, 0.47\} + 0.34 + 0.58 + 0.01 = 4.82 + 1.08 + 0.34 + 0.58 + 0.01$. These numbers correspond to the entries in Table 4.3.

Table 4.3 Time usage of the various components

Components	Runtime (seconds per frame)
amc saliency	0.22
gbmr saliency	1.25
optical flow	4.82
superpixel segmentation	0.15
SIFT feature extraction	0.80
Texton feature extraction	1.15
gc saliency	1.08
w saliency	0.47
saliency fusion	0.34
graph construction	0.58
MRF inference	0.01
total w/o parallelization	10.86
total with parallelization	6.84

From Table 4.3 it can be seen that the efficiency bottleneck of our method is the optical flow computation, saliency estimation, and feature extraction. The graph construction and inference only contribute 5% of the total computational time. Hence, the efficiency of our method can be further improved with the recent advancement in GPU accelerated optical flow, *e.g.*, 0.2 seconds per frame in [7]. For the compared methods, [64, 151, 65] are significantly slower because they employ the more advanced but time-consuming region proposals [24] as the primitive input.

4.4 Conclusion

In this work, we propose an efficient and effective appearance modeling technique in the MRF framework for automatic thematic video object segmentation. The proposed method uses histogram features to characterize the local regions and embed the global appearance constraint into the graph by auxiliary nodes and connections. Compared with many existing appearance models, the optimization process of our method is non-iterative. Experimen-

tal evaluations show that our method is faster than many of the alternatives and the segmentation accuracy is also better than or comparable with the state-of-the-art methods.

Chapter 5

Temporally Enhanced Image Object Proposals for Videos

Despite the recent success of image object proposals (IOPs) for image applications, the per-frame IOPs are also important for video applications. However, the existing IOPs are extracted from each frame separately and may exhibit inconsistencies across the frames. In this work, we propose to improve the existing IOPs by enforcing the temporal consistency through a video sequence in an online manner. To achieve this, we propose a novel spatiotemporal objectness measure considering both the frame level objectness as well as the temporal consistency across frames. An online dynamic programming technique is proposed to efficiently compute such spatiotemporal objectness. In addition, compared with the spatiotemporal video object proposals(VOPs), the proposed method supports online applications and provides more accurate per-frame localizations. Experiments on benchmark datasets validate its superior performance compared with the

existing IOPs and VOPs. This work has been published in the IEEE International Conference on Multimedia and Expo [141]

5.1 Introduction

In recent years, image object proposals (IOPs) [164, 123] have been actively studied in the literature. They generate image bounding boxes or segments to capture candidate object locations in each frame. Despite the popularity of IOPs in image applications [34, 33, 103], it is also useful for video applications as the per-frame localization is still important in videos. For example, in online video object or action detection [109], the localization is performed in each current frame while streaming the video. Moreover, IOPs are also commonly used as the primitive input to many offline video object or action detections [28, 144]. However, the existing IOPs are extracted from each frame separately and may exhibit inconsistencies across the frames. In this work, we propose to improve the existing IOPs by enforcing the temporal consistency through a video sequence in an online manner. We name this improved IOPs as temporally enhanced image object proposals (TE-IOPs).

The task is challenging due to four reasons. First, the original IOPs are noisy as they are just proposals instead of accurate detections. Indeed, only a few out of hundreds of proposals capture the true object locations. Secondly, the IOPs extracted from different frames are not associated with each other. It is not a trivial task to associate them to explore temporal consistency in the presence of viewpoint/scale variation and fast motion. Thirdly, there are hundreds or thousands of frames in each video and hundreds of proposals in each frame. The method must be efficient. Last but not the least, it needs to run online to support online video object or action detections, *i.e.*, only frame 1 to t can be used while processing frame t.

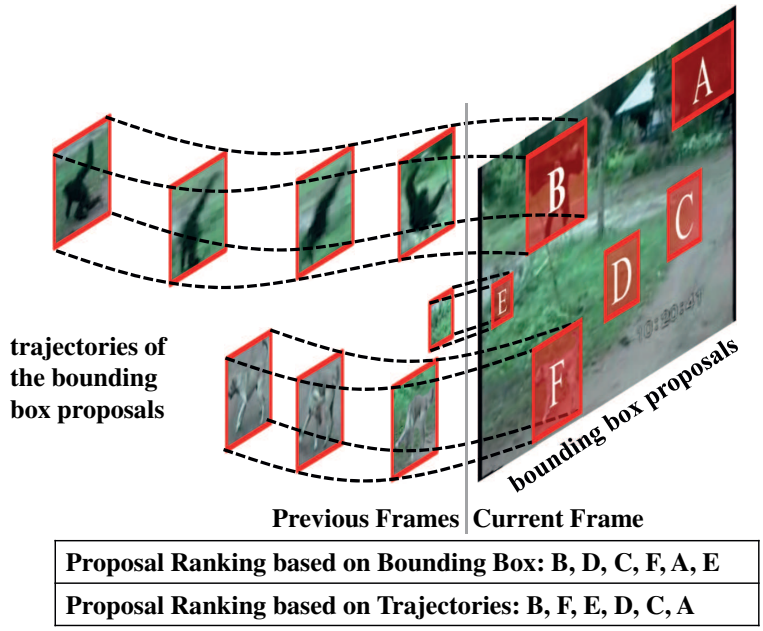


Fig. 5.1 Illustration of how the proposed TE-IOPs improves the original IOPs.

To tackle these challenges, for each IOP, we propose to trace it backward in the past frames to form a short video tube that captures the temporal context of the object. We achieve this by linking the bounding box proposals at current frames to those in the past frames. We then propose a novel spatiotemporal objectness to re-rank the original IOPs by considering the per-frame objectness as well the temporal coherency along the traced tubes. A graphical illustration is shown in Fig. 5.1. However, generating TE-IOPs from IOPs by tracing the moving trajectories of all bounding box proposals could be computationally expensive. We have to trace hundreds of bounding boxes for each frame and the number of possible trajectories for each IOP is exponential to the number of frames. As a result, a brute force solution is not feasible. To address this problem, we first design a measure, namely spatiotemporal objectness, to assess the quality of the trajectories based on both appearance and motion cues. We then design a spatiotemporal dynamic

programming algorithm to efficiently search for the best trajectory of each bounding box, with optimality guarantee and linear time complexity.

In addition, spatiotemporal video object proposals (VOPs) [144, 45, 95] have also been studied in recent years. However, they have to see the whole video to generate proposals, while the proposed TE-IOP runs online. Moreover, different from the proposed TE-IOPs which provide per-frame localization proposals, VOPs generate spatiotemporal tubes to capture the whole spatiotemporal extent of the objects throughout the video. As a result, it may not provide good per-frame localizations especially for the objects/events appearing for a short duration. Furthermore, some VOP methods like [144] use the per-frame IOPs as input to perform spatiotemporal localizations in videos. There are also some CNN based VOP approaches such as [57, 135]. However, these methods only use the CNN to produce per-frame detections or spatiotemporal detections on a small fixed temporal window, *e.g.*, 10 frames. Postprocessing like temporal linking or tracing is still needed to generate full-length proposals.

5.2 Proposed Method

Given a video sequence $\mathcal{V} = \{I_1, I_2, \dots I_N\}$ with N frames and a set of image object proposal (IOP) bounding boxes $\mathbb{B} = \{\mathcal{B}_1, \mathcal{B}_2, \dots \mathcal{B}_N\}$ in all the frames, the goal is to assign each IOP a spatiotemporal objectness scores by tracing their temporal context, $\mathbb{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots \mathcal{T}_N\}$, in the past frames. $\mathcal{B}_t$ is a set of IOPs at frame I_t and $\mathcal{T}_t$ is a set of tubes associated with each bounding box in $\mathcal{B}_t$, *i.e.*, $\mathcal{T}_t = \{T_b \mid b \in \mathcal{B}_t\}$. T_b denotes a temporal sequence of IOPs in the past frames ending at b, *i.e.*, $T_b = \{b, b_{-1}, b_{-2}, \dots\}$ where b_{-k} represents the box of T_b at frame I_{t-k}. T_b is expected to provide temporal context to b by

tracing the movement of b in the past frames. Note that, we can only use $\{I_k \mid 1 \leq k \leq t\}$ to generate $\mathcal{B}_t$ and $\mathcal{T}_t$ due to the online constraint.

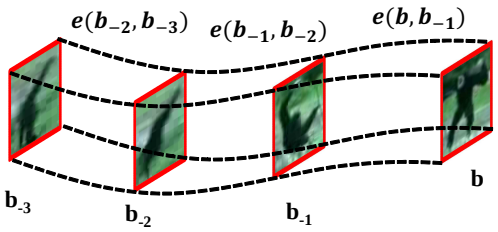
5.2.1 Problem Formulation

Let us denote the per-frame objectness score of image object proposal b as $s(b)$. This score represents how likely a bounding box contains a real object and is usually used to rank the proposals. An IOP method is regarded better if it can reach a recall level using less number of proposals based on the objectness ranking. After obtaining IOPs on each frame, the next step is to find the trajectories, T_b ending at each bounding box b. In order to ensure the temporal smoothness of the traces, the bounding boxes along a trajectory need to satisfy the following constraints:

$$\begin{aligned} \mathrm{IoU}(b_{-i}, b_{-(i+1)}) &> \tau, \\ TD(b_{-i}, b_{-(i+1)}) &> d, \end{aligned} \tag{5.1}$$

where b_{-i} and $b_{-(i+1)}$ denote two consecutive bounding boxes along the trajectory, IoU denotes the intersection over union ratio between two bounding boxes and TD denotes the dense trajectory density used in [88] between two bounding boxes. $TD(b_{-i}, b_{-(i+1)})$ is computed as the number of dense trajectories [127] passing through both b_{-i} and $b_{-(i+1)}$ normalized by the summation of the two bounding boxes' areas. τ is a fixed parameter and d is adaptively chosen for each frame pair to keep the top 10% of the pairwise connections satisfying the IoU constraint.

However, there usually exists more than one candidate trajectories ended at bounding box b satisfying Eq.(5.1), and we denote them as $\mathcal{T}_b = \{T_b^{(1)}, T_b^{(2)}, T_b^{(3)}, ...\}$. Our target is to find a single best trajectory $T_b^* \in \mathcal{T}_b$ that is most likely to trace the bounding box b. In order to find $T_b^* \in \mathcal{T}_b$, we propose



$\mathbf{b}$	$s(b)$
$\mathbf{b}_{-1}$	$s(b_{-1}) \times \alpha \times e(b, b_{-1})$
$\mathbf{b}_{-2}$	$s(b_{-2}) \times \alpha^2 \times e(b_{-1}, b_{-2}) \times e(b, b_{-1})$
$\mathbf{b}_{-3}$	$s(b_{-3}) \times \alpha^3 \times e(b_{-2}, b_{-3}) \times e(b_{-1}, b_{-2}) \times e(b, b_{-1})$

Fig. 5.2 An example to illustrate Eq.(5.2). The top figure shows an example trajectory and the bottom table shows the contribution of each bounding box to $Q(T_b)$.

a spatiotemporal objectness measure to qualify each trajectory T_b. One similar measure is proposed in [119] which simply sums up all the bounding boxes' confidence scores. However, it is not the best choice for our per-frame online localization scenario due to the following reasons. First, we are looking for the trajectory for a particular bounding box instead of the whole video. We need to emphasize more on the boxes that are nearer to the current frame. Secondly, the tracing is more likely to drift in the further away frames. Thirdly, if the trajectory can track the object, the appearance should also vary smoothly along the trajectory. Thus, we propose a new spatiotemporal objectness measure for a trajectory T_b:

$$Q(T_b) = s(b) + \sum_{i=1}^{L-1} \alpha^i \times \beta_i \times s(b_{-i}), \tag{5.2}$$

where L is the length of the trajectory including b, $\alpha \in [0, 1]$ is a decay factor to emphasize more on the most recent frames, and β_i describes the geodesic appearance variations between b and b_{-i} along the trajectory. β_i is defined as

$$\beta_i = \prod_{j=1}^{i} e(b_{-(j-1)}, b_{-j}), \tag{5.3}$$

where $e(b_i, b_j)$ represents the visual similarity between bounding box b_i and b_j. We define $e(b_i, b_j)$ based on the color histogram features of b_i and b_j:

$$e(b_i, b_j) = \exp(-\frac{\|H_i - H_j\|^2}{2 \times \gamma_{ij}^2}), \tag{5.4}$$

where H_i and H_j denote the normalized RGB color histograms of bounding box b_i and b_j, respectively. γ_{ij} is a normalization factor computed as the average of the ℓ_2 distances of all the bounding box pairs satisfying the IoU constraint in Eq.(5.1) between the two frames containing b_i and b_j, respectively. Note that, similar to α, the value of $e(b_i, b_j)$ is also between 0 and 1. A graphical illustration of $Q(T_b)$ is shown in Fig. 5.2. In the experiment section, we will show the benefit of this newly proposed spatiotemporal objectness measure compared with the one used in [119].

With this new spatiotemporal objectness measure $Q(T_b)$, we can then select the trajectory with the highest spatiotemporal objectness score as the optimal trajectory T_b^* of the bounding box proposal b. Formally, we formulate this process as the following optimization problem:

$$T_b^* = \arg\max_{T_b \in \mathcal{T}_b} Q(T_b). \tag{5.5}$$

To solve this optimization problem, we can naively enumerate all the candidate trajectories for each bounding box proposal. However, this is computationally intractable because the number of candidate trajectories ending at each bounding box proposal is exponential to the number of frames, *i.e.*, $1 + n + n^2 + n^3 + \cdots + n^{t-1}$ where n is the average number of neighboring bounding box proposals satisfying Eq.(5.1) and t is the index of the current frame. Moreover, the algorithm proposed in [119] is not directly applicable here because we use a different quality measure for trajectories. In the

next section, we propose an online dynamic programming approach to solve Eq.(5.5) efficiently.

5.2.2 Online Search of Video Object Proposals

It is observed that, the candidate traces for the IOPs in frame I_t highly overlaps with those in frame I_{t-1}. Hence, we seek to re-use the generated trajectories in the previous frame to speed up the search process in the current frame. We first rewrite the spatiotemporal objectness measure defined in Eq.(5.2) in recurrence form by assuming the length of the trajectory is at least 2, *i.e.*, b_{-1} exists:

$$Q(T_b) = s(b) + \alpha \times e(b, b_{-1}) \times Q(T_{b_{-1}}). \tag{5.6}$$

Based on this idea, we propose an efficient online dynamic programming approach to solve Eq.(5.5) in linear time complexity. Formally, we take the following three steps defined by Eq.(5.7), (5.8) and (5.9) to search for the optimal trajectory of b.

$$b^*_{-1} = \arg \max_{b_{-1} \in \mathcal{N}(b)} e(b, b_{-1}) \times Q(T^*_{b_{-1}}) \tag{5.7}$$

$$T^*_b = \begin{cases} \{b, T^*_{b^*_{-1}}\}, & \mathcal{N}(b) \neq \emptyset \text{ and } Q(T^*_{b^*_{-1}}) \geq 0 \\ \{b\}, & \text{otherwise} \end{cases} \tag{5.8}$$

$$Q(T^*_b) = s(b) + \max\{0, \alpha \times e(b, b^*_{-1}) \times Q(T^*_{b^*_{-1}})\} \tag{5.9}$$

$\mathcal{N}(b)$ denotes the collection of bounding boxes that can be connected to b in the previous frame according to Eq.(5.1). The step in Eq.(5.7) is to find the bounding box $b^*_{-1} \in \mathcal{N}(b)$ with the highest spatiotemporal objectness score after decaying by the appearance term. The step in Eq.(5.8) is to generate

the optimal trajectory for bounding box b based on $\mathcal{N}(b)$ and $Q(T^*_{b^*_{-1}})$. If $\mathcal{N}(b)$ is not empty and $Q(T^*_{b^*_{-1}})$ is greater than zero, b will be connected to b^*_{-1} to produce the optimal trajectory for bounding box b. Otherwise, the optimal trajectory T^*_b will just be the bounding box b itself. The step in Eq.(5.9) is to compute the spatiotemporal objectness score $Q(T^*_b)$ of the newly generated trajectory T^*_b.

