Video Co-summarization: Video Summarization by Visual Co-occurrence

Wen-Sheng Chu[1] **Yale Song**[2] **Alejandro Jaimes**[2]
[1]Robotics Institute, Carnegie Mellon University [2]Yahoo Labs, New York

Abstract

We present video co-summarization*, a novel perspective to video summarization that exploits visual co-occurrence across multiple videos. Motivated by the observation that important visual concepts tend to appear repeatedly across videos of the same topic, we propose to summarize a video by finding shots that co-occur most frequently across videos collected using a topic keyword. The main technical challenge is dealing with the sparsity of co-occurring patterns, out of hundreds to possibly thousands of irrelevant shots in videos being considered. To deal with this challenge, we developed a Maximal Biclique Finding (MBF) algorithm that is optimized to find sparsely co-occurring patterns, discarding less co-occurring patterns even if they are dominant in one video. Our algorithm is parallelizable with closed-form updates, thus can easily scale up to handle a large number of videos simultaneously. We demonstrate the effectiveness of our approach on motion capture and self-compiled YouTube datasets. Our results suggest that summaries generated by visual co-occurrence tend to match more closely with human generated summaries, when compared to several popular unsupervised techniques.*

1. Introduction

The amount of online videos has been growing at an exponential rate; the need for easier video browsing has increased considerably. With the goal of providing an efficient way to overview the large collection of videos, video summarization has attracted intensive attention over the past decade [27, 34]. Several approaches have been proposed to summarize videos by leveraging domain-specific knowledge [11, 23, 32] or training a supervised model with a labeled database [21, 31, 33]. However, it still remains as a challenge to formulate the right model able to deal with the large diversity of video content without human supervision.

We present a novel perspective to video summarization, termed as *video co-summarization*. We observe that, given a collection of videos sharing the same topic (*e.g.*, videos retrieved using a query term), important visual concepts tend to appear repeatedly across the videos; the frequency of visual co-occurrence can thus serve as a proxy to measure the shot importance. Our goal is, therefore, to summarize a video by including shots that co-occur most frequently across videos of the same topic. Fig. 1 illustrates our main idea: Given an input video belonging to a query *surfing*, we identify visually co-occurring shots, *e.g.*, *surfing*, *sunset*, and *palm tree*, across additional videos retrieved using the same query. In this way, video co-summarization can identify important shots without assuming domain-specific knowledge or training a supervised model with labeled examples; this makes our approach particularly suitable for dealing with the content diversity in online videos.

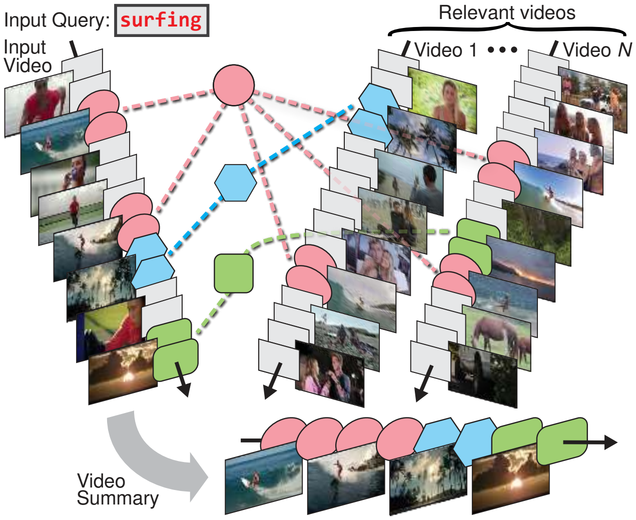


Figure 1. An illustration of video *co-summarization* as identifying visually most similar events shared across N videos. Different colors and shapes indicate relevant events discovered by our algorithm: surfing (red circles), sunset (green rectangles), and palm tree (blue hexagons), as shown in the selected video frames. Dashed lines represent correspondence between shots.

Our work builds upon the idea of unsupervised commonality analysis, which has been successfully applied to image co-segmentation [4, 6], image/video co-localization [17], temporal commonality discovery [7], and object discovery [35] with different techniques. Unlike the previous tasks, however, video co-summarization has an additional challenge of dealing with the sparsity of co-occurring patterns: A set of videos can have hundreds to thousands of shots; often there are only a few common shots that appear *jointly* across videos. To deal with this challenge, we propose a novel *Maximal Biclique Finding* (MBF) algorithm, which formulates the problem as finding complete bipartite subgraphs (*i.e*., bicliques) that maximize the total visual co-occurrence within a bipartite graphical representation of shots and videos. Unlike the standard clustering-based approaches that assign labels to all existing shots, our MBF algorithm *sparsely* assigns labels to just a few shots with maximum joint similarities. This allows our algorithm to effectively discard irrelevant shots that appear only within a single video, even if they are dominant in that video. We develop a parallelizable learning algorithm with closed-form updates, allowing us to scale up to handle a large number of videos simultaneously. Our contributions are three-fold:

