Object Instance Search in Videos via Spatio-Temporal Trajectory Discovery

Jingjing Meng, *Member, IEEE*, Junsong Yuan, *Senior Member, IEEE*, Jiong Yang, Gang Wang, *Member, IEEE*, and Yap-Peng Tan, *Senior Member, IEEE*

***Abstract*—Given a specific object as query, object instance search aims to not only retrieve the images or frames that contain the query, but also locate all its occurrences. In this work, we explore the use of spatio-temporal cues to improve the quality of object instance search from videos. To this end, we formulate this problem as the spatio-temporal trajectory search problem, where a trajectory is a sequence of bounding boxes that locate the object instance in each frame. The goal is to find the top-K trajectories that are likely to contain the target object. Despite the large number of trajectory candidates, we build on a recent spatio-temporal search algorithm for event detection to efficiently find the optimal spatio-temporal trajectories in large video volumes, with complexity linear to the video volume size. We solve the key bottleneck in applying this approach to object instance search by leveraging a randomized approach to enable fast scoring of any bounding boxes in the video volume. In addition, we present a new dataset for video object instance search. Experimental results on a 73-hour video dataset demonstrate that our approach improves the performance of video object instance search and localization over the state-of-the-art search and tracking methods.**

***Index Terms*—Object instance search, spatio-temporal trajectory, video.**

I. Introduction

OBJECT instance search aims to search for and locate instances of a particular object in large image or video datasets. This problem arises from the needs for object-instance-level annotation and retrieval in big visual data, which are crucial for applications such as contextual advertising and embedded marketing in online videos [1]. Specifically, knowing exactly which frames and locations in these frames where a particular product appears in a video not only enables closer correlation of the content of an ad to the content of the video, but also allows for accurate measurements of the relevance of the product of interest in specific contexts by tracking users' interactions with it (e.g., click-through rate). Consequently, better models for allocating and pricing advertising inventory can be built to maximize revenues for both the content providers and the seller of the product.

Although much progress has been made in object instance search on large-scale image datasets in the past decade [2]–[11], little has been done on videos, which concerns about locating instances of the query object in the spatio-temporal video volume. Although named *Video Google*, [6] treats each video frame independently and returns a ranked list of keyframes or shots of a video that contain the object. The temporal consistency between the video frames is largely ignored except for when rejecting unstable regions across frames in a shot. Similarly, the TRECVID object instance search challenge searches for shots that contain a topic (i.e., object) without concerning about its location in each frame or how many frames in the shot contain the topic [12]–[15]. Therefore, in essence these approaches are still tailored for images.

In this work, we explore how to efficiently utilize the spatio-temporal cues to improve object instance search in 3D video volumes. Inspired by the bounding box search that locates object instances in images [7], [3], we propose to formulate object instance search in videos as spatio-temporal trajectory discovery problem, where each trajectory is a temporal series of bounding boxes that locate the object of interest across frames. Our goal is therefore to find the top-K trajectories that locate and track the object instances in videos. The trajectories are ranked by the summations of its bounding box scores.

As an object instance can appear in any frame and at any location, the number of trajectory candidates is huge. We build on the recent dynamic programming approach of Max-Path search [16] to locate the top-K trajectories efficiently in large video corpus, with complexity linear to the video volume size. However, Max-Path search requires a 3D trellis structure, which connects per frame bounding boxes that are scored prior to the spatio-temporal search. Such a trellis is usually built per frame by a sliding-window based scoring of all considered bounding boxes across different scales and aspect ratios, which is computationally expensive. While this may be less of a concern for detection problems, as real-time processing is usually good enough, it is prohibitive when searching tens of hours of videos. This bottleneck is the key challenge in adapting Max-Path search for detection problems to object instance search in large video volumes. We address this challenge by leveraging a randomized scheme to efficiently generate pixel-wise confidence scores. It is achieved by averaging the matching score of each pixel over a pool of randomly generated image patches, thus avoiding a sliding-window scanning of

Manuscript received July 22, 2015; revised October 08, 2015; accepted October 31, 2015. Date of publication November 13, 2015; date of current version December 14, 2015. This work was supported in part by the Singapore Ministry of Education Academic Research Fund (AcRF) Tier 1 under Grant M4011272.040. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shu-Ching Chen.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jingjing.meng@ntu.edu.sg; jsyuan@ntu.edu.sg; wanggang@ntu.edu.sg; eyptan@ntu.edu.sg; yang0374@e.ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2015.2500734

Fig. 1. Example videos from the 73-hour video dataset (only one keyframe is shown for each video). Videos with annotated object locations are from NTU-VOI. The remaining are distractors from Stanford I2V [18]. Three example queries are given on the left: KittyB, NTU, and Pokka (Table I). Their ground truth locations are annotated by magenta, blue, and red bounding boxes, respectively. The NTU-VOI dataset comprises diverse scenes, most of which are cluttered with small target objects. This makes NTU-VOI a challenging dataset for object instance search and localization. Best viewed in color and magnification.

all pixels, which is time consuming. Consequently, the score of any bounding box can be calculated as the summation of pixel scores inside the bounding box, which is only $O(1)$ using integral images [17].

In this paper, we formulate the problem of object instance search in videos as the problem of finding the top-K spatio-temporal trajectories in videos. Experimentally it shows that by enforcing spatio-temporal consistency via trajectory search, we can improve search accuracy over other state-of-the-art methods that treat frames independently. To efficiently find the top-K trajectories, we extend the Max-Path search to large video volumes by utilizing a randomized scheme to quickly obtain per pixel confidence scores. Therefore we can efficiently score all possible bounding boxes to build the 3D trellis that supports Max-Path search. In addition, we make a new video dataset available to the research community, called NTU Video-Object-Instance (NTU-VOI): https://sites.google.com/site/jingjingmengsite/research/ntu-voi/data. It consists of 146 clips captured by mobile cameras or downloaded from YouTube, with frame-wise bounding box annotations of object instances. In total 33,018 frames are annotated. More information on this dataset can be found in Section V-A. To our best knowledge, this is the first video dataset available that provides per frame bounding box annotations of object instances. The YouTube-Objects dataset is another dataset of annotated object videos from YouTube [19]. However it annotates object classes instead of specific object instances. Also, for each class, the bounding box annotations are only provided for one frame per shot on 100–290 different shots, rather than for all ground truth frames. Recent Stanford I2V dataset [18] is a large-scale video dataset with annotated ground-truth video clips and precise temporal segments, but the bounding box locations in ground truth frames are not provided.

We have experimentally evaluated our proposed approach on a 73-hour video dataset (Fig. 1). It is shown that our approach improves the performance of object instance search and localization when compared with the state-of-the-art search and tracking methods [20], [2], [21]. Its effectiveness in incorporating spatio-temporal cues into search is also demonstrated by its ability to find object occurrences in cluttered scenes regardless of large appearance variations due to motion blur, occlusions, and changes in viewpoint, illumination and color. The top 100 trajectories of an object instance in the 73-hour dataset can be found in 200 seconds.

