AN OBJECT BASED GRAPH REPRESENTATION FOR VIDEO COMPARISON

Xin Feng[⋆] *Yuanyi Xue*[†] *Yao Wang*[†]

[⋆] College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China.
[†] Dept. of Electrical and Computer Engineering, NYU Tandon School of Engineering, New York, USA.
email: xfeng@cqut.edu.cn, {yxue,yw523}@nyu.edu

ABSTRACT

This paper develops a novel object based graph model for semantic video comparison. The model describes a video with detected objects as nodes, and relationship between the objects as edges in a graph. We investigated several spatial and temporal features as the graph node attributes, and different ways to describe the spatial-temporal relationship between objects as the edge attributes. To tackle the problem of erratic camera motion on the detected object, a global motion estimation and correction approach is proposed to reveal the true object trajectory. We further propose to evaluate the similarity between two videos by establishing the object correspondence between two object graphs through graph matching. The model is verified on a challenging user generated video dataset. Experiments show that our method outperforms other video representation frameworks in matching videos with the same semantic content. The proposed object graph provides a compact and robust semantic descriptor for a video, which can be used for applications such as video retrieval,clustering and summarization. The graph representation is also flexible to incorporate other features as node and edge attributes.

Index Terms— Video representation, object graph, graph matching, video comparison

1. INTRODUCTION

Given a video, the natural way of human understanding begins from the perceived objects. Based on the appearances and moving patterns of individual objects and mutual relations among objects, a human observer can easily recognize what/who are the objects, what are they doing and where are they. However, in most of the current video analysis works, video information is generally characterized by some spatial and temporal features, such as color, texture, SIFT, HOG, HOF (Histogram of Optical Flow), trajectory features. The bag-of-words model (BoW), which treats the low level features as bag of visual words and describes an image/video using the occurrence frequency of words is often used as image/video descriptor. Although much effort has been devoted to develop good feature descriptors, using low level feature descriptors, could not sufficiently differentiate videos at semantic level and fail to consider spatial and temporal relationships of the feature patches for video. More recently, researchers have investigated how to learn spatial and temporal features using deep convolutional neural networks [1, 2], which have shown good performance on video retrieval, action and object recognition. However, these learning based models need large scale annotated data, and the model output cannot provide meaningful and well visualized representation.

In this paper, we present an object based graph model to semantically represent the content of a video, which we term as video object graph (VOG). We mainly target on user generated videos, because these videos are usually with semantic scene content and specific objects, but also accompanied by erratic camera motion, which leverage more challenge for the representation. In order to reveal the true object motion, we propose a global motion estimation method based on minimization of the L1 norm of the error between each two frames of video after global motion compensation. Besides, to efficiently describe the object motion attribute, we propose a couple of trajectory descriptors by analyzing trajectories in the frequency domain to describe the motion pattern. As we will demonstrate in this paper, the proposed VOG model will describe a video mimicing the way of human understanding, and facilitate video matching to measure video similarity in the semantic level.

The remainder of the paper will first present the general idea of the object graph for video representation in Sec. 2. All the node and edge attributes involved in the object graph are then discussed in Sec. 3. The matching of two object graph is given in Sec. 4. Dataset and experiment results are analyzed in Sec. 5. Finally, conclusions are drawn in Sec. 6.

2. THE VIDEO OBJECT GRAPH FRAMEWORK

Graph representation has been extensively exploited for object action recognition in computer vision [3–5], where usually different parts of an object form the nodes, and their relative positions are the edges of a graph. However, the graph is often built from each frame of a video, which leads to a redundant video representation with many repetitive nodes and edges.

In this work, we propose an object based graph representation for describing the content of a video. As illustrated in Fig. 1, two object graphs are constructed from two "sliding" videos, the nodes in the graph encode different objects in a video, described by different spatial and temporal attributes, and the edges encode the relationships between different objects. Here, the original videos are first segmented into separate shots, so that the content in each shot is relatively consistent. To detect the objects, we use the R-CNN framework in [6] trained on Microsoft COCO dataset [7] to find the relevant objects on an exemplar video frame of each video.

The two object graphs, as shown in Fig. 1, facilitate the comparison between the corresponding two videos by using graph matching technique, where the node correspondence could be established by considering the similarities of both the node attributes and the edge features in solving the quadratic assignment problem.