- We present *video co-summarization*, a novel perspective to summarizing videos by exploiting visual co-occurrence across additional videos sharing the same topic. To the best of our knowledge, our work is the first to propose and demonstrate the effectiveness of video co-summarization.
- Our approach determines the shot importance by visual co-occurrence across multiple videos sharing the same topic, without assuming domain-specific knowledge or training supervised learning models; this makes our model generalizable to web-scale videos with high content diversity.
- The proposed Maximal Biclique Finding (MBF) algorithm can naturally handle the sparsity of co-occurring shots by discarding the ones that appear only within a single video. The algorithm is parallelizable with closed-form updates, and thus can handle a large number of videos simultaneously.

2. Related Work

Video summarization has been tackled from various perspectives [27, 34]. Below, we review the most representative works in three common approaches – domain-specific, supervised, and unsupervised – and differentiate our work from the previous work.

Domain-specific video summarization: Domain-specific knowledge can help identify important shots of a video. For instance, sports videos contain canonical scenes, such as “home run” in baseball [12] and “touch down” in football [5]; those shots can be used to generate sports highlights. Similarly, trajectories can be used to summarize tactic information in soccer games [42]. For surveillance videos, most frames contain stationary background, and thus can be summarized into *synopsis* [11, 32]. News videos contain rich textual information, and can be auto-documented with the correspondence between topic themes and visual-textual concepts [39], or with spatial image salience and temporal content variation [23].

Supervised video summarization: Much work has been proposed to measure the shot importance through supervised learning. Egocentric videos can be summarized by learning important faces, hands, and objects [21], or learning the overall energy of storiness, importance, and diversity of selected video shots [24]. To predict per-frame interestingness, low-level, high-level, and spatial-temporal features were combined to train a linear regression model [15]. Similarly, shot importance was measured with a pre-trained topic-specific binary SVM classifier [31] or a SVM ranker [33]. Furthermore, with a small number of labels, a hierarchical model was learned to generate a video summary that contains objects of interests [22].

Compared to video co-summarization, the above approaches require either prior knowledge about a certain domain (*e.g*., sports, news), or labeled examples that are difficult to collect. Because domain-specific knowledge does not generalize across different contents, and labels are expensive to obtain, it is difficult to apply these approaches to web-scale video with diverse content. Our method, on the other hand, exploits visual co-occurrence across videos without strict supervision, and thus can be easily applied to any video collection that shares the same topic.

Unsupervised video summarization: The closest to our approach is unsupervised video summarization, which do not require domain-specific knowledge or labeled examples, but instead seek low-level visual relevance or leverage additional resources to determine shot importance. One popular approach is reducing visual redundancy by learning a dictionary of basis frames or shots [8, 40], or performing a hierarchical clustering analysis [25]. Other works have explored human attention during video watching in order to capture the perceptual quality of video shots for selecting content highlights [30]. Multiple videos can be summarized using a set of keyframes selected [37]. Another recent trend is to summarize videos with online images, such as an image set with canonical views [18] or a photo stream that are taken consecutively [19]. Such methods generate keyframe summaries using correlations between video frames and an image collection. While images carry visual information that could help determine shot importance, our approach uses videos and their visual co-occurrence, which better preserve spatio-temporal information for summarizing videos. Also, our proposed MBF algorithm can handle the sparsity of co-occurring patterns, which is crucial in leveraging online videos.

3. Video Co-summarization

Video co-summarization aims to identify shots that co-occur frequently across videos of the same topic. This section describes our solution to tackle this problem.

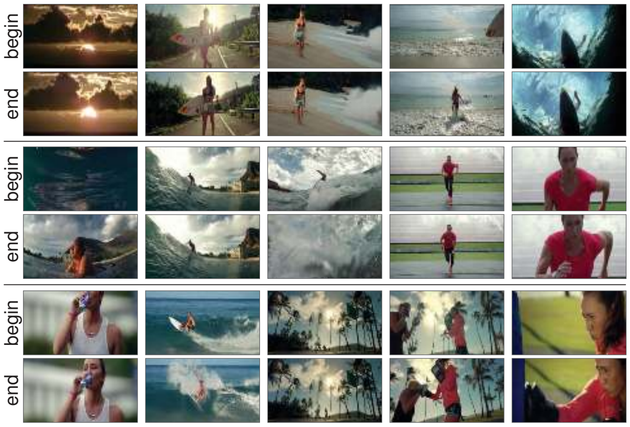


Figure 2. Our video segmentation is simple yet effective. Example segmentation results show that our method performs well on a video retrieved by a query *Surfing*. Each column indicates a shot, where shot boundaries are denoted as begin and end, respectively.