II. Related Work

A. Object Instance Search in Images Versus in Videos

Object instance search in images has received much interest in recent years [3]–[6]. The seminal work [6] first recasts object search as text retrieval. It introduces inverted file index on quantized descriptors and the Bag-of-Words (BoW) model to make fast matching possible on large image datasets. Since then, it has spawn much work on instance search in large image datasets as well as in video shots. For instance, TRECVID instance search challenge aims to locate video shots that most likely to contain a query topic. Most earlier systems first down-sample videos to keyframes, and consider the task as an image retrieval or instance search problem to find keyframes that are most likely to contain the query topics [12]–[15]. Recently, aggregation over several keyframes per shot into a single global signature has been shown to boost the search performance, such as representing each shot by the average of the BoW vectors of its multiple keyframes [22]–[24].

In comparison, object instance search in videos is at a much finer grain compared with the problem above. The goal is to pinpoint the spatio-temporal locations of the object across the 3D video volume in order to enable finer-grain user interaction. The main drawback of directly applying instance search approaches for images to videos is the loss of spatio-temporal context across video frames, which results in sub-optimal performance, as will be shown in our experiments (Section VI-A). Existing approaches focus on the use of temporal continuity to improve the quality of feature descriptors [6], [25], [26]. The regions detected in each frame within a shot are first tracked and then aggregated to describe this scene region throughout the track. In comparison, we enforce spatio-temporal consistency at the trajectory level instead of feature descriptor level. Also

we do not rely on tracking to find correspondences (between bounding boxes in our case) across frames.

Nevertheless, our approach does benefit from the design of existing image-based instance search systems in the way we adopt the feature descriptor quantization and inverted file index to enable efficient frame-wise filtering and matching [6]. Recently, approaches based on higher order statistics [27]–[30] have shown to achieve better search performance than the classical BoW model and its variants [31]. Furthermore, by decomposing the global appearance model and similarity measure of Vector of Locally Aggregated Descriptors (VLAD) and Fisher vector, [3] permits the application of much larger vocabularies in these high order descriptors, which were earlier prohibited by the amount of processing memory it requires. This results in a substantial improvement of performance in generic object instance search in images with a single example. Although we employ the simple quantized SIFT descriptors [32] in this work, our approach does not forbid the use of decomposed VLAD and Fisher vectors [3].

B. Object Localization

For instance search in images, locations of object instances are usually obtained by post-processing through geometric verification, such as Random sample consensus (RANSAC) [8]. However RANSAC requires sufficient number of matched points to reliably estimate a transformation. Alternatively, efficient subimage retrieval (ESR) [7] and efficient subwindow search (ESS) [33] have been proposed to find the subimage with maximum similarity to the query, which are much faster than the exhaustive sliding window approach. In addition, spatial random partition is proposed in [34] to discover and locate visual common objects. The common limitation of the above approaches is that the localization is performed independently on individual images or frames. Therefore, these approaches do not enforce spatio-temporal consistency across video frames. Recently, category-independent object proposals have become a popular approach, which enhances the efficiency of object detection by only evaluating bounding box locations that are likely to contain an object [35]–[38]. References [39] and [40] extend recent 2D object proposal methods to generate spatio-temporal video tube proposals for action detection and localization. As [39] uses optical flow as a cue to derive similarity for both supervoxel and proposal generation, it is more effective when the object is in motion with respect to the background. Similarly, [40] relies on the estimation of the dominant motion and an independent motion evidence map to compute initial supervoxels.

Besides video proposal methods [39], [40], Max-Path search [41], [16] has also been proposed for spatio-temporal event detection. It uses dynamic programming to search for the optimal path over the 3D trellis connecting bounding boxes that are scored prior to the spatio-temporal search. The key challenge to extend Max-Path to the problem of object instance search in videos, however, is the computational cost of scoring all considered bounding boxes in order to construct such a trellis. A sliding-window based scoring of bounding boxes is viable for detection problems, as real-time processing is usually good enough [16]. But it is infeasible for any practical search systems. We tackle this challenge in this work by leveraging a randomized approach to quickly obtain pixel-wise confidence scores, which in combination with integral images [17] permits the application of Max-Path search to large video datasets.

We are the first to extend Max-Path search to the problem of finding object trajectories in large video volumes. Some preliminary results on a 5.5-hour consumer video dataset have been published in [42]. Our previous work [43] is closest to this work in that it combines Hough Voting and Max-Path search to locate object centers in a video sequence. However, it can only produce trajectories of the object center rather than the object itself. Moreover, it is not efficient for large scale videos.

III. Spatio-Temporal Search of Object Instances

A. Problem Formulation

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

Denote a trajectory $T = \{\mathcal{B}_i\}_{i=i_1}^{i_k}$ as a temporal sequence of spatial bounding boxes that locate the object instance in each frame $\mathcal{B}_i = \{x_i, y_i, s_i, a_i, t_i\}$, where x_i and y_i are the center coordinates, s_i and a_i are the scale and aspect ratio of bounding box $\mathcal{B}_i$, and t_i is its temporal index. Then each trajectory $T = \{\mathcal{B}_i\}_{i=i_1}^{i_k}$ should satisfy the following constraints: 1) the smoothness constraint of the trajectory: $x_i - k \leq x_{i+1} \leq x_i + k$, $y_i - k \leq y_{i+1} \leq y_i + k, t_{i+1} = t_i + 1$, where $(2k+1) \times (2k+1)$ is the valid neighborhood size that the center of one bounding box can move to in the next frame; 2) the smoothness constraint of the bounding box scale: $s_i / s_{i+1} \in \{1 - r, 1, 1 + r\}$, where $0 < r < 1$ and a bounding box is only allowed to change to one of these three relative scales in the next frame.

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

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

$$s(T) = \sum_{\mathcal{B}_i \in T} s(\mathcal{B}_i). \tag{1}$$

Similarly, a large positive score $s(T)$ indicates that the object is highly likely to appear along T, and a negative score indicates otherwise. Given a video database $\mathcal{V}$ and a query object Q, our objective is to find the top K trajectories that capture

K instances of the object query. Our top-K trajectory search is then formulated as the following optimization problem:

$$\begin{aligned} \mathbf{T}^* &= \arg\max_{\mathbf{T}\subseteq\mathcal{T}} \sum_{T\in\mathbf{T}} s(T) \\ \text{s.t.} \quad \forall T_i, T_j \in \mathbf{T}, T_i \cap T_j &= \emptyset \\ |\mathbf{T}| &= K. \end{aligned} \tag{2}$$

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

$$T^* = \arg\max_{T\in\mathcal{T}} s(T). \tag{3}$$

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

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

B. Fast Spatio-Temporal Trajectory Search

Exhaustively searching for the globally optimal trajectory $T^* \in \mathcal{V}$ (2) is infeasible due to the exponential complexity of the problem. To reduce the search complexity, we propose to employ the Max-Path search approach proposed in [41], [16], which uses dynamic programming to obtain the globally optimal trajectory.