Thanks to National Natural Science Foundation of China for Young Scientists (Grant No. 61502065), Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (Grant No. cstc2015jcyjBX0127), Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant No. KJ1500922) for funding.

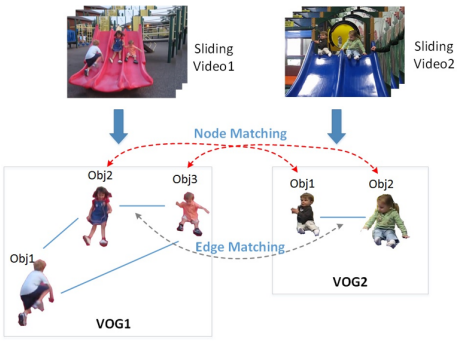

Fig. 1: Illustration of object graph based video representation. Two object graphs are constructed for two videos. The solid blue line in each graph is the edges between adjacent objects, the red dash line denotes the node correspondence, and the gray dash line indicates the edge correspondence.

As seen from Fig. 1, the proposed framework is a compact descriptor, where only several nodes and edges are sufficient to represent a video shot. Besides, the model is in the bottom-up data driven manner, any complicated learning process and large annotated data are not involved.

Formally, given a video segment V, we define an object graph OG to be a tuple graph $OG\{V\} = (O, E)$, where $O = [o_1, ...o_n]$ is a set of objects detected from the video and $E = [e_{1,1}, e_{1,2}...e_{n,n}]$ is a set of edges. Each object has multiple spatial and temporal features, and $e_{i,j}$ describes the relationship between object i and object j.

3. FEATURES IN OBJECT GRAPH

In this section, we give the description of each feature used in the proposed object graph framework.

3.1. Spatial descriptors of objects

We consider two types of features for describing the spatial characteristics of the objects. The first type of feature concerns the contour of the object, which we use the classic shape context descriptor by [8]. Specifically, an exemplar object frame is selected for each object in each video sequence, and the resultant shape context feature is a histogram of 60 bins for the contour pixels in the log-polar diagram.

The other spatial feature characterizes the spatial appearance of the objects in the video. In this work, we leverage the same R-CNN model for object detection described in the previous section and use the output of the first fully-connected layer (*fc6*), which is a 4096 dimension vector to be the other spatial descriptor.

As we found in the experiment, RCNN feature could capture the general appearance of the object, while the contour of the object, e.g. the shape of the "cake", may include distinguish shape information of objects, so the two spatial features are compensated with each other.

3.2. Motion trajectories of objects

Temporal dynamics of different video objects play a central role in differentiating different contents. Naturally, this can be described by the motion trajectories of the objects present in the video.

3.2.1. Object Tracking

Based on R-CNN detected objects (each indicated by a bounding box), we utilize an object tracking algorithm [9] to derive the trajectories of these objects in subsequent frames. Here, instead of updating all 15 PCA components incrementally as the tracking proceeds

(a) (b)

Fig. 2: Trajectories of two objects in a wedding video sequence, (a) shows the original trajectories from object tracking, (b) shows the true object trajectories after global motion correction.

(as done in [9]), we keep the top 5 PCA components derived from the object in the first 15 frames unchanged and only incrementally update the remaining 10 basis. By keeping a part of initial set of basis, it is shown to prevent the algorithm from losing track of the original object of interest, particularly when the object has very fast motion and/or the video footage has low quality.

For multiple objects, we do tracking for each of them separately. The resultant trajectory is a time-series, where each point represents the center coordinate of the object bounding box in a particular frame.

3.2.2. Correcting global motion for object trajectory

The trajectories from the object tracking are affected heavily by the presence of global motion in the video, especially when the original footage was shot by amateurs using hand-held cameras/smartphones. To illustrate this, Fig. 2(a) shows the trajectories obtained for the bride and groom who were walking down the aisle, overlayed on an exemplar frame of the video. It is obvious that the true motion trajectories of the couple shall have a long translation stride along the aisle. The reason that this is not shown in the trajectories produced by the tracking algorithm, is because the person who shot this video was following the couple, which effectively produced a global motion that offset the true motion of the couple.