3.1. Video pre-processing

Video segmentation: We first perform video segmentation by measuring the amount of changes (sum-squared pixel-wise difference) between two consecutive frames in the RGB and the HSV color spaces. A shot boundary is determined at a certain frame when the portion of total change is greater than 75%. We then merge shots with less than 10 frames with their subsequent shot, and divide lengthy shots evenly so that each shot contains at most 150 frames. This approach is simple yet effective (see Fig. 2 for an illustration), and serves as the building block throughout the paper.

Shot-level feature mapping: We represent a shot with two types of features: *observation features* extracted from a single frame, and *interaction features* extracted from two consecutive frames [16]. Suppose the j-th frame is described as a feature vector $\mathbf{x}_j$ (Sec. 4.1 describes our choice of feature descriptors). We design the observation feature $\phi^{\text{obs}}(\mathbf{x}_j)$ to capture the pseudo-probability that $\mathbf{x}_j$ belongs to a state, and the interaction feature $\phi^{\text{int}}(\mathbf{x}_j)$ to capture the transition probability of the states between two consecutive frames. Formally, for the i-th shot $\mathbf{X}_i = \{\mathbf{x}_{b_i}, ..., \mathbf{x}_{e_i}\}$ between the b_i-th and the e_i-th frames (see notation[1]), we consider a shot-level feature mapping:

$$\phi(\mathbf{X}_i) = \frac{1}{|\mathbf{X}_i|} \sum_{j=b_i}^{e_i} \begin{bmatrix} \phi^{\text{obs}}(\mathbf{x}_j) \\ \phi^{\text{int}}(\mathbf{x}_j) \end{bmatrix}, \tag{1}$$

where $|\mathbf{X}_i|$ is the number of frames in shot $\mathbf{X}_i$. We perform a k-means clustering to find K centroids $\{\mathbf{c}_k\}_{k=1}^{K}$ as the hidden states; we set $K = 200$. The observation feature vector is represented as $\phi^{\text{obs}}(\mathbf{x}_j) \in [0,1]^K$ with the i-th element computed as $\exp(-\gamma\|\mathbf{x}_j - \mathbf{c}_i\|^2)$ and γ chosen as an inverse of the median distance of all samples to the centroids. The interaction feature vector $\phi^{\text{int}}(\mathbf{x}_j) \in [0,1]^{K^2}$ is defined as:

$$\phi^{\text{int}}(\mathbf{x}_j) = \phi^{\text{obs}}(\mathbf{x}_j) \otimes \phi^{\text{obs}}(\mathbf{x}_{j+1}), \tag{2}$$

where $\otimes$ denotes a Kronecker product of two observation vectors. As a result, we represent a video shot as a feature vector $\phi(\mathbf{X}_i) \in \mathbb{R}^{(K^2+K)}$.

3.2. Bipartite graph construction

We model a collection of videos and their associated shots as a weighted bipartite graph. Suppose we are given two videos $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ and $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ with m and n shot-level features, respectively. We model the video pair as a weighted bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V} = \mathbf{A} \cup \mathbf{B}$ is the vertex set, $\mathcal{E} = \{(\mathbf{a}_i, \mathbf{b}_j) | \mathbf{a}_i \in \mathbf{A}, \mathbf{b}_j \in \mathbf{B}\}$ is the edge set, and $\mathbf{W} = \begin{bmatrix} 0 & \mathbf{C} \\ \mathbf{C}^\top & 0 \end{bmatrix}$ is the weight matrix. We encode the co-occurrence relationship between a pair of videos with a *co-occurrence matrix* $\mathbf{C} \in \mathbb{R}^{|\mathbf{A}| \times |\mathbf{B}|}$. Each entry C_{ij} of the matrix is computed as $\exp(-\rho d(\mathbf{a}_i, \mathbf{b}_j))$. We use the χ^2 distance to compute $d(\cdot, \cdot)$; ρ is the bandwidth value, set to the median of all distance values. Given a set of more than two videos, we apply the same method for each pair of videos to construct the entire graph.

3.3. Visual co-occurrence as co-clusters

This section describes a co-clustering approach to tackle video co-summarization. In the next section, we explain the limitations of this approach and propose our novel solution.

