Video Object Co-Segmentation via Subspace Clustering and Quadratic Pseudo-Boolean Optimization in an MRF Framework

Chuan Wang, Yanwen Guo, Jie Zhu, Linbo Wang, and Wenping Wang, *Member, IEEE*

***Abstract*—Multiple videos may share a common foreground object, for instance a family member in home videos, or a leading role in various clips of a movie or TV series. In this paper, we present a novel method for co-segmenting the common foreground object from a group of video sequences. The issue was seldom touched on in the literature. Starting from over-segmentation of each video into Temporal Superpixels (TSPs), we first propose a new subspace clustering algorithm which segments the videos into consistent spatio-temporal regions with multiple classes, such that the common foreground has consistent labels across different videos. The subspace clustering algorithm exploits the fact that across different videos the common foreground shares similar appearance features, while motions can be used to better differentiate regions within each video, making accurate extraction of object boundaries easier. We further formulate video object co-segmentation as a Markov Random Field (MRF) model which imposes the constraint of foreground model automatically computed or specified with little user effort. The Quadratic Pseudo-Boolean Optimization (QPBO) is used to generate the results. Experiments show that this video co-segmentation framework can achieve good quality foreground extraction results without user interaction for those videos with unrelated background, and with only moderate user interaction for those videos with similar background. Comparisons with previous work also show the superiority of our approach.**

***Index Terms*—Co-segmentation, subspace clustering, video.**

I. Introduction

UPON its release in 2012, the famous *Titanic 3D*, as its 2D version released in 1997, achieved critical and commercial success. Rolling Stone film critic Peter Travers rated the reissue 3.5 stars out of 4, and said *"The 3D intensifies Titanic."* and *"Caught up like never before in an intimate epic that earns its place in the movie time capsule"*. Behind this success the huge efforts are the cooperative endeavors of a technical team consisting of hundreds of artists and computer engineers. It was reported that the conversion from 2D to 3D took about 60 weeks and $18 million, most of which were spent on segmenting the video frames and extracting prominent foreground and background objects for assigning them depth. Undoubtedly, segmentation of video data into semantic parts is a fundamental research topic for its wide applications, such as visual tracking, video retrieval, compression, and human-computer interaction.

Most existing efforts [1]–[6] concentrate on a single video as input. Segmentation of a single video is a challenging problem, and usually needs user assistance for supplying the sampling of foreground and background or correcting segmentation errors. Moreover, within an individual video accidental similarities in appearance or motion among foreground and background might be so deceptive that lead to ambiguous and even incorrect results easily. For the production of *Titanic 3D*, different clips in the movie generally share common foreground objects with similar appearance. Such a scenario holds for many other video data, for instance a family member in home videos, a hero/heroine in different shots of a TV series, and a famous player in sports videos. This induces us to explore the possibility of segmenting simultaneously the common foreground object from a group of videos. Exploiting the common or similar appearance across different videos may facilitate segmentation, since a group of related videos may provide more information which can be applied to better inference of the foreground. We refer to this problem as video co-segmentation, and co-segmentation of the common foreground objects from multiple related videos is the goal of this paper.

Video co-segmentation, as an emerging research problem, is receiving increasing attention recently. To the best of our knowledge, there are only a few methods [7]–[9] specifically designed to address this problem till now. These methods, however, either phrase it as a multi-class labeling problem [9] thus are not competent for the task of the common object co-segmentation we concern, or make strong assumptions about object motions which significantly limit the applicability. For instance, they usually overuse motions by assuming that several videos contain the common object with similar motions in addition to similar appearance [7], or these videos can be roughly grouped into foreground and background regions according to motion similarity within each video [8]. They generally ignore the fact that

Manuscript received September 02, 2013; revised December 13, 2013; accepted January 27, 2014. Date of publication February 17, 2014; date of current version May 13, 2014. This work was supported in part by the National Natural Science Foundation of China under Grants 61373059 and 61321491, the National Basic Research Program of China (2010CB327903), and the Jiangsu Green Blue Project. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sheng-Wei (Kuan-Ta) Chen.

C. Wang and W. Wang are with the Department of Computer Science, The University of Hong Kong, Hong Kong, China (e-mail: cwang@cs.hku.hk; wenping@cs.hku.hk).

Y. Guo, J. Zhu, and L. Wang are with the National Key Lab for Novel Software Technology, Nanjing University, Nanjing, China (e-mail: ywguo@nju.edu.cn; magickuang@126.com; wanglb.2005@gmail.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org. The material is a video that is 39.3 MB in size.

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.2014.2306393

in multiple videos the common object rarely implies consistent motions as the appearance. Therefore, video object co-segmentation is still a problem that needs to be intensively explored.

In this paper, we propose a novel framework for segmenting the common foreground objects in a group of videos consistently. This is accomplished by subspace clustering on Temporal Superpixels (TSPs) of the input videos and a subsequent Quadratic Pseudo-Boolean Optimization (QPBO) procedure using a Markov Random Field (MRF) model defined on videos. Our framework is applicable to the generic scenario of video object co-segmentation since we do not make further assumptions on the appearance and motions of video foreground and background as previous methods have done.

The common foreground object of different videos should have similar statistics of appearance features. To model foreground appearance, we develop an appearance-motion-fused subspace clustering algorithm to yield initial multi-label clustering results. Multiple appearance and motion features are fed into the subspace clustering algorithm. Although foreground objects across different videos may have quite different motion characteristics, motions can be used to better distinguish different parts within each video, making accurate foreground extraction within each video easier, especially around those ambiguous and low contrast object boundaries. We then for each video build a bag-of-words like descriptor. This implicitly links the common foreground objects with the same semantics across videos. We further formulate video object co-segmentation as an MRF model which imposes the constraint of appearance model of the foreground and is optimized by QPBO.

Contributions: To summarize, our contributions are:

- A novel framework that consistently segments the common foreground objects in a group of videos, and is applicable to generic videos.
- An appearance-motion-fused (amf) co-segmentation algorithm that leverages the appearance and motion features, based on subspace clustering.

The remainder of the paper is organized as follows. The related work is introduced in Section II. Section III gives a high-level overview of our framework. The key components of our framework, preprocessing, subspace clustering, and object co-segmentation by QPBO are presented in Sections IV, V and VI respectively. We evaluate our method in Section VII and conclude the paper finally.

II. Related Work

Our work is inspired by previous work on video segmentation and co-segmentation, image co-segmentation, as well as 3D shape co-segmentation.

A. Video Segmentation and Co-Segmentation

Extraction of video object has received considerable attention over the past decade [10]–[13]. User interactions are more or less required to specify the sampling of foreground and background, and to remove errors caused by inseparable statistics of the foreground and background and temporal discontinuities. Early video cutout methods are generally based on global classifiers [13][14]. Recent efforts seek to extract accurate object boundaries by using local, directional, or combined classifiers [1], [15], followed by foreground matting used to remove remaining errors.

Video co-segmentation has received increasing attention recently. Multi-class video co-segmentation is enabled in [9] by a generative multi-video model. This method realizes multi-class clustering where the number of classes is unknown beforehand, but cannot provide sufficiently accurate segmentation for a common foreground object of different videos. The problem of video object co-segmentation is addressed in [7][8]. Co-segmentation is posed as an optimization problem under a probabilistic framework in [7]. However, its applicability is limited by the dependency on objectness and saliency based initial estimation of foreground as well as the requirement that the foreground object undergoes similar motions across different videos. We compare our video co-segmentation approach with [9], [7] through experimenting with a variety of video examples in the experiments. In [8], the intra-video motion cues and the inter-video appearance model together are taken into account for segmenting the common object in a pair of video sequences. The method may fail for the videos with similar foreground and background motions since it relies on motion-based video grouping for identifying candidate object within each video first. It can be seen that video object co-segmentation is still an emerging research problem to be intensively investigated.

B. Image and 3D Shape Co-Segmentation

The problem of image co-segmentation was first studied by Rother *et al.* [16] and has gained considerable attention in the last few years [17]–[22]. The basic goal is to segment a common salient foreground object from two or more images. Consistency between the extracted object regions is ensured by imposing a global constraint which penalizes variations between the objects' respective histograms or appearance models. However, direct generalization of image co-segmentation methods to video co-segmentation is infeasible since it remains challenging to build a comprehensive foreground model for videos. Furthermore, the scale of video co-segmentation problem itself makes direct extension of image co-segmentation unpractical.

More recently, the concept of co-segmentation has been generalized to the 3D shape segmentation problem [23]–[26]. In [26], Hu *et al.* consider co-segmentation as a clustering problem, which is well solved by subspace clustering fed with multiple geometric features. This inspires us to explore the possibility of realizing video co-segmentation by subspace clustering.

III. Overview

Our video object co-segmentation framework takes a group of videos as input and produces their common foreground as output. It runs in the following steps (See Fig. 1).

Preprocessing: To improve the efficiency of subsequent steps, we over-segment each video into Temporal Superpixels (TSPs). We also compute statistical appearance and motion features on each TSP as its descriptors as will be described in Section IV.

