Object Instance Search in Videos

Jingjing Meng, Junsong Yuan, Gang Wang
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
Email: {jingjing.meng, jsyuan, wanggang}@ntu.edu.sg

Jianbo Xu
School of Computer Science
Hunan University of Science and Technology
Xiangtan, Hunan, China
Email: jbxu@hnust.edu.cn

Abstract—**In this paper, we propose a novel approach for object instance search in videos. Employing discriminative mutual information score and inferring the location of target object centers from matched local feature descriptors using Hough voting, we achieve robust matching and per-frame localization despite orientation and scale variations. We then leverage Max-Path search [1] to efficiently find the globally optimal spatio-temporal trajectory of the object center in each video sequence. Experimental results on a collection of mobile-captured videos in real-world environments demonstrate the effectiveness and accuracy of our method.**

I. Introduction

With the proliferation of user-created video content, searching object instances in videos has started to gain tractions in recent years. Different from object detection, which concern about a class of objects, such as cars and pedestrians, object instance search is interested in finding a specific object instance, such as a Dettol logo. This fine grain discriminability is useful for many real world applications, such as finding a missing suitcase from surveillance videos (vs. finding any suitcases), augmented reality and context-aware advertisement. Moreover, object instance search ususally uses a single query and directly matches it against each video frame. Therefore, it avoids the time-consuming training step in traditional object detection problems and is especially suited for situations where limited training examples are available.

Searching object instances in videos aims to find spatio-temporal trajectories of the target object in the 3D video volume. Therefore, we can decompose it into two sub-problems: per-frame matching and spatio-temporal localization.

For per-frame matching, motivated by the successes of invariant local features [2] and nearest neighbor classifiers in object detection, classification and search [3]–[5], we propose a similarity measure using discriminative point-wise mutual information score employing the non-parametric nearest neighbor classifier [5].

For spatio-temporal localization, we first estimate per-frame object center locations using Hough voting, then we utilize Max-Path search [1] to find the globally optimal spatio-temporal trajectory of the object center in each video sequence. To accommodate object scale and orientation variations in videos, existing Hough-based methods either sample at different scales and orientations [3], [6] or painstakingly derive all parameters in the Hough space [2]. Contrarily, we directly estimate the center location of the target by comparing the Gaussian scale levels and dominant orientations of matched local feature descriptors. Consequently, our Hough voting is efficient and robust to scale and orientation variations. Although our per-frame confidence map can be noisier than results from strict geometric verification [2], [7], Max-Path [1] serves as an effective spatio-temporal filter to improve localization accuracy and gives the globally optimal trajectories in test video sequences. Different from the original Max-Path search [1], which has to test a number of window scales and aspect ratios if the size of the target is unknown, we care about the object centers rather than the object itself. Consequently, we only apply a Gaussian filter at each predicted center location for spatial smoothing and run Max-Path on each pixel (or a small window centered at each pixel, for speed consideration). Our preliminary results on a small video collection captured by a mobile device show the promise of our approach.

II. Method

Given a query object Q and a video sequence $\mathcal{V} = \{\mathcal{I}_1, \mathcal{I}_2 ... \mathcal{I}_n\}$, where $\mathcal{I}_i \in \mathcal{V}$ is a $w \times h$ sized frame with temporal index i, our goal is to find in $\mathcal{V}$ all spatio-temporal trajectories of instances of Q in $\mathcal{V}$. We denote a sequence of object center locations as $\mathcal{C} = \{c_{i_1}, c_{i_2} ... c_{i_k}\}$, where $1 \leq i_1 \leq i_k \leq n$. Each c_i in $\mathcal{C}$ is represented as a spatio-temporal location (x_i, y_i, i), where i is the temporal index.