To reveal the true object motion from the trajectories obtained by the tracking algorithm, we propose a global motion correction procedure. The idea is to find the parametric global motion in each pair of neighboring frames, and iteratively applying the mapping from the old coordinate to the new coordinate. Specifically, denote the temporal midpoint of the video as frame f_{M}. For all frames preceding f_{M}, we find a parametric global motion between two successive frames with the destination frame being the later frame of the pair; for all frames succeeding f_{M}, we find a parametric global motion between two successive frames with the destination frame being the earlier frame of the pair. Fig. **??** illustrates the above procedure. The object center coordinate in each frame is however mapped iteratively to the midpoint frame f_{M}. For example, if frame i is preceding f_{M}, and the object center coordinate in frame i is $\mathbf{X}_i = [u, v]^\top$, the object coordinate with respect to the midpoint frame after global motion correction shall be

$$\begin{aligned} &\mathbf{X}_{i,corr} = \tau_{\mathrm{M}-1}(\tau_{\mathrm{M}-2}(\cdots \tau_{i+1}(\tau_i(\mathbf{X}_i)))) \\ &\text{where} \quad \tau_i(\mathbf{X}_i) = \tau_i\left(\begin{bmatrix} u \\ v \end{bmatrix}\right) = \begin{bmatrix} \alpha_1^i u + \alpha_2^i v + \alpha_3^i \\ \beta_1^i u + \beta_2^i v + \beta_3^i \end{bmatrix}, \\ &\text{and} \quad \{\alpha_{1\ldots3}^i, \beta_{1\ldots3}^i\} \text{ are affine model parameters} \\ &\text{between frame } i \text{ and } i+1. \end{aligned} \tag{1}$$

To solve the parametric global motion between a pair of frames, we propose an L1-norm minimization scheme inspired from [10].

Denote the source frame as $\mathrm{R}(p)$ and the destination frame as $\mathrm{F}(p)$, where p is the pixel coordinates. The global motion estimation problem is essentially to find a motion field $\tau(p)$ that minimizes some cost function of $\mathrm{F}(p) - \mathrm{R}(p \circ \tau)$.

We assume the motion field $\tau(p)$ can be parameterized by an affine model. Furthermore, we only want to consider pixels corresponding to objects with motion different from global motion, so a mask Ω is defined to only count the pixels that do not belong to the object. The mask is obtained by taking the complement of the object masks derived from the tracking algorithm described in the previous section. The resulting parametric global motion estimation problem now becomes:

$$\underset{\tau}{\operatorname{argmin}} \quad \|\mathrm{F}(p) - \mathrm{R}(p) \circ \tau\|_1 \qquad \text{where} \quad p \in \Omega. \tag{2}$$

Following [10], the warping operator τ can be approximated by linearizing around its current status τ_0, i.e. $\mathrm{R}(p) \circ (\tau_0 + \Delta\tau) \approx \mathrm{R}(p) \circ \tau_0 + J\Delta\tau$ assuming $\Delta\tau$ is small. Specifically, in the case of an affine model, we define $\Delta\tau$ and Jacobian J as follows:

$$\begin{aligned} \Delta\tau &= \begin{bmatrix} \Delta\alpha_1 & \Delta\alpha_2 & \Delta\alpha_3 & \Delta\beta_1 & \Delta\beta_2 & \Delta\beta_3 \end{bmatrix}^\top \\ J &\triangleq \frac{\partial}{\partial\tau} \mathrm{vec}(\mathrm{R} \circ \tau)|_{\tau=\tau_0} \in \mathbb{R}^{N\times 6} \\ &= \begin{bmatrix} \frac{\partial R}{\partial\alpha_1} & \frac{\partial R}{\partial\alpha_2} & \frac{\partial R}{\partial\alpha_3} & \frac{\partial R}{\partial\beta_1} & \frac{\partial R}{\partial\beta_2} & \frac{\partial R}{\partial\beta_3} \end{bmatrix}, \end{aligned} \tag{3}$$

where N is the total number of pixels in R.