Given multiple items from different classes, co-clustering represents their relationship using an "incidence matrix" and performs clustering by generating a subset of rows and columns of the matrix that exhibits certain mutual behavior [10]. The classical example of this technique is joint document-word clustering [10], where the incidence matrix represents a document collection with columns representing documents and rows representing words.

Applied to video co-summarization, we model the incidence matrix by constructing a bipartite graph $\mathcal{G}$ (see Sec. 3.2), representing a video collection with rows and columns that correspond to shots of respective videos. We then formulate video co-summarization as the graph bipartition problem, *i.e.*, partitioning the graph $\mathcal{G}$ into *co-clusters* such that each cluster contains pairs of correlated shots with a high visual similarity.

To solve the graph bi-partition problem, similar to spectral clustering [29], we first construct a graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & 0 \\ 0 & \mathbf{D}_2 \end{bmatrix}$ is the degree

[1] Bold capital letters denote a matrix $\mathbf{X}$; bold lower-case letters denote a column vector $\mathbf{x}$. $\mathbf{X}_{i:}$ and $\mathbf{X}_{:j}$ represent the i-th row and the j-th column of the matrix $\mathbf{X}$, respectively. $\mathbf{e}_n$ denotes an n-dimensional column vector of ones. All non-bold letters represent scalar variables. X_{ij} and x_i denote the (i, j)-th element of $\mathbf{X}$ and the i-th element of $\mathbf{x}$, respectively.

matrix with $\mathbf{D}_1 = \mathrm{diag}(\mathbf{C}\mathbf{e}_n)$ and $\mathbf{D}_2 = \mathrm{diag}(\mathbf{C}^\top\mathbf{e}_m)$. We then apply the efficient spectral solution [10] to the generalized eigenvalue problem $\mathbf{LZ} = \lambda\mathbf{DZ}$. Let $\widehat{\mathbf{C}} = \mathbf{D}_1^{-1/2}\mathbf{C}\mathbf{D}_2^{-1/2}$ be the normalized co-occurrence matrix. It has been proved that the solution to the eigenvalue problem becomes $\mathbf{Z} = [\mathbf{D}_1^{-1/2}\mathbf{U}; \mathbf{D}_2^{-1/2}\mathbf{V}]$, where $\mathbf{U} \in \mathbb{R}^{m\times\ell}$ and $\mathbf{V} \in \mathbb{R}^{n\times\ell}$ are top ℓ largest singular vectors of $\widehat{\mathbf{C}}$, and $\ell = \lceil \log_2 k \rceil$, *i.e.*, $\widehat{\mathbf{C}} = \mathbf{U}\Sigma\mathbf{V}^\top$ [10]. As a result, the optimal k co-clusters are extracted by performing k-means on the ℓ-dimensional data $\mathbf{Z}$. Each co-cluster contains a subset of shot-pairs that exhibit high visual co-occurrence.

3.4. Visual co-occurrence as maximal bicliques

While co-clustering groups similar pairs of shots into co-clusters, it does not provide a robust way to deal with shots that co-occur only sparsely. For example, given multiple videos with a total of hundreds to thousands of shots, often case there are only a few shots that are truly related to the topic, while the rest is unrelated and specific to a single video. In such case, as confirmed by our experiment in Sec. 4 with the Mocap data, co-clustering would fail to capture the sparsely co-occurring shots because the co-occurrence matrix will be dominated by a majority of unrelated pairs of shots.

To remedy this problem, we formulate video co-summarization as finding complete bipartite subgraphs, or *bicliques*. Each biclique represents a compact set of video shots that are visually similar to each other. Specifically, given the co-occurrence matrix $\mathbf{C}$, we look for two binary selection vectors $\mathbf{u}$ and $\mathbf{v}$ that identify the bicliques with maximal visual correlation:

$$\begin{aligned} \max_{\mathbf{u},\mathbf{v}} \quad & \sum_{ij} C_{ij} u_i v_j & (3) \\ \text{subject to} \quad & u_i + v_j \leq 1 + I(C_{ij} \geq \epsilon), \forall i,j, \\ & \mathbf{u} \in \{0,1\}^m, \mathbf{v} \in \{0,1\}^n, \end{aligned}$$

where $I(x)$ is an indicator function that returns 1 if the statement x is true, and 0 otherwise. The first constraint ensures that a biclique contains only shots with sufficient visual similarity, *i.e.*, if $C_{ij} < \epsilon$, either u_i or v_j equals to zero. Because solving the 0-1 integer programming in Eqn. (3) is NP-hard, we relax the second constraint to the interval $[0,1]$. In addition, to avoid a trivial solution that contains all shots as a biclique, we reformulate Eqn. (3) by imposing the sparsity-inducing norm on $\mathbf{u}$ and $\mathbf{v}$:

$$\begin{aligned} \max_{\mathbf{u},\mathbf{v}} \quad & \sum_{ij} C_{ij} u_i v_j - \lambda_u \|\mathbf{u}\|_1 - \lambda_v \|\mathbf{v}\|_1 & (4) \\ \text{subject to} \quad & u_i + v_j \leq 1 + I(C_{ij} \geq \epsilon), \forall i,j, \\ & \mathbf{u} \in [0,1]^m, \mathbf{v} \in [0,1]^n, , \end{aligned}$$

where λ_u and λ_v are trade-off terms controlling the sparsity in $\mathbf{u}$ and $\mathbf{v}$; we set $\lambda_u = \lambda_v = 10$. Problem (4) is non-concave, so we use block coordinate descent [13] by alternating between $\mathbf{u}$ and $\mathbf{v}$. Suppose we solve for $\mathbf{u}$ with $\mathbf{v}$ fixed, Problem (4) becomes:

```
Algorithm 1: Maximal Biclique Finding (MBF)
Input : Bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathbf{W}$ is
        described by the co-occurrence matrix $\mathbf{C}$;
        parameters $\lambda_u \geq 0$, $\lambda_v \geq 0$, and $\epsilon$.
Output: Maximal biclique indicated by $\mathbf{u}$ and $\mathbf{v}$
Initialize $\mathbf{v} \leftarrow \mathtt{rand}(n) \in [0,1]^n$;
while not converged do
    Compute $\widehat{u}_i = \min\{I(\mathbf{C}_{ij} \geq \epsilon) - v_j\}_{j=1}^n$;
    Update $u_i = \min(I(\mathbf{C}_{i:}\mathbf{v} \geq \lambda_u), 1 + (\widehat{u}_i)_-)$;
    Compute $\widehat{v}_j = \min\{I(\mathbf{C}_{ij} \geq \epsilon) - u_i\}_{i=1}^m$;
    Update $v_j = \min(I(\mathbf{u}^\top\mathbf{C}_{:j} \geq \lambda_v), 1 + (\widehat{v}_j)_-)$;
```

$$\begin{aligned} \max_{\mathbf{u}\in[0,1]^m} \quad & \sum_i (\mathbf{C}_{i:}\mathbf{v} - \lambda_u) u_i & (5) \\ \text{subject to} \quad & u_i \leq 1 + I(C_{ij} \geq \epsilon) - v_j, \forall i,j. \end{aligned}$$

Problem (5) is linear in $\mathbf{u}$; we solve it using linear programming. Importantly, we can derive an update rule in a closed-form because u_i's are independent of each other. Denoting $\widehat{u}_i = \min\{I(C_{ij} \geq \epsilon) - v_j\}_{j=1}^n$, and $(x)_- = \min(0,x)$ as a non-positive operator, we obtain a closed-form update $u_i = \min\left(I(\mathbf{C}_{i:}\mathbf{v} \geq \lambda_u), 1 + (\widehat{u}_i)_-\right)$. Similarly, we have a closed form update for $v_j = \min(I(\mathbf{u}^\top\mathbf{C}_{:j} \geq \lambda_v), 1 + (\widehat{v}_j)_-)$, where $\widehat{v}_j = \min\{I(\mathbf{C}_{ij} \geq \epsilon) - u_i\}_{i=1}^m$.

Compared to standard maximal biclique finding algorithms (*e.g.*, [2, 28]), our algorithm has two nice properties: (1) the updates are expressed in a closed form, and (2) the algorithm can be parallelized due to the element-wise update. Both properties suggest high scalability of our method. Algorithm 1 summarizes the maximal biclique finding (MBF) algorithm. Compared to co-clustering that requires an SVD and costs $\mathcal{O}(mn^2 + n^3)$ [14], MBF requires only $\mathcal{O}(m+n)$ operations per iteration. The main computational cost lies in the matrix-vector product $\mathbf{C}_{i:}\mathbf{v}$ and $\mathbf{u}^\top\mathbf{C}_{:j}$. The rest requires only $\mathcal{O}(\max(m,n))$.