Amf-co-segmentation via subspace clustering: With the feature descriptors, our amf-co-segmentation algorithm integrates

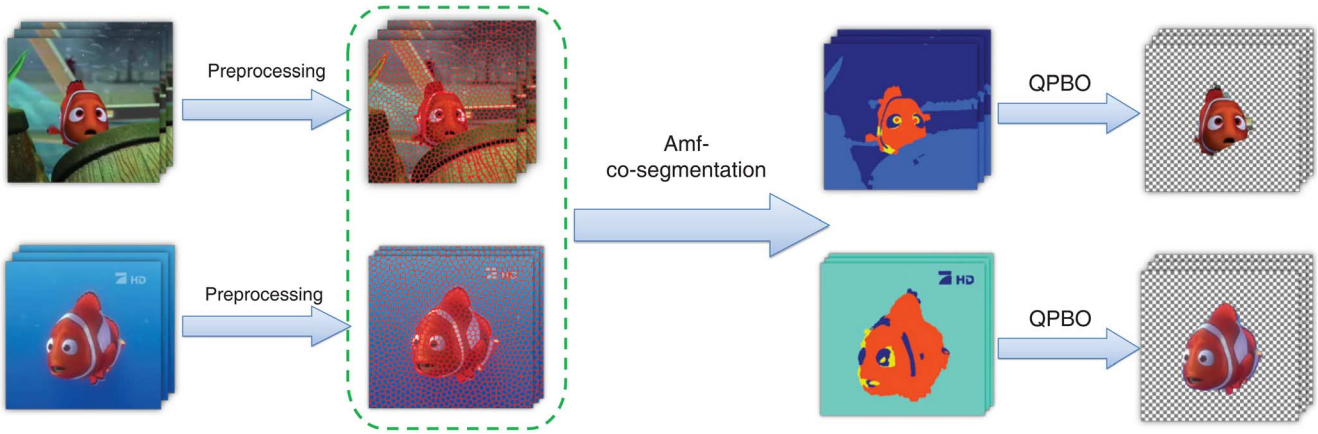


Fig. 1. Workflow of our object co-segmentation framework. Columns from left to right: a video set as input, over-segmentation by TSP, clustering results by our amf-co-segmentation algorithm and object cutout results.

the appearance and motion information, and uses subspace clustering to cluster all TSPs by solving an objective function defined over a unified affinity matrix. This yields multi-label clustering results which in essence are similar to the output of multi-class video co-segmentation problem defined and pursued by [9]. However, we further rely on the output of this step to build a bag-of-words like histogram descriptor for each video, by which the common foreground is linked across different videos implicitly. Our amf-co-segmentation algorithm will be introduced in Section V.

Object co-segmentation via QPBO: We formulate video object co-segmentation as an MRF model which imposes constraint of the common foreground model. Upon the MRF model, we build an undirected graph whose nodes are all TSPs and edges are constructed by considering spatio-temporal consistency in both appearance and motion. We extract the common foreground by Quadratic Pseudo-Boolean Optimization in this MRF framework, and the details are described in Section VI.

IV. Preprocessing

In the preprocessing stage, we over-segment each video into Temporal Superpixels (TSPs) [27] over which multiple statistical appearance and motion features are computed. This process also allows the subsequence steps to operate efficiently.

A. Over-Segmentation with TSP

Even short video sequences contain a large number of pixels. The scale of this problem makes it computationally infeasible to process the data at pixel level. Normally superpixel or supervoxel, as an important preprocessing step, is applied to videos by various algorithms. However, as indicated in [27], off-the-shelf superpixel algorithms running independently on each frame will produce superpixels that are unrelated across time, and supervoxel is not specifically designed for video data.

In this paper, we use the generative probabilistic model [27] to over-segment each video into Temporal Superpixels (TSPs). Each TSP is a set of local video pixels in space and tracks the same part of an object across time. Note that, to accommodate large motions, SIFT flow [28] instead of optical flow originally applied by [27] is used to relate segments between two successive frames. This benefits us since SIFT flow establishes more robust feature correspondences than optical flow. Thus more consistent TSPs across frames and more accurate motion trajectories of them can be obtained.

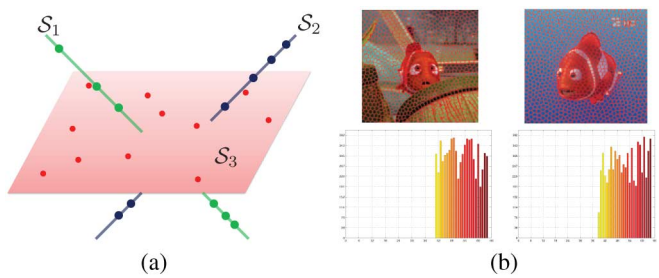


Fig. 2. Subspace clustering. (a) An illustration of subspace clustering in $\mathbb{R}^3$. (b) The appearance feature, i.e. HSV histograms of two TSPs belonging to the orange clownfish. Due to the similar distributions, these two feature vectors are in a common subspace spanned by standard basis corresponding to the nonzero bins.

B. Feature Description

We extract for each TSP the appearance features and its motion trajectory.

Appearance features: We compute the raw features including 17-D texture by Winn filter bank [29] as well as HSV color at pixel level. Gaussian mixture model and discretization of feature space are then applied to the texture and color features, separately. We thus have two kinds of feature distributions, each of which describes the TSP with a histogram. An appearance feature vector is formed on each TSP by concatenating the two histograms. It is observed that TSPs from the common object of interest have similar feature distributions. As a result, these feature vectors are likely to be in common subspaces generated by standard basis corresponding to the nonzero bins. Fig. 2(b) shows an example.

Motion trajectory: Multiple spatio-temporal regions corresponding to different motions in a video can be separated by subspace clustering [30] because the 2D trajectories associated with a single rigid motion live in a 3D subspace under the affine camera model. In this paper, given a video sequence with F

frames, the 2D trajectories of all TSPs are represented as a $2F \times N$ matrix

$$T = \begin{bmatrix} c_{11} & \cdots & c_{1N} \\ \vdots & \vdots & \vdots \\ c_{F1} & \cdots & c_{FN} \end{bmatrix} \tag{1}$$

where i-th column is the trajectory of its corresponding TSP, and c_{fi} is the center $[c_{fi_x}, c_{fi_y}]^{\mathrm{T}}$ of the intersection of TSP and frame-f. Noted that TSPs produced by [27] cannot guarantee each one passing through all frames due to newly appeared or vanished objects, or large motions not always tracked. For the remaining incomplete trajectories, we simply fill the empty entries by copying their nearest non-empty entry in the corresponding column. Our experiments show this scheme simple but works well in most cases.

V. Appearance-motion-fused Co-segmentation via Subspace Clustering

In spite of possible variance due to changes in illumination, view angles, and non-rigid object motions, the foreground objects in different videos should have similar statistics of appearance features, which is a vital cue to relate them with each other. Besides, object motions within each video can facilitate differentiating foreground from background even though they may be inconsistent across different videos. To fully utilize the two kinds of features, we have developed an appearance-motion-fused (amf) co-segmentation algorithm whose core is subspace clustering. The TSPs of all videos are grouped into clusters with which a bag-of-words like histogram descriptor for the common video object can be obtained.

We first briefly introduce the background of subspace clustering. Then our appearance-motion-fused co-segmentation algorithm which is applicable to a group of videos is described in detail.

A. General Subspace Clustering

The problem of subspace clustering aims at clustering data vectors into multiple subspaces and finding a low-dimensional subspace fitting each cluster of vectors [31].

Notation: Given a feature data matrix $X = [x_1, x_2, \ldots, x_N]$ each column of which is a feature sample $x_i \in \mathbb{R}^D$ drawn from a union of P subspaces $\{\mathcal{S}_i\}_{p=1}^{P}$ of unknown dimensions, subspace clustering aims to segment the data vectors into their respective subspaces. Fig. 2(a) illustrates subspace clustering in $\mathbb{R}^3$.

Low-Rank Representation (LRR) solution: Since the data vectors of X are drawn from a union of P subspaces, each of them can be represented by the linear combination of X itself as the basis such that $X = XZ$ with $Z = [z_1, z_2, \ldots, z_N]$. The LRR algorithm [32] is based on the observation that Z should be low-rank because each data vector can always be represented by a sparse linear combination of the ones belonging to the same linear subspace. Consequently, the low-rank representation can be obtained by solving the following problem

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_{2,1}, \qquad \text{s.t.} \quad X = XZ + E \tag{2}$$

where $\|\cdot\|_*$ represents the nuclear norm (sum of the singular values) and $\|\cdot\|_{2,1}$ denotes the $\ell_{2,1}$-norm [32] for characterizing noise E. $\lambda > 0$ is a parameter balancing the influences of the two parts.

According to [32], an affinity matrix that encodes the pairwise affinities among data vectors naturally derives from the solution Z^* of problem (2). Specifically, the affinity $(\mathbf{S})_{ij}$ of two data vectors x_i and x_j can be calculated by

$$(\mathbf{S})_{ij} = |(Z^*)_{ij}| + |(Z^*)_{ji}| \tag{3}$$

where $(\cdot)_{ij}$ denotes the (i,j)-th entry of the matrix. With such an affinity matrix, spectral clustering algorithm NCut [33] is applied to get the final clustering result.

B. Amf-Co-Segmentation to a Video Group

LRR algorithm as mentioned above, is originally designed to handle a single type of feature. In our formulation of the video co-segmentation problem, since appearance and motion features need to be taken into account simultaneously but treated unequally, it cannot be directly used here. For ease of exposition, we first formulate the video co-segmentation problem as follows.

Given L videos, each of which has been over-segmented into n_l TSPs, $l = 1, 2, \ldots, L$. Thus there are $N = \sum_{l=1}^{L} n_l$ TSPs in total. For each TSP, we compute K features including appearance and motion as mentioned in Section IV-B, so as to get K feature matrices $\{X_k\}(k = 1, 2, \ldots, K)$, where X_k is $D_k \times N$ and D_k is the dimension of k-th feature. Recall that we compute the 17-D texture, HSV color histogram and motion trajectories as the features of each TSP, then K is set to 3 in all our experiments. To distinguish X_k and Z_k belonging to appearance and motion features, we tag them with $\mathcal{A}$ or $\mathcal{M}$ so that X_k or Z_k belonging to appearance/motion feature can be written as X_k or $Z_k \in \mathcal{A}/\mathcal{M}$. The goal of our amf-co-segmentation is to separate all TSPs into P clusters with all X_k as input, such that the common foreground has consistent labels of clusters across different videos.