A. Discriminative mutual information score

We represent query Q and each frame I_i as collections of invariant local feature points, p, $d_q \in \mathbb{R}^N$ as the feature descriptor of points in Q, and $d_p \in \mathbb{R}^N$ as the feature descriptor of points in I_i. We randomly pick one frame from a video sequence outside the testing video dataset as the negative example, and also describe that frame as a collection of local invariant features. Following [5], we measure the similarity between a feature point in Q and that in I_i by the discriminative point-wise mutual information. Denote the set of positive feature points as Ω^+ (i.e. feature points in Q) and the set of negative features as Ω^-, the point-wise score $s(p)$ can be derived as:

$$s(p) = \log \frac{1}{P(\mathbf{\Omega}^+) + \frac{P(d_p|\mathbf{\Omega}^-)}{P(d_p|\mathbf{\Omega}^+)} P(\mathbf{\Omega}^-)} \tag{1}$$

For simplicity, we assume equal probability of Ω^+ and Ω^-. If the number of feature points in Ω^- is much larger than

978-1-4799-0434-1/13/$31.00 ©2013 IEEE

that in Ω^+ (the query image), we can balance the sizes of Ω^+ and Ω^- by clustering negative feature points via K-means [5]. Alternatively, the probability of Ω^+ and Ω^- can be more accurately estimated by the number of feature points in Ω^+ and Ω^-, respectively. According to [4], if we apply the kernel density estimation based on Ω^+ and Ω^- and use the nearest neighbor approximation for the Gaussian kernel estimation, we have:

$$\begin{aligned}\frac{P(d_p|\mathbf{\Omega}^-)}{P(d_p|\mathbf{\Omega}^+)} &= \frac{\frac{1}{|\mathbf{\Omega}^-|}\sum_{d_j\in\mathbf{\Omega}^-}K(d_p-d_j)}{\frac{1}{|\mathbf{\Omega}^+|}\sum_{d_j\in\mathbf{\Omega}^+}K(d_p-d_j)} \\ &\approx \exp^{-\frac{1}{2\sigma^2}(\|d_p-d^-_{NN}\|^2-\|d_p-d^+_{NN}\|^2)}. \end{aligned} \tag{2}$$

In the above, d^+_{NN} and d^-_{NN} are the nearest neighbors of d_p in Ω^+ and Ω^-, respectively, measured in Euclidean distance.

Different from [5], observing that most d_p in I_i should be dissimilar to $d_q \in \Omega^+$, and the object center in I_i should only be estimated by high-confidence matches to the query, we only calculate $s(p)$ of the k nearest neighbors in I_i of each $d_q \in \Omega^+$, instead of calculating $s(p)$ for every d_p in I_i. This not only greatly reduces the computational cost for nearest neighbor search, but also reduces noise in the Hough voting step in the next section.

Consequently, to calculate $\frac{P(d_p|\mathbf{\Omega}^-)}{P(d_p|\mathbf{\Omega}^+)}$ for each k nearest neighbor d_p of $d_q \in \Omega^+$, we replace d^+_{NN} in Eq.2 by d_q. Note that the distance between d_p and d^+_{NN} is equal to or smaller than the distance between d_p and d_q. Although we use a variant of Eq.2, this modification makes sense. If d^+_{NN} (i.e. the nearest neighbor of d_p) is different from d_q, our modification will result in a greater $\frac{P(d_p|\mathbf{\Omega}^-)}{P(d_p|\mathbf{\Omega}^+)}$, hence a lower score $s(p)$. Therefore, although we would cast $s(p)$ to the wrong center location (if d_q is far from the real best match of d_p), the confidence score of the voted center location would be low. Contrarily, if we follow the original Eq.2, a high positive vote will be cast to the wrong center location, which is undesirable.

B. Spatial localization using Hough voting

A problem with the above similarity measure is the Nave Bayes assumption [4], [5], which assumes independence among the local feature points. In reality, this assumption is often violated. When two local feature points come from the same object, they must satisfy certain geometric constraints such as their relative distances to the object center and relative orientations. To alleviate this problem, we propose to transfer the geometric constraints on the query to its potential matches. We derive these constraints from invariant local features. Take SIFT features [2] for example, our observation is that if two matched SIFT points correspond to the same part of the target object, the difference in their dominant orientations should reflect the relative rotation between the two object instances. Similarly, the relative scale of the two object instances can be estimated from the corresponding Gaussian scale levels in which the two SIFT keypoints are extracted.