Here we briefly explain the procedure of Max-Path search. Max-Path search runs on a 3D trellis structure that connects per frame bounding boxes that are scored prior to the spatio-temporal search. The Max-Path search starts from all possible bounding boxes in the first frame, where each bounding box initiates a trajectory and tries to propagate it from the current frame to the next. Given the smoothness constraint of the trajectory, each bounding box will only search for bounding boxes in its neighborhood in the next frame. Whether the trajectory can continue or cannot depends on the accumulated score of the trajectory $s(T)$. If $s(T)$ is positive, then the trajectory continues to grow; otherwise, a new path will be initiated from the current bounding box. During the search, each bounding box $\mathcal{B}$ carries the positive score $s(T)$, which is the best score of all possible trajectories ended at $\mathcal{B}$, and passes it to the next bounding box in the trajectory. Once the search reaches the final frame, we can find the best trajectory by the highest score $s(T)$. Multiple trajectories can be found by removing the current best trajectory at the end of each round before searching for the next best trajectory. As proven in [41], [16], such a dynamic programming strategy can guarantee to find the best trajectory among all candidates, with complexity linear to the video volume size $(O(whn(2k + 1)^2))$.

Despite the successes of applying the Max-Path search to the action detection problem, its extension to object instance search is however not straightforward. As we are searching a large video corpus, it will be computationally expensive to use a sliding-window approach to obtain the bounding box score $s(\mathcal{B})$ at all locations and scales [41]. This is less of a concern for detection problems, where real-time processing is usually good enough. Particularly, note that with a single example, we do not train a discriminative classifier, but directly match the query against the dataset. Exhaustively matching the query at $w \times h$ locations and varying scales in each frame is prohibitive for large video volumes.

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

C. Efficient Confidence Map Generation

We observe that efficient calculation of any bounding box score requires an additive scoring scheme. In other words, if the score of any bounding box can be calculated as the sum of the scores of its containing pixels or interest points [10], we can quickly obtain the score of any bounding box using integral images [17]. Note that image search methods based on global descriptors cannot serve this purpose, as they do not provide matching scores at such a fine level. Hence, we propose to use Randomized Visual Phrases (RVP) to efficiently obtain pixel-wise confidence scores in each frame [4]. Moreover, scoring on the resulting confidence maps from RVP satisfies the additive property.

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

$$\bar{s}(p) = \frac{1}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} s(P_{p,k}) \tag{4}$$

where $\mathcal{K}$ is the total number of RVPs (i.e., patches) covering pixel p, which is equal to the partition rounds, and $s(P_{p,k})$ is the matching score of a patch. In our implementation, we use histogram intersection $(HI(.))$ as the similarity measure to compute $s(P_{p,k})$. Given the histogram representation of the query, h_Q, and that of a patch, h_P, which describe the respective visual word frequency, the histogram intersection $HI(.) = \sum_v \min({h_Q}^v, {h_P}^v)$.

This randomized approach offers two benefits. First, it results in more accurate matching, as spatial context of varying sizes has been taken into consideration. Second, it generates pixel-wise confidence scores efficiently, as only a few patches in each image are evaluated and aggregated to produce the confidence map. However, the original RVP relies on a heuristic seg-

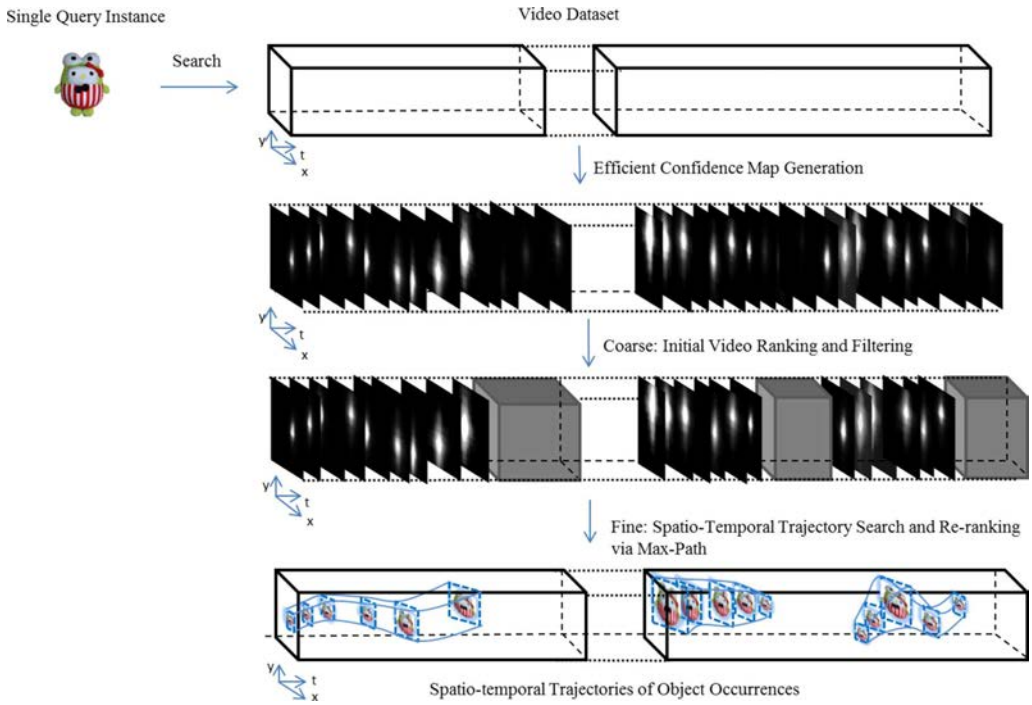


Fig. 2. Overview of proposed approach to find object trajectories in videos. Video keyframes are first matched with the query to produce confidence maps with pixel-wise matching score (2^{nd} row, Section III-C). Next, low confidence videos (the gray cuboids in the 3^{rd} row) are filtered by ranking videos based on the keyframe scores (Section III-D). Max-Path search then finds the globally optimal object trajectories in top ranked videos and re-ranks them according to the trajectory scores (Section III-B). Ground truth locations are highlighted in dashed blue boxes in the found trajectories (4^{th} row). Best viewed in color and magnification.

mentation coefficient α to rank images and locate target objects in each image [4]. On the contrary, in this work we use Max-Path search to jointly evaluate confidence maps across video frames (Section III-B), instead of segmenting individual frames independently. Our approach not only removes the dependency of search and localization on the segmentation coefficient α, but also boosts the search performance by leveraging the spatio-temporal cues (Section VI-A3).