Lemma 5.2.1. *Eq.(5.7), (5.8) and (5.9) give the global optimal solution to Eq.(5.5).*

Proof. We use mathematical induction to prove that Eq.(5.7), (5.8) and (5.9) give the global optimal solution to Eq.(5.5). The proof is in spirit similar to [119] and [100]. In the following we will use the term $\widehat{T^*_b}$ to denote the generated trajectory from Eq.(5.7), (5.8) and (5.9), and prove that $\widehat{T^*_b}$ is indeed the global optimal solution to Eq.(5.5). Note that, we can prove this by either directly proving $\widehat{T^*_b} = T^*_b$ or indirectly proving $Q(\widehat{T^*_b}) = Q(T^*_b)$. In the first frame, we always have $\mathcal{N}(b) = \emptyset$, and $T^*_b = \{b\}\ \forall b \in \mathcal{B}_1$. Hence, $\widehat{T^*_b} = T^*_b$ is true. Let us assume that $\widehat{T^*_b} = T^*_b$ is true in the k^{th} frame. There are three conditions in the $(k+1)^{th}$ frame. The first condition is $\mathcal{N}(b) = \emptyset$. In this case the trajectory can only be the bounding box itself, and hence $\widehat{T^*_b} = T^*_b$. The second condition is $\mathcal{N}(b) \neq \emptyset$ and $Q(T^*_{b^*_{-1}}) < 0$. $Q(T^*_{b^*_{-1}}) < 0$ implies $Q(T_{b_{-1}}) < 0\ \forall b_{-1} \in \mathcal{N}(b)$, $\forall T_{b_{-1}} \in \mathcal{T}_{b_{-1}}$, and connecting b to any bounding box in $\mathcal{N}(b)$ will make the trajectory objectness score less than $s(b)$. Hence T^*_b will still be the bounding box itself and $\widehat{T^*_b} = T^*_b$. The last condition is $\mathcal{N}(b) \neq \emptyset$ and $Q(T^*_{b^*_{-1}}) \geq 0$. We prove $Q(\widehat{T^*_b}) = Q(T^*_b)$ in this

situation. Based on Eq.(5.6), (5.7) and (5.9) we have

$$\begin{aligned} Q(\widehat{T_b^*}) =& s(b) + \alpha \times e(b, b_{-1}^*) \times Q(T_{b_{-1}^*}^*) \\ \geq& s(b) + \alpha \times e(b, b_{-1}) \times Q(T_{b_{-1}}^*), \forall b_{-1} \in \mathcal{N}(b) \\ \geq& s(b) + \alpha \times e(b, b_{-1}) \times Q(T_{b_{-1}}), \\ & \forall b_{-1} \in \mathcal{N}(b), \forall T_{b_{-1}} \in \mathcal{T}_{b_{-1}}. \end{aligned} \tag{5.10}$$

This implies that $Q(\widehat{T_b^*})$ is equal to or greater than the spatiotemporal objectness scores of any possible trajectories ending at b, *i.e.*, $Q(\widehat{T_b^*}) \geq Q(T_b)$, $\forall T_b \in \mathcal{T}_b$. As a result, $Q(\widehat{T_b^*}) = Q(T_b^*)$ and $\widehat{T_b^*}$ is the global optimal solution for the $(k+1)^{th}$ frame. □

The time complexity of the dynamic programming algorithm to process a single frame is $O(B \times n)$ where B is the number of bounding box proposals in this frame and n is the average number of neighborhood bounding box proposals in the previous frame satisfying Eq.(5.1). Its memory footprint is also $O(B \times n)$ to process a single frame. It is worth noting that both complexities are independent of the number of past frames, which implies that the proposed dynamic programming algorithm does not impose any limit on the video length. Finally, we complete the steps to generate the proposed TE-IOPs for a given input video in Algorithm 2. The operation $ImProp$ in Algorithm 2 means to extract image object proposals from a video frame.

Algorithm 2 The Generation of Temporally Enhanced Image Object Proposals (TE-IOPs)

Input: video frames $\{I_1, I_2, ..., I_N\}$
Output: TE-IOPs, *i.e.*, $\mathbb{B}$ and $\mathbb{T}$
$\mathbb{B} \leftarrow \emptyset$
$\mathbb{T} \leftarrow \emptyset$
for $t = 1 \rightarrow N$ **do**
    $\mathcal{B}_t \leftarrow ImProp(I_t)$
    $\mathcal{T}_t \leftarrow \emptyset$
    for each b in $\mathcal{B}_t$ **do**
        $T_b^* \leftarrow \{b\}$
        $Q(T_b^*) \leftarrow s(b)$
        if $\mathcal{N}(b) \neq \emptyset$ **then**
            $b_{-1}^* \leftarrow \arg\max_{b_{-1} \in \mathcal{N}(b)} e(b, b_{-1}) \times Q(T_{b_{-1}}^*)$
            if $Q(T_{b_{-1}^*}^*) \geq 0$ **then**
                $T_b^* \leftarrow \{b, T_{b_{-1}^*}^*\}$
                $Q(T_b^*) \leftarrow s(b) + \alpha \times e(b, b_{-1}^*) \times Q(T_{b_{-1}^*}^*)$
            end if
        end if
        $\mathcal{T}_t \leftarrow \mathcal{T}_t \cup \{T_b^*\}$
    end for
    $\mathbb{B} \leftarrow \mathbb{B} \cup \{\mathcal{B}_t\}$
    $\mathbb{T} \leftarrow \mathbb{T} \cup \{\mathcal{T}_t\}$
end for

5.3 Experiments

5.3.1 Datasets and Evaluation Criteria

Three datasets are used in the experiments, *i.e.*, *NTU-Adobe* [142] dataset, *UCF-Sports-Action* [104] dataset and the 24-class localization subset of the *UCF101* [110] dataset. *NTU-Adobe* is a multi-category salient object detection dataset comprising 51 videos with dynamic scenes and camera motions. *UCF Sports Action* and *UCF101* are the benchmarking action detection datasets containing 150 and 3204 videos, respectively. Per-frame bounding box annotations are provided in all the datasets to denote the objects or actors. We extract the top 300 edge box proposals [164] from each video frame as the initial IOPs for experimentation. The number 300 is chosen to

ensure that at least 90% of the ground truth object localizations are recalled in all the datasets.

Following the criterion of evaluating IOPs and VOPs, we use the best intersection over union (bIoU) scores to evaluate how well a ground truth location is recalled when a certain number of proposals is returned for each frame. We then obtain the average bIoU (abIoU) scores of all the ground truth locations in all the frames and videos. The correct localization ratio (CorLoc) is also used. A ground truth location is regarded as recalled when its bIoU score exceeds a threshold, and the CorLoc ratio measures what percentage of the ground truth localizations are recalled. As a common practice, we set the threshold to 0.5, *i.e.*, CorLoc@50%.

5.3.2 Comparisons with Baselines

We first compare TE-IOPs with the original IOPs on all the datasets to see how the spatiotemporal objectness improves the original objectness ranking. We also compare with a baseline of our TE-IOPs without using the decay terms defined in Eq.(5.2), *i.e.*, set $\alpha = 1$ and $\beta = 1$. We denote this non-decay baseline as TE-IOPs-ND in the comparisons. Finally we compare with some state-of-the-art offline VOP methods on the benchmarking *UCF Sports Action* and *UCF101* datasets. In addition to the original edge box objectness scores, we also use the saliency measure of each bounding box proposal proposed in [142, 139]. We denote these methods as SIOPs, TE-SIOPs, and TE-SIOPs-ND, respectively. In all the experiments, we set $\tau = 0.5$ for *UCF Sports Action* and *UCF101* datasets, $\tau = 0.75$ for *NTU-Adobe* dataset, and $\alpha = 0.8$ for all three datasets. A smaller τ is used for the action detection datasets as they usually contain fast motions. These parameters will be discussed later in more detail.

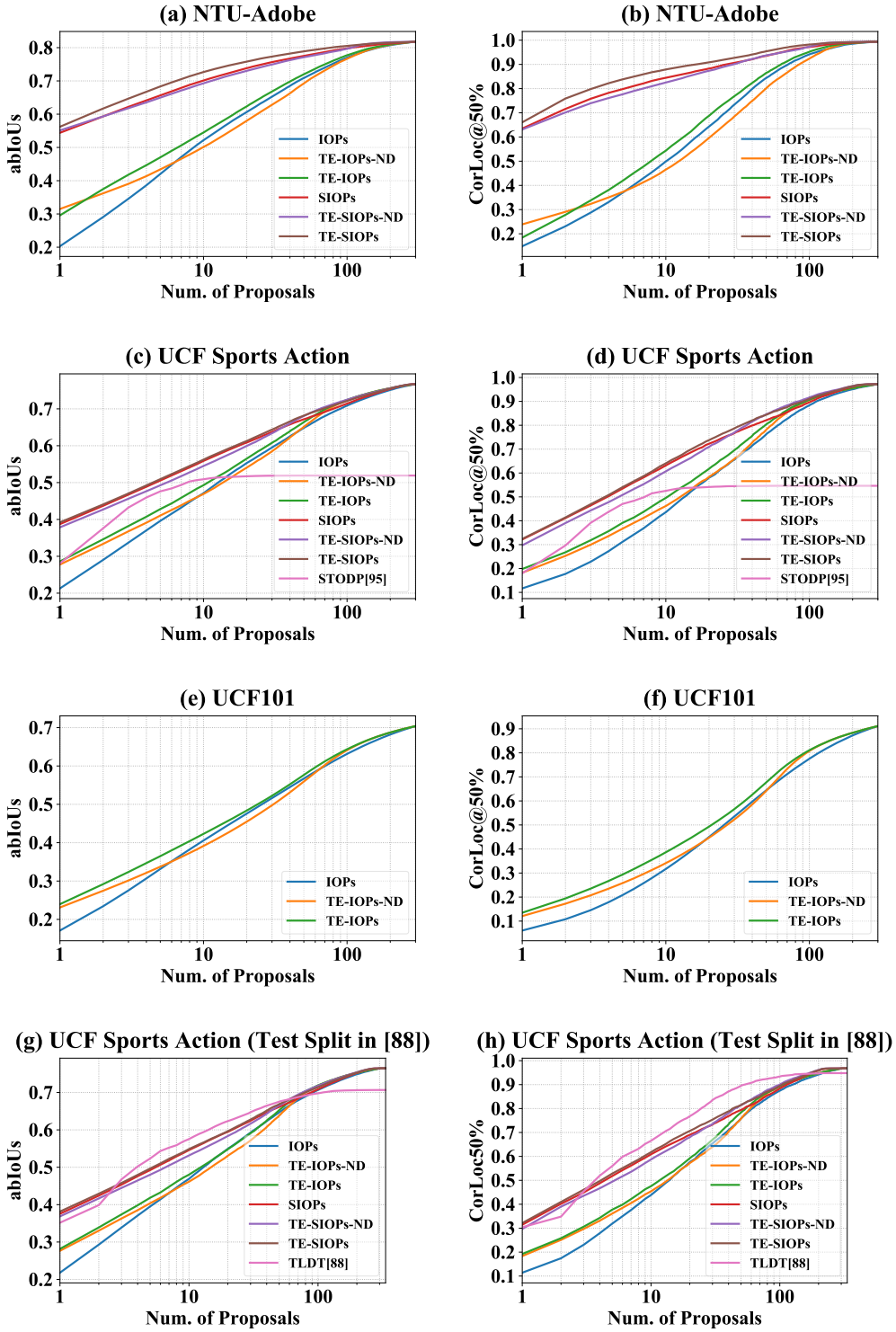


Fig. 5.3 Quantitative comparisons with the baselines and the state-of-the-art offline VOPs. Best viewed in color. Top 300 proposals are shown in each plot.

The quantitative comparisons are shown in Fig. 5.3. We draw the abIoU and CorLoc@50% *vs.*Number of Proposal curves because they reveal which method can use fewer proposals to achieve the same recall. This is crucial especially for videos because IOPs are generated on a per frame basis instead of per video basis, thus a less number of proposals can largely save the computational cost in the later classification stage. By comparing *SIOPs* with *IOPs*, it can be seen that the saliency in [139] is a useful cue to improve the proposal ranking in videos. However, saliency computation in [139] is much more time-consuming. It is worth noting that the proposed method is independent of the initial objectness measures. Even when the original IOPs are already very good, *i.e.*, *SIOPs*, the proposed method, *TE-SIOPs*, can still improve it. By comparing *TE-IOPs* and *TE-SIOPs* with *IOPs* and *SIOPs*, respectively, it can be seen that finding the trajectories in the previous frames can improve the ranking and decrease the number of proposals required to achieve the same recall. This is because our method can downgrade false positives if they have a weak trace in the past frames and upgrade false negatives if they have a strong trace in the past frames. By comparing *TE-IOPs* and *TE-SIOPs* with *TE-IOPs-ND* and *TE-SIOPs-ND*, respectively, it can be seen that the decay terms in Eq.(5.2) are important for per-frame localization. Furthermore, when the initial IOPs are already very good, *i.e.*, *SIOPs*, the non-decay version hurts the performance significantly. Some qualitative results illustrating how our TE-IOPs improve the original IOPs are shown in Fig. 5.4. In the presented examples, our online proposals can already capture the location of the object using as few as top 2 object proposals.

It is worth noting that our TE-SIOPs can be naively extended to perform online salient object detection by using the top 1 proposal in each frame.

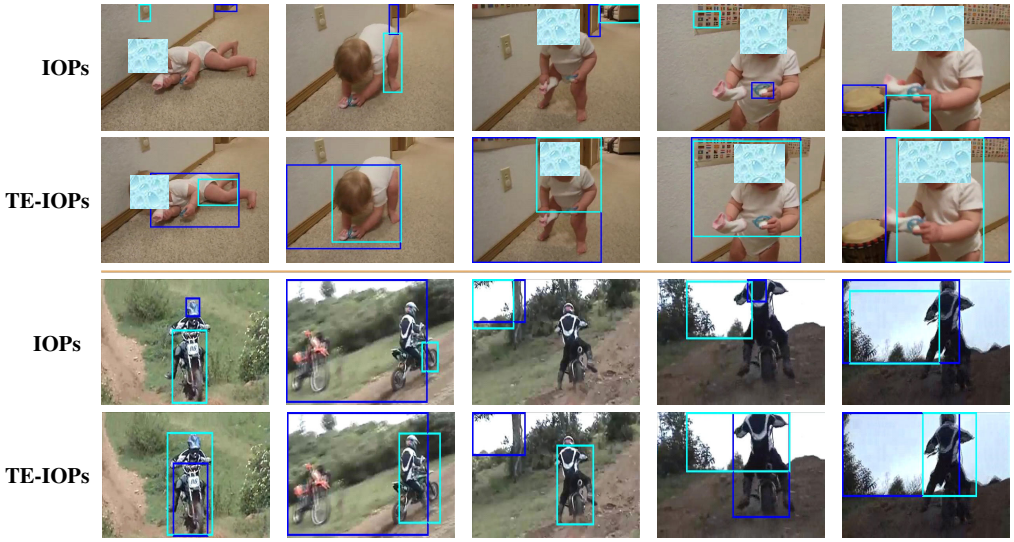


Fig. 5.4 Comparisons between *TE-IOPs* and *IOPs*. Only top 2 proposals are drawn on each frame for clarity.