Specifically, for each feature point in Ω^+, we calculate its offset from the object center and dominant orientation. We also record the scale level from which this point is extracted. For each of its matched k-NN feature points in I_i, we estimate the target object center through an affine transform by the difference between this neighbors dominant orientation and scale level and those of the query feature. The mutual information score in Eq.1 is then cast to the predicted center location instead of the feature location. Similarly, a negative vote is cast to the predicted object center for each nearest neighbor of points in Ω^-. We apply a Gaussian filter centered at each predicted center for spatial smoothing. This results in a confidence map of frame I_i, where a high positive score at location c_i implies a high likelihood that the target object center locates at c_i, while a negative score indicates that the target center is unlikely to appear at c_i.

C. Spatio-temporal Localization via Max-Path Search

Once we obtain the confidence map of target object center for each frame I_i, we search for the optimal spatio-temporal center location in the entire video sequence using Max-Path search [1].

As we locate the object center rather than the object, we do not need to test a number of window scales and aspect ratios as [1] does. If we denote by $s(c_i)$ the discriminative confidence score of location c_i, our goal is to find a smooth 3D spatio-temporal path p^* with the highest accumulative confidence score. In our experiments, we evaluate the per-frame confidence score in the 2×2 neighborhood of c_i, simply to enhance the local response. Similar to [1], multiple paths can be located by repeating the search after removing the current best path from the confidence map sequence.

Once p^* is found, we have the spatio-temporal trajectory of the object. We can then estimate the scale of the object in each frame by pooling all feature points that have voted for the detected center in this frame. The estimation is based on the scale of Q and relative Gaussian scale levels of matched feature points.

III. Experiment

We capture 9 short video sequences by an iPhone in two local supermarkets. The lengths of these video sequences vary between 164 and 320 frames, with a total number of $2,277$ frames. The resolution of all sequences is 1280×720. Our detection targets are 9 logos: Clorox, Dettol, Dumex, Ferrero Rocher, Kopiko, Oral-B, Pokka, Sara Lee and Tefal. Each sequence contains one of the nine target objects, except for the Tefal sequence, which also contains Clorox and Dettol. As can be seen from Fig.1 and Fig.2, these video sequences are captured with cluttered background in a supermarket setting, and the target objects usually only occupy a small portion of the frames, which make the detection task challenging. In addition, zoom, camera rotation, partial occlusion and motion blur further worsen the detection condition.

We use Google image search and input the logo names to find clean-background logos as queries. It is worth noting that because our Hough voting step adaptively estimates a matched features distance to object center based on keypoint scales, we do not need to constrain the size of the query logos. In our experiments, sizes of the logos vary between 271×140 and

TABLE I
DETECTION RESULTS

Video	Total #Frs	#Frs Containing the Object[a]	#Correctly Detected Frs	#Falsely Detected Frs	#Missed Frs	Localization Accuracy
Clorox	164	164	164	0	0	100%
Dettol	258	241	241	4	0	100%
Dumex	287	287	287	0	0	100%
Ferrero Rocher	238	238	238	0	0	100%
Kopiko	289	289	0	0	289	NA
Oral-B	234	198	151	26	47	98.01%
Pokka	181	129	101	34	28	99.01%
Sara Lee	306	306	302	0	4	100%
CDT-C[b]	320	83	0	0	83	NA
CDT-D	320	305	269	0	36	98.51%
CDT-T	320	320	294	0	26	100%

[a] A frame with partial occlusion is counted as containing the object if $> 1/2$ of the object is visible.

[b] CDT stands for the video sequence that contains three objects: Clorox, Dettol and Tefal. CDT-C is the detection result for Clorox, while CDT-D and CDT-T are the detection results for Dettol and Tefal, respectively.

1100×410. For all video sequences, we use a random frame from a commercial video as the negative example.

We compute the SIFT keypoints and descriptors for each example logo using VLFeat package. For each SIFT point in a logo, we calculate its spatial distance to the logo center and record its corresponding scale and orientation. We use 5 nearest neighbors in each frame for each SIFT in the query logo. For each neighbor, we calculate the object center location by comparing its scale and orientation with that of the query SIFT, and cast a vote to the estimated center. The weight of its vote is calculated as the mutual information score defined in Eq.1. For each voted center, we apply a Gaussian filter with a variance of 2.5 in its 5×5 neighborhood for smoothing.