As the resulting confidence maps are not discriminative for Max-Path search [41]. Similar to [41], we add a negative threshold to each confidence map to introduce negative values. To accommodate object appearance variations in different video, instead of a fixed threshold [41], we set the threshold adaptively to be proportional to the average pixel-wise confidence score of each video (excluding zero confidence maps). If we denote the total number of non-zero confidence maps in video V_i as $\mathcal{N}_{NZ}$, the threshold is calculated as

$$MP_{\text{thres}} = \beta \frac{\sum_{\mathcal{F} \in V_i} \overline{s_{\mathcal{F}}}}{\mathcal{N}_{NZ}} \tag{5}$$

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

$$\overline{s_{\mathcal{F}}} = \frac{\sum_{p \in \mathcal{F}} \bar{s}(p)}{|\mathcal{F}|} \tag{6}$$

where $|\mathcal{F}|$ is the total number of non-zero pixels in frame $\mathcal{F}$, and $\bar{s}(p)$ indicates the confidence score of pixel $p \in \mathcal{F}$ calculated by (4). The final confidence score of pixel $p \in \mathcal{F}$ is computed as

$$s(p) = \bar{s}(p) - \overline{s_{\mathcal{F}}}. \tag{7}$$

We shall examine how the negative coefficient β affects the search performance in Section VI-A3.

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

D. Coarse-to-Fine Search

To further improve efficiency, we perform the search in two steps. Given a query, we first match it against coarsely sampled keyframes to fast rank videos based on the best matched keyframes. Only for those top ranked videos, we refine the search and localization using Max-Path to find the globally optimal object trajectories, which is more computationally expensive. In fact, any image search methods besides RVP can be used to generate the initial ranking of all videos, such as Scalable compressed Fisher Vectors (SCFV) tested in our experiments (Section VI-A1). We choose to use RVP because it can both produce the initial video ranking and generate the confidence maps for Max-Path in a single run.

Fig. 2 summarizes our propose approach for spatio-temporal trajectory discovery.

IV. Evaluation Metrics

We use the widely adopted average precision (AP) and mean average precision (mAP) measures to evaluate the search performance. Given a ranked list of R retrieved results, the AP is calculated as the area under the precision-recall curve

$$\text{AP} = \frac{\sum_{r=1}^{R} (\text{Prec}(r) \times \text{rel}(r))}{\#\text{Ground Truth}} \tag{8}$$

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

A. Video mAP

Video mAP evaluates how effective a search system finds the ground truth clips that contain the query object, without considering the exact locations of the object within each retrieved clip. Similar to image retrieval systems [44], we calculate the video average precision (video AP) as the area under the precision-recall curve given a ranked list of videos, and video mAP is defined as the mean of video APs over all queries. Precision is the number of returned positive videos relative to the total number of returned videos. Recall is the number of returned positive videos relative to the total number of positives in the dataset. A video is considered positive if it indeed contains the query object, and negative if otherwise.

B. Trajectory mAP

To evaluate the search quality more precisely, we also introduce the trajectory mAP metric. Different from video mAP, trajectory AP and mAP are calculated based on a ranked list of object trajectories instead of videos. Precision is defined as the number of returned positive trajectories relative to the total number of returned trajectories. Recall is the number of returned positive trajectories relative to the total number of positives in the dataset. As we assume that the trajectories are non-overlapping (Section III-A), we only find one best trajectory for each video and rank them.

However, to obtain the *trajectory-wise* precision-recall curves, we need a criterion to judge whether a returned trajectory is *positive*. It is straightforward for images and videos, but not so obvious for trajectories. Therefore, we introduce Path IoU [16] to measure this relevance, which is the overlapped volume of two paths divided by the union volume of the two. Formally

$$PIoU(T_i, T_j) = \frac{\sum_{t=1}^{N} \mathcal{B}_i^{(t)} \cap \mathcal{B}_j^{(t)}}{\sum_{t=1}^{N} \mathcal{B}_i^{(t)} \cup \mathcal{B}_j^{(t)}} \tag{9}$$

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

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

TABLE I
STATISTICS OF GROUND TRUTH (GT) OBJECT INSTANCE TRAJECTORIES IN VOI DATASET

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

V. EXPERIMENTAL SETUP

A. Dataset

We evaluate the proposed approach on a 73-hour video dataset, comprising 26-minute ground truth videos and a subset of Stanford I2V dataset [18] as distractors.

The ground truth set, called NTU Video-Object-Instance dataset (NTU-VOI), consists of 146 video clips captured by mobile cameras or downloaded from YouTube, with per-frame bounding box annotations of target object locations (i.e., the 10 queries in Table I). Each clip consists of a single shot and contains up to two ground truth trajectories. The average duration of these clips is 10.54 seconds. The annotations are obtained by four trained student helpers using the LabelMe Video tool [45]. In total 151 trajectories with a total of 33,018 frames are annotated. As can be seen in Fig. 1, the VOI dataset covers diverse scenes, most of which are cluttered with the target objects occupying only a small portion of the frame. This makes VOI a challenging dataset for object instance search and localization. We release VOI for future research, which can be accessed at: https://sites.google.com/site/jingjingmengsite/research/ntu-voi/data.

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

All 73-hour videos are first resized to a spatial resolution of 800×450 pixels, then sampled uniformly at 1 fps, resulting in 263,180 keyframes. Although advanced keyframe detection methods [46], [47] can be applied instead to obtain the

keyframes, uniform sampling works reasonably well for this dataset.

B. Queries

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

VI. Experimental Results

A. Video mAP

As video mAP does not concern about the exact object locations within retrieved clips, image search methods can be directly applied to rank videos and calculate video mAP. To this end, we evaluate two existing visual search systems in comparison with our proposed trajectory discovery approach, which we call as Randomized Visual Phrases with Max-Path search (RVP-MP). The two baseline methods we compare with are: (1) Scalable compressed Fisher vector with RANSAC (SCFV-RANSAC), which is based on global descriptor matching followed by RANSAC geometric verification [20], and (2) the baseline RVP (RVP-Baseline) [2], which is based on bundled local descriptors. It is worth mentioning that both baseline systems rank videos based on the score of the best matched keyframe, while the proposed RVP-MP ranks videos by trajectory scores. For each query, we compute video AP based on the top 100 retrieved videos. And we average the APs over the 10 queries to get the video mAP. In the following, we first explain the implementation of the two baseline systems and our proposed RVP-MP, then we present our results.

1) SCFV-RANSAC: Scalable compressed Fisher vector (SCFV) is a state-of-the-art global image descriptor [20], which has been adopted by the MPEG standardization of compact descriptor for visual search (CDVS). In our experiments, the performance of SCFV followed by RANSAC geometric verification is tested using the API from the authors. The dimension of local descriptors is reduced to 32 using PCA, and the number of Gaussian mixture components is set to 512. As in [20], a separate set of images (also from the authors) is used to train the Guassian mixture model, PCA and correlation weights.