Table 5.1 Comparison with APT when we use similar number of proposals as APT under different parameter settings. The numbers in the parentheses indicate the per-frame average numbers of bounding box proposals.

		abIoUs	CorLoc@50%
UCF Sports Action	APT-200 (78)	57.72%	69.16%
	TE-IOPs (top 78)	70.74%	87.87%
	APT-50 (937)	75.95%	95.98%
	TE-IOPs (top 300)	76.72%	97.24%
UCF 101	APT-700 (75)	50.89%	54.35%
	TE-IOPs (top 75)	61.87%	76.42%
	APT-50 (894)	71.18%	90.82%
	TE-IOPs (top 300)	70.39%	91.12%

From Fig. 5.3, it can be seen that this simple setting can already detect around 66% of the ground truth objects in *NTU-Adobe* dataset.

5.3.3 Comparison with Video Object Proposals

We compare our TE-IOPs with four offline video object proposal (VOP) methods, *i.e.*, STODP [95], APT [124], TLDT [88] and Tubelets [45] on the

[1]As shown in Table 5.1, the abIoU of our TE-IOPs does not reach 71.18% even using all 300 proposals. Here we get this number by using 400 edge box proposals as the initial input.

Table 5.2 Number of proposals needed by our method to reach the same abIoU and CorLoc@50% scores as APT under different parameter settings. The numbers in the parentheses indicate the per-frame average numbers of bounding box proposals for APT method.

		abIoUs	CorLoc@50%
UCF Sports Action	APT-200 (78)	23	30
	APT-50 (937)	222	217
UCF 101	APT-700 (75)	27	27
	APT-50 (894)	346[1]	286

benchmark *UCF Sports Action* and *UCF101* datasets. In order to convert these offline VOPs to IOPs, we map the offline spatiotemporal proposals to each frame to obtain bounding box proposals and duplicated bounding box proposals in each frame are removed to ensure fair comparisons. The ranking of the mapped bounding box proposals on each frame is determined by their tubes' original rankings. For [95] and [88], we directly plot the abIoUs and CorLoc@50% *vs.* Number of Proposal curves in Fig. 5.3. For [124], we cannot draw these curves because its proposals are obtained by clustering and thus not ranked. Instead, we compare with it in two different ways. The first is to compare the abIoUs and CorLoc50% scores when both our method and APT use similar number of bounding box proposals, and the second is to compare the number of proposals required by our method to reach the same abIoU and CorLoc50% scores as APT. The default parameter setting of APT, *i.e.*, APT-50 where the number 50 is the value of the parameter in APT controlling the number of generated proposals, produces 937 and 894 bounding box proposals on average after mapping to each frame for the *UCF Sports Action* and *UCF101* datasets, respectively. We also tune the parameter of APT to generate less number of proposals, *i.e.*, APT-200 for *UCF Sports Action* which produces 78 bounding box proposals on average for each frame, and APT-700 for *UCF101* which produces 75 bounding box proposals on average for each frame. In Table 5.1 we compare our

TE-IOPs with the APT method under different parameter settings in terms of the final abIoUs and CorLoc@50%. In Table 5.2, we show the number of proposals needed by our TE-IOPs method to reach the same recall level of APT under different parameter settings. For [45], their proposals are not ranked either. On the *UCF Sports Action* dataset, it produces 108.06 bounding box proposals on each frame after mapping and reaches a recall level of 78.21% in terms of CorLoc@50% and 63.81% in terms of abIoUs. The proposed TE-IOPs only need 45 and 40 bounding box proposals in each frame to reach the same CorLoc@50% and abIoUs levels, respectively.

Overall, it can be seen that the performance of the proposed TE-IOPs/TE-SIOPs is on par with or superior to these state-of-the-art offline VOPs, even though they have used information from the entire video while generating the proposals. One reason is that the goal of this study is to improve the per-frame image object bounding box proposals and thus we evaluate the performance in terms of per-frame localization accuracy, *i.e.*, how good can the per-frame bounding box localization proposals capture the objects in each frame. The goal of these offline VOPs are to generate spatiotemporal localization tubes, *i.e.*, sequences of bounding boxes, to capture the complete moving trajectories of the objects throughout the video and they may not provide accurate per-frame localization.

5.3.4 Parameter Evaluation

The two major parameters involved in our method is the IoU threshold τ and decay term α. We evaluate the sensitivity of these two parameters by fixing one and varying another. For parameter evaluation purpose, we average the abIoU and CorLoc@50% scores of the top 20 bounding box proposals to give a single value evaluation of the performance. This is conceptually

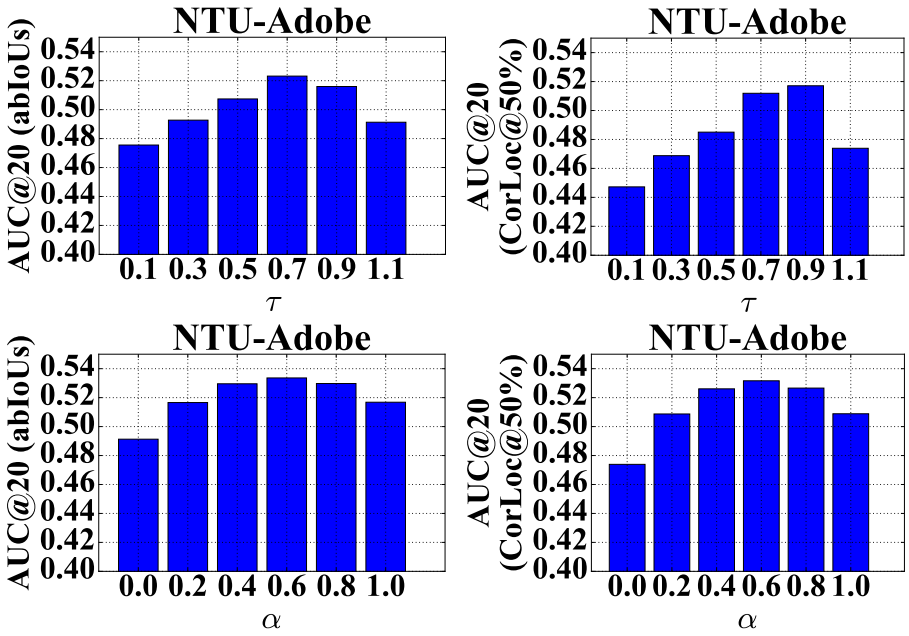


Fig. 5.5 Parameter sensitivity evaluations of the IoU threshold τ and decay term α on *NTU-Adobe* dataset. The top two curves are obtained by varying τ with $\alpha = 0.5$, and the bottom two curves are obtained by varying α with $\tau = 0.8$.

similar to the commonly used Area Under the Curve (AUC) concept in the literature. The evaluation results on *NTU-Adobe* dataset are shown in Fig. 5.5. It can be seen that the performance varies smoothly with these two parameters. It is worth noting that $\alpha = 0.0$ or $\tau = 1.1$ corresponds to the performance of the original IOPs. This is because when $\alpha = 0.0$, the new spatiotemporal objectness score will be the same to the original 2D objectness scores regardless of the traced trajectory, and when $\tau > 1$, there will be no edges between consecutive frames. This comparison demonstrates the importance of the proposed spatiotemporal objectness scores. Note that, in the evaluation of α, making $\alpha = 1.0$ does not correspond to the non-decay version of our method as compared in Fig. 5.3 because we have two decay terms as shown in Eq.(5.2), *i.e.*, one is the fixed constant α, and the other is the adaptive appearance decay term β_i. The compared non-decay version in Fig. 5.3 turns off both decays.

Moreover, since we have two decay terms in total and the comparisons in Fig. 5.3 turn off both decays, *i.e.*, the fixed decay α and appearance adaptive decay β, we perform an ablation study on the two decay terms separately. The comparison results on the NTU-Adobe dataset are shown in Fig. 5.6. The baseline "TE-IOPs (No Decays)" corresponds to the baseline "TE-IOPs-ND" in Fig. 5.3. It can be seen that the successive addition of the two decay terms consistently improve the performance and their combination produces the best result.

5.3.5 Runtime Analysis

Thanks to the simplicity of the proposed dynamic programming algorithm, the proposed method is very efficient given the input per-frame IOPs. Most of the time is spent on edge score computation and dense trajectory extraction. For instance, in a video of the *NTU-Adobe* dataset with 601 frames and 300 edge box proposals in each frame, the edge score computation, and IOU based edge pruning take 27.53 seconds, while the dynamic programming algorithm only takes 0.15 second. Overall, we achieve near real-time, *i.e.*, 21.65 frames per second, throughput. In a video of the *UCF101* dataset with 900 frames, the dense trajectory extraction takes 39.08 seconds, the edge score computation takes 21.47 seconds, the IoU and trajectory density based edge pruning takes 14.83 seconds, and the dynamic programming algorithm takes 0.69 seconds. Since the edge score computation and dense trajectory extraction do not depend on each other, they can be run concurrently. With this assumption, the processing throughput is around 16.48 frames per second. It can be seen that, on the analyzed examples, our method can achieve real-time performance when the video frame rate is not high, *e.g.*, below 21 fps without dense trajectory and below 16fps with dense trajectory. To achieve

real-time performance for videos with high frame rate, we can either perform a downsampling on the frame rate or implement a faster version of dense trajectory extraction algorithm through parallel computation, *e.g.*, GPU. Note that, all the computation times are measured using a desktop computer equipped with a 4-core Intel i5 processor and 32GB of memory.

5.3.6 Limitations

Despite the shown good performance, there are a few limitations of the proposed TE-IOPs. First, the method only links the per-frame detections from adjacent frames. If an object exits the scene for a moment and re-enters later, our method may not be able to associate the re-entered object to its previous existence. It will tend to treat the new appearance as a new object. In future, we plan to alleviate this issue by exploring skip connections between distant frames. Secondly, in our formulation, we have used a fixed IoU threshold to determine possible edge connections. This is not optimal especially in extreme cases where the video contains very fast or slow motion. In future, we will design a variable threshold to automatically adapt to different motion speeds. Thirdly, since our method is independent of the initial input proposals, we have no control over their quality. For example, if the noisy detections are also very consistent over a long sequence of frames, their spatiotemporal scores will also be high. An underlying assumption of the proposed method is that the detections on true objects should be more temporally consistent compared with noisy detections. Although the overall good performance showed previously validates that the Edge Box proposals mostly satisfy this assumption, some strong and consistent noisy detections do exist and some failure cases are shown in Fig.5.7.

5.4 Conclusion

In this work, we propose a novel form of temporally enhanced image object proposals (TE-IOPs) for videos. The proposed method improves per-frame IOPs in videos by enforcing temporal consistency. To achieve this, we propose a new spatiotemporal objectness measure that considers appearance consistency among the trace, and also penalizes bounding boxes that are far away from the current frame. An online dynamic programming scheme is proposed to search for the trace that brings the maximum spatiotemporal objectness in linear time complexity. Experiment evaluations show that the obtained TE-IOPs are superior to the original IOPs as well as some state-of-the-art offline VOPs.

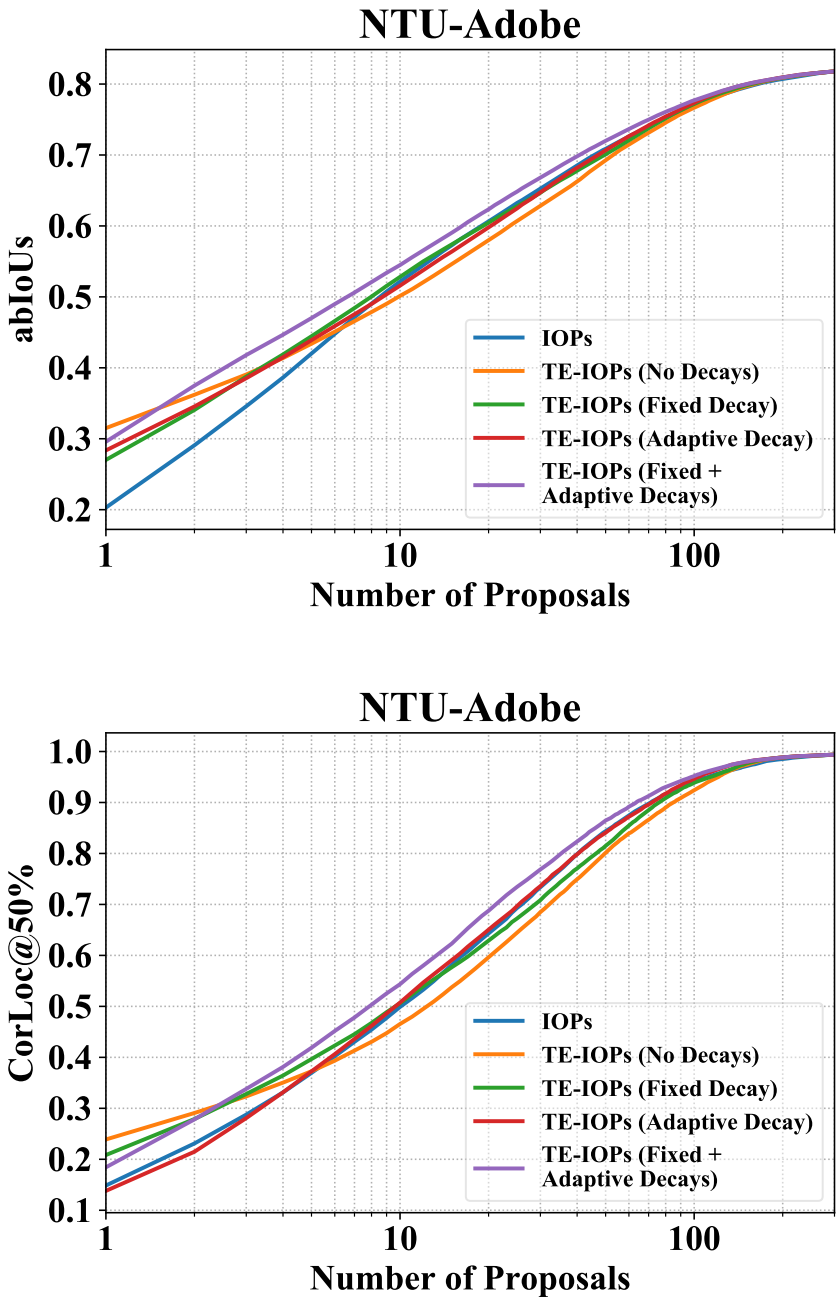


Fig. 5.6 An ablation study on the two decay terms, *i.e.*, fixed decay α and adaptive decay β, on the NTU-Adobe dataset.

Fig. 5.7 Some failure cases of our TE-IOPs. Only top 2 proposals are shown for clarity. The four frames are from the same video playing from left to right. In the fourth frame, our method assigns higher spatiotemporal objectness scores to the noisy detections due to the strong and consistent noisy detections in previous frames.

Chapter 6

Thematic Action Discovery and Localization in Video Collections

Thematic actions in a video collection refer to the common actions that appear frequently in the collection. Similar to common object discovery in images or videos, it is of great interests to discover and locate common actions in videos, which can benefit many video analytics applications such as video summarization, search, and understanding. In this work, we tackle the problem of common action discovery and localization in unconstrained videos, where we do not assume to know the types, numbers or locations of the common actions in the videos. Furthermore, each video can contain zero, one or several common action instances. To perform automatic discovery and localization in such challenging scenarios, we first generate action proposals using human prior. By building an affinity graph among all action proposals, we formulate the common action discovery as a subgraph density

maximization problem to select the proposals containing common actions. To avoid enumerating in the exponentially large solution space, we propose an efficient polynomial time optimization algorithm. It solves the problem up to a user-specified error bound with respect to the global optimal solution. The experimental results on several datasets show that even without any prior knowledge of common actions, our method can robustly locate the common actions in a collection of videos. This work has been accepted by the IEEE International Conference on Computer Vision [140].