To fully utilize appearance and motion features jointly within each video but treat them differently across videos, our amf-co-segmentation algorithm is proposed to consider multiple features with a penalty term imposed on them by solving the following optimization problem:

$$\begin{aligned} \min_{\substack{Z_1,\ldots,Z_K \\ E_1,\ldots,E_K}} & \sum_{k=1}^{K} (\|Z_k\|_* + \lambda \|E_k\|_{2,1}) + \alpha\, \mathcal{P}_{\mathrm{amf}}(Z_1, \ldots, Z_K) \\ & \text{s.t.} \quad X_k = X_k Z_k + E_k,\ k = 1, \ldots, K \end{aligned} \tag{4}$$

where $\alpha > 0$ is a parameter set to 1×10^{-5} in our experiments and $\mathcal{P}_{\mathrm{amf}}$ is the consistent penalty term. The part of summation in Problem (4) is to apply LRR to each feature. If no other constraint is introduced, these features will actually work independently, causing each pair of data vectors to have various affinities corresponding to K features. The introduced $\mathcal{P}_{\mathrm{amf}}$ aims to infer a unified affinity matrix by seeking the sparsity-consistent

low-rank affinities Z_k over appearance and motion feature matrices jointly but freezing the effect of motion feature within each video simultaneously.

The structure of X_k and Z_k: Since every feature matrix X_k is constructed with concatenated features $X_k^{(l)}$ belonging to the l-th video, it can be represented as a block matrix as follows

$$X_k = \left[X_k^{(1)}, X_k^{(2)}, \ldots, X_k^{(L)}\right] \tag{5}$$

Then $X_k Z_k$ can be written as

$$X_k Z_k = \left[X_k^{(1)}, \cdots, X_k^{(L)}\right] \begin{bmatrix} Z_k^{(1,1)} & \cdots & Z_k^{(1,L)} \\ \vdots & \ddots & \vdots \\ Z_k^{(L,1)} & \cdots & Z_k^{(L,L)} \end{bmatrix} \tag{6}$$

where Z_k is also represented as a block matrix. The diagonal blocks in Z_k, i.e. $Z_k^{(l,l)}(l = 1, 2, \ldots, L)$ encode the affinities of data vectors within l-th video; and the non-diagonal ones, i.e. $Z_k^{(l,m)}(l \neq m)$ encode the affinities of data vectors across the l-th and m-th videos. As aforementioned, across multiple videos appearance features should be similar but there is no guarantee on motion, therefore for $Z_k \in \mathcal{A}$, its diagonal and non-diagonal blocks should have equal status, which is however not true for $Z_k \in \mathcal{M}$. Specifically, in this case, since the motion similarity across videos is not reliable, all the affinity information in blocks $Z_k^{(l,m)}(l \neq m)$ cannot be treated equally as $Z_k^{(l,l)}$. To reserve the valid information only, we use a mask matrix which is of the same size as Z_k

$$(M)_{ij} = \begin{cases} 1 & \text{if } (\mathrm{Z_k})_{\mathrm{ij}} \text{ locates at diagonal block,} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Let $\circ$ be the element-wise product, then $M \circ Z_k$ sets all non-diagonal blocks in Z_k to zero, meaning that motion affinities across different videos are filtered out.

The penalty $\mathcal{P}_{\mathrm{amf}}$: Within each video, the affinity of a pair of data vectors should be consistent under all feature descriptors including motion. Across different videos, the affinity should be consistent only on appearance features. We therefore define the penalty term $\mathcal{P}_{\mathrm{amf}}$ as follows,

$$\mathcal{P}_{\mathrm{amf}} = \|\mathbf{Z}\|_{2,1} \tag{8}$$

where

$$\mathbf{Z} = \begin{bmatrix} (\mathcal{T}Z_1)_{11} & (\mathcal{T}Z_1)_{12} & \cdots & (\mathcal{T}Z_1)_{NN} \\ (\mathcal{T}Z_2)_{11} & (\mathcal{T}Z_2)_{12} & \cdots & (\mathcal{T}Z_2)_{NN} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathcal{T}Z_K)_{11} & (\mathcal{T}Z_K)_{12} & \cdots & (\mathcal{T}Z_K)_{NN} \end{bmatrix} \tag{9}$$

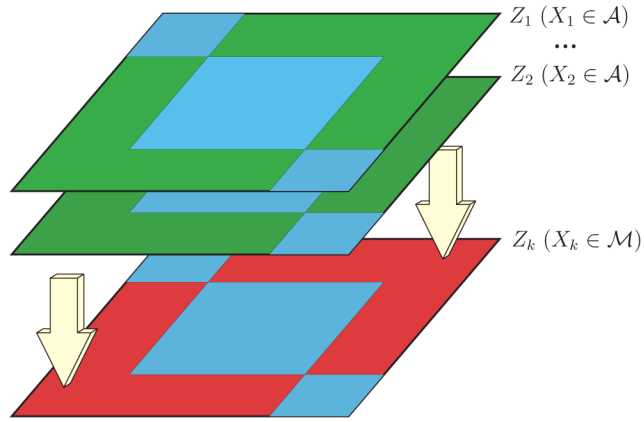


Fig. 3. Construction of the matrix $\mathbf{Z}$. Blue represents the diagonal blocks corresponding to intra-video affinities $Z_k^{(l,l)}$. Green and red denote inter-video affinities $Z_k^{(l,m)}(l \neq m)$ corresponding to appearance and motion features. When constructing $\mathbf{Z}$, the inter-video affinities of matrices Z_k corresponding to appearance features replace those belonging to Z_k of motion features, which means that motion affinities take effect as appearance ones within each video only.

is a $K \times N^2$ matrix formed by concatenating $\mathcal{T}Z_k$. $\mathcal{T}Z_k$ is defined as

$$\mathcal{T}Z_k = \begin{cases} Z_k & \text{if} \quad Z_k \in \mathcal{A} \\ M \circ Z_k + \frac{(\mathbf{J}-M)\circ\sum_{X_l \in \mathcal{A}} Z_l}{|\mathcal{A}|} & \text{otherwise} \end{cases} \tag{10}$$

where $\mathbf{J}$ is an all-one matrix and $|\cdot|$ is the cardinality of a set.

$\mathcal{T}$ operates on Z_k in the following manner. For the affinities $Z_k \in \mathcal{A}$, the output is actually Z_k itself, meaning that all the entries in appearance affinity matrix are equally valid. However for $Z_k \in \mathcal{M}$,since its inter-video affinities are not reliable, we first use the mask defined by Equation (7) to filter out the intra-video motion affinities, setting the inter-video affinities to zero. Then we further use the average inter-video appearance affinities to fill up the yielded zero entries. The filling step after filtering is necessary because the yielded zero affinity in $Z_k \in \mathcal{M}$ will strongly keep other $Z_k \in \mathcal{A}$ from taking effect across videos, due to its strong potential tend informing that the corresponding data vectors are not similar. With the operation $\mathcal{T}$, the intra-video parts of $Z_k \in \mathcal{M}$ remain unchanged while the inter-video parts (non-diagonal blocks) resemble their counterparts in $Z_k \in \mathcal{A}$.

The $\ell_{2,1}$ norm on $\mathbf{Z}$ is defined by

$$\|\mathbf{Z}\|_{2,1} = \sum_{j=1}^{N^2} \|\mathbf{Z}(*, j)\|_2 \tag{11}$$

where $\mathbf{Z}(*, j)$ is the j-th column of $\mathbf{Z}$ and $\|\cdot\|_2$ is the ℓ_2 norm. $\ell_{2,1}$ norm can induce column sparsity of $\mathbf{Z}$, meaning each pair of data vectors should have consistent affinities in terms of appearance and motion features. As a result, the sparsity-consistency of $\mathcal{T}Z_k(k = 1, 2, \ldots, K)$ is guaranteed. At the same time, the inter-video motion affinities are shielded, only leaving

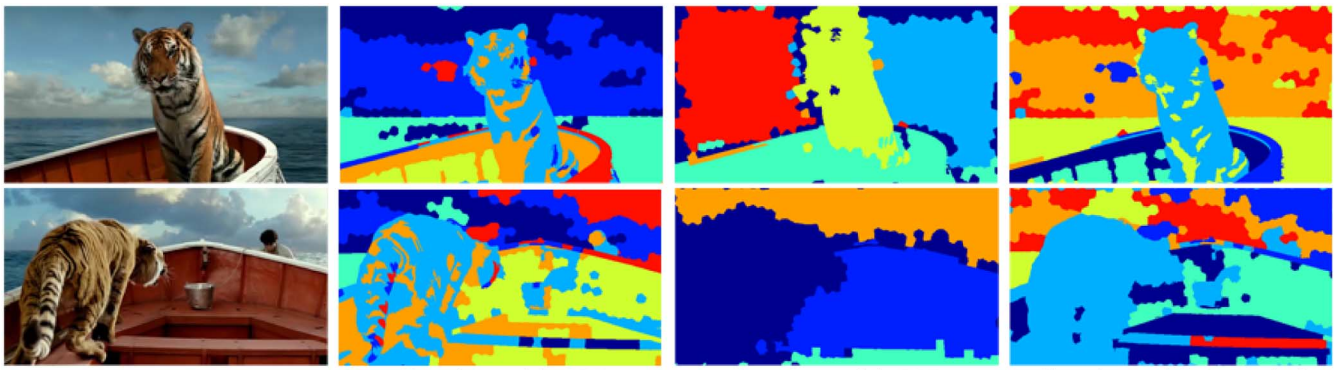


Fig. 4. An example of the amf-co-segmentation results on a pair of video shots from the film *Life of Pi*. Columns from left to right: two frames from each video, clustering results by LRR with appearance feature only, results by LRR with motion feature only, and the results by our amf-co-segmentation algorithm. In each column, the same color represents the same class. Obviously, our amf-co-segmentation generates more consistent clustering results on the common foreground tiger.

the intra-video ones assist to distinguish foreground from background.