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

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

TABLE II
Video MAP on Top 100 Retrieved Clips

	SCFV-RANSAC [20]	RVP-Baseline [2] α				
		1.0	2.0	3.0	4.0	5.0
mAP	0.379	0.357	0.366	0.364	0.375	0.396

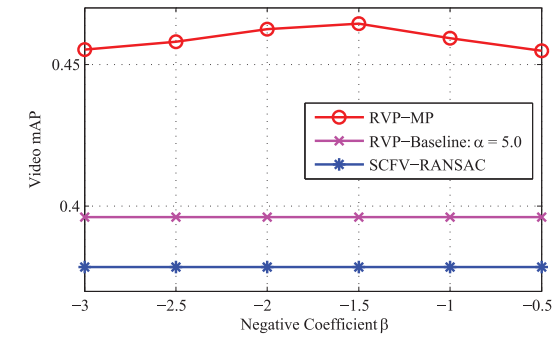


Fig. 3. Impact of β on video mAP: our RVP-MP method consistently outperforms the RVP-Baseline [2] and SCFV-RANSAC [20] as β changes from -3 to -0.5.

In our experiments, interest points are first extracted using Hessian-Affine detectors [48] and represented as SIFT local descriptors [32]. SIFTs are sampled every 10th keyframe to build a vocabulary of 1 M words using FLANN [49]. Once all SIFTs are quantized and indexed into an inverted file for each video, we run RVP to rank keyframes of each video. We use $HI(.)$ as the matching kernel for RVP and the partition parameters is set to 200 rounds of partitions with 8×4-sized random templates.

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

3) RVP-MP: Different from the above two baseline approaches, RVP-MP uses the best trajectory score instead of the best keyframe score as video score to rank database videos. The best trajectory score is calculated as the Max-Path score of each video [i.e., $s(T^*)$, (3)].

As mentioned in Section III-C, we only apply Max-Path search on top ranked videos for efficiency. Therefore we first follow the baseline RVP Section VI-A2, with $\alpha = 5.0$) to obtain a ranked list of database videos based on the best keyframe score. Then for each of the top 100 clips, we run Max-Path search on the resulting confidence maps from RVP to find the best trajectory T^*, and use $s(T^*)$ to re-rank the top 100 clips accordingly. We use the Max-Path algorithm with multi-scale extension and a starting height of 60 pixels. The aspect ratio is fixed to be the same as that of the query. The spatial step is set to 10 pixels and the temporal step is synchronized with the keyframe sample rate (i.e., 1 fps).We compute the summations of multi-scale bounding boxes using integral images [17].

The video mAP of RVP-MP is evaluated under varying negative coefficient β (5). Fig. 3 illustrates the video mAP of RVP-MP in comparison with RVP-Baseline ($\alpha = 5.0$) and SCFV-RANSAC, as β increases from -3.0 to -0.5, at an

TABLE III
VIDEO MAP ON TOP 100 RETRIEVED CLIPS. RVP-MP CONSISTENTLY IMPROVES AP ON ALL QUERIES OVER RVP-BASELINE, EXCEPT FOR FERRARI (SAME AP). OVERALL RVP-MP IMPROVES RVP-BASELINE BY 16.71%.

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

interval of 0.5. It is observed that our RVP-MP consistently outperforms the other two methods regardless of changing β. When $\beta = -2.0$, RVP-MP improves RVP-Baseline by 16.71% and SCFV-RANSAC by 22.17%. It validates that spatio-temporal consistency is beneficial to object instance search in videos.

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

B. Trajectory mAP

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

We use Kernelized Correlation Filters (KCF) [21] with the source code provided by the authors. The chosen kernel is Gaussian. To make a fair comparison with RVP-MP, we start with the same top 100 videos obtained by the baseline RVP, with a segmentation threshold $\alpha = 5.0$ (Section VI-A2). To initialize the tracker in each of the top 100 clips, we fit a bounding box to the segmented region from the best matched keyframe in this clip. We consider the resulting bounding box as valid if it has a minimum height of 60 pixels. Otherwise we check the box from the second best keyframe, and so on. Empirically we observe that this unsupervised approach produces reasonable initialization results. Once we obtain the initial bounding box of a clip, we run the KCF tracker from the initialization frame forward and backward, and concatenate the two trajectories as the final trajectory. Note that although SCFV-RANSAC (Section VI-A1) can also provide object bounding box locations to initialize the tracker, we observe that it produces much fewer reliable bounding boxes compared to RVP. This is because the target objects are usually small; hence the number of matched points are insufficient to reliably estimate a transformation using RANSAC.

As the KCF tracker runs on every frame instead of just the keyframes, for RVP-MP, we also run the Max-Path algorithm across all frames. The other Max-Path parameters remain the same as in previous Section VI-A3. Following (9), for the computation of the precision-recall curves and consequently the trajectory mAP, we consider a trajectory T^* positive if $PIoU(T^*, G) > \delta_p$, where G is the annotated ground truth trajectory and δ_p is the overlap threshold.

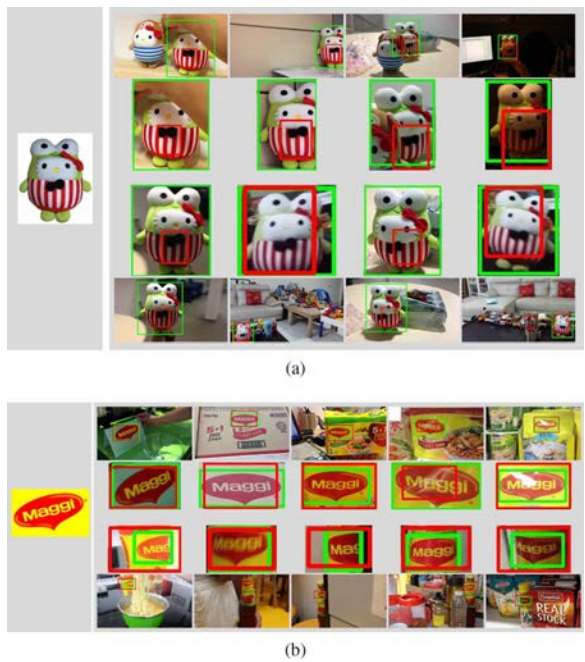


Fig. 4. Two example queries (left) and top trajectories returned by our method. Each trajectory is represented by one keyframe with the magnified object region below or above the keyframe. The red boxes indicate our results, and the green boxes indicate the ground truth annotations. Given a single query example, our approach is able to find its occurrences in cluttered scenes regardless of appearance variations due to occlusions, motion blur, and changes in viewpoint, illumination, and color. Best viewed in color and magnification.