6.1 Introduction

Given a collection of unlabeled videos as shown in Fig. 6.1, can we automatically discover and locate the thematic or common actions that frequently appear in these videos? It is worth noting that the video collection may contain multiple types of common actions which are not known in advance, and each video can contain zero, one or several common action instances. Similar to common object discovery in images [74, 113, 148] or videos [54], finding common actions can benefit many video analytics applications such as video summarization [91], search [145, 146] and labeling.

However, compared with the previous success of common object discovery in images and videos [48, 54, 63, 121], common action discovery is much less explored due to the following challenges. First, as we do not know in advance the types or locations of the actions that are common in the given dataset, we have to perform the discovery and localization simultaneously. Given a collection of unlabeled videos, we need to automatically identify a set of spatiotemporal bounding boxes that capture the common actions. Second,

Fig. 6.1 Assuming that we are given a set of unlabeled videos (each frame represents a video), we would like to automatically discover and locate the common human actions in these videos. The common actions to be discovered and located are denoted by bounding boxes. Some videos contain one or multiple common actions, while some videos contain no common actions.

similar actions may also appear differently due to viewpoint variation, scale variation or camera motion. It is not a trivial task to automatically associate these common actions. Last but not the least, besides common actions, videos may also contain dynamic backgrounds or uncommon actions, it is thus critical to differentiate such "noisy motions" from common actions.

To address the above challenges of common action discovery and localization, we first use human prior, *i.e.*, human detector, to generate spatiotemporal action proposals in each video. However, it is inevitable that some proposals may contain dynamic background, uncommon actions or only partially capture the common actions. In order to stand out the proposals containing common actions from the initial proposal corpus, we build an affinity graph of the action proposals and formulate the common action discovery as a subgraph density maximization problem. Instead of using the average degree subgraph density [36] in which the average regularization is usually too strict for our co-localization problem, we propose a different

subgraph density measure that relaxes the average regularization to recall more common actions. To avoid enumerating in the exponentially large solution space, we propose an efficient polynomial time algorithm to effectively find the optimal subgraph that captures common actions. The proposed algorithm solves the proposed formulation within a user-specified error bound with respect to the global optimal solution. A tighter bound requires more computation.

The experimental results on several datasets show that even without any prior knowledge of common actions, the proposed method can robustly locate common actions in unconstrained videos, where each video can contain zero, one or several common actions. The extensive comparisons with other graph-based pattern discovery methods, *e.g.*, [8, 36, 98, 155, 158], as well as one recent video object co-localization method [48] validate the effectiveness of our method in the problem of common action co-localization.

6.2 Proposed Method

In this section, we introduce the proposed action co-localization framework. The input is a collection of unlabeled videos, and the output is spatiotemporal localizations of the common actions appearing frequently in the videos. The proposed method is comprised of two steps. The first step is to extract action proposals from the input video collection. Each action proposal is a spatiotemporal tube, *i.e.*, a temporal sequence of bounding boxes, that locates an action instance. However, besides capturing the common actions, the proposals may also contain noisy background or actions that are not common in the dataset. Hence, the second step is to select the action proposals containing common actions from the initial proposal corpus.

In order to stand out the proposals containing common actions, we utilize the confidence score of each proposal as well as the similarities among the proposals. The former helps to reject the proposals containing non-action background and the latter helps to identify proposals containing common actions from those containing non-common actions. To integrate these two cues in a unified framework, we formulate it as a node selection problem in a graph $\mathbb{G} = (\mathcal{T}, \mathcal{E})$, where $\mathcal{T}$ denotes the collection of nodes, *i.e.*, action proposals, and $\mathcal{E}$ denotes the collection of edges. Node weights represent the quality scores of the proposals, and edge weights represent the semantic similarities between action proposals. Intuitively, the purpose is to select a subgraph in which most of the nodes have high quality scores and are densely connected to each other. Let t_i^j denote the i^{th} proposal in the j^{th} video, s_i^j denote the node weight of t_i^j, and $w(t_i^j, t_p^q)$ denote the edge weight between node t_i^j and t_p^q. A classic formulation to make such a selection is the average degree density maximization formulation in [36]:

$$\mathcal{A}^* = \arg\max_{\mathcal{A}\subseteq\mathcal{T}} D(\mathcal{A}), \tag{6.1}$$

where $\mathcal{A}$ is the subgraph containing the selected nodes and $D(\mathcal{A})$ is the average degree subgraph density defined as

$$D(\mathcal{A}) = \frac{\sum_{t_i^j\in\mathcal{A}} s_i^j + \sum_{\{t_i^j, t_p^q\}\subseteq\mathcal{A}} w(t_i^j, t_p^q)}{|\mathcal{A}|}, \tag{6.2}$$

where $|\cdot|$ is the cardinality of a set. The nominator computes the total node and edge weights of the selected subgraph $\mathcal{A}$, and the denominator regularizes the subgraph size. Although it can be solved efficiently using the method in [36], it is not suitable for our problem because the pure average regularization in the denominator is too strong. The selection of a larger subgraph will

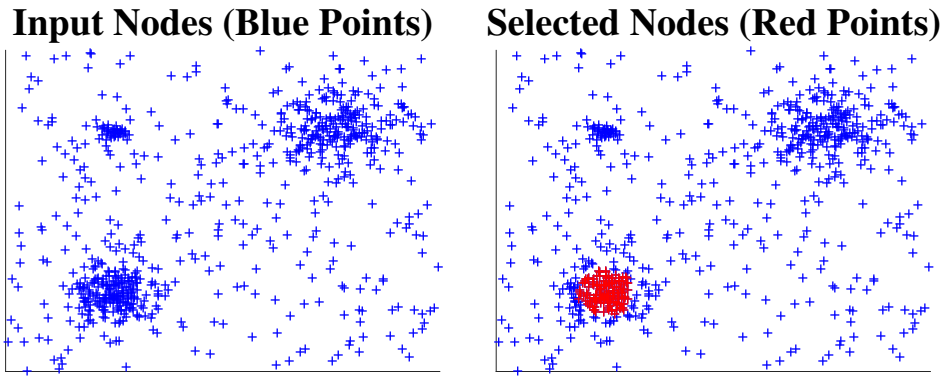


Fig. 6.2 An illustration on the selection results of the classic average degree density maximization formulation defined by Eq. (6.1) and (6.2). Node weights are set to zero for simplicity.

significantly decrease the subgraph density. Thus, it always favors very small subgraphs and leads to low recall. An example is shown in Fig. 6.2. The selection completely misses the top two modes as well as the outer region of the bottom mode. Hence, to overcome this problem, we propose to use a relaxed regularization in the subgraph density definition:

$$D(\mathcal{A}) = \frac{\lambda \times \sum_{t_i^j \in \mathcal{A}} s_i^j + \sum_{\{t_i^j, t_p^q\} \subseteq \mathcal{A}} w(t_i^j, t_p^q)}{\eta + |\mathcal{A}|}. \tag{6.3}$$

Here λ is a parameter balancing the node and edge weights, and η weakens the subgraph size regularization which allows the selection of more nodes to improve recall. We name this new density definition as η-density. When η is zero, it is equivalent to the classic average degree density and only a small number of nodes will be selected. When η approaches to infinity, the regularization vanishes and all the nodes will be selected. However, the addition of η invalidates the optimization algorithm in [36], and enumerating all possible solutions is not feasible as the number of possible subgraphs is exponential to the problem size.

In this work, we propose a polynomial time algorithm to solve Eq. (6.1) and (6.3). The proposed method is applicable to any η values and solves

the problem within a user-specified error bound with respect to the globally optimal solution, although a tighter bound requires more computation.

In the following, we first present the construction of the affinity graph in Section 6.2.1, and then introduce the proposed optimization algorithm to Eq. (6.1) and (6.3). in Section 6.2.2. The details of action proposal generation and description are presented in Section 6.2.3.

6.2.1 Affinity Graph Construction

Given all the action proposals $\{t_i^j\}$ and their feature descriptions $\{f_i^j\}$, we build a ϵ-neighborhood [126] affinity graph, $\mathbb{G} = (\mathcal{T}, \mathcal{E})$, using all the proposal nodes. Node t_i^j and t_p^q will be linked only if $\|f_i^j - f_p^q\|_2 \leq \epsilon$, where $\|\cdot\|_2$ denotes the ℓ_2 distance and ϵ is the bandwidth for graph construction. The edge weight $w(t_i^j, t_p^q)$ is computed as

$$w(t_i^j, t_p^q) = \exp\left(\frac{-\|f_i^j - f_p^q\|_2^2}{2 \times \beta^2}\right), \tag{6.4}$$

where β is computed as

$$\beta = \frac{\sum_{(t_i^j, t_p^q) \in \mathcal{E}} \|f_i^j - f_p^q\|_2}{|\mathcal{E}|}, \tag{6.5}$$

and $|\mathcal{E}|$ is the number of edges in the graph.

6.2.2 Density Maximization Optimization

When $\eta = 0$, Eq. (6.1) and (6.3) are the classic average degree density maximization formulation which can be solved efficiently by the method in [36]. However, the addition of a non-zero parameter η in Eq. (6.3) invalidates the original approach in [36], and enumerating in the exponentially large solution space is computationally infeasible. Hence, in this section, we

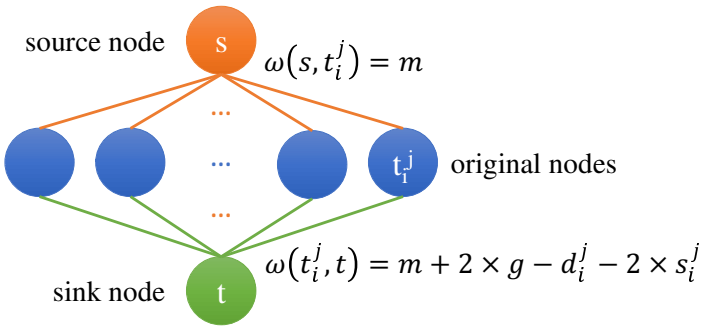


Fig. 6.3 An illustration of the affinity graph with the addition of the source and sink nodes. The edges between the original nodes are omitted for simplicity.

propose a polynomial time algorithm that generalizes the approach in [36] to any nonzero η values. The proposed method uses a binary search strategy to find the optimal density $D(\mathcal{A}^*)$, *i.e.*, given the current lower bound l and upper bound u on $D(\mathcal{A}^*)$, we first check if our new guess $g = \frac{u+l}{2}$ defines a lower or upper bound on $D(\mathcal{A}^*)$ and then shrink the search space by half. A candidate subgraph whose η-density falls within the current bound is maintained during the bound update. In the following, we introduce how to perform this bound check in the binary search process. We ignore λ in Eq. (6.3) without loss of generality.

Bound Check: Inspired by the discovery of [36], the cut capacity of a graph cut operation is related to the average density, *i.e.*, η-density for $\eta = 0$, of the resultant subgraph. In this work, we generalize this relationship to $\eta > 0$ and use it to check the bound of η-density for any non-negative η values. To perform graph cut, we add two auxiliary nodes, *i.e.*, source node s and sink node t, to the original affinity graph. Both s and t are connected to all the original nodes as shown in Fig. 6.3. The newly added source to node weight $\omega(s, t_i^j)$ and node to sink weight $\omega(t_i^j, t)$ are set such that the relationship between the cut capacity and η-density can be used for bound

check. Let d_i^j denote the degree of node t_i^j, we set

$$w(s, t_i^j) = m, \tag{6.6}$$

$$w(t_i^j, t) = m + 2\times g - d_i^j - 2\times s_i^j, \tag{6.7}$$

where the variable m is defined as

$$m = \max_{t_i^j \in \mathcal{T}} (d_i^j + 2\times s_i^j). \tag{6.8}$$

Note that all the newly added edge weights are non-negative based on the definition. In the following, we will show why and how this particular setting can be used to perform the bound check on the η-density for any $\eta \geq 0$.

Now let's cut the new graph by dividing the nodes into two disjoint subgraphs in which one subgraph contains the source node and another subgraph contains the sink node. Given an arbitrary cut, we denote the subgraph containing the source and sink node as $\mathcal{A}_s$ and $\mathcal{A}_t$, respectively. The cut capacity $c(\mathcal{A}_s, \mathcal{A}_t)$ is defined as the summation of the edge weights along the cut boundary. Interestingly, $c(\mathcal{A}_s, \mathcal{A}_t)$ is related to the η-density of

subgraph $\mathcal{A}_s$, *i.e.*, $D(\mathcal{A}_s)$, as follows:

$$\begin{aligned}
& c(\mathcal{A}_s, \mathcal{A}_t) \\
= & \sum_{t_i^j \in \mathcal{A}_t} w(s, t_i^j) + \sum_{t_i^j \in \mathcal{A}_s} w(t_i^j, t) + \sum_{t_i^j \in \mathcal{A}_s, t_p^q \in \mathcal{A}_t} w(t_i^j, t_p^q) \\
= & \; m \times |\mathcal{A}_t| + \Big(m \times |\mathcal{A}_s| + 2 \times g \times |\mathcal{A}_s| \\
& - \sum_{t_i^j \in \mathcal{A}_s} d_i^j - 2 \times \sum_{t_i^j \in \mathcal{A}_s} s_i^j \Big) + \sum_{t_i^j \in \mathcal{A}_s, t_p^q \in \mathcal{A}_t} w(t_i^j, t_p^q) \\
= & \; m \times |\mathcal{T}| + 2 \times g \times |\mathcal{A}_s| - 2 \times g \times \eta + 2 \times g \times \eta \\
& - \sum_{t_i^j \in \mathcal{A}_s} d_i^j - 2 \times \sum_{t_i^j \in \mathcal{A}_s} s_i^j + \sum_{t_i^j \in \mathcal{A}_s, t_p^q \in \mathcal{A}_t} w(t_i^j, t_p^q) \\
= & \; m \times |\mathcal{T}| - 2 \times g \times \eta + 2 \times (|\mathcal{A}_s| + \eta) \times \Big(g \\
& - \frac{\sum_{t_i^j \in \mathcal{A}_s} s_i^j}{|\mathcal{A}_s| + \eta} - \frac{\sum_{t_i^j \in \mathcal{A}_s} d_i^j - \sum_{t_i^j \in \mathcal{A}_s, t_p^q \in \mathcal{A}_t} w(t_i^j, t_p^q)}{2 \times (|\mathcal{A}_s| + \eta)} \Big) \\
= & \; m \times |\mathcal{T}| - 2 \times g \times \eta \\
& + 2 \times (|\mathcal{A}_s| + \eta) \times (g - D(\mathcal{A}_s)).
\end{aligned} \tag{6.9}$$

Let $c^* = c(\mathcal{A}_s^*, \mathcal{A}_t^*)$ denote the minimum cut capacity on the current graph:

$$(\mathcal{A}_s^*, \mathcal{A}_t^*) = \arg \min_{\mathcal{A}_s \cap \mathcal{A}_t = \emptyset, \mathcal{A}_s \cup \mathcal{A}_t = \mathcal{T}, s \in \mathcal{A}_s, t \in \mathcal{A}_t} c(\mathcal{A}_s, \mathcal{A}_t). \tag{6.10}$$

Eq. (6.10) can be solved in polynomial time using the min-cut algorithm proposed in [12]. We then perform the bound check on the current guess g based on Theorem 6.2.1.