Fig. 3 illustrates the construction of $\mathbf{Z}$.

Optimization: Problem (4) is convex and can be solved with the augmented Lagrange multiplier (ALM) method [34]. First, it is converted into the following equivalent problem

$$\min_{\substack{\{J_k\},\{S_k\}\\ \{Z_k\},\{E_k\}}} \sum_{k=1}^{K}(\|J_k\|_* + \lambda\|E_k\|_{2,1}) + \alpha\|\mathbf{Z}\|_{2,1} \tag{12}$$

$$\text{s.t.} \quad X_k = X_k S_k + E_k,\ Z_k = J_k,\ Z_k = S_k,\ k = 1, 2, \ldots, K \tag{13}$$

Then Problem (12) can be solved by the so-called alternating direction method (ADM) [34], listed in Algorithm 1. Note that the sub-problems of the algorithm are convex with closed-form solutions. Step 1 is solved via the singular value thresholding operator [35], while steps 3 and 4 are solved via Lemma 4.1 of [32].

Let $(Z_1^*, Z_2^*, \ldots, Z_K^*)$ represent the optimal solution to the objective function (4), the unified affinity matrix $\mathbf{S}$ is constructed with

$$(\mathbf{S})_{ij} = \frac{1}{2}\left(\sqrt{\sum_{k=1}^{K}(\mathcal{T}Z_k^*)_{ij}^2} + \sqrt{\sum_{k=1}^{K}(\mathcal{T}Z_k^*)_{ji}^2}\right) \tag{14}$$

NCut is applied to this affinity matrix to produce the co-segmentation result that groups all TSPs into clusters corresponding to the subspaces.

Fig. 4 shows an simple example by our algorithm. Note that in column 2, multiple classes are produced on the tiger, when only HSV color histogram is used as feature. In column 3, when only motion trajectory is taken as the feature, it distinguishes different semantic regions in each video, but lacks consistent labels across the input videos. The tiger is labeled as yellow in the top row but as deep blue in the bottom. This shows that the common foreground do not necessarily presents consistent motions across different videos. In column 4, our amf-co-segmentation yields more consistent clustering results across the two videos. The tiger is labeled as blue in both videos. We further compare the final cutout results by motion-excluded and motion-included features in the experiment of unsupervised object co-segmentation with more examples, please refer to Section VII-B1 for more details.

Algorithm 1 Solving Problem (12) by ADM

Inputs: Feature Matrices $\{X_k\}(k = 1, 2, \ldots, K)$, parameters λ and α

while not converged **do**

1. Fix the others and update $\{J_k\}$ $(k = 1, 2, \ldots, K)$ by

$$J_k = \arg\min_{J_k} \frac{\|J_k\|_*}{\mu} + \left\|J_k - \left(Z_k + \frac{W_k}{\mu}\right)\right\|_F^2$$

2. Fix the others and update $\{S_k\}$ $(k = 1, 2, \ldots, K)$ by

$$S_k = \left(\mathbf{I} + X_k^T X_k\right)^{-1}\left(X_k^T(X_k - E_k) + Z_k + \frac{X_k^T Y_k + V_k - W_k}{\mu}\right)$$

3. Fix the others and update $\mathbf{Z}$ by

$$\mathbf{Z} = \arg\min_{\mathbf{Z}} \frac{\alpha}{\mu}\|\mathbf{Z}\|_{2,1} + \|\mathbf{Z} - \mathbf{Q}\|_F^2$$

where $\mathbf{Q}$ is a $K \times N^2$ matrix formed as follows:

$$\mathbf{Q} = \begin{bmatrix} (\mathcal{T}Q_1)_{11} & (\mathcal{T}Q_1)_{12} & \cdots & (\mathcal{T}Q_1)_{NN} \\ (\mathcal{T}Q_2)_{11} & (\mathcal{T}Q_2)_{12} & \cdots & (\mathcal{T}Q_2)_{NN} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathcal{T}Q_K)_{11} & (\mathcal{T}Q_K)_{12} & \cdots & (\mathcal{T}Q_K)_{NN} \end{bmatrix}$$

where $Q_k = (J_k + S_k - (W_k + V_k)/\mu)/2,\ k = 1, 2, \ldots, K$ and $\mathcal{T}$ is the same defined as Equation (10).

4. Fix the others and update $\{E_k\}$ $(k = 1, 2, \ldots, K)$ by

$$E_k = \arg\min_{E_k} \frac{\lambda}{\mu}\|E_k\|_{2,1} + \left\|E_k - \left(X_k - X_k S_k + \frac{Y_k}{\mu}\right)\right\|_F^2$$

5. Update the multipliers

$$\begin{aligned} Y_k &\leftarrow Y_k + \mu(X_k - X_k S_k - E_k) \\ W_k &\leftarrow W_k + \mu(Z_k - J_k) \\ V_k &\leftarrow V_k + \mu(Z_k - S_k) \end{aligned}$$

6. $\mu \leftarrow \min(1.1\mu, 10^{10})$.

7. Check the convergence conditions:

$$\begin{aligned} (X_k - X_k S_k - E_k) &\to 0 \\ (Z_k - J_k) &\to 0 \\ (Z_k - S_k) &\to 0, \quad k = 1, 2, \ldots K \end{aligned}$$

end while

Output: Z

VI. Object Co-Segmentation

The amf-co-segmentation stage actually yields a bag-of-words like histogram description for each video. We further formulate video object co-segmentation as a binary labeling problem that aims to extract the common foreground simultaneously. It is achieved by defining a Markov Random Field (MRF) model on TSPs, while imposing the constraint of foreground model. Object co-segmentation is achieved by Quadratic Pseudo-Boolean Optimization (QPBO) under this MRF framework.

A. Estimation of Video-Level Foreground Histogram

All TSPs of the video group have been clustered into P classes corresponding to the P subspaces after the amf-co-segmentation stage. This results in a bag-of-words [36] like histogram description for each video. Let us define a histogram matrix H^l of size $P \times n_l$ for video-l, $l = 1, 2, \ldots, L$ such that

$$(H^l)_{bj} = \begin{cases} 1 & \text{if the } j\text{-th TSP is in the } b\text{-th class} \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Its histogram description h^l is thus $h^l = H^l\mathbf{1}$ where $\mathbf{1}$ is a $n_l \times 1$ column vector of all ones. The foreground histogram h_f^l is $H^l y^l$ where y^l is a binary $n_l \times 1$ column vector for $(y^l)_i = 1$ if the i-th TSP is foreground and $(y^l)_i = 0$ otherwise.

The common foreground of all videos is supposed to be nearly identical by eliminating the scale difference. That is to say, if we stack all the latent foreground histograms into a matrix $\mathbf{H}_f = [h_f^1, h_f^2, \ldots, h_f^L]$, it should be closely rank-one. We further apply Rank One Decomposition to $\mathbf{H} = [h^1, h^2, \ldots, h^L]$ to get the approximation of $\mathbf{H}_f$ noted as $\hat{H} = [\hat{h}^1, \hat{h}^2, \ldots, \hat{h}^L]$, where each column is an estimation of the foreground histogram of video-l.

B. QPBO in the MRF Framework

For each video sequence l, an undirect graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is built whose nodes $\mathcal{V}$ are TSPs and edges $\mathcal{E}$ connect relevant TSPs. There are two cases that a pair of TSPs are linked by an edge. The first is that they fall into the same class by our amf-co-segmentation algorithm. The second is that they are spatio-temporally adjacent to each other.

The binary labelling problem is to assign a unique label $y_i \in \{0(\text{background}), 1(\text{foreground})\}, i = 1, 2, \ldots, n_l$ for each node such that the energy $E^l(y)$ can be minimized:

$$\min_{y^l} E^l(y^l) = \underbrace{\sum_{i\in\mathcal{V}} E_{\mathrm{d}}^l(y_i^l) + \sum_{(i,j)\in\mathcal{E}} E_{\mathrm{s}}^l(y_i^l, y_j^l)}_{\text{MRF Gibbs Energy}} + \gamma \left\|H^l y^l - \hat{h}^l\right\|_2^2 \tag{16}$$

where the first two terms on the left are data and smooth terms respectively, derived from the traditional MRF Gibbs energy. $\gamma > 0$ is a parameter penalizing the variation between the latent foreground histogram $H^l y^l$ and the given estimation $\hat{h}^l$. Obviously, the rightmost term aims to reinforce the similarity between the foreground histogram $H^l y^l$ induced by the optimal y^l and $\hat{h}^l$. With this term we show that with moderate or even no user intervention, the common foreground can be extracted simultaneously from a group of videos.

Equation (16) is called the Quadratic Pseudo-Boolean function, which is sub-modular and can be solved with roof duality by the QPBO algorithm. We refer readers to [37] for a detailed description of QPBO. Note that roof duality may produce unlabeled TSPs. In implementation as we found unlabeled TSPs usually belong to background, we just labelled them as background in all our experiments. This method is simple but works well in practice.