Multiple bicliques: Given the selection vectors $\mathbf{u}$ and $\mathbf{v}$, we are now able to identify one biclique $\mathcal{B} \subseteq \mathcal{G}$. Once a biclique is discovered, we remove its edges from $\mathcal{G}$. We obtain the k maximal bicliques by performing Algorithm 1 k times. To avoid the manual choice of parameter k, we design a quality measurement for a discovered biclique:

$$q(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{ij} C_{ij} u_i v_j, \qquad (6)$$

where $|\mathcal{B}|$ is the size of the biclique. Compared to standard clustering approaches that consider all shots in the objective (such as co-clustering in Sec. 3.3), our method *greedily*

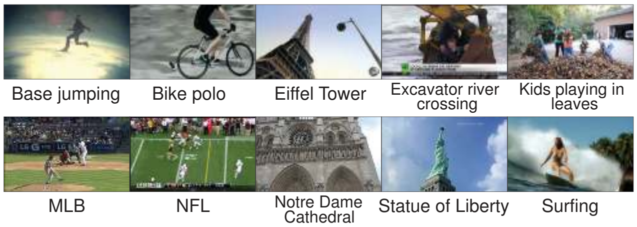


Figure 3. 10 retrieved video categories retrieved from YouTube.

finds maximal bicliques until the quality of a discovered biclique is less than a pre-determined threshold. The quality function allows us to reject visually dissimilar shots and to avoid assigning a cluster label to every shot. We set the threshold to 0.3 throughout the paper, which provides consistent visual similarities within each biclique. Note that the quality function can also be applied to co-clusters to describe their qualities.

Connection to Non-negative Matrix Factorization (NMF): Problem (3) is closely related to NMF [20]. Particularly, we show that the objective of (3) can be interpreted as a special case of NMF. Suppose $\mathbf{u}$ and $\mathbf{v}$ are non-negative and unitary, *i.e.*, $\mathbf{u} \geq 0$, $\mathbf{v} \geq 0$, and $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$. The solution to Problem (3) can be rewritten as $\arg\max_{\mathbf{u},\mathbf{v}} \sum_{i,j} \mathbf{C}_{ij}\mathbf{u}_i\mathbf{v}_j = \arg\max_{\mathbf{u},\mathbf{v}} \mathbf{u}^\top\mathbf{C}\mathbf{v} + \mathrm{tr}(\mathbf{C}^\top\mathbf{C}) + \mathrm{tr}(\mathbf{v}\mathbf{u}^\top\mathbf{u}\mathbf{v}^\top) = \arg\max_{\mathbf{u},\mathbf{v}} \|\mathbf{C} - \mathbf{u}\mathbf{v}^\top\|_F^2$, which shows a rank-one NMF.

Differences from ACA [41] and TCD [7]: Our MBF algorithm has similarities with recent techniques in unsupervised temporal analysis. In particular, both Aligned Cluster Analysis (ACA) and Temporal Commonality Discovery (TCD) aim to discover visually similar shots in an unsupervised manner. However, ACA is a clustering-based algorithm, and by nature, considers all shots in its objective. As we will show in experiments, ACA includes irrelevant shots that generally reduce the discovery quality. Both TCD and MBF focus on discovering only similar shots, while TCD aims to locate one pair of shots at once. On the other hand, MBF finds a group of shot pairs at once, and ensures each biclique contains only shots that are similar to each other.

3.5. Generation of video summaries

Above, we described how we measure the visual importance of a shot by discovering visual co-occurrences as co-clusters (Sec. 3.3) or bicliques (Sec. 3.4). To generate a video summary, we compute a score for each shot, and select top-ranked shots as the final summary. In particular, for both co-clusters and bicliques, the score of a shot is computed as the *quality* measure in Eqn. (6). Given a set of more than two videos, we compute the shot importance score for each pair of videos in the set, and sum up the scores across all the possible pairs. Note that we can parallelize the computation of scores across video pairs because each video pair is independent of other pairs; our method can thus process a large number of videos simultaneously.

Table 1. Descriptive statistics of our YouTube data set.

Video query	Length	#Vid	#Frm	#Shot
Base jumping	10m54s	5	17960	241
Bike polo	14m08s	5	22490	341
Eiffel Tower	25m47s	7	43729	381
Excavators river xing	10m41s	3	16019	112
Kids playing in leaves	15m40s	6	27972	238
MLB	12m11s	6	21271	201
NFL	13m28s	3	23179	405
Notre Dame Cathedral	11m26s	5	20110	196
Statue of Liberty	10m44s	5	18542	164
Surfing	22m40s	6	34790	483
Total	147m40s	51	246062	2762

4. Experiments

4.1. Query-specific video summarization

We demonstrate the effectiveness of our method on a query-specific video summarization scenario, where the goal is to provide the users with video summaries that are adaptive to the query term.

Dataset: To evaluate video co-summarization, we need a dataset of multiple videos organized into groups with a topic keyword. However, since there exists no such dataset that fits our need, we self-compiled a dataset from the web. We queried the YouTube website with 10 search queries from the SumMe dataset [15], *i.e.*, each video set is collected using a certain query term, *e.g.*, *Statue of Liberty*. Note that the SumMe dataset contains only one video for each category, and thus is not suitable for our purpose. We used a duration filter "Short (∼4 minutes)" on YouTube search engine, and sampled first few videos from the search results such that each video set contained at least 10 minutes of videos. See Fig. 3 for an illustration of the 10 video categories, and Table 1 for descriptive statistics.