Theorem 6.2.1. *Assume subgraph $\mathcal{A}_s^*$ and $\mathcal{A}_t^*$ give the minimum cut c^* and subgraph $\mathcal{A}^*$ solves Eq. (6.1) and (6.3). If $c^* > m \times |\mathcal{T}| - 2 \times g \times \eta$, then $g > D(\mathcal{A}^*)$; if $c^* < m \times |\mathcal{T}| - 2 \times g \times \eta$, then $g < D(\mathcal{A}^*)$; if $c^* = m \times |\mathcal{T}| - 2 \times g \times \eta$, then $g = D(\mathcal{A}_s^*)$ and $D(\mathcal{A}_s^*) = D(\mathcal{A}^*)$.*

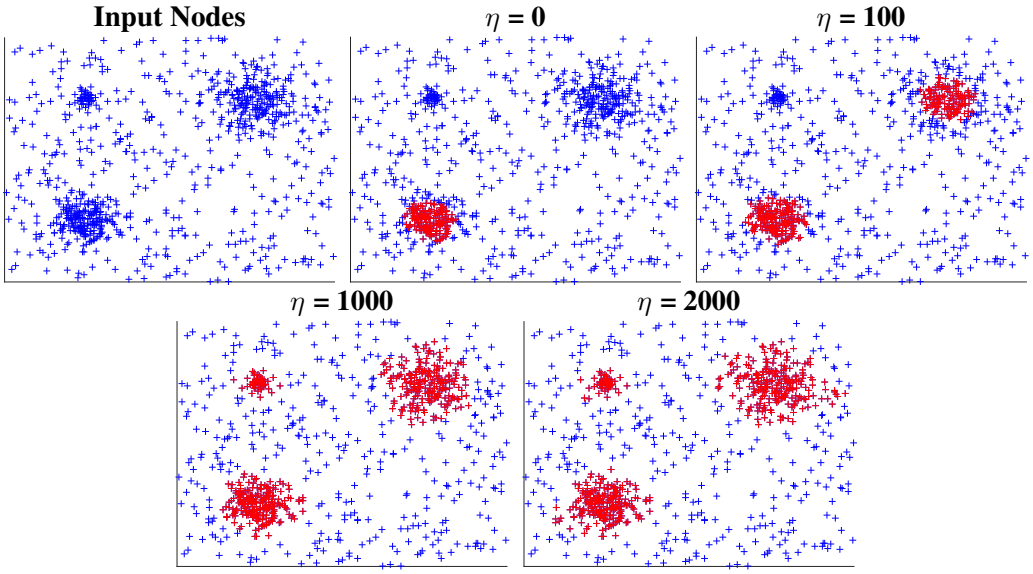


Fig. 6.4 Dense subgraph selection results in the simulation experiment. The red dots denote the selected nodes under different η settings.

Proof. Notice that if $c^* > m \times |\mathcal{T}| - 2{\times}g \times \eta$, then $g > D(\mathcal{A})\ \forall \mathcal{A} \subseteq \mathcal{T}$ based on Eq. (6.9) since c^* is the minimum cut capacity. We have g as an upper bound of the optimal density $D(\mathcal{A}^*)$. If $c^* < m \times |\mathcal{T}| - 2{\times}g \times \eta$, then $\exists \mathcal{A} \subseteq \mathcal{T}$ such that $g < D(\mathcal{A})$ based on Eq. (6.9) since c^* is the minimum cut capacity. We have g as an lower bound of the optimal density $D(\mathcal{A}^*)$. If $c^* = m \times |\mathcal{T}| - 2{\times}g \times \eta$, then $g{\geq}D(\mathcal{A})\ \forall \mathcal{A} \subseteq \mathcal{T}$ and $g = D(\mathcal{A}_s^*)$ based on Eq. (6.9) since c^* is the minimum cut capacity. Hence, $D(\mathcal{A}_s^*){\geq}D(\mathcal{A}) \forall \mathcal{A} \subseteq \mathcal{T}$ and $D(\mathcal{A}_s^*) = D(\mathcal{A}^*)$. □

Algorithm: Based on Theorem 6.2.1, we implement our binary search strategy to iteratively shrink the lower and upper bound of the optimal density $D(\mathcal{A}^*)$. However, since both our node and edge weights are continuous values, $D(\mathcal{A})$ is also continuous for $\mathcal{A} \subseteq \mathcal{T}$. We have to search infinitely to find $\mathcal{A}^*$. In practice, we can specify an error bound to stop the search. For example, let $\hat{\mathcal{A}}$ be our candidate solution, u and l be the current upper and lower bound. If a solution satisfying $\frac{D(\mathcal{A}^*)-D(\hat{\mathcal{A}})}{D(\mathcal{A}^*)} \leq \beta$ is good enough, we can safely stop the search when $\frac{u-l}{l} \leq \beta$. A tighter bound needs more iterations.

The entire algorithm is summarized in Algorithm 3. Note that, we update the candidate subgraph $\hat{\mathcal{A}}$ to $\mathcal{A}_s^*$ when g is determined to be the new lower bound because $D(\mathcal{A}_s^*)$ is greater than the new lower bound g and smaller than the current upper bound u. This is not true when g is the upper bound as $D(\mathcal{A}_s^*)$ may be smaller than the current lower bound. The initial lower bound is set to the maximum η-density of any 2-node subgraph of $\mathbb{G}$ to avoid zero lower bound, and the initial upper bound is set to the summation of all the node and edge weights in the graph. A loose bound like this does not affect the efficiency of our algorithm as binary search shrinks the bounds exponentially. Let U denote the initial upper bound in Algorithm 3, we need to perform $O(log(\frac{U}{\beta \times D(\mathcal{A}^*)}))$ graph cut operations throughout the algorithm. It is logarithmic in terms of the problem size.

Algorithm 3 Maximum η-Density Optimization

Input: Graph $G = (\mathcal{T}, \mathcal{E})$, error bound β
Output: Subgraph $\hat{\mathcal{A}} \subseteq \mathcal{T}$ achieving the maximum η-density defined by Eq. (6.3) within the given error bound
$l \leftarrow \frac{\max_{t_i^j \in \mathcal{T}, t_p^q \in \mathcal{T}, t_i^j \neq t_p^q} s_i^j + s_p^q + w(t_i^j, t_p^q)}{\eta + 2}$
$u \leftarrow \sum_{t_i^j \in \mathcal{T}} s_i^j + \sum_{(t_i^j, t_p^q) \in \mathcal{E}} w(t_i^j, t_p^q)$
$\hat{\mathcal{A}} \leftarrow$ the two-node subgraph achieving l
while $\frac{u-l}{l} > \beta$ **do**
    $g \leftarrow \frac{u+l}{2}$
    Find c^*, $\mathcal{A}_s^*$ and $\mathcal{A}_t^*$ in Eq. (6.10) by max flow [12]
    if $c^* > m \times |\mathcal{T}| - 2 \times g \times \eta$ **then**
        $u \leftarrow g$
    else if $c^* < m \times |\mathcal{T}| - 2 \times g \times \eta$ **then**
        $\hat{\mathcal{A}} \leftarrow \mathcal{A}_s^*$
        $l \leftarrow g$
    else
        $\hat{\mathcal{A}} \leftarrow \mathcal{A}_s^*$
        break
    end if
end while

Simulation Experiment: In order to visualize the effectiveness of the proposed density maximization approach, a test experiment is performed

on simulated 2D data points. These data points are drawn from three 2D Gaussian distributions and one 2D uniform distribution, as shown in the first plot of Fig. 6.4. In this simulation experiment, the points' unary scores are set to zero as they are difficult to visualize. The affinity graph is constructed in the same way as described in Section 6.2.1. The selection results using the proposed method are shown in the subsequent plots of Fig. 6.4. It can be seen that the selection is quite conservative when $\eta = 0$. It completely misses the two dense modes at the top as well as the outer region of the mode at the bottom. A larger η relaxes the strict average regularization and the algorithm selects all three modes. It is worth noting that the selection does not change much when we increase η from 1000 to 2000, which indicates the proposed method is not particularly sensitive to η at this range.

6.2.3 Action Proposal Generation and Description

Compared with generic video object proposals [45, 95, 122] or motion-based action proposals [124], human prior has shown to be a much more accurate cue to detect human actions [60, 134, 144]. In this work, we also start with per-frame human detections to build spatiotemporal action proposals. The Faster R-CNN (VGG16) object detection framework [103] is used here for its good performance. However, a critical drawback of human detection is that it usually misses the humans undergoing severe pose variations or occlusions while performing an action. Hence, to improve the recall, we fuse per-frame detection results of two Faster-RCNN models. One is trained on the VOC2007 [25] train-validation subset containing daily human photos with moderate pose variation, and the other is trained on the MPII human activity dataset [134] containing human photos with large pose variations. Spatiotemporal action detection proposals are then generated by linking the per-frame human

detections using the method proposed in [144]. Furthermore, since the linking process does not produce new human detections, we also apply tracking to generate more spatiotemporal action proposals to improve the recall [134].

After extracting spatiotemporal action proposals, we break each proposal tube into a sequence of tubelets with 16 frames long and 8 frames overlap. These tubelets are then described by the 4096-dimensional fc-6 activations of the C3D network [118] trained on Sport-1M dataset [59]. A proposal tube's feature vector is computed as the average of all its tubelets' C3D features followed by ℓ_2 normalization. PCA is also used to reduce the feature dimensions to 512.

It is worth noting that, although the proposed method is applicable to any action features for the description of spatiotemporal action tubes, the feature selection is still important as we have no prior knowledge about the actions we want to discover. There are two options to choose effective features. The first option is to hand-craft some features to capture the low-level human motion instead of high-level semantic information, e.g., the traditional dense trajectory features [127]. Another option is to use huge action dataset to train the features. The assumption is that the huge action dataset can cover most of the possible human actions. In this work, we choose the second option, i.e., the C3D feature [118] trained on 1 million action videos, due to its superior performance in the action detection literature. Our experiment shows that this feature can well adapt to the datasets we have tested although they are different from the 1 million training videos.

6.3 Experiments

6.3.1 Datasets

To evaluate the proposed common action co-localization method in unconstrained scenarios, we first build two datasets. In both datasets, some videos contain one or multiple common actions, while some videos contain no common actions, *i.e.*, outlier videos. Many outlier videos also contain actions but they are not common in the dataset.

The first dataset is the *UCF Sports Plus* dataset. It includes all the 150 videos (10 action classes) in the *UCF Sports Action* [104] dataset as common actions. We re-annotate the videos containing multiple common actions to include them all. We also add 70 outlier videos containing no common actions. The second dataset is the *SVW Mini* dataset. It includes the annotated bowling and golfing videos in the SVW (Sports Action in the Wild) [106] dataset as common actions. Besides adding the 70 outlier videos in the *UCF Sports Plus* dataset, we also pick one video from each of the rest action classes in the SVW dataset as additional outlier videos. In total, there are 216 videos in this dataset, 120 of which contain common actions. Some example frames in these two datasets are shown in Fig. 6.5.

6.3.2 Evaluation Criteria

Similar to previous video object co-localization works [48, 54, 63], our co-localization method returns a ranked list of localizations for each video. A ground truth is recalled if its intersection over union (IOU) ratio with a localization is greater than a threshold. Most previous works have evaluated the co-localization performance separately for each video, *i.e.*, flag a video as correctly localized if the common object in the video is covered by the

Fig. 6.5 Sample frames in the newly proposed *UCF Sports Plus* (left two columns) and *SVW Mini* (right two columns) datasets. The bounding boxes denote the common action annotations.

top 1 detection. While this metric is meaningful in the constrained case where each video contains exactly one common object, it is not suitable for our unconstrained action co-localization scenario where a video may contain zero or multiple common actions. Indeed, this evaluation criterion implicitly excludes all the outlier videos containing no common actions. Thus, in this work, we use a different evaluation criterion to test the performance. We first put all the detections, including outlier videos, in a single ranked listed, and then compute the precision-recall curve and average precision to evaluate the localization performance. Note that, a ground truth common action can only be recalled once and all subsequent detections are treated as false positives.

6.3.3 Comparison with Baselines

In this section, we compare with several baselines and other subgraph selection methods to validate the advantage of the proposed co-localization method. In

Table 6.1 The average precisions of the proposed method as well as several baselines on the action co-localization task.

	UCF Sports Plus		*SVW Mini*
	IOU = 0.5	IOU = 0.25	IOU = 0.25
random	8.67%	21.73%	7.78%
original	31.27%	35.62%	25.60%
$\eta = 0$ [36]	5.88%	7.41%	0.83%
graph cut [8]	31.28%	35.63%	28.83%
k-means	32.24%	37.99%	33.83%
CSGM [155]	31.38%	35.82%	25.68%
DomSet [98]	41.79%	49.65%	46.75%
ours	**50.29%**	**58.49%**	**48.17%**
w/o outliers	71.23%	80.41%	71.30%

the following description, we use K to denote the actual number of common action classes in the dataset.

- Select all proposals and assign them random scores.
- Select all proposals and use their original scores.
- Set $\eta = 0$, *i.e.*, the average degree subgraph density formulation proposed in [36].
- Use the graph cut based subgraph selection formulation proposed in [8].
- Use the Cohesive Subgraph Mining (CSGM) formulation proposed in [155].
- Use K-Means to cluster the proposals into $K + 1$ clusters, and remove the cluster with the most number of outlier tubes.
- Use the Dominant Set clustering method with the proposed peeling off strategy in [98] to cluster the proposals and remove those un-clustered proposals.
- Select all proposals after excluding the videos containing no common actions.

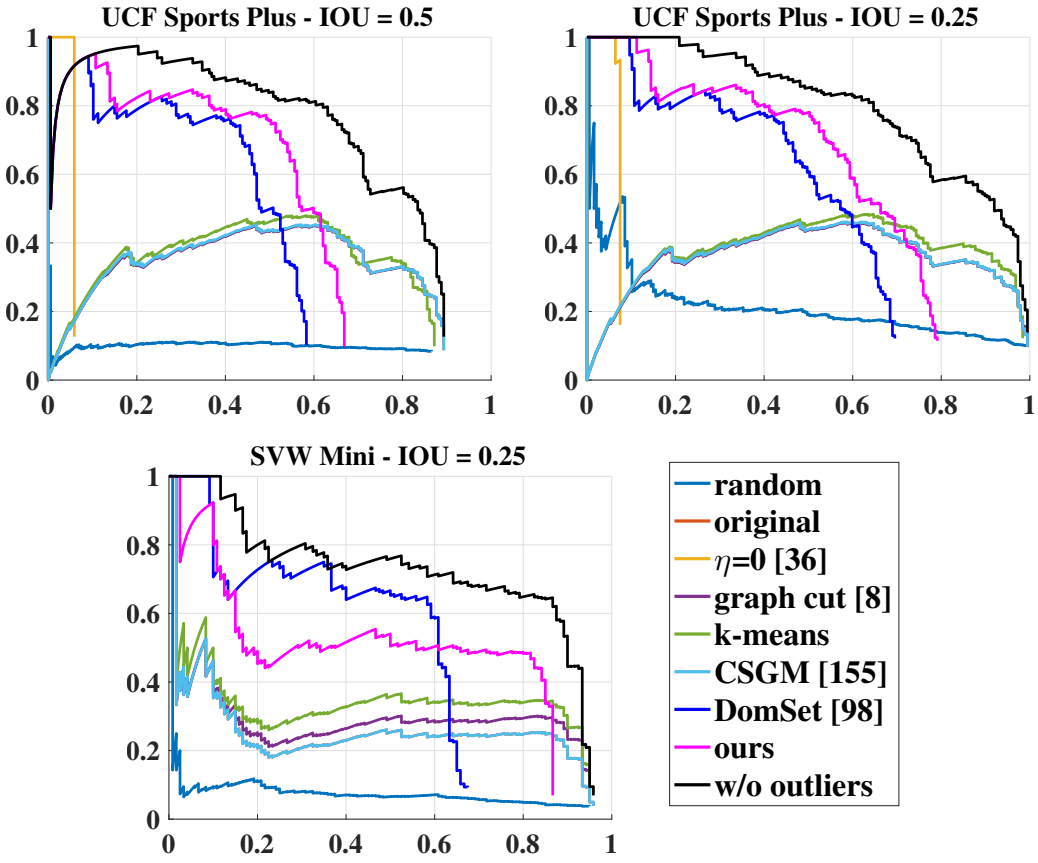


Fig. 6.6 Precision recall curves of the proposed method as well as several baselines on the action co-localization task. The curve for "original" is not quite visible because it largely overlaps with the "graph cut" or "CSGM" curve.