In our implementation, the smooth term is set as follows

$$E_{\mathrm{s}}^l(y_i^l, y_j^l) = \beta_{ij}|y_i^l - y_j^l| \tag{17}$$

where β_{ij} is set to the average color similarity of TSP i and TSP j. The data term is set to be a constant if no user interaction is involved. While for the sake of interactivity of our program, it will be set by the following manner if the user specifies strokes indicating the sampling of foreground and background on frames

$$E_{\mathrm{d}}^l(y_i^l) = \begin{cases} p(x_i^l|\mathcal{F})/(p(x_i^l|\mathcal{F}) + p(x_i^l|\mathcal{B})) & \text{if } y_i^l = 0 \\ 1 - E_{\mathrm{d}}^l(0) & \text{otherwise} \end{cases} \tag{18}$$

where $\mathcal{F}$ and $\mathcal{B}$ represent foreground and background models learned from user-specified samples and $p(x_i^l|\cdot)$ denotes the likelihood of TSP i under the corresponding model, based on its color feature x_i^l.

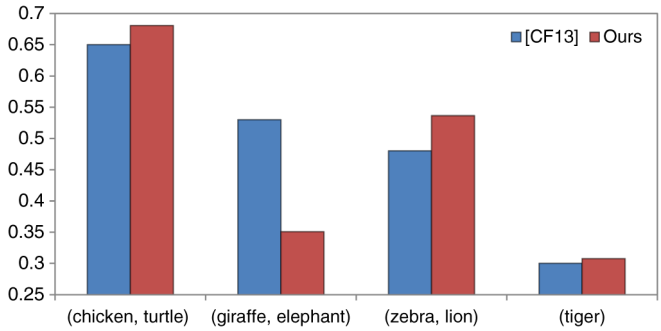


Fig. 5. Comparison of co-segmentation accuracies between our method and [9] on MOViCS dataset.

TABLE I
RUNTIME COMPARISON BETWEEN [9] AND OUR METHOD. #: INCLUDING OPTICAL FLOW + SUPERPIXEL + FEATURES. *: INCLUDING SIFT FLOW + TSP + FEATURES. THE DATA OF [9] IS GOT FROM ITS SUPPLEMENTARY MATERIAL

	Preprocessing		Clustering	
	[9]#	Ours*	[9]	Ours
(chicken, turtle)	1h 40m	1h 10m	1h 12m	1h 04m 13s
(giraffe, elephant)	1h 33m	54m	1h 22m	13m 15s
(zebra, lion)	3h 19m	2h 14m	3h 13m	1h 23m 20s
(tiger)	1h 11m	48m	59m	30m 18s

VII. EXPERIMENTS

A. Evaluation on the Clustering Results

Since we formulate video co-segmentation as a clustering problem in Section V, even though it is not the final goal of our purposed algorithm, we still evaluate the effectiveness of our approach by comparing our results against those by [9] at the very beginning.

In [9], a multi-class video co-segmentation dataset MOViCS which contains 11 videos belonging to 4 groups is released and an evaluation method that measures the average accuracy of the best matching clusters to each classes in ground truth is proposed. Specifically, the metric is defined as

$$\text{Score} = \frac{1}{C}\sum_{j}\max_{i}\frac{S_i \cap G_j}{S_i \cup G_j} \tag{19}$$

where S_i is a set of segments belonging to Class i, G_j is the set of segments of Class j in ground truth, and C is the number of classes in ground truth.

In our experiments, we use the same criterion to compare the clustering results. Besides, we compare the runtime under the same computer configuration Intel Core 2 Duo E8500 @ 3.16 GHz with 8 GB RAM as used in [9]. Fig. 5 illustrates the performance scores and Table I shows the runtime. From Fig. 5, we can see for 3 out of 4 video groups our method produces better, or at least comparable clustering results except the 2nd video group, which will be further discussed as a failure case in the later subsection. However, as our method treats the TSPs instead of superpixels in all frames as the basic units so that data size is much reduced, it runs much faster than [9], as shown in Table I.

TABLE II
PRECISION AND RECALL OF CUTOUT RESULTS BY MOTION-EXCLUDED AND MOTION-INCLUDED FEATURE

	Motion-excluded		Motion-included	
	Precision	Recall	Precision	Recall
BlackCar	0.8423	0.8957	0.9634	0.9703
FuzzyToy	0.7989	0.8943	0.9821	0.9857
Hobbits	0.7960	0.8480	0.9729	0.9729
Baby	0.8916	0.8137	0.9748	0.9904

B. Unsupervised Object Co-Segmentation

Our framework can produce object co-segmentation results for videos with unrelated backgrounds, in a fully unsupervised manner. Figs. 6 and 7 show eight groups of results on a wide range of videos by our system, without any user guidance. Each of the video groups *BlackCar*, *FuzzyToy*, *Hobbits*, *Baby* and *Nemo* consists of two source videos, while each of the rest ones comprises three videos. The resolutions of input videos range from 344×327 to 1280×536 and numbers of frames range from 30 to 127. Most foreground objects have distinct motions from the background, nevertheless, foreground motions across different videos in the same group are not necessarily similar. Rows 7 and 8 in Fig. 6 show such an example where the father and his baby present quite different motions in the two source videos. All the results are shown in our accompanying video.

1) Motion-Excluded vs. Motion-Included: As a complement to appearance, motion information is fed into the subspace clustering algorithm for better differentiating foreground and background within each video. To validate this, Fig. 6 shows four groups of results in each group of which we compare the performance of subspace clustering and the final cutout results with and without motion feature used. Note that in the video group *BlackCar*, the black running car in the 2nd video has similar colors and textures to the ground due to the impact of backlight shot, causing it difficult to differentiate them with appearance features only. Accordingly, we can see from the 2nd column showing the clustering results, the ground and the car are clustered together. Furthermore, they are extracted simultaneously in the final cutout results as shown in the 4th column. By comparison, they are grouped into different clusters by our motion-included subspace clustering, and as a result, the car is successfully segmented from the video. Such a case also holds for the rest three video groups. Table II shows that incorporating motion into our framework improves both precision and recall on the four video groups over the motion-excluded implementation. The performance of both clustering and cutout is improved by ten to twenty percents, varying according to different video groups.

2) The method in [7] vs. ours: We further compare our algorithm with the video object co-segmentation method purposed in [7]. Unlike [7] which only accepts the videos depicting an object with similar motions in addition to similar appearance, and heavily relies on an initial estimation of the foreground and background labelling based on objectness and saliency detection, our method is more suitable for generic video data which only requires similar visual appearance of the common foreground in the video group.

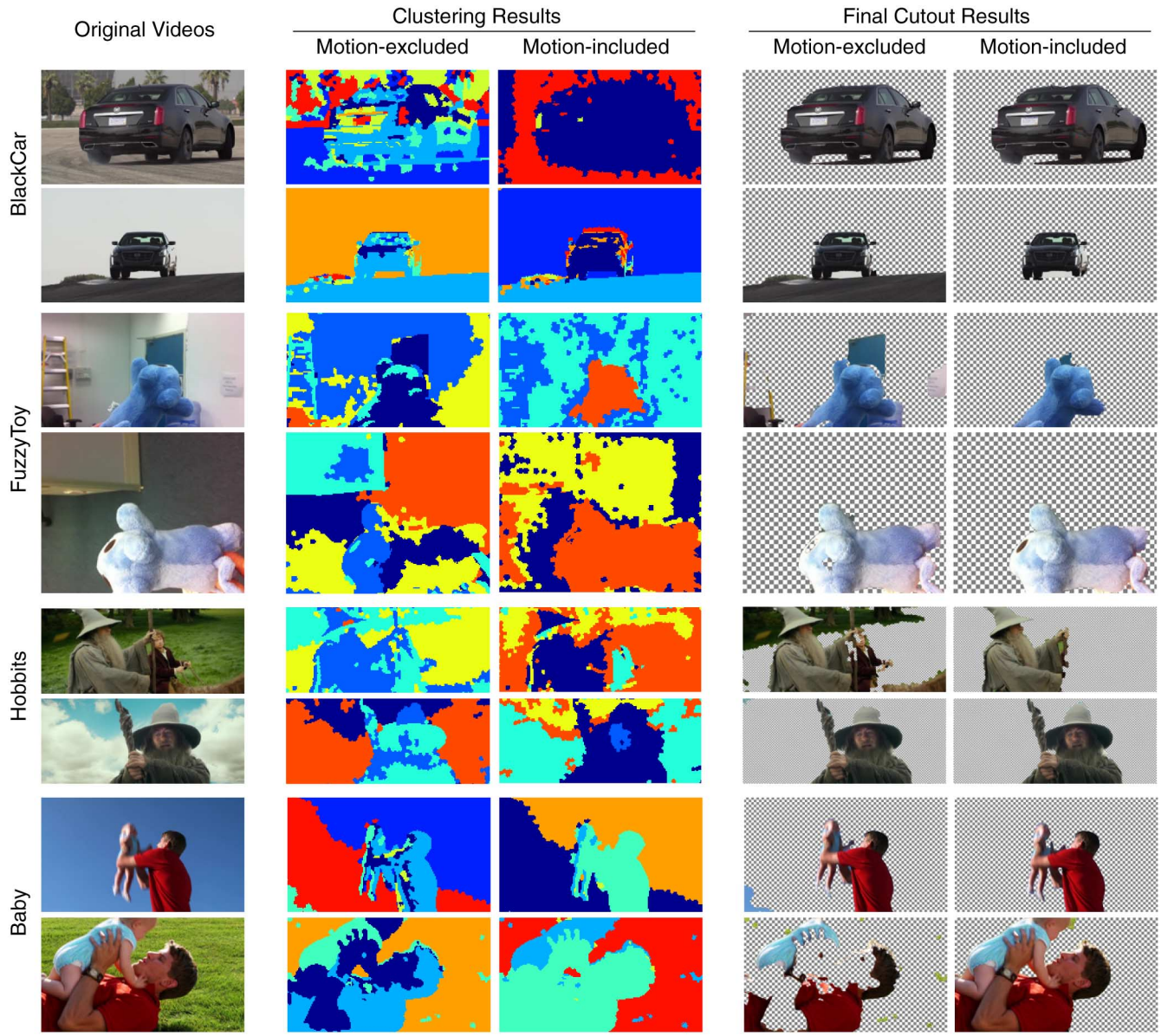


Fig. 6. Unsupervised object co-segmentation results by motion-excluded vs. motion-included features on four video groups. From top to bottom: video groups *BlackCar*, *FuzzyToy*, *Hobbits* and *Baby*. In each group, from left to right: original videos, clustering as well as final cutout results with motion-excluded and motion-included features separately.