Fig. 4 demonstrates that even on the highly cluttered NTU-VOI dataset, the proposed RVP-MP is able to find object trajectories with large appearance variations due to viewpoint changes, illumination changes, color changes, motion blur and occlusions.

Fig. 5 shows visual comparisons of example trajectory search results from RVP-MP and KCF. We can see that a typical tracker like KCF fails to terminate the trajectory when the object exits the scene [1st row, Fig. 5(a)]. On the contrary, our RVP-MP approach can discover the start and end points of each trajectory automatically [2nd row, Fig. 5(a)]. In addition, as the initial object location for KCF is given by the best segmented bounding box generated by the baseline RVP, an inaccurate initial estimation of the object size (i.e., size of initial bounding box) will be carried on by the tracker, thus affecting the accuracy of the resulting trajectory [1st row, Fig. 5(b)]. On the other hand, using multi-scale Max-Path search, RVP-MP can search for and adjust the sizes of bounding boxes along the trajectory [2nd row, Fig. 5(b)].

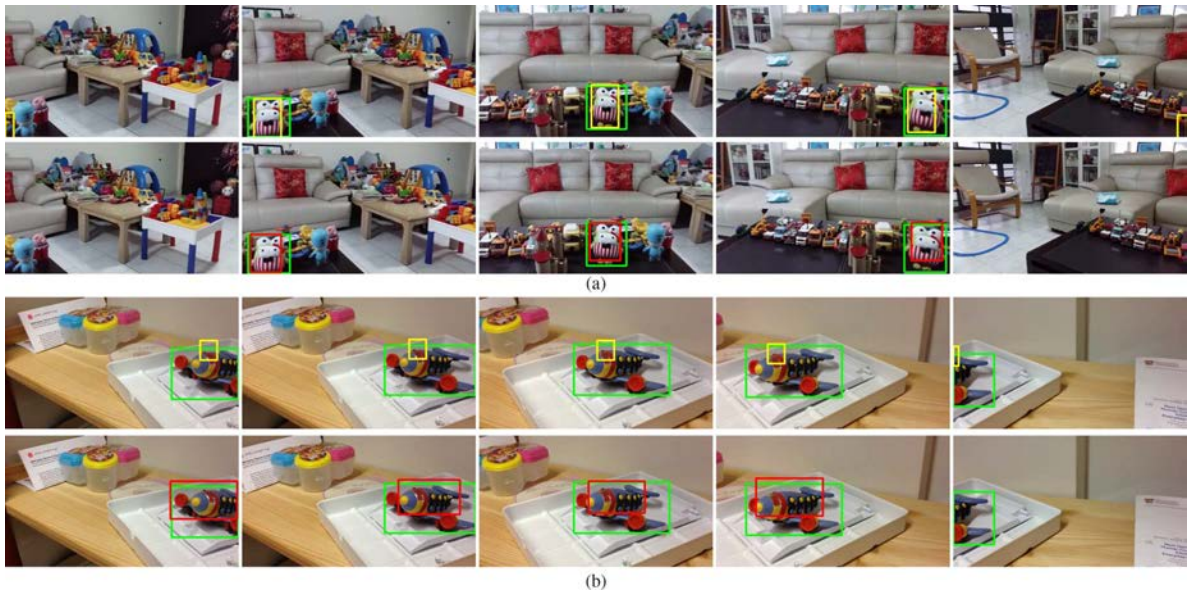


Fig. 5. Example search results of (a) "KittyG" and (b) "Plane". Results of KCF tracker [21] are denoted in yellow bounding boxes in the top rows. Results of RVP-MP are in red in the bottom rows. Ground truth bounding boxes are in green. (a) KCF fails to find the correct start and end points of the trajectory when the object enters and exits the scene, while RVP-MP discover the start and end points automatically. (b) An inaccurate initial estimation of bounding box size will be carried on by KCF, while RVP-MP searches for the best bounding box size given the confidence maps. Best viewed in color and magnification.

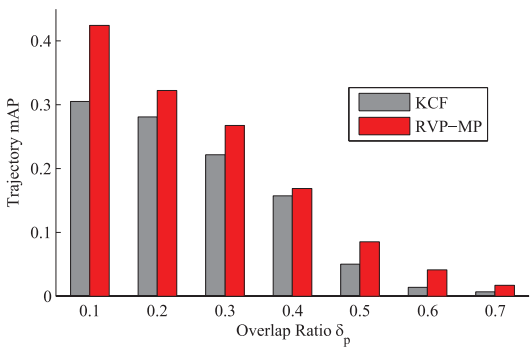


Fig. 6. Trajectory mAP: our RVP-MP method consistently outperforms KCF tracker [21] in a wide range of overlap ratio δ_p.

Fig. 6 compares the trajectory mAP of RVP-MP and KCF with different overlap ratio δ_p. It shows that RVP-MP consistently outperforms KCF on trajectory mAP as δ_p increases from 0.1 to 0.7. The improvement is between 7 and 197% (39.05%, 14.63%, 20.76%, 7.13%, 70.46%, 197.10%, 163.08%, respectively).

Fig. 7 presents more example trajectories resulting from RVP-MP. Our approach accurately locates the objects regardless of varying viewpoint (1st and 2nd row) and scale (2nd rows), reflection (2nd row) and occlusions (3rd row). In addition, it can automatically discover the start and end points of each trajectory (Fig. 7, 4th–6th rows).

2) Parameter Sensitivity: We test the Max-Path search at different temporal granularities: 1 fps, 2 fps, 3 fps, 5 fps, 10 fps, up to every frame. Fig. 8 plots the trajectory mAP of RVP-MP with different temporal sample intervals, with respect to the overlap ratio δ_p. As can be seen, the performance of RVP-MP degrades gracefully as the temporal step increases.