Table 6.2 Clustering accuracies (F-Measures) on the original proposals and our selected proposals on the *UCF Sports Plus* dataset.

	dive	golf	kick	lift	horse ride	run	skateboard	swing	angle swing	walk	average
original	0.55	0.048	0.22	0.19	0.39	0.35	0.036	0.57	0.35	0.26	0.29
ours	0.62	0.074	0.11	0.50	0.46	0.38	0.053	0.67	0.46	0.33	0.37
w/o outlier videos	0.61	0.20	0.26	0.36	0.63	0.39	0.014	0.62	0.42	0.38	0.39
w/o outlier tubes	0.92	0.32	0.52	0.78	0.82	0.60	0.33	0.87	0.90	0.53	0.66

Table 6.3 Clustering accuracies (F-Measures) on the original proposals and our selected proposals on the *SVW Mini* dataset.

	bowling	golf	average
original	0.20	0.15	0.17
ours	0.43	0.22	0.32
w/o outlier videos	0.34	0.25	0.30
w/o outlier tubes	0.96	0.97	0.97

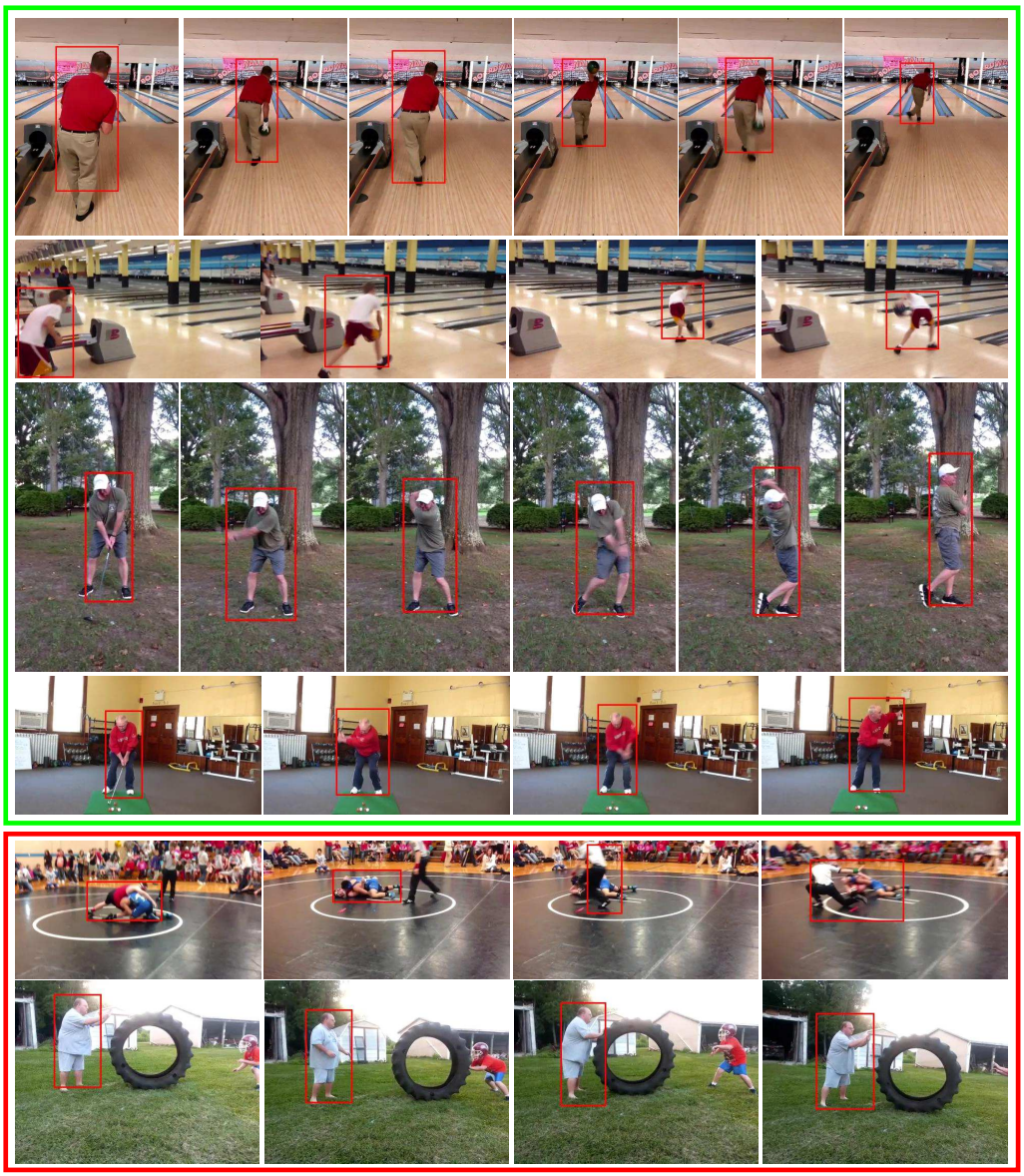


Fig. 6.7 Although all the shown proposals capture valid human actions, some are selected (top four rows) by our method as they contain common actions in the dataset, while some are rejected (bottom two rows) as they contain non-common actions.

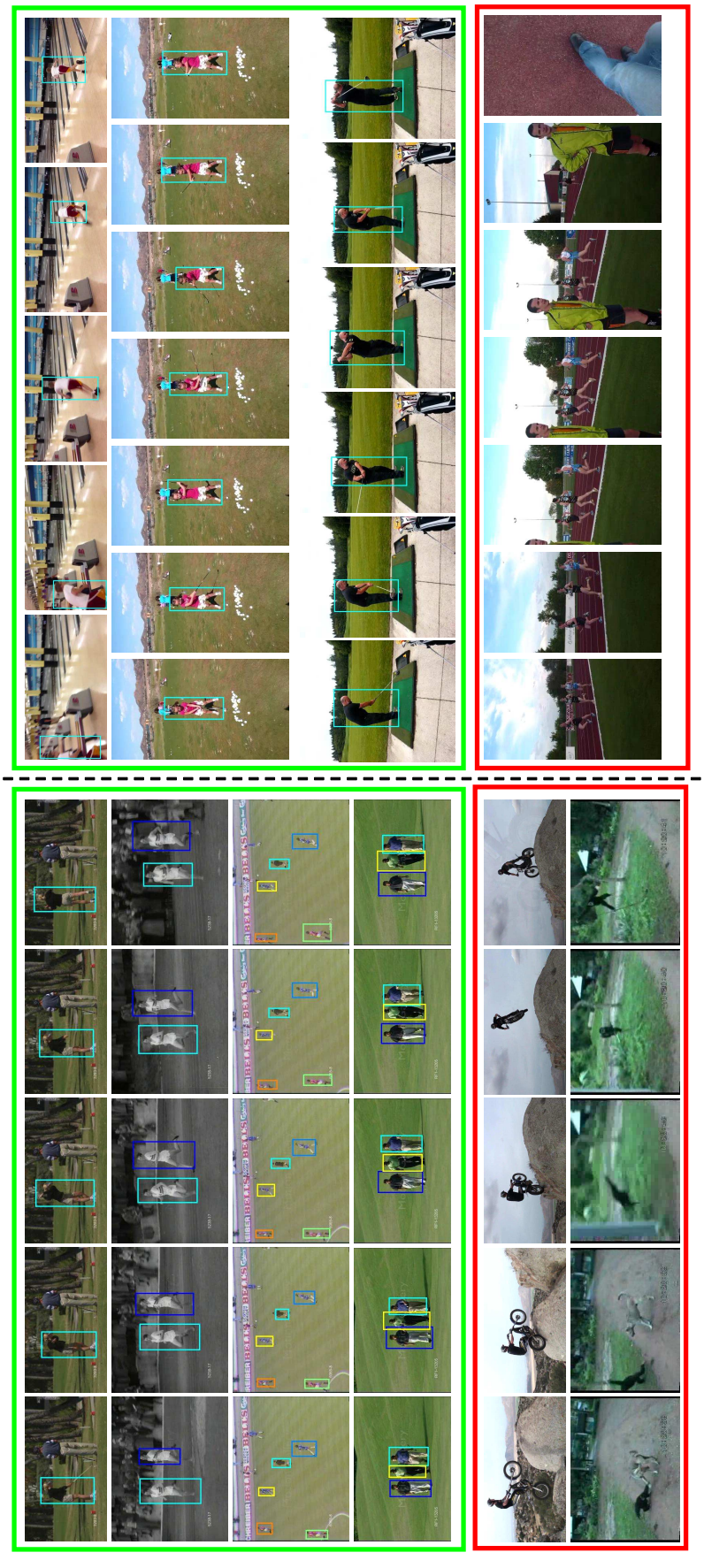


Fig. 6.8 Our action co-localization results on the *UCF Sports Plus* (left) and *SVW Mini* (right) datasets. The top rows in the green blocks contain videos with common actions. The bottom rows in the red blocks contain videos that do not have common actions. It can be seen that, the proposed method can handle the cases when there are zero, one or multiple common actions in the videos.

The precision-recall curves of these methods are shown in Fig. 6.6, and the average precisions are shown in Table 6.1. We use a lower threshold, *i.e.*, 0.25, for *SVW Mini* as its ground truth annotation is loose, as shown in Fig. 6.5. Note that, if we manually exclude the outlier videos, *i.e.*, the method "w/o outliers", we can achieve a high average precision just using the original proposals. This demonstrates the good quality of the initial action proposals. After adding outlier videos, *i.e.*, the method "original", the average precision drops significantly. This shows the importance of the proposed action co-localization problem in unconstrained scenarios. The proposed method successfully selects the proposals containing common actions and improves the average precision by more than 20% percent in all the datasets and IOU settings. It is worth noting that, when $\eta = 0$, the average precision is even lower than the random case but the initial precision is almost perfect in the first two precision-recall curves. The lower average precision is due to the extremely low recall as the selection is too strict. This further demonstrates the importance of a relaxed regularization. The graph cut based formulation [8] has only marginally improved the original baseline. This is because the graph cut formulation only minimizes the edge weights between selected and unselected nodes, while does not enforce strong edges within the selected nodes. Hence, this formulation is useful only when the node weights are mostly reliable, which is not true in our case due to the existence of "non-common" actions. The method "K-Means" does not perform well because it assumes the outlier proposals can also form a compact cluster. The "CSGM" method is not performing well as their formulation can only be solved approximately [155]. The method "DomSet" is the best among the compared methods but still outperformed by ours, especially on the *UCF Sports Plus* dataset where its PR curve is consistently lower than

ours. On *SVW Mini*, it has a better initial precision but the final recall is lower. In addition, "DomSet" is much slower than ours in our experiment.

Some qualitative results are shown in Fig. 6.7 to demonstrate how our method successfully selects proposals containing common actions and rejects proposals containing uncommon actions. We also show some actual action co-localization results in Fig. 6.8. It is worth noting that, in the top left row of Fig. 6.8, there are actually two humans. Indeed, the proposal highlighting the right person has a higher proposal confidence score, but our method selects the left person as there are many golfing actions in this dataset.

To further demonstrate the effectiveness of the proposed co-localization approach. We perform K-Means to cluster the selected proposals into $K+1$ clusters. We assign each cluster an action label based on majority voting and compute the precision, recall, and f-measure of each action class. The comparisons before and after the node selection are shown in Table 6.2 and 6.3. The method "w/o outlier videos" means we manually exclude the videos containing no common actions. The method "w/o outlier tubes" means we manually exclude the proposal tubes capturing no common actions. It can be seen that the proposed method apparently improves the baseline and are comparable to the method "w/o outlier videos".

In order to show our method can also generalize to larger video sets. We perform extra experiments on the JHMDB [49] and UCF101 (test set of the 24-class detection subset) [110] datasets with added outlier videos. In total, there are 1010 videos (45K frames), and 997 videos (200K frames), respectively. The mAP for the input proposals, [36] and our method are 64.83%, 1.94%, 69.00% for JHMDB with an IOU threshold of 0.5, and 26.07%, 3.43%, 37.92% for UCF101 with an IOU threshold of 0.25. This shows that our method can improve the baselines in these larger sets.

Table 6.4 Comparison with [48] using correct detection ratio metric.

	[48]	ours
UCF Sports Plus (IoU=0.5)	31.85%	49.20%

6.3.4 Comparison with Video Object Co-localization

In this section, we compare with [48] to show that it is non-trivial to apply existing video object co-localization methods to common action co-localizations. In [48], it generates one localization for each video (including outlier videos) because it assumes each video contains exactly one common object and the detection scores between different videos are not comparable. For a fair comparison, we use [48]'s correct detection ratio metric, exclude all outlier videos and only use the top 1 detection of our method in each video. The comparison results are shown in Table 6.4. It can be seen that our method produces more accurate localization results. Furthermore, [48] cannot identify outlier videos due to their assumption.

6.3.5 Running Time

The detailed running time of the proposed co-localization approach on the *UCF Sports Plus* dataset is shown in Table 6.5. For this dataset, there are 23013 frames in total and the affinity graph contains 2142 nodes and 41102 edges. Furthermore, for the larger UCF101 dataset with 200K frames, affinity graph construction and common proposal selection take 138 and 1.1 seconds, respectively. It can be seen that most of the time is spent on the proposal generation and feature extraction steps, while the proposed optimal subgraph selection algorithm is efficient.

Table 6.5 The running time of each step in the proposed co-localization approach on a dataset containing 23013 frames.

	time	% of total time
human detection	200 min	48.78%
action proposal generation	130 min	31.70%
feature extraction + PCA	79 min	19.27%
affinity graph construction	6.1 sec	0.025%
subgraph selection	1.2 sec	0.0049%
total	410 min	-

6.3.6 Limitations and Future Work

It is worth noting that, due to human prior, the proposed generation method we use cannot handle untrimmed case where a human performs common action only during part of his/her presence. However, the proposed common action selection method still has great potential of handling untrimmed videos as long as we can obtain reasonable action proposals. In the future, we will consider proposing more robust action proposals to handle untrimmed videos. We will also explore the potential of our selection method to directly refine temporally untrimmed proposals.

Furthermore, in this study, we have only explored the discovery and localization of human actions, but the proposed method is general enough to also discover and locate non-human actions. To maintain good performance, two modifications are suggested for non-human action discovery and localization. Firstly, the initial human detector should be replaced by the corresponding object detector to generate appropriate action detection proposals. Secondly, the features to describe the action detection proposals should be adjusted to better describe the actions of the interested object. For example, the CNN features we have used are trained on 1 million sports videos since we are only interested in human actions, but, it may not be suitable to describe other objects' actions.

In addition, the main parameters of the proposed method are the weight adjustment term λ and regularization adjustment term η. These two parameters are in spirit similar to the number of clusters K in the commonly used clustering algorithms such as k-means or spectral clustering. Hence, unfortunately, it is difficult to have the same parameter setting for different datasets, and in the current experiment, we have used different λ and η values for different datasets. It is worth noting that, to ensure a fair comparison with the methods shown in Table 6.1 and Fig. 6.6, we also used different parameter settings in these methods for different datasets. In the future, we will explore how to automatically choose the parameters for different datasets.