Fig. 7 shows another four groups of videos, and each group is co-segmented by [7] and our method. Experimental results show that our method can produce cutout with much higher accuracy than [7]. Note that the videos *IceSkater* and *KiteSurfer* are from [7] and both video groups present little appearance similarities on foreground, which are not fully suitable for our formulation of video co-segmentation problem. In [7], an initial common foreground was detected by saliency and objectness and was used to guide the following co-segmentation task. So here for fair comparison, we also provide an initial foreground estimation for our implementation using the same manner to avoid the ambiguity of foreground and background. Yet due to heterogeneous foreground involved, the third term in Equation (16) is violated to some extent. Even so, the experimental results clearly show that the algorithm of [7] presents large segmentation errors on most video examples. Table III compares the precision and recall on the video examples of [7] and our method, based on the ground truth we manually obtained using Adobe After Effects [38]. We also conduct the comparison using the video group *Baby*, and illustrate more frames in Fig. 8. Please see our accompanying video for the live demo. A high resolution version can be browsed or downloaded from the link YouTube and GoogleDrive.

C. Object Co-Segmentation with Moderate User Guidance

Our framework can also produce object co-segmentation results for those videos with similar background, under which circumstance user guidance has to be involved to assist the segmentation. However, our system does not require the user to draw strokes on every video sequence in a group. For each video

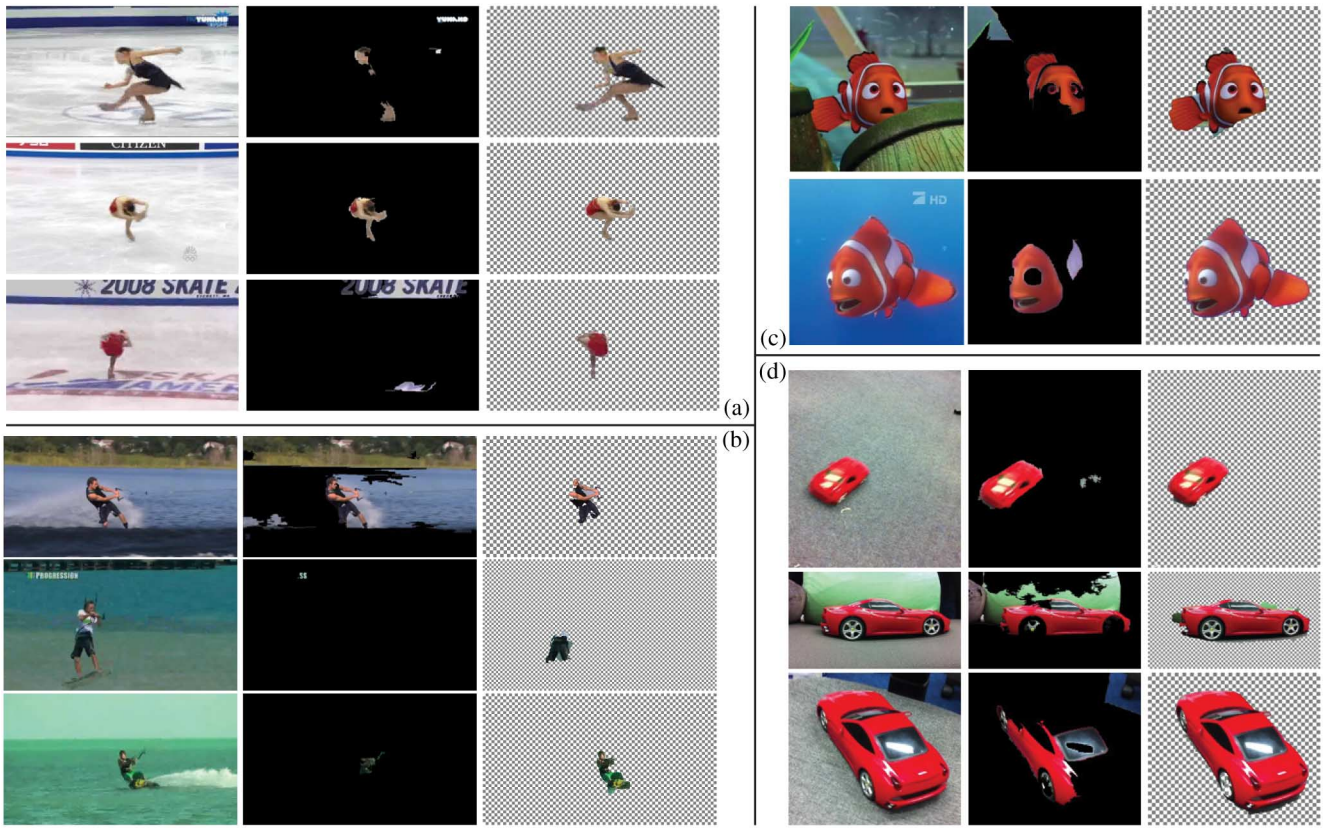


Fig. 7. Unsupervised object co-segmentation results on four groups of videos. (a)–(d) : video groups *IceSkater*, *KiteSurfer*, *Nemo* and *Toycar*. In each group, each row corresponds to a video and columns from left to right show raw frames, results by [7] (black means background) and by our method. See the accompanying video.

TABLE III
PRECISION AND RECALL OF UNSUPERVISED OBJECT CO-SEGMENTATION RESULTS BY [7] AND OUR METHOD

	[7]		Ours	
	Precision	Recall	Precision	Recall
Nemo	0.2073	0.7113	0.9571	0.9808
IceSkater	0.3907	0.6308	0.9206	0.8784
KiteSurfer	0.0696	0.5822	0.7919	0.5364
Toycar	0.4785	0.5593	0.9111	0.9635

TABLE IV
PRECISION AND RECALL OF OBJECT CO-SEGMENTATION RESULTS BY OUR METHOD, WITH MODERATE USER GUIDANCE

	Precision	Recall
Girl	0.9861	0.7926
Cheetah	0.9716	0.6358
Flowers	0.9537	0.9596

group, very limited number of strokes on a few frames in one input video are enough for our system to generate satisfactory results. This avoids much repetitive work, and significantly reduces users efforts.

Fig. 9 shows some of the co-segmentation results with moderate user guidance. In the video group *Flowers*, the upper-left video contains scattered foreground objects with different sizes, some of which are small and hard to draw strokes on. Traditional video cutout tools like Rotobrush of Adobe After Effects often require users to specify strokes on all the foreground objects, and this work needs to be done on all videos separately. While with our video object co-segmentation method, users just need to draw strokes on one frame in any video source, and then the rest videos will be co-segmented, guided by the prior knowledge on foreground thus obtained. For the group *Girl*, although background clutter and color ambiguity are apparent in the source videos, we only draw strokes on 3 frames of one video source even though the background motion is more complex. The cutout results are comparable to ground truth (see Table IV).

In Fig. 10, we also show cutout results of more frames of one of the examples, the video group *Girl*. Please see our accompanying video for the live demo.

We also compare our video cutout results with the ones generated by applying one of the state-of-the-art image co-segmentation approaches [18] to all the video frames in each of the video groups. For fair comparison, the publicly available code[1] is slightly modified for accepting input foreground/background labels. We feed the implementation of this approach with ground-truth of the frames we draw strokes on as the guidance for segmentation. Furthermore, due to the high computational complexity of this approach, each video frame is down-sampled to a resolution so that width and height do not exceed 160 pixels. As shown in Fig. 9, the foreground and background produced by the image co-segmentation approach are shown in green and red separately. Obviously, our video co-segmentation framework performs consistently better on the three video groups even though we give aforementioned advantages to

[1]http://ai.stanford.edu/ ajoulin/index.php?page=coseg

Fig. 8. Cutout results of more frames in video group *Baby*. Rows 1, 4 are the original video frames, rows 2, 5 are the results by [7] and rows 3, 6 are the results by our method.