Equation (2) can then be solved in an alternating steps recursively: at each iteration, the outer loop updates τ and the inner loop solve the linear version of Eq. 2, with respect to $\Delta\tau$.

$$\underset{\Delta\tau}{\operatorname{argmin}} \quad |\mathrm{F} - \mathrm{R} \circ \tau_0 - J\Delta\tau|_1 \tag{4}$$

In this work, we solve Eq. 4 using inexact ADMM. Rewrite the prediction residual as $\mathrm{S} = \Delta\mathrm{F} - J\Delta\tau$, with $\Delta\mathrm{F} = \mathrm{F} - \mathrm{R} \circ \tau_0$. The unconstrained augmented Lagrangian functional $\mathcal{L}$ can be written as follows:

$$\underset{\Delta\tau,\mathrm{S}}{\operatorname{argmin}} \quad \mathcal{L} = |\mathrm{S}|_1 + Y^\top(J\Delta\tau + \mathrm{S} - \Delta\mathrm{F}) + \frac{\mu}{2}\|J\Delta\tau + \mathrm{S} - \Delta\mathrm{F}\|_2^2 \tag{5}$$

In inexact ADMM, Equation (5) can be decomposed into two sub-problems w.r.t S and $\Delta\tau$ respectively, and a Lagrangian update. The two sub-problems are easy to solve, which involve a one-step soft-thresholding and a least squares update.

To illustrate the effectiveness and the necessity of the global motion correction, Fig. 2(b) shows the corrected real object trajectories for the same wedding video as discussed in the beginning of this section. It is clear that after removing the global motion, the resulting trajectories show a clear translation motion as the couple walked down the aisle. We use this corrected object trajectories to derive motion descriptors of the objects as explained next.

3.2.3. Descriptors for object trajectory

From each of the two corrected object trajectories (for the vertical and horizontal directions, respectively), we derive two descriptors based on the Fourier spectrum of the trajectory as our temporal features. Fourier spectrum based descriptors are selected because they are invariant to the trajectory length. Figure 3 shows one object's trajectories, the corresponding Fourier spectra of the two video sequences in the dataset. The top row shows the trajectories of the object in vertical (X-trajectory) and horizontal (Y-trajectory). The image coordinates are normalized to the range of 0 to 1, and their temporal resolution are all 30Hz. The second and the last rows show the first 30 non-DC components in the Fourier spectrum, which shows very distinct differences between the motion of two objects. Specifically, in Fig. 3(b), we see a peak of spectrum at 0.3Hz for X-trajectory and 0.65Hz for Y-trajectory, both of which match the period of the near sinusoidal shape of the trajectories (the video is shot at 30 frames per second). Additionally, in Fig. 3(a), we also observe a peak in Y-trajectory at 0.15Hz that matches the sliding down motion present in the video. Comparing spectrum for X- and Y-trajectories in Fig. 3(b), both show sinusoidal shape but with different scale, the magnitude of the two peaks differed as expected. From this perspective, we use both the peak frequencies, and the magnitude spectrum as our temporal feature descriptors.

We first perform a median filtering of length 5 on each trajectory to smooth the trajectory, and then transformed into two spectra using fast Fourier transform (FFT), one with 200 points (coarse spectrum) and the other with 1000 points (fine spectrum). The coarse spectrum gives less frequency leakage thus better representation of the general trend of the motion, while the fine spectrum helps to identify the peak frequency accurately, which represents the dominant motion pattern of the object. We select the first 30 non-DC components of the coarse spectrum as one descriptor of the motion trajectory.

To find the peak frequencies, the fine spectrum is used and a first-order center difference of the spectrum is calculated. We identify the top three local extrema by ranking the absolute value of the first-order difference. The frequencies corresponding to these three local extrema are recorded and used as another descriptor for the motion trajectory. Overall, for each object, we generate 4 motion descriptors, 2 for the X-trajectory, and 2 for the Y-trajectory with a total dimension of 66.

3.3. Describing object relations on edge attributes

In this work, we choose the trajectory relative difference between two connecting objects as the edge attributes. The trajectory difference is able to reveal the consistency of motion between two objects as well as their spatial displacement. For example, if there are two people moving coherently like dancing together, the trajectory difference of these two objects is likely to be smaller than two people walking along random directions. Similar to the descriptors used for object trajectory, we also use the magnitude of Fourier spectrum of trajectory difference and the top three peak frequencies of spectrum as edge feature descriptors.