6.4 Conclusion

In this work, we tackle the problem of automatic common action discovery and localization in unconstrained videos. We are unaware of which types of action are common, and each video may contain zero, one or several common action instances. In the proposed method, we first generate action proposals and then select the proposals containing common actions by solving a subgraph density maximization problem. A polynomial time algorithm is also proposed to solve it. The evaluation results on several datasets demonstrate the effectiveness of the proposed method.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

A systematic study has been conducted in this thesis to tackle the problem of automatic thematic object discovery and localization in videos. It is of great importance to many real-world applications such as video summarization, search, editing, and labeling. However, despite the recent studies, this problem remains to be unsolved. Most of the challenges come from the fundamental fact that there is no initialization or prior knowledge about the thematic objects in the videos. In this thesis, we have studied the utilization of various unsupervised cues to tackle this problem, such as per-frame spatiotemporal saliency, temporal smoothness of spatial location and appearance, and global appearance consistency.

For the spatiotemporal saliency estimation, we propose two methods to estimate motion saliency in Chapter 3. The first method highlights the motion of the thematic object by removing camera motion from the optical flow fields. The second method resorts to the global motion contrast of local pixels to find salient motions. Considering the strong complementation between different saliency estimations, Chapter 3 presents a learning-based

adaptive saliency fusion scheme that can effectively fuse different saliency estimations to improve the overall performance. Furthermore, in special cases where we can narrow down the domain of the thematic objects to be discovered, semantic information can help to provide more accurate initialization compared with general saliency estimation. For example, we can use the semantic information of humans if we are trying to discover and localize human actions. In Chapter 6, we employ human detectors for the thematic action discovery and localization in video collections. We first use a human detector to provide candidate human detections on each video frame and then produce spatiotemporal human detection proposals by linking and tracking the per-frame detections.

Without prior knowledge or initialization, the per-frame detections are usually noisy. By assuming that the object always moves or changes smoothly from frame to frame, temporal smoothness is an important cue to suppress the false positives and recover missed detections. For instance, a detection is likely to be a false positive if most of the nearby frames do not have detections at similar locations. To utilize temporal smoothness, Chapter 3 presents a method to locate the thematic object by finding temporally smooth bounding box trajectories capturing highly salient regions. The optimal path can be found using the efficient max-path search algorithm in [119]. In Chapter 5, we also propose an online version of this search algorithm to improve the per-frame image object proposals in videos. Besides temporal smoothness of spatial locations, we also incorporate the temporal smoothness of appearance variation in the formulation. Furthermore, for the problem of thematic object segmentation studied in Chapter 4, we impose the temporal smoothness of spatial location and appearance by modeling the spatiotemporal proximity and appearance similarity between superpixels as edge scores in the superpixel graph.

Besides local saliency and temporal smoothness, appearance consistency is also an important cue to accurately discover and localize thematic objects in videos. For thematic video object discovery in single videos, it ensures the appearance coherency of each localization. For the thematic video object discovery in collections of videos, it also enforces the appearance or motion pattern similarity among the localizations from different videos. Chapter 3 presents an iterative appearance modeling technique for thematic object discovery and localization in single videos. In each iteration, we use image patches inside and outside the current localization as positive and negative training samples, respectively, to train a linear SVM model. The trained model is then used to refine the localization for the next iteration. Chapter 4 presents a non-iterative appearance modeling technique for thematic video object segmentation in the Markov random field segmentation framework. It constrains the appearance coherency and disparity within and between the thematic object and background regions, respectively, by adding auxiliary nodes to the original superpixel graph. Chapter 6 presents a method using the motion pattern consistency among the action detection proposals to select the proposals containing thematic (common) actions in a video collection. It builds an affinity graph connecting action proposals and models the semantic similarity between proposals as edge scores. A subset of action proposals with similar motion patterns is then discovered by solving a maximum density subgraph selection problem.

7.2 Future Work

Despite the active studies in this field, there are still many open problems. One problem is that, in the thematic object discovery and localization, most of the existing works have assumed that the whole video is available

before processing. However, in many real-world applications, the videos are only available in a streaming manner, *e.g.*, real-time video surveillance. It is of great importance to study how to discover and localize thematic objects in streaming videos. Another problem is that the discovery of the thematic object in video collections requires simultaneous processing of all the videos, and the development of algorithms that are scalable to web-scale video datasets, *e.g.*, Youtube Video Collection, are in demand. Furthermore, with the popularization of wearable cameras, *e.g.*, Google Glass or Go Pro, in recent years, egocentric videos have emerged in the computer vision research community. Discovering thematic objects in egocentric videos provides effective tools for the study of the behavior and interest of the wearer. However, egocentric videos are different from traditional ones. For instance, egocentric videos usually contain significant motion distortion and fast motion due to head movement.

Author's Publications

- **Jiong Yang**, Junsong Yuan, "Temporally Enhanced Image Object Proposals for Online Video Object and Action Detections", Journal of Visual Communication and Image Representation (**JVCI**), Accepted with Minor Revision, 2017
- **Jiong Yang**, Junsong Yuan, "Common Action Discovery and Localization in Unconstrained Videos", IEEE International Conf. Computer Vision (**ICCV**), 2017
- **Jiong Yang**, Junsong Yuan, "Temporally Enhanced Image Object Proposals for Video", IEEE International Conf. on Multimedia and Expo (**ICME**), 2017
- **Jiong Yang**, Brian Price, Xiaohui Shen, Zhe Lin, Junsong Yuan, "Fast Appearance Modeling for Automatic Primary Video Object Segmentation", IEEE Trans. on Image Processing (**T-IP**), 2016
- **Jiong Yang**, Gangqiang Zhao, Junsong Yuan, Xiaohui Shen, Zhe Lin, Brian Price, Jonathan Brandt, "Discovering Primary Objects in Videos by Saliency Fusion and Iterative Appearance Estimation", IEEE Trans. on Circuits and Systems for Video Technology (**T-CSVT**), 2016

- Jingjing Meng, Junsong Yuan, **Jiong Yang**, Gang Wang, Yap-Peng Tan, "Object Instance Search in Videos via Spatio-Temporal Trajectory Discovery", IEEE Trans. on Multimedia (**T-MM**), 2016

- Gangqiang Zhao, Junsong Yuan, Gang Hua, and **Jiong Yang**, "Topical Video Object Discovery from Key Frames by Modeling Word Co-occurrence Prior", IEEE Trans. on Image Processing (**T-IP**), 2016

- Kang Dang, **Jiong Yang**, and Junsong Yuan, "Adaptive Exponential Smoothing for Online Filtering of Pixel Prediction Maps", IEEE International Conf. on Computer Vision (**ICCV**), 2015

References

[1] Achanta, R., Hemami, S., Estrada, F., and Susstrunk, S. (2009). Frequency-tuned salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604. IEEE.

[2] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Aand Machine Intelligence*, 34(11):2274–2282.

[3] Alexe, B., Deselaers, T., and Ferrari, V. (2012). Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202.

[4] Badrinarayanan, V., Galasso, F., and Cipolla, R. (2010). Label propagation in video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3265–3272. IEEE.

[5] Bai, X., Wang, J., Simons, D., and Sapiro, G. (2009). Video snapcut: Robust video object cutout using localized classifiers. *ACM Transactions on Graphics*, 28(3):70.

[6] Banica, D., Agape, A., Ion, A., and Sminchisescu, C. (2013). Video object segmentation by salient segment chain composition. In *Proceedings of the International Conference on Computer Vision*, pages 283–290. IEEE.

[7] Bao, L., Yang, Q., and Jin, H. (2010). Fast edge-preserving patchmatch for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Patter Recognition*, pages 2141–2148. IEEE.

[8] Bhattacharjee, S. D., Yuan, J., Tan, Y.-P., and Duan, L.-Y. (2016). Query-adaptive small object search using object proposals and shape-aware descriptors. *IEEE Transactions on Multimedia*, 18(4):726–737.

[9] Borji, A., Cheng, M.-M., Jiang, H., and Li, J. (2015). Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722.

[10] Borji, A. and Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207.

[11] Borji, A., Sihite, D. N., and Itti, L. (2013). Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1):55–69.

[12] Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.

[13] Chen, X., Ma, H., Wang, X., and Zhao, Z. (2015). Improving object proposals with multi-thresholding straddling expansion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2587–2595. IEEE.

[14] Cheng, M.-M., Mitra, N. J., Huang, X., and Hu, S.-M. (2014). Salientshape: Group saliency in image collections. *The Visual Computer*, 30(4):443–453.

[15] Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H. S., and Hu, S.-M. (2015). Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582.

[16] Cheng, M.-M., Warrell, J., Lin, W.-Y., Zheng, S., Vineet, V., and Crook, N. (2013). Efficient salient region detection with soft image abstraction. In *Proceedings of the IEEE International Conference on Computer vision*, pages 1529–1536.

[17] Cho, M., Kwak, S., Schmid, C., and Ponce, J. (2015). Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1201–1210. IEEE.

[18] Chu, W.-S., Zhou, F., and De la Torre, F. (2012). Unsupervised temporal commonality discovery. In *Proceedings of the European Conference on Computer Vision*, pages 373–387. Springer.

[19] Chu, W.-T. and Tsai, M.-H. (2012). Visual pattern discovery for architecture image classification and product image search. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, page 27. ACM.

[20] Del Pero, L., Ricco, S., Sukthankar, R., and Ferrari, V. (2015). Articulated motion discovery using pairs of trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2151–2160. IEEE.

[21] Del Pero, L., Ricco, S., Sukthankar, R., and Ferrari, V. (2016a). Behavior discovery and alignment of articulated object classes from unstructured video. *International Journal on Computer Vision*, pages 1–23.

[22] Del Pero, L., Ricco, S., Sukthankar, R., and Ferrari, V. (2016b). Discovering the physical parts of an articulated oclass from multiple videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 714–723.

[23] Dollár, P. and Zitnick, C. L. (2015). Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570.

[24] Endres, I. and Hoiem, D. (2010). Category independent object proposals. In *Proceedings of the European Conference on Computer Vision*, pages 575–588. Springer.

[25] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2013). (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[26] Fang, Y., Lin, W., Chen, Z., Tsai, C.-M., and Lin, C.-W. (2014). A video saliency detection model in compressed domain. *IEEE transactions on circuits and systems for video technology*, 24(1):27–38.

[27] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

[28] Fragkiadaki, K., Arbeláez, P., Felsen, P., and Malik, J. (2015). Learning to segment moving objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4083–4090. IEEE.

[29] Fu, H., Xu, D., Zhang, B., and Lin, S. (2014). Object-based multiple foreground video co-segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173. IEEE.

[30] Fu, H., Xu, D., Zhang, B., Lin, S., and Ward, R. K. (2015). Object-based multiple foreground video co-segmentation via multi-state selection graph. *T-IP*, 24(11):3415–3424.

[31] Fukuchi, K., Miyazato, K., Kimura, A., Takagi, S., and Yamato, J. (2009). Saliency-based video segmentation with graph cuts and sequentially updated priors. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 638–641. IEEE.

[32] Furnari, A., Farinella, G. M., and Battiato, S. (2014). An experimental analysis of saliency detection with respect to three saliency levels. In *ECCV Workshops (3)*, pages 806–821.

[33] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Confernece on Computer Vision*, pages 1440–1448. IEEE.

[34] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587. IEEE.

[35] Goferman, S., Zelnik-Manor, L., and Tal, A. (2012). Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926.

[36] Goldberg, A. V. (1984). Finding a maximum density subgraph. Technical report, University of California at Berkeley, Berkeley, CA, USA.

[37] Gong, M. (2011). Foreground segmentation of live videos using locally competing svms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2105–2112. IEEE.

[38] Guo, J., Li, Z., Cheong, L.-F., and Zhiying Zhou, S. (2013). Video co-segmentation for meaningful action extraction. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2232–2239.

[39] Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.-M., Hicks, S. L., and Torr, P. H. (2016). Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2096–2109.

[40] Harris, R. (2017). Improving our brand safety controls. https://www.blog.google/topics/google-europe/improving-our-brand-safety-controls. Accessed: 2017-06-28.

[41] Hong, R., Tang, J., Tan, H.-K., Yan, S., Ngo, C., and Chua, T.-S. (2009). Event driven summarization for web videos. In *Proceedings of the SIGMM workshop on Social media*, pages 43–48. ACM.

[42] Hosang, J., Benenson, R., and Schiele, B. (2014). How good are detection proposals, really? In *Proceedings of the British Machine Vision Conference*. BMVA Press.

[43] Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

[44] Jain, M., Jegou, H., and Bouthemy, P. (2013). Better exploiting motion for better action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2555–2562.

[45] Jain, M., Van Gemert, J., Jégou, H., Bouthemy, P., and Snoek, C. G. (2014). Action localization with tubelets from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 740–747. IEEE.

[46] Jain, S. D. and Grauman, K. (2014). Supervoxel-consistent foreground propagation in video. In *Proceedings of the European Conference on Computer Vision*, pages 656–671. Springer.

[47] Jang, W.-D., Lee, C., and Kim, C.-S. (2016). Primary object segmentation in videos via alternate convex optimization of foreground and background distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 696–704.

[48] Jerripothula, K. R., Cai, J., and Yuan, J. (2016). Cats: Co-saliency activated tracklet selection for video co-localization. In *Proceedings of the European Conference on Computer Vision*, pages 187–202. Springer.

[49] Jhuang, H., Gall, J., Zuffi, S., Schmid, C., and Black, M. J. (2013). Towards understanding action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3192–3199. IEEE.

[50] Jiang, B., Zhang, L., Lu, H., Yang, C., and Yang, M.-H. (2013a). Saliency detection via absorbing markov chain. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1665–1672.

[51] Jiang, H., Wang, J., Yuan, Z., Liu, T., Zheng, N., and Li, S. (2011). Automatic salient object segmentation based on context and shape prior. In *Proceedings of the British Machine Vision Conference*, volume 6, page 9.

[52] Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., and Li, S. (2013b). Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2083–2090.

[53] Jiang, Q., Wu, Z., Tian, C., and Liu, T. (2015). Msr: A simple and effective metric for visual saliency map fusion. In *International Symposium on Computational Intelligence and Design*, volume 2, pages 432–435. IEEE.

[54] Joulin, A., Tang, K. D., and Li, F.-F. (2014). Efficient image and video co-localization with frank-wolfe algorithm. In *Proceedings of the European Conference on Computer Vision*, pages 253–268. Springer.

[55] Jun Koh, Y., Jang, W.-D., and Kim, C.-S. (2016). Pod: Discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1068–1076.

[56] Jung, C. and Kim, C. (2012). A unified spectral-domain approach for saliency detection and its application to automatic object segmentation. *IEEE Transactions on Image Processing*, 21(3):1272–1283.

[57] Kang, K., Li, H., Xiao, T., Ouyang, W., Yan, J., Liu, X., and Wang, X. (2017). Object detection in videos with tubelet proposal networks. In *Proc. CVPR*, volume 2, page 7.

[58] Kang, K., Ouyang, W., Li, H., and Wang, X. (2016). Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 817–825. IEEE.

[59] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732. IEEE.

[60] Kläser, A., Marszałek, M., Schmid, C., and Zisserman, A. (2010). Human focused action localization in video. In *Proceedings of the European Conference on Computer Vision*, pages 219–233. Springer.

[61] Kohli, P., Torr, P. H., et al. (2009). Robust higher order potentials for enforcing label consistency. *International Journal Computer Vision*, 82(3):302–324.

[62] Kuo, W., Hariharan, B., and Malik, J. (2015). Deepbox: Learning objectness with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2479–2487. IEEE.