Once per-frame confidence maps are obtained for all logos in a video sequence, we run Max-Paths search for each logo and calculate its average per-frame confidence score. If the average per-frame score is below 1.7, the sequence is considered not containing the target logo. Otherwise, we evaluate the frame-wise detection performance (number of correctly detected frames, number of falsely detected frames, number of missed frames) as in columns $4 - 6$ of Table I.

To evaluate the localization accuracy, we manually check each correctly detected frame (column 4, Table I) to see if the detected object center is indeed located inside the object. The localization accuracy of each sequence is calculated as the ratio of frames with correct center location in the correctly detected frames. Localization accuracy is listed in the last column of Table I. It can be seen that our algorithm achieves an average localization accuracy of 99.50%.

When taking a close look at one of the missed detection sequences, CDT-C, we notice that Clorox only appears in frames $238 - 320$ in this sequence. Moreover, most of the time there is strong reflection on the object, which makes it difficult to detect keypoints (Fig.3). Both explain why this sequence is filtered by the average per-frame score threshold 1.7. Nevertheless, when we check the detected Max-Path in CDT-C, we find that our method still locates the object center with an accuracy of 53.25%. However, as this sequence is filtered by the threshold, no localization accuracy is reported in Table I. This could be addressed by replacing the simple thresholding over the entire sequence with per-frame significance test, similar to [3].

IV. CONCLUSION

Our contributions are two-fold. First, we achieve robust matching and per-frame localization using Hough voting. By comparing the orientations and Gaussian scale levels of matched local features, we innovatively infer the object instance center in each frame regardless of scale and orientation changes. It is worth noting that the robustness to scale changes permits the use of high resolution queries even when the targets are small, which can improve accuracy over low resolution queries, as more local feature points are used in matching. Second, we find the globally optimal object trajectories leveraging Max-Path search, which leads to accurate spatio-temporal localization. Although per-frame response can be noisy as we solely rely on local features, Max-Path ensures spatio-temporal smoothness regardless of partial occlusions and classifier noise.

ACKNOWLEDGMENT

This research is in part supported by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority, and the Nanyang Assistant Professorship (SUG M58040015) to Dr. Junsong Yuan.

REFERENCES

[1] D. Tran and J. Yuan, "Optimal spatio-temporal path discovery for video event detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision*, 2004.

[3] H. J. Seo and P. Milanfar, "Training-free, generic object detection using locally adaptive regression kernels," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[4] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[5] J. Meng, J. Yuan, Y. Jiang, N. Narasimhan, V. Vasudevan, and Y. Wu, "Interactive visual object search through mutual information maximization," in *Proc. ACM Multimedia*, 2010.

[6] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[7] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2003.