4. COMPARING TWO VIDEOS THROUGH MATCHING TWO OBJECT GRAPHS

An essential step in video retrieval, clustering, and summarization is the evaluation of the similarity between any two given video. With the VOG representation of each video $OG\{V_1\}, OG\{V_2\}$, we can reduce this problem to a graph matching process. Graph matching in general is a quadratic assignment problem, where a linear term in the objective function encodes node compatibility functions and a quadratic term encodes edge compatibility functions. We compute two affinity matrices, $\mathbf{K}_o \in \mathbb{R}^{n_1\times n_2}$ and $\mathbf{K}_e \in \mathbb{R}^{m_1\times m_2}$ where $m_1 = \frac{n_1\times(n_1-1)}{2}, m_2 = \frac{n_2\times(n_2-1)}{2}$, for the undirected graph, to respectively measure the similarity of each node pair (o_{i_1}, o_{i_2}) and edge pair $(e_{i_1,j_1}, e_{i_2,j_2})$. The problem of graph matching is to find the optimal binary assignment matrix $\mathbf{X} = [x_{i_1,i_2}]$, such that the summation of corresponding node and edge similarity is maximized:

$$\mathcal{M}(\mathbf{X}) = \mathtt{vec}(\mathbf{X})^T \mathbf{K}\, \mathtt{vec}(\mathbf{X}) \tag{6}$$

Where, $\mathtt{vec}()$ is the vectorization operation, and $\mathbf{K}$ is the large affinity matrix that integrates $\mathbf{K}_o$ and $\mathbf{K}_e$. In this work, we use the

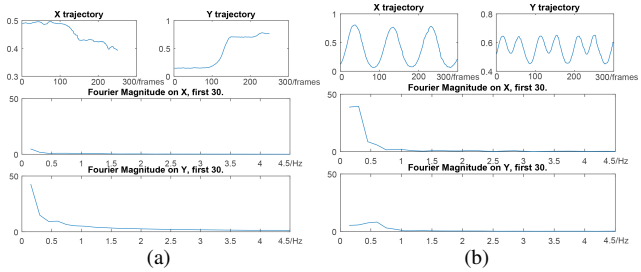

Fig. 3: Trajectories of the objects and their corresponding Fourier spectrums, shown the first 30 non-DC components. All videos are at 30 fps, and the horizontal axis for trajectory plots is frame index. (a) for a boy sliding down in a playground slide video; (b) for a girl swinging in a playground swing video.

Factorized Graph Matching (FGM) [11] method to solve the problem.

To derive the node affinity matrix $\mathbf{K}_o$, we define the following distance metrics for each node feature descriptor. For the spatial CNN feature, and the temporal coarse-scale Fourier spectrum, we directly measure the distance as the Euclidean distance between feature vectors. For temporal fine-scale Fourier spectrum peak feature, we compute the sum of absolute difference. For shape context feature, we compute the similarity using symmetrical Kullback-Leibler divergence.For affinity matrix $\mathbf{K}_e$, we also use Euclidean distance and absolute difference as the distance metric for these the two Fourier spectrum based descriptors, respectively. Finally, all the feature distances are converted to similarities by a exponential decay mapping and fused into a single score as an entry of the affinity matrices.

In this work, the solution of the maximized objective (Eq. 6) is used as the matching score of two videos.

5. EXPERIMENT RESULTS

In this section, we perform the experiment by using the proposed framework on finding semantically similar content from a set of user generated videos, and also compare with two relevant works.

5.1. Dataset

User generated videos usually record the special moments spent with the family and friends (e.g. birthday, wedding, etc.), or some exciting events being witnessed (e.g.soccer plays, kids performance, etc.). Hence, the majority of user generated videos contain human interaction, or interactions with other well-defined objects. In this study, 9 videos belonging to 4 event categories ("Birthday", "Sliding", "Swing", "Wedding") are used in our experiment. The videos are selected from the Columbia Consumer Video (CCV) database [12]We preprocess the selected videos such that each clip contains a single shot with around 10-second length. Most of the videos are in VGA (640×480) at 25 frames per second. Figure 4 shows the exemplar frames of some test videos from the dataset. We can see the videos from the dataset contain both indoor and outdoor environments with a variety of content and motion patterns.