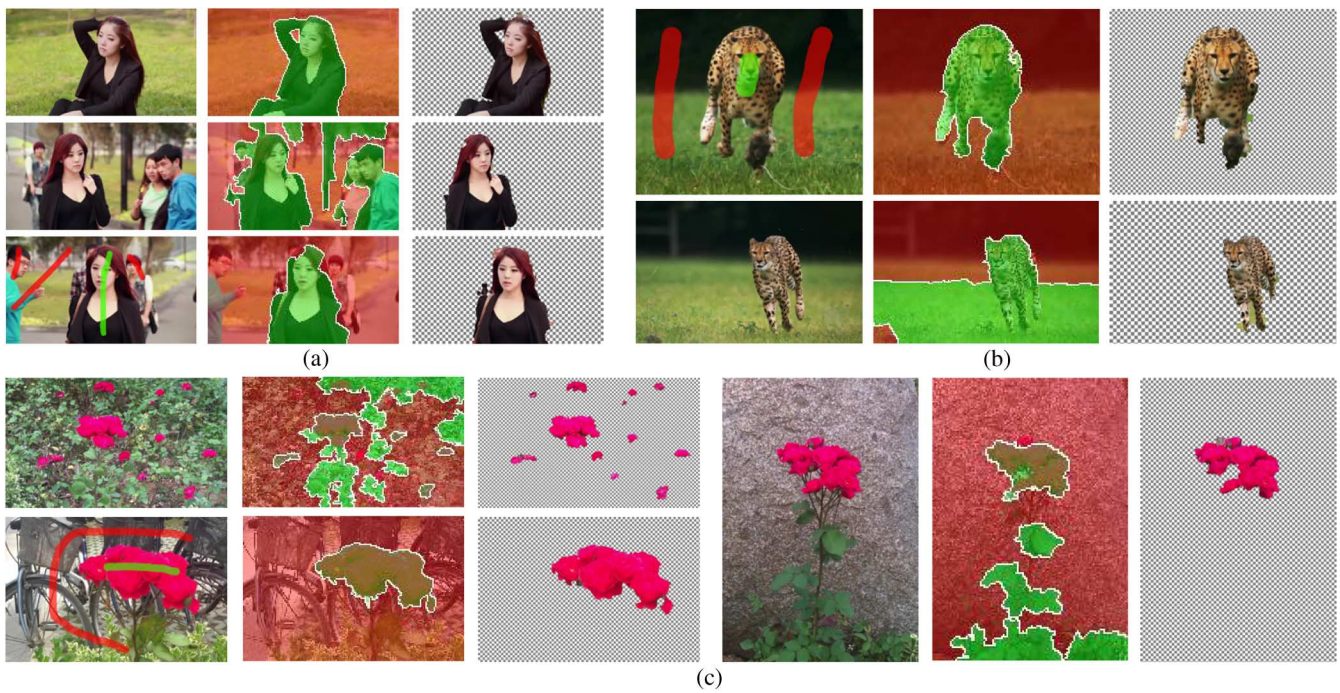


Fig. 9. Object co-segmentation with moderate user guidance. (a)–(c) : video groups *Girl*, *Cheetah* and *Flowers*. In each group, from left to right: original video frame, results by image co-segmentation method [18] and by ours. For each video group, we just draw strokes (green: foreground, red: background) on no more than 3 frames of one video only. See also the accompanying video.

image co-segmentation approach. Besides the potential lack of robustness, the reason might also be that the image co-segmentation working in this manner ignores the intrinsic nature of spatio-temporal consistency of the video, and pays little attention to the consistency of cutout results of neighboring video frames. In comparison, working on TSPs which is locally consistent over frames, our approach successfully generates high-quality results by leveraging the motion coherence within each video and foreground consistency across different videos.

D. Limitations

Our method takes TSP as the basic unit during the procedures of clustering and the successive object cutout. This highly reduces the amount of data to be processed so that the efficiency

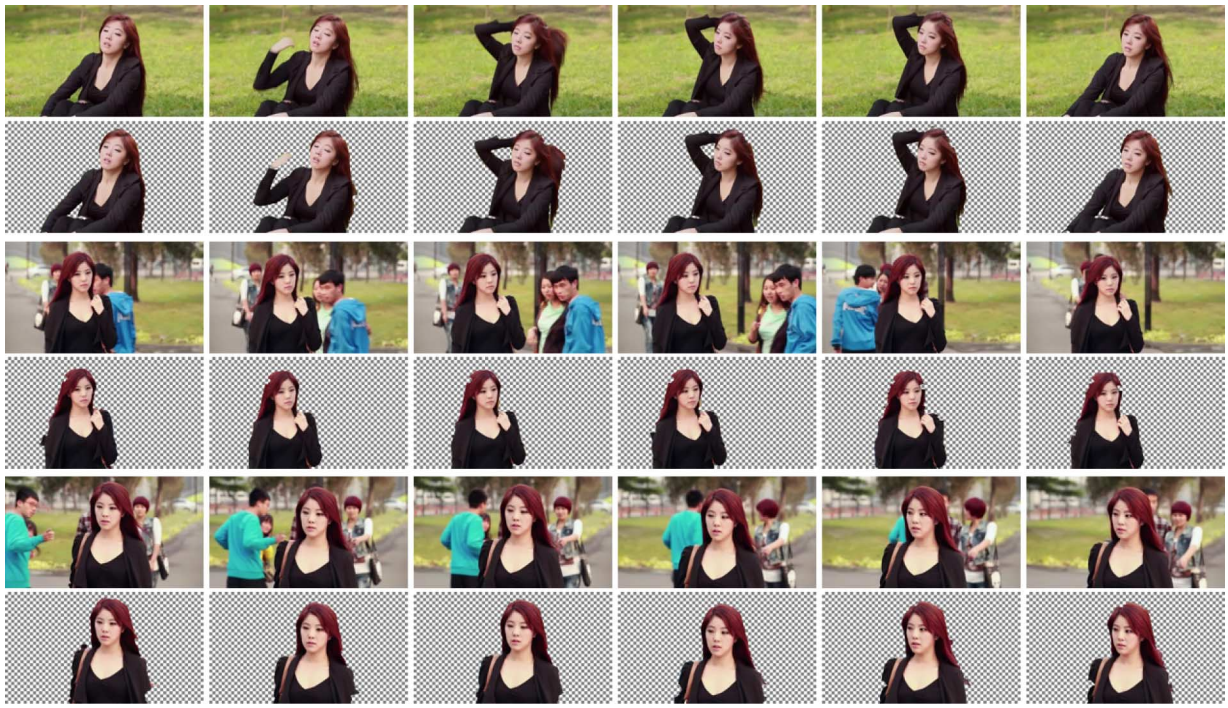


Fig. 10. Cutout results of more frames in video group *Girl*. Rows 1, 3, 5 are original video frames, and rows 2, 4, 6 are results by our method.

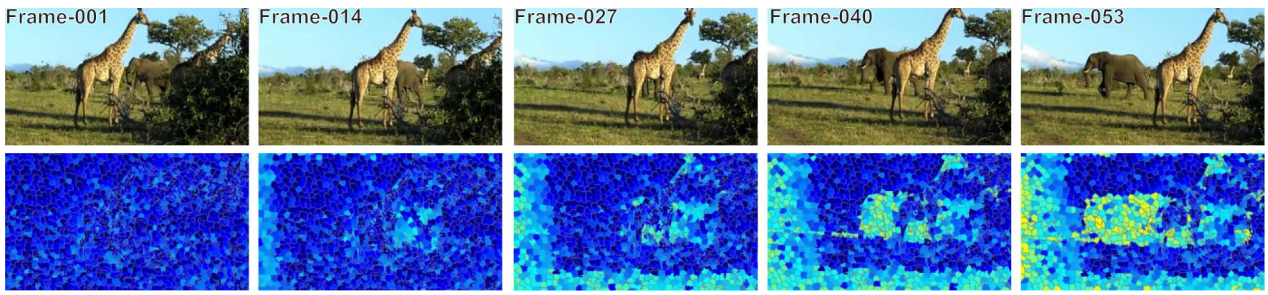


Fig. 11. Label changes of the TSPs on the elephant over time, in the second video of the group. The second row visualizes the TSPs grouped into different clusters. When the elephant is occluded gradually by the giraffe, the TSPs on it vanish. New TSPs are assigned to it when it re-appears in the scene. This causes failure in obtaining the complete and consistent trajectories for the TSPs on the elephant, and leads to the poor performance of clustering.

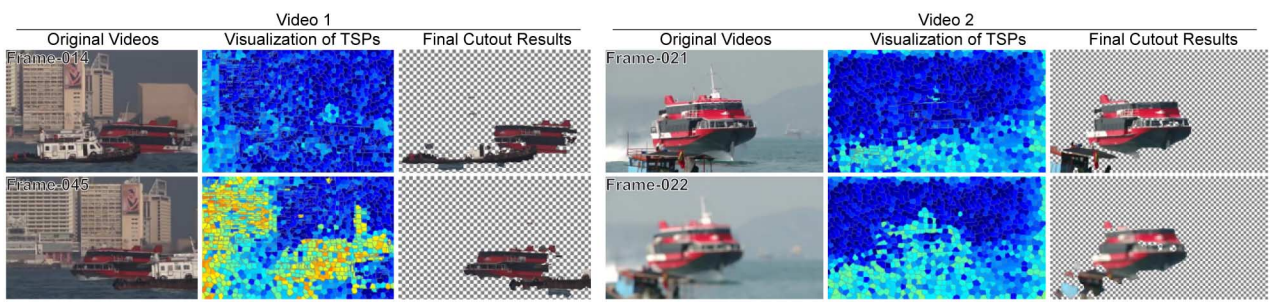


Fig. 12. Object co-segmentation results on video group *Jetfoil* by our method. In Video 1, the ship occludes the jetfoil when it passes by; In Video 2, defocus blur occurs in Frame 22; Each situation causes the changing of TSPs on the jetfoil, as visualized in columns 2 and 4 respectively. Besides, color similarity commonly exists in parts of the jetfoil and the background. All of the factors result in bad cutout results as shown in columns 3 and 6.

is improved. Motion information, characterized by the trajectories of TSPs, is incorporated into the subspace clustering for better differentiating foreground from background. However, if severe occlusion happens, it is often challenging to get a complete trajectory which is consistent over all video frames. This will degrade the performance of clustering. Fig. 11 visualizes the changing of all TSPs over time for the second video in the video group *giraffe, elephant*. Besides the fact that the elephant, giraffe and meadow have similar color distribution, TSPs on the elephant have inconsistent labels before and after the occlusion by the giraffe. Moreover, the giraffe is always static throughout the sequence, resulting in motion unable to distinguish it from

the background. These factors lead to the poor performance of clustering.