C. Efficiency

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

D. Discussions

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

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

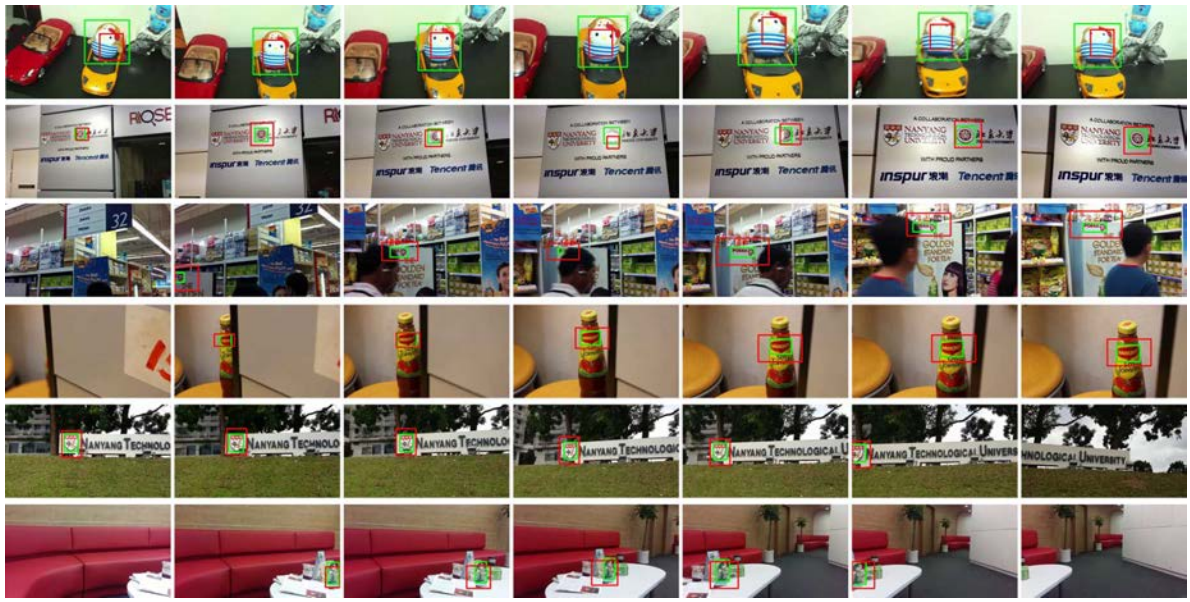


Fig. 7. Example trajectories returned by our proposed RVP-MP in the presence of varying viewpoint (1st and 2nd rows) and scale (2nd row), reflection (2nd row) and occlusions (3rd row). In addition, the start and end points of each trajectory are discovered automatically (4th–6th rows). RVP-MP results are marked in red, and the ground truth bounding boxes are in green. Best viewed in color and magnification.

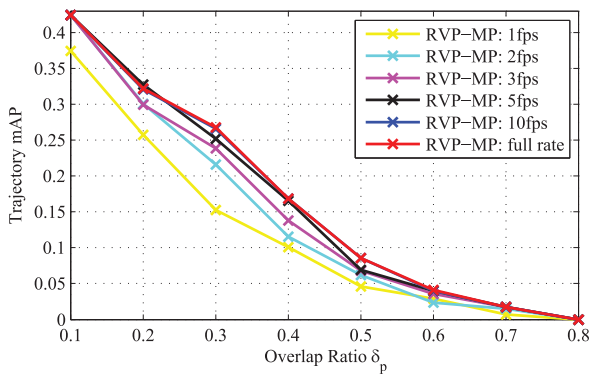


Fig. 8. Trajectory mAP of RVP-MP for different temporal steps and overlap ratio δ_p. Best viewed in color and magnification.

VII. Conclusion

In this work, we explore the use of spatio-temporal cues to boost the performance of object instance search in videos, and propose to formulate the problem as finding the top-K spatio-temporal object trajectories. By utilizing RVP to quickly obtain per pixel confidence scores, thus enabling fast scoring of any bounding boxes in a video, we are the first to extend the Max-Path search from detection to the search domain. Experiments on a 73-hour video dataset validates that the proposed approach is effective in improving search quality compared with state-of-the-art methods that treat frames independently. Compared with individual frames, the resulting spatio-temporal trajectories are better indicators of video relevance because of their ability to better handle false alarms and missed detections across frames caused by appearance variations or cluttered backgrounds. Our approach also improves the spatio-temporal localization of objects compared with the state-of-the-art KCF tracker. In addition, we make available a new video dataset, NTU-VOI, to facilitate future research on object instance search in videos.

Acknowledgment

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Singapore, under its Interactive Digital Media (IDM) Strategic Research Programme.

References

[1] X.-S. Hua, T. Mei, and A. Hanjalic, *Online Multimedia Advertising: Techniques and Technologies*. Hershey, PA, USA: IGI Global, 2010.
[2] Y. Jiang, J. Meng, J. Yuan, and J. Luo, "Randomized spatial context for object search," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1748–1762, Jun. 2015.
[3] R. Tao, E. Gavves, C. Snoek, and A. Smeulders, "Locality in generic instance search from one example," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* , Jun. 2014, pp. 2099–2106.
[4] Y. Jiang, J. Meng, and J. Yuan, "Randomized visual phrases for object search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 3100–3107.
[5] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object retrieval and localization with spatially-constrained similarity measure and k-NN reranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 3013–3020.
[6] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, vol. 2, pp. 1470–1477.
[7] C. H. Lampert, "Detecting objects in large image collections and videos by efficient subimage retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep.–Oct. 2009, pp. 987–994.
[8] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.
[9] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, Apr. 2009.
[10] J. Meng *et al.*, "Interactive visual object search through mutual information maximization," in *Proc. ACM Multimedia*, 2010, pp. 1147–1150.
[11] J. Revaud, M. Douze, and C. Schmid, "Correlation-based burstiness for logo retrieval," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 965–968.