5.2. Performance Evaluation

We evaluate the performance of video matching using the proposed object graph based representation, and compare with the widely used BoW framework and the recent learned spatio-temporal features from 3D convolutional network (C3D) [1].

Fig. 4: Snapshot of sample videos from our dataset.

For BoW, we use the 3D-SIFT descriptor [13] as the low-level visual feature to incorporate both spatial and temporal information. We randomly select 50 frames from each video as training dataset, and a spatial-temporal vocabulary composed of 500 visual words is formulated by using k-means. For C3D, we extract features from fully-connected *fc6* layer (C3D_ *fc6*), based on the recommended setting provided in the paper [1].

For proposed model, the similarity is the maximized objective of graph matching in Eq. 6. For C3D, we use Euclidean distance between compared feature vectors, and for BoW, we measure the distance by KL divergence. The distance measurements are converted to similarity by using the exponential decay mapping.

We report the ratio of intra-class average similarity over inter-class average similarity (*SimRatio*) in Table 1, where classes are the four event categories in our dataset. We can see, the proposed VOG obtains higher *SimRatio* than C3D_ *fc6* and BoW based representations, which indicates its good performance in differentiating videos of different content. To let each video as a query, we make further comparison by computing precision and recall to measure the relevance. The threshold of retrieval numbers is configured as the value of 30% of the descending sorted scores. The final averaged precision and recall over all videos for the three methods are shown in Table 1. The VOG representation also gets better performance in both *precision* and *recall*. The entire experiment demonstrates the advantage of our proposed VOG representation. On the other hand, in terms of computational complexity we compare the time cost of matching two test videos for the three models, from feature extraction to similarity measurement. We can see, because the inference time of the trained C3D model(on GPU) is very fast, it is an efficiency feature representation. BoW is time consuming in feature extraction and vector quantization. The proposed VOG takes some time in the graph matching process, but without the large-scale training steps. Overall, the proposed VOG is more powerful in the semantic level representation of a video.

Table 1: Performance of different video representation methods

	BoW	**C3D_ *fc6***	**VOG (proposed)**
SimRatio	0.87	1.10	1.47
Precision	14.4%	41.7%	68.9%
Recall	33.3%	77.8%	100%
Time Cost(s)	0.59	0.06	0.61

6. CONCLUSIONS

In this paper, we propose a new object based graph representation for videos. It semantically describes a video using a graph, wherein graph nodes describe objects present in a video, and graph edges characterize relationships among objects. Each object is described by a descriptor that specifies its shape, appearance, and motion characteristics. We propose to measure the semantic similarity of two video by solving a graph matching problem. Experiment results demonstrate the advantage of our object graph representation in discriminating different video events than the methods of using deep generic and low-level features.

7. REFERENCES

[1] Fergus R et al Tran D, Bourdev L, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE International Conference on Computer Vision. IEEE, 2015:4489-4497.*, 2014.

[2] Vignesh Ramanathan, Kevin Tang, Greg Mori, and Li Fei-Fei, "Learning temporal embeddings for complex video analysis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4471–4479.

[3] JeongKyu Lee, JungHwan Oh, and Sae Hwang, "Clustering of video objects by graph matching," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 394–397.

[4] Shugao Ma, Leonid Sigal, and Stan Sclaroff, "Space-time tree ensemble for action recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[5] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei, "Image retrieval using scene graphs," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 3668–3678.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014*, pp. 740–755. Springer, 2014.

[8] Serge Belongie, Jitendra Malik, and Jan Puzicha, "Shape context: A new descriptor for shape matching and object recognition," in *NIPS*, 2000, vol. 2, p. 3.

[9] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang, "Online object tracking with sparse prototypes," *Image Processing, IEEE Transactions on*, vol. 22, no. 1, pp. 314–325, 2013.

[10] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma, "Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2233–2246, 2012.

[11] Feng Zhou and Fernando De la Torre, "Factorized graph matching," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 127–134.

[12] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel Ellis, and Alexander C Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. ACM, 2011, p. 29.

[13] Paul Scovanner, Saad Ali, and Mubarak Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 357–360.