We further apply our approach to a more challenging video group *Jetfoil* (Fig. 12). In this video group, besides complicated background such as the ships and buildings which contain similar black color as the jetfoil, severe occlusion (the ship occludes the jetfoil in Video 1) and defocus blur (Frame 22 in Video 2) exist, so that the TSPs are more likely to be inconsistent. Under these conditions, the degraded motion trajectory loses its power to distinguish various object motions, causing some of the background and foreground regions are grouped and extracted together. Although in our current system the quality of results may be improved by adding more user inputs, more advanced motion segmentation techniques such as exploring the application of low-rank representation to the contaminated motion trajectories can be potentially coupled with the amf-co-segmentation to achieve more robust results in the future.

VIII. Conclusions and Future Work

We have presented a novel framework for video object co-segmentation. Our experiments show that with moderate or even no user guidance, common foreground can be segmented simultaneously from a group of videos. Our approach can make full use of the appearance and motion information embodied in the videos for co-segmentation. This is realized by a new appearance-motion-fused video co-segmentation algorithm via subspace clustering, which yields consistent labeling of the common foreground across different videos. Furthermore, we define video object co-segmentation as a binary labeling problem that can be solved by QPBO in an MRF framework. The framework imposes the constraint of the foreground model automatically computed or specified with little user effort.

Our framework realizes video object co-segmentation in a manner of global optimization on the videos, but it lacks a fine-tuning mechanism for locally refining the results. Since the common foreground of different videos would complement each other, in the future we plan to take a co-refinement step that corrects the errors and refines the result in a video by exploiting the good result in another one. For example, the user first drags a rectangle around the area of segmentation errors in a video frame. The rectangle is then propagated to the successive frames, and the result is refined within these frames by local optimization, while imposing the constraint of good results of other videos in the same group.

Another solution is to adopt the local classifiers as [1][15] to handle the problem of inseparable statistics. Exploring the possibility of combining our framework with the local classifier-based video segmentation techniques is another interesting work. In addition, we plan to solve the matte of the foreground and remove remaining errors around object boundaries through an additive step of coherent matting which has been proven effective by previous video cutout systems [1]–[15].

Acknowledgment

The authors would like to thank the reviewers for their constructive comments which helped improve this paper greatly. Yanwen Guo is the corresponding author.

References

[1] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video snapcut: Robust video object cutout using localized classifiers," *ACM Trans. Graph.*, vol. 28, no. 3, p. 70, 2009.

[2] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proc. IEEE CVPR*, 2010, pp. 2141–2148.

[3] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *Proc. IEEE ICCV*, 2011, pp. 1995–2002.

[4] P. Ochs and T. Brox, "Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions," in *Proc. IEEE ICCV*, 2011.

[5] O. Peter and B. Thomas, "Higher order motion models and spectral clustering," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2012*, 2012, pp. 614–621.

[6] C. Xu, C. Xiong, and J. J. Corso, "Streaming hierarchical video segmentation," in *Proc. ECCV*, 2012, pp. 626–639, Springer.

[7] J. C. Rubio, J. Serrat, and A. López, "Video co-segmentation," in *Proc. ACCV*, ser. 7725. : LNCS, 2012, pp. 13–24.

[8] D.-J. Chen, H.-T. Chen, and L.-W. Chang, "Video object cosegmentation," in *Proc. ACM Multimedia*, 2012, pp. 805–808.

[9] W.-C. Chiu and M. Fritz, "Multi-class video co-segmentation with a generative multi-video model," in *Proc. IEEE CVPR*, Portland, OR, USA, 2013.

[10] R.-F. Tong, Y. Zhang, and M. Ding, "Video brush: A novel interface for efficient video cutout," in *Computer Graphics Forum*, 2011, vol. 30, no. 7, pp. 2049–2057, Wiley Online Library.

[11] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski, "Video matting of complex scenes," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 243–248, 2002.

[12] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz, "Keyframe-based tracking for rotoscoping and animation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 584–591.

[13] Y. Li, J. Sun, and H.-Y. Shum, "Video object cut and paste," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 595–600, 2005.

[14] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen, "Interactive video cutout," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 585–594.

[15] F. Zhong, X. Qin, Q. Peng, and X. Meng, "Discontinuity-aware video object cutout," *ACM Trans. Graph. (SIGGRAPH Asia)*, vol. 31, no. 6, pp. 175:1–175:10, 2012.

[16] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs," in *Proc. IEEE CVPR*, 2006, vol. 1, pp. 993–1000.

[17] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "icoseg: Interactive co-segmentation with intelligent scribble guidance," in *Proc. IEEE CVPR*, 2010, pp. 3169–3176.

[18] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2012*, 2012, pp. 542–549.

[19] L. Mukherjee, V. Singh, and J. Peng, "Scale invariant cosegmentation for image groups," in *Proc. IEEE CVPR*, 2011, pp. 1881–1888.

[20] M. D. Collins, J. Xu, L. Grady, and V. Singh, "Random walks based multi-image segmentation: Quasiconvexity results and gpu-based solutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, Jun. 2012 [Online]. Available: http://pages.cs.wisc.edu/mcollins/pubs/cvpr2012.html

[21] L. Mukherjee, V. Singh, J. Xu, and M. D. Collins, "Analyzing the subspace structure of related images: Concurrent segmentation of image sets," in *Proc. Computer Vision–ECCV 2012*, 2012, pp. 128–142, Springer.

[22] Y. Fu and Y. Guo, "Content-sensitive collection snapping," in *Proc. IEEE International Conf. Multimedia and Expo (ICME), 2011*, 2011, pp. 1–6.

[23] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3d mesh segmentation and labeling," *ACM Trans. Graph.*, vol. 29, no. 4, p. 102, 2010.

[24] A. Golovinskiy and T. Funkhouser, "Consistent segmentation of 3d models," *Comput. Graph.*, vol. 33, no. 3, pp. 262–269, 2009.

[25] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or, "Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering," *ACM Trans. Graph.*, vol. 30, no. 6, p. 126, 2011.

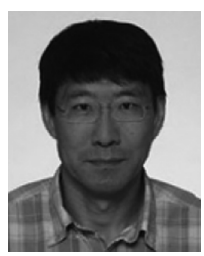
[26] R. Hu, L. Fan, and L. Liu, "Co-segmentation of 3d shapes via subspace clustering," in *Computer Graphics Forum*, 2012, vol. 31, no. 5, pp. 1703–1713, Wiley Online Library.
[27] J. Chang, D. Wei, and J. W. F. , III, "A video representation using temporal superpixels," in *Proc. IEEE CVPR*, Jun. 2013.
[28] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.
[29] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Proc. IEEE ICCV*, 2005, vol. 2, pp. 1800–1807.
[30] R. Vidal and Y. Ma, "A unified algebraic approach to 2-d and 3-d motion segmentation," in *Proc. ECCV*, 2004, pp. 1–15, Springer.
[31] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, 2010.
[32] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 171–184, 2013.
[33] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
[34] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," arXiv preprint arXiv:1009.5055, 2010.
[35] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optimiz.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[36] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Computer Vision, 2003*, 2003, pp. 1470–1477.
[37] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," in *Proc. IEEE CVPR*, 2007, pp. 1–8.
[38] Adobe after effects [Online]. Available: http://www.adobe.com/products/aftereffects.html

Chuan Wang received his B.Eng degree from University of Science and Technology of China in 2010. He is currently a Ph.D. candidate in Department of Computer Science, The University of Hong Kong. He worked as a visiting scholar in National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, China in 2009 and State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China in 2010. His research interests include image/video processing/analysis and computer vision.

Yanwen Guo received the Ph.D. degree in applied mathematics from the State Key Lab of CAD&CG, Zhejiang University, China, in 2006. He is currently an associate professor at the National Key Lab for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Jiangsu, China. He worked as a visiting professor in the Department of Computer Science and Engineering, The Chinese University of Hong Kong, in 2006 and 2009, respectively, and the Department of Computer Science, The University of Hong Kong, in 2008, 2012, and 2013, respectively. He has been a visiting scholar in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, since 2013. His research interests include image and video processing, vision, and computer graphics. He is the corresponding author of this paper.

Jie Zhu received the B.Eng degree from Jiangnan University, Wuxi in 2012. He is now a master candidate in the Department of Computer Science and Technology at Nanjing University. He worked as a visiting scholar in The University of Hong Kong in 2013. His research interests include computer vision and computer graphics.

Linbo Wang received his B.S. degree in Computer Science and Technology from Shandong University, China, in 2005. He is currently a 4th year Ph.D. candidate at the National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University. His research interests include computer vision and digital image processing, specifically, image and video co-segmentation, matting, object recognition, matching, etc.



Wenping Wang received the Ph.D. degree from the University of Alberta, Edmonton, Canada. He is a professor and the department head of the Department of Computer Science, The University of Hong Kong, Pokfulam. His research interests include computer graphics, visualization, and geometric computing. His current research interests include mesh generation and surface modeling for architectural design. He is a journal associate editor of Computer Aided Geometric Design, Computers and Graphics, and IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, and the program cochair of several international conferences, including Pacific Graphics 2003, ACM Symposium on Physical and Solid Modeling (SPM '06), Conference on Shape Modeling (SMI '09), and the conference chair of Pacific Graphics 2012 and SIGGRAPH Asia 2013. He is a member of the IEEE.