Features: We computed three types of visual feature descriptors for each frame: CENTRIST [38], Dense-SIFT (D-SIFT) [36], and HSV color moments [8]. CENTRIST generates a 254-D descriptor that checks whether the value of a center pixel is greater than its neighbors [38]. To capture the magnitude of pixel intensity differences and orientation gradients, we resized images to 620×420 resolution, and extracted a 3840-D D-SIFT with bin sizes 32 and 64 (2 scales) and step sizes as 3 times the bin size. To introduce color information, we divided a frame into 3×4 spatial cells, and for each cell extracted color moments in HSV color space (*i.e.*, mean, standard deviation and skewness), resulting in a 108-D descriptor. Each descriptor was L_2-normalized. For each frame, we concatenated three descriptors into one vector, and reduced the dimension to 400 using PCA. Shot-level feature was computed as mentioned in Sec. 3.1.

Evaluation: We evaluated the quality of query-specific summaries compared to human judgement. In particular, given the videos that were pre-processed into shots, we had three judges see the query term (*e.g.*, *Statue of Liberty*), and

Table 2. Mean average precision on top 5 and 15 results. * abbreviates video query for display convenience. See Table 1 for full names.

	Methods	Base*	Bike*	Eiffel*	Excavators*	Kids*	MLB	NFL	Notre Dame*	Statue*	Surfing	*Avg.*
Top 5	k-means	0.432	0.427	0.422	0.289	0.791	0.556	0.663	0.392	0.543	0.550	0.507
	LL	0.226	0.305	0.413	0.667	0.744	0.508	0.710	0.568	0.763	0.334	0.524
	COC	0.495	**0.802**	0.580	**0.713**	**0.859**	0.561	**0.762**	**0.803**	0.378	0.668	0.662
	MBF	**0.680**	0.788	**0.596**	0.690	0.798	**0.638**	0.680	0.715	**0.810**	**0.684**	**0.707**
Top 15	k-means	0.397	0.369	0.422	0.338	0.772	0.485	0.562	0.442	0.597	0.481	0.487
	LL	0.318	0.459	0.468	0.671	0.710	0.499	0.737	0.592	0.653	0.337	0.545
	COC	0.496	**0.795**	0.561	0.656	0.852	0.503	**0.823**	0.676	0.458	0.586	0.641
	MBF	**0.747**	0.663	**0.562**	**0.674**	**0.859**	**0.755**	0.760	**0.680**	**0.661**	**0.652**	**0.701**

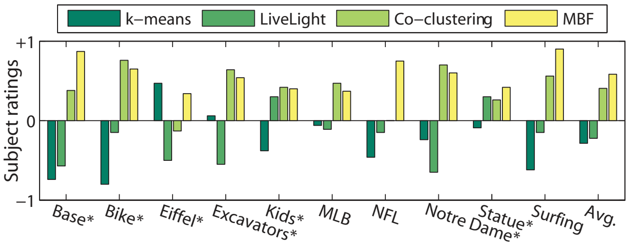


Figure 4. Concept visualization: Subject ratings.

select at least 10%, but no more than 50%, of shots for each video. The selected shots compiled individual preferences that the judges agreed to be relevant to the query. The ground truth was constructed by pooling together those shots selected by at least two judges. As an evaluation metric, we used the standard mean average precision (mAP), *i.e.*, the mean of average precision over all categories.

Competitive methods: We compare our method (MBF) against three baseline methods: k-means, LiveLight (LL) [40], and co-clustering (COC) [10]. For k-means, we generate a summary by selecting shots closest to each cluster centroid; we empirically set $k = 20$ that works well on a subset of videos. LiveLight generates a summary using online dictionary learning; we implemented it using the SPAMS library [26]. As reported in [40], we generated an initial dictionary of size 200 using the first three shots, and set the threshold for reconstruction error $\epsilon_0 = 0.15$. A video summary was generated as the shots with high reconstruction errors. For COC and MBF, we ranked the shots by their quality scores as described in Sec. 3.5. A final summary was selected as the shots with the highest quality scores, indicating a high degree visual co-occurrence.