[12] T. Kawanishi, A. Kimura, K. Kashino, S. Satoh, D.-D. Le, X. Wu, and S. Poullot, "Ntt communication science laboratories and nii in TRECVID 2010 instance search task," in *Proc. TRECVID*, 2010, pp. 1–1.
[13] D.-D. Le, C.-Z. Zhu, S. Poullot, V. Q. Lam, V. H. Nguyen, N. C. Duong, T. D. Ngo, D. A. Duong, and S. Satoh, "National Institute of Informatics, Japan at TRECVID 2012," in *Proc. TRECVID Workshop*, 2012.
[14] C.-Z. Zhu and S. Satoh, "Large vocabulary quantization for searching instances from videos," in *Proc. 2nd ACM Int. Conf. Multimedia Retrieval*, 2012, pp. 52:1–52:8.
[15] Y. Yang and S. Satoh, "Efficient instance search from large video database via sparse filters in subspaces," in *Proc. 20th IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 3972–3976.
[16] D. Tran, J. Yuan, and D. Forsyth, "Video event detection: From subvolume localization to spatiotemporal path search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 404–416, Feb. 2014.
[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Dec. 2001, vol. 1, pp. I-511–I-518.
[18] A. Araujo, J. Chaves, D. Chen, R. Angst, and B. Girod, "Stanford I2V: A news video dataset for query-by-image experiments," in *Proc. ACM Multimedia Syst.*, 2015, pp. 237–242.
[19] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 3282–3289.
[20] L.-Y. Duan, J. Lin, J. Chen, T. Huang, and W. Gao, "Compact descriptors for visual search," *IEEE Multimedia*, vol. 21, no. 3, pp. 30–40, Jul.–Sep. 2014.
[21] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans.Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
[22] C.-Z. Zhu, Y.-H. Huang, and S. Satoh, "Multi-image aggregation for better visual object retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2014, pp. 4304–4308.
[23] N. Ballas *et al.*, "Irim at TRECVID 2014: Semantic indexing and instance search," in *Proc. TRECVID*, 2014.
[24] H. J. Cai-Zhi and S. Satoh, "Nii team: Query-adaptive asymmetrical dissimilarities for instance search," in *Proc. TRECVID*, 2013.
[25] A. Araujo *et al.*, "Efficient video search using image queries," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 3082–3086.
[26] A. Araujo, J. Chaves, R. Angst, and B. Girod, "Temporal aggregation for large-scale query-by-image video retrieval," in *Proc. 22nd IEEE Int. Conf. Image Process.*, to be published.
[27] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.
[28] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 3384–3391.
[29] H. Jegou *et al.*, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
[30] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 1578–1585.
[31] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2006, vol. 2, pp. 2169–2178.
[32] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 6, no. 2, pp. 91–110, 2004.
[33] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[34] J. Yuan, Y. Wu, and M. Yang, "Discovery of collocation patterns: From visual words to visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.
[35] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans.Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.
[36] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.
[37] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014 [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=220569
[38] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?," *CoRR*, 2014 [Online]. Available: http://arxiv.org/abs/1406.6962
[39] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid, "Spatio-temporal object detection proposals," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014 [Online]. Available: http://hal.inria.fr/hal-01021902
[40] M. Jain, J. van Gemert, H. Jegou, P. Bouthemy, and C. G. Snoek, "Action localization with tubelets from motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 740–747.
[41] D. Tran and J. Yuan, "Optimal spatio-temporal path discovery for video event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 3321–3328.
[42] J. Meng, J. Yuan, Y.-P. Tan, and G. Wang, "Fast object instance search in videos from one example," in *Proc. 22nd IEEE Int. Conf. Image Process.*, to be published.
[43] J. Meng, J. Yuan, G. Wang, Y.-P. Tan, and J. Xu, "Object instance search in videos," in *Proc. Int. Conf. Inform., Commun. Signal Process.*, 2013, pp. 1–4.
[44] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[45] J. Yuen, B. Russell, C. Liu, and A. Torralba, "LabelMe video: Building a video database with human annotations," in *Proc. 12th IEEE Int. Conf. Comput. Vis.*, Sep.–Oct. 2009, pp. 1451–1458.
[46] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput., Commun., Appl.* vol. 3, no. 1, p. 3, Feb. 2007.
[47] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 66–75, Feb. 2012.
[48] M. Perd'och, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 9–16.
[49] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2009, pp. 331–340.

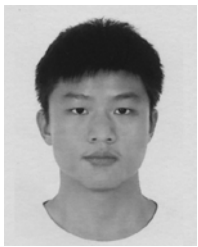
Jingjing Meng (M'09) received the B.E. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2003, the M.S. degree in computer science from Vanderbilt University, Nashville, TN, USA, in 2006, and is currently working toward the Ph.D. degree in electrical and electronic engineering at Nanyang Technological University (NTU), Singapore.

From 2007 to 2010, she was a Senior Research Staff Engineer with Motorola Applied Research Center, Schaumburg, IL, USA. She is currently a Researcher with the School of Electrical and Electronic Engineering, NTU. Her research interests include computer vision and big image and video data analysis.

Junsong Yuan (S'06–M'08–SM'14) received the Ph.D. degree from Northwestern University, Evanston, IL, USA, in 2009.

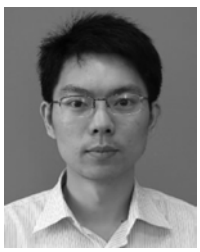
He is currently an Associate Professor and Program Director of Video Analytics with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Dr. Yuan served as the Program Co-Chair of the IEEE Visual Communications and Image Processing Conference in 2015. He served as the Organizing Co-Chair of the Asian Conference on Computer Vision (ACCV) in 2014, the Area Chair of the IEEE Winter Conference on Computer Vision in 2014, the IEEE Conference on Multimedia Expo (ICME) in 2014 and 2015, and the ACCV in 2014. He is also a Guest Editor of the *International Journal of Computer Vision*, and an Associate Editor of *The Visual Computer Journal* and the *IPSJ Transactions on Computer Vision and Applications*. He was the recipient of the Nanyang Assistant Professorship from Nanyang Technological University, the Outstanding EECS Ph.D. Thesis Award from Northwestern University, and the Best Doctoral Spotlight Award from the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2009.



Jiong Yang received the B.Eng. (Hons.) degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2013, and is currently working toward the Ph.D. degree at Nanyang Technological University.

His research interests include computer vision and machine learning.



Gang Wang (M'11) received the B.S. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA, in 2010.

He is currently an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. His research interests include computer vision and machine learning, particularly object recognition, scene analysis, and deep learning.

Prof. Wang is an Associate Editor of *Neurocomputing*. He was a recipient of the Harriett & Robert Perry Fellowship (2009–2010) and the CS/AI Award (2009) at UIUC.

Yap-Peng Tan (S'95–M'98–SM'04) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1993, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, USA, in 1995 and 1997, respectively, all in electrical engineering.

From 1997 to 1999, he was with the Intel Corporation, Chandler, AZ, USA, and Sharp Laboratories of America, Camas, WA, USA. In November 1999, he joined the Nanyang Technological University of Singapore, where he is currently Associate Professor and Associate Chair (Academic) of the School of Electrical and Electronic Engineering. He is the principal inventor or co-inventor on 15 U.S. patents in the areas of image and video processing. His current research interests include image and video processing, content-based multimedia analysis, computer vision, and pattern recognition.

Dr. Tan served as the Chair of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society from 2010 to 2014, a member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society from 2009 to 2013, a voting member of the IEEE International Conference on Multimedia & Expo (ICME) Steering Committee from 2011 to 2012, and Chairman of the IEEE Signal Processing Singapore Chapter from 2009 to 2010. He is currently serving as Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (since 2014) and IEEE ACCESS (since 2013), an Editorial Board Member of the EURASIP *Journal on Advances in Signal Processing* and the EURASIP *Journal on Image and Video Processing*, and an elected member of the Multimedia Systems and Applications Technical Committee (MSA TC) and Visual Signal Processing and Communications Technical Committee (VSPC TC) of the IEEE Circuits and Systems Society. He has also served as Guest Editor for special issues of several journals, including the IEEE TRANSACTIONS ON MULTIMEDIA. He is the General Co-Chair of the 2015 IEEE International Conference on Visual Communications and Image Processing (VCIP 2015), Tutorial Co-Chair of the 2016 IEEE International Conference on Multimedia and Expo (ICME 2016), and Technical Program Co-Chair of the 2019 IEEE International Conference on Image Processing (ICIP 2019), and was the General Co-Chair of the 2010 IEEE International Conference on Multimedia and Expo (ICME 2010) and Finance Chair of the 2004 IEEE International Conference on Image Processing (ICIP 2004).