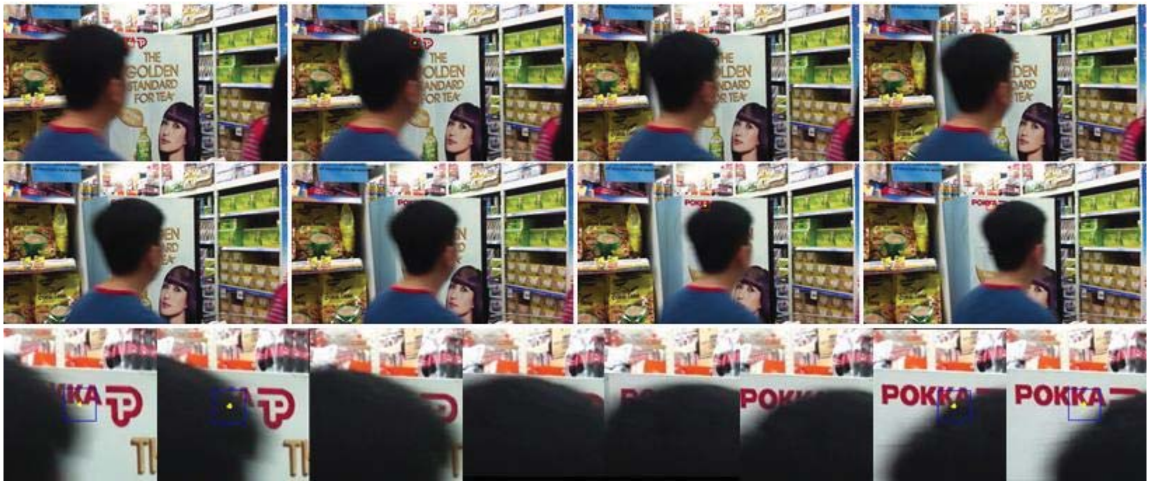

Fig. 1. Object detection with partial occlusions. The first two rows are video key frames and the third row is the enlarged view of the detected locations of Pokka logo.

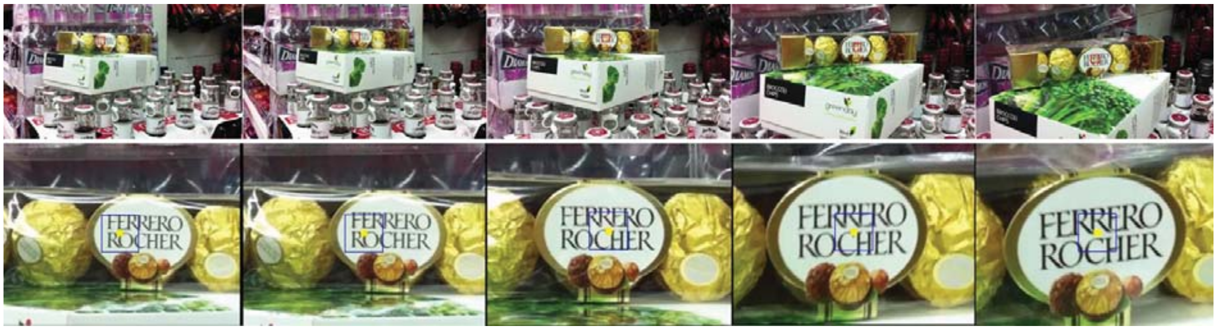

Fig. 2. Object detection with changing view angle and camera zoom. The top row are the video key frames and the bottom row is the enlarged view of the detected locations of Ferrero Rocher logo.

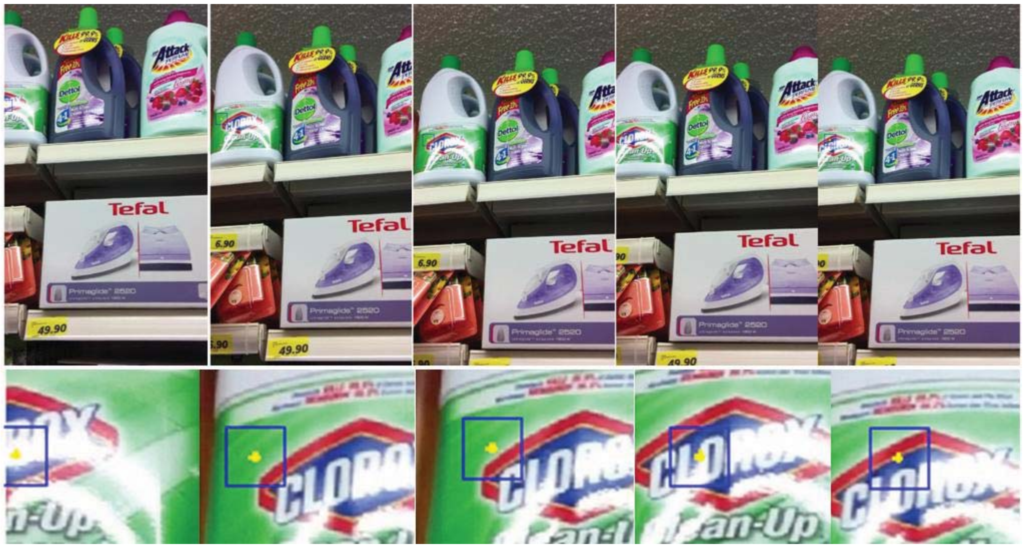

Fig. 3. Object detection with partial occlusions and reflections. The top row are the video key frames and the bottom row is the enlarged view of the detected locations of Clorox logo.