[63] Kwak, S., Cho, M., Laptev, I., Ponce, J., and Schmid, C. (2015). Unsupervised object discovery and tracking in video collections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3173–3181. IEEE.

[64] Lee, Y. J., Kim, J., and Grauman, K. (2011). Key-segments for video object segmentation. In *Proceedings of the International Conference on Computer Vision*, pages 1995–2002. IEEE.

[65] Li, F., Kim, T., Humayun, A., Tsai, D., and Rehg, J. M. (2013a). Video segmentation by tracking many figure-ground segments. In *Proceedings of the International Conference on Computer Vision*, pages 2192–2199. IEEE.

[66] Li, W.-T., Chang, H.-S., Lien, K.-C., Chang, H.-T., and Wang, Y. (2013b). Exploring visual and motion saliency for automatic video object extraction. *IEEE Transactions on Image Processing*, 22(7):2600–2610.

[67] Li, X., Zhao, L., Wei, L., Yang, M.-H., Wu, F., Zhuang, Y., Ling, H., and Wang, J. (2016). Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing*, 25(8):3919–3930.

[68] Li, Y., Hou, X., Koch, C., Rehg, J. M., and Yuille, A. L. (2014). The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287.

[69] Li, Y., Sun, J., and Shum, H.-Y. (2005). Video object cut and paste. *ACM Transactions on Graphics*, 24(3):595–600.

[70] Liu, C. et al. (2009). *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology.

[71] Liu, D. and Chen, T. (2007). A topic-motion model for unsupervised video object discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

[72] Liu, D., Hua, G., and Chen, T. (2010). A hierarchical visual model for video object summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2178–2190.

[73] Liu, H. and Yan, S. (2010). Common visual pattern discovery via spatially coherent correspondences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1609–1616. IEEE.

[74] Liu, J. and Liu, Y. (2013). Grasp recurring patterns from a single view. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2003–2010. IEEE.

[75] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110.

[76] Lu, S., Mahadevan, V., and Vasconcelos, N. (2014). Learning optimal seeds for diffusion-based salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797.

[77] Luo, Y. and Yuan, J. (2013). Salient object detection in videos by optimal spatio-temporal path discovery. In *Proceedings of the ACM International Conference on Multimedia*, pages 509–512. ACM.

[78] Luo, Y., Yuan, J., and Lu, J. (2016). Finding spatio-temporal salient paths for video objects discovery. *Journal of Visual Communication and Image Representation*, 38:45–54.

[79] Luo, Y., Yuan, J., Xue, P., and Tian, Q. (2011a). Saliency density maximization for efficient visual objects discovery. *IEEE Transactions on Circuits and Systems for video Technology*, 21(12):1822–1834.

[80] Luo, Y., Yuan, J., Xue, P., and Tian, Q. (2011b). Salient region detection and its application to video retargeting. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1–6. IEEE.

[81] Luo, Y., Zhao, G., and Yuan, J. (2013). Thematic saliency detection using spatial-temporal context. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 347–353.

[82] Ma, T. and Latecki, L. J. (2012). Maximum weight cliques with mconstraints for vobject segmentation. In *Proceedings of IEEE International Conference on Computer Vision and Patter Recognition*, pages 670–677. IEEE.

[83] Mai, L. and Liu, F. (2014). Comparing salient object detection results without ground truth. In *Proceedings of the European Conference on Computer Vision*, pages 76–91. Springer.

[84] Mai, L., Niu, Y., and Liu, F. (2013). Saliency aggregation: A data-driven approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1131–1138. IEEE.

[85] Manerba, F., Benois-Pineau, J., Leonardi, R., and Mansencal, B. (2008). Multiple moving object detection for fast video content description in compressed domain. *EURASIP Journal on Advances in Signal Processing*, 2008:5.

[86] Marat, S., Phuoc, T. H., Granjon, L., Guyader, N., Pellerin, D., and Guérin-Dugué, A. (2009). Modelling spatio-temporal saliency to predict gaze direction for short videos. *International journal of computer vision*, 82(3):231.

[87] Margolin, R., Tal, A., and Zelnik-Manor, L. (2013). What makes a patch distinct? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1139–1146.

[88] Marian Puscas, M., Sangineto, E., Culibrk, D., and Sebe, N. (2015). Unsupervised tube extraction using transductive learning and dense trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1653–1661. IEEE.

[89] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767.

[90] Mei, T., Li, L., Hua, X.-S., and Li, S. (2012). Imagesense: Towards contextual image advertising. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 8(1):6.

[91] Meng, J., Wang, H., Yuan, J., and Tan, Y.-P. (2016). From keyframes to key objects: Video summarization by representative object proposal selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1039–1048. IEEE.

[92] Mettes, P., van Gemert, J. C., and Snoek, C. G. (2016). Spot on: Action localization from pointly-supervised proposals. In *Proceedings of the European Conference on Computer Vision*, pages 437–453. Springer.

[93] Mosabbeb, E. A., Cabral, R., De la Torre, F., and Fathy, M. (2014). Multi-label discriminative weakly-supervised human activity recognition and localization. In *Proceedings of Asian Conference on Computer Vision*, pages 241–258. Springer.

[94] Nguyen, T. V., Xu, M., Gao, G., Kankanhalli, M., Tian, Q., and Yan, S. (2013). Static saliency vs. dynamic saliency: a comparative study. In *Proceedings of the ACM International Conference on Multimedia*, pages 987–996. ACM.

[95] Oneata, D., Revaud, J., Verbeek, J., and Schmid, C. (2014). Spatio-temporal object detection proposals. In *Proceedings of the European Conference on Computer Vision*, pages 737–752. Springer.

[96] Papazoglou, A. and Ferrari, V. (2013). Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784. IEEE.

[97] Papoutsakis, K., Panagiotakis, C., and Argyros, A. A. (2017). Temporal action co-segmentation in 3d motion capture data and videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[98] Pavan, M. and Pelillo, M. (2007). Dominant sets and pairwise clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1).

[99] Perazzi, F., Krähenbühl, P., Pritch, Y., and Hornung, A. (2012). Saliency filters: Contrast based filtering for salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–740. IEEE.

[100] Pirsiavash, H., Ramanan, D., and Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1201–1208. IEEE.

[101] Prest, A., Leistner, C., Civera, J., Schmid, C., and Ferrari, V. (2012). Learning object class detectors from weakly annotated video. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3282–3289. IEEE.

[102] Price, B. L., Morse, B. S., and Cohen, S. (2009). Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *Proceedings of the International Conference on Computer Vision*, pages 779–786. IEEE.

[103] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99.

[104] Rodriguez, M. D., Ahmed, J., and Shah, M. (2008). Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8. IEEE.

[105] Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314.

[106] Safdarnejad, S. M., Liu, X., Udpa, L., Andrus, B., Wood, J., and Craven, D. (2015). Sports videos in the wild (svw): A video dataset for sports analysis. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, volume 1, pages 1–7. IEEE.

[107] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analyais and Machine Intelligence*, 22(8):888–905.

[108] Siva, P. and Xiang, T. (2011). Weakly supervised action detection. In *Proceedings of British Machine Vision Conference*, volume 2, page 6.

[109] Soomro, K., Idrees, H., and Shah, M. (2016). Predicting the where and what of actors and actions through online action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

[110] Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. Technical report, CRCV, UCF.

[111] Sun, J., Lu, H.-H., and Liu, X. (2015). Saliency region detection based on markov absorption probabilities. *IEEE Trans. Image Processing*, 24(5):1639–1649.

[112] Tan, H.-K. and Ngo, C.-W. (2009). Localized matching using earth mover's distance towards discovery of common patterns from small image samples. *Image and Vision Computing*, 27(10):1470–1483.

[113] Tang, K., Joulin, A., Li, L.-J., and Fei-Fei, L. (2014). Co-localization in real-world images. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1464–1471. IEEE.

[114] Tang, M., Gorelick, L., Veksler, O., and Boykov, Y. (2013). Grabcut in one cut. In *Proceedings on the IEEE International Conference on Computer Vision*, pages 1769–1776. IEEE.

[115] Tarashima, S., Irie, G., Tsutsuguchi, K., Arai, H., and Taniguchi, Y. (2013). Fast image/video collection summarization with local clustering. In *Proceedings of the ACM International Conference on Multimedia*, pages 725–728. ACM.

[116] Tighe, J. and Lazebnik, S. (2010). Superparsing: scalable nonparametric image parsing with superpixels. *Proceedings of the European Conference of Computer Vision*, pages 352–365.

[117] Toet, A. (2011). Computational versus psychophysical bottom-up image saliency: A comparative evaluation study. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2131–2146.

[118] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497. IEEE.

[119] Tran, D., Yuan, J., and Forsyth, D. (2014). Video event detection: From subvolume localization to spatiotemporal path search. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):404–416.

[120] Tsai, D., Flagg, M., Nakazawa, A., and Rehg, J. M. (2012). Motion coherent tracking using multi-label mrf optimization. *International Journal of Computer Vision*, 100(2):190–202.

[121] Tsai, Y.-H., Zhong, G., and Yang, M.-H. (2016). Semantic co-segmentation in videos. In *Proceedings of the European Conference on Computer Vision*, pages 760–775. Springer.

[122] Tu, Z., Guo, Z., Xie, W., Yan, M., Veltkamp, R. C., Li, B., and Yuan, J. (2017). Fusing disparate object signatures for salient object detection in video. *Pattern Recognition*.

[123] Uijlings, J. R., van de Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International Journal on Computer Vision*, 104(2):154–171.

[124] van Gemert, J. C., Jain, M., Gati, E., and Snoek, C. G. (2015). Apt: Action localization proposals from dense trajectories. In *Proceedings of the British Machine Vision Conference*, volume 2, page 4.

[125] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.

[126] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

[127] Wang, H. and Schmid, C. (2013a). Action recognition with improved trajectories. In *Proceedings of the IEEE international Conference on Computer Vision*, pages 3551–3558. IEEE.

[128] Wang, H. and Schmid, C. (2013b). Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia.

[129] Wang, L., Hua, G., Sukthankar, R., Xue, J., Niu, Z., and Zheng, N. (2016a). Video object discovery and co-segmentation with eweak supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[130] Wang, L., Hua, G., Sukthankar, R., Xue, J., and Zheng, N. (2014). Video object discovery and co-segmentation with extremely weak supervision. In *Proceedings of the European Conference on Computer Vision*, pages 640–655. Springer.

[131] Wang, L., Lu, H., Ruan, X., and Yang, M.-H. (2015). Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192.

[132] Wang, L., Wang, L., Lu, H., Zhang, P., and Ruan, X. (2016b). Saliency detection with recurrent fully convolutional networks. In *European Conference on Computer Vision*, pages 825–841. Springer.

[133] Wei, Y., Wen, F., Zhu, W., and Sun, J. (2012). Geodesic saliency using background priors. *Computer Vision–ECCV 2012*, pages 29–42.

[134] Weinzaepfel, P., Martin, X., and Schmid, C. (2016). Towards weakly-supervised action localization. *arXiv preprint arXiv:1605.05197*.

[135] Weinzaepfel, P., Martin, X., and Schmid, C. (2017). Human action localization with sparse spatial supervision.

[136] Wu, Y., Lim, J., and Yang, M.-H. (2013). Online object tracking: A benchmark. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2411–2418. IEEE.

[137] Xie, H., Gao, K., Zhang, Y., Li, J., and Ren, H. (2011). Common visual pattern discovery via graph matching. In *Proceedings of the 19th ACM international conference on multimedia*, pages 1385–1388. ACM.

[138] Yang, C., Zhang, L., Lu, H., Ruan, X., and Yang, M.-H. (2013). Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173.

[139] Yang, J., Price, B., Shen, X., Lin, Z., and Yuan, J. (2016a). Fast appearance modeling for automatic primary video object segmentation. *IEEE Transactions on Image Processing*, 25(2):503–515.

[140] Yang, J. and Yuan, J. (2017a). Common action discovery and localization in unconstrained videos. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE.

[141] Yang, J. and Yuan, J. (2017b). Temporally enhanced image object proposals for videos. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE.

[142] Yang, J., Zhao, G., Yuan, J., Shen, X., Lin, Z., Price, B., and Brandt, J. (2016b). Discovering primary objects in videos by saliency fusion and iterative appearance estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(6):1070–1083.

[143] Yeo, D., Han, B., and Han, J. H. (2016). Unsupervised co-activity detection from multiple videos using absorbing markov chain. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 3662–3668.

[144] Yu, G. and Yuan, J. (2015). Fast action proposals for human action detection and search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1302–1311. IEEE.

[145] Yu, G., Yuan, J., and Liu, Z. (2011). Unsupervised random forest indexing for fast action search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 865–872. IEEE.

[146] Yu, G., Yuan, J., and Liu, Z. (2013). Action search by example using randomized visual vocabularies. *IEEE Transactions on Image Processing*, 22(1):377–390.

[147] Yuan, J., Meng, J., Wu, Y., and Luo, J. (2008). Mining recurring events through forest growing. *T-CSVT*, 18(11):1597–1607.

[148] Yuan, J. and Wu, Y. (2007). Spatial random partition for common visual pattern discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8. IEEE.

[149] Yuan, J., Zhao, G., Fu, Y., Li, Z., Katsaggelos, A. K., and Wu, Y. (2012). Discovering thematic objects in image collections and videos. *IEEE Transactions on Image Processing*, 21(4):2207–2219.

[150] Zhai, Y. and Shah, M. (2006). Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 815–824. ACM.

[151] Zhang, D., Javed, O., and Shah, M. (2013). Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635. IEEE.

[152] Zhang, D., Javed, O., and Shah, M. (2014). Video object co-segmentation by regulated mweight cliques. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 551–566. Springer.

[153] Zhang, J. and Sclaroff, S. (2013). Saliency detection: A boolean map approach. In *Proceedings of the IEEE international conference on computer vision*, pages 153–160.

[154] Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., and Mech, R. (2015). Minimum barrier salient object detection at 80 fps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1404–1412.

[155] Zhao, G. and Yuan, J. (2011). Discovering thematic patterns in videos via cohesive sub-graph mining. In *Proceedings of the IEEE International Conference on Data Mining*, pages 1260–1265. IEEE.

[156] Zhao, G., Yuan, J., and Hua, G. (2013). Topical video object discovery from key frames by modeling word co-occurrence prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1602–1609. IEEE.

[157] Zhao, G., Yuan, J., Hua, G., and Yang, J. (2016). Topical video object discovery from key frames by modeling word co-occurrence prior. *IEEE Transactions Image Processing*, 24(12).

[158] Zhao, G., Yuan, J., Xu, J., and Wu, Y. (2011). Discovering the thematic object in commercial videos. *IEEE Transactions on MultiMedia*, 18(3):56–65.

[159] Zhao, R., Ouyang, W., Li, H., and Wang, X. (2015). Saliency detection by multi-context deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1265–1274.

[160] Zheng, L., Wang, S., Liu, Z., and Tian, Q. (2014). Packing and padding: Coupled multi-index for accurate image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Patter Recognition*, pages 1947–1954. IEEE.

[161] Zhou, X., Liu, Z., Sun, G., and Wang, X. (2016a). Adaptive saliency fusion based on quality assessment. *Multimedia Tools and Applications*, pages 1–25.

[162] Zhou, X., Liu, Z., Sun, G., Ye, L., and Wang, X. (2016b). Improving saliency detection via multiple kernel boosting and adaptive fusion. *IEEE Signal Processing Letters*, 23(4):517–521.

[163] Zhu, W., Liang, S., Wei, Y., and Sun, J. (2014). Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2814–2821.

[164] Zitnick, C. L. and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *Proceedings of the European Conference on Computer Vision*, pages 391–405. Springer.