Results: Table 2 shows the mAP on top 5 and top 15 shots included in the summaries. We can see that MBF achieved the highest mAP for both top 5 and top 15 results. For the top 15 results, MBF outperformed COC in 7 out of the 10 video sets. We note, however, that for cases where the video contains mostly repetitive events, *e.g.*, *Excavator river crossing* and *Kids playing in leaves*, MBF performed slightly worse than COC because MBF encourages the sparsity in co-occurring shot selection. LL performed slightly better than k-means. LL selects shots with large reconstruction errors; we believe this made the resulting summary less relevant to human-generate summaries. Both COC and MBF consistently outperformed k-means and LL, showing that the summaries of visually co-occurred shots are closer to human's selection. Our runtime analysis revealed that it took about 8 hours to extract image features and compute shot-level representations, while it took less than 0.5s to generate a summary using MBF. We used MATLAB implementation on a PC (Intel i7 3.5GHz).

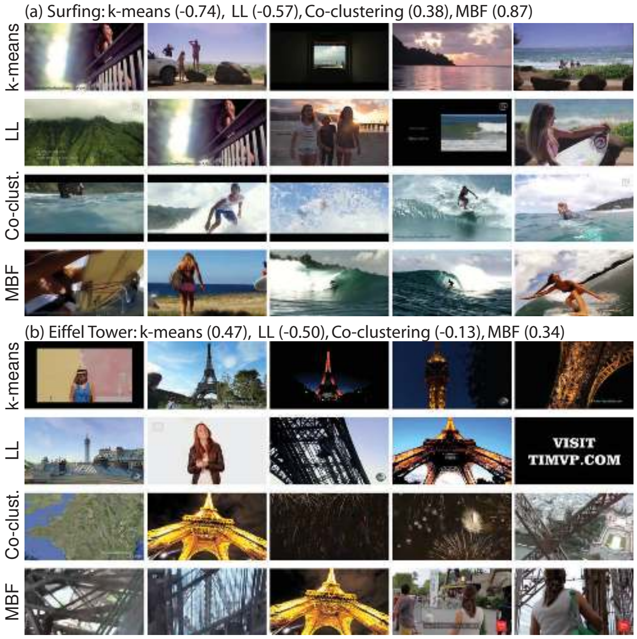
4.2. Concept visualization

A natural extension of video co-summarization is visualizing concept(s) from a collection of videos, *e.g.*, videos from the same channel. This section demonstrates the effectiveness of our approach on multi-video concept visualization, *i.e.*, given a collection of videos sharing the same topic, our goal is to generate a summary that describes the collection altogether.

We used our YouTube dataset for this experiment. From each video category, we generated a summary using the top 5 ranked shots. Note that we put together the shots according to their importance scores in a descending order, regardless of their actual temporal order. How to maintain temporal consistency in multi-video summarization remains as an open question [9]; we leave this as a future work.

Evaluation: We developed an AMT-like webpage similar to [19]. We designed the evaluation task as a quadruplet comparison, where each quadruplet consisted of 4 summaries generated by different methods. 20 subjects (14 males and 6 females, 23 to 33 years old) were shown a query term (*e.g.*, *Statue of Liberty*), and then were asked to label each summary as *good* (+1), *neutral* (0) or *bad* (-1) to describe the relevance between the query term and the video summary. One had to choose at least one *good* and one *bad* summary to continue. For each category, a summary that consists of top 5 shots were evaluated. A *subject rating* was computed as the averaged ratings from all subjects.

Fig. 4 shows that MBF outperformed competitive methods in terms of the average subject ratings across all video sets. Fig. 5(a) shows example summaries of *Surfing*, where MBF performed particularly well compared to other methods. We can see that the canonical scenes of *Surfing* (*e.g.*, surfing on the wave and walking on beach) were captured well, perhaps due to its high level of co-occurrence across



(a) Surfing: k-means (-0.74), LL (-0.57), Co-clustering (0.38), MBF (0.87)

(b) Eiffel Tower: k-means (0.47), LL (-0.50), Co-clustering (-0.13), MBF (0.34)

Figure 5. Concept visualization: Top-5 results on video collections of (a) *Surfing* and (b) *Eiffel Tower*.

videos. Fig. 5(b) shows, on the other hand, a less successful example where k-means performed better on capturing diverse shots of day and night views of the *Eiffel Tower*. As can be seen, MBF included the steel structure of Eiffel Tower and the tourist spots. Overall, our method generates summaries that better estimates human's visual concepts.

4.3. Objective evaluation on CMU-Mocap dataset

The two experiments above demonstrate the effectiveness of our approach via subjective evaluation. This section evaluates our method's ability to discover visual co-occurrence in an objective manner, with clear-cut ground-truth labels, using the CMU-Mocap dataset [1].

We used the *Subject 86* data that contains 14 sequences labeled with shot boundaries [3], where each sequence contains 4000$\sim$8000 frames and up to 10 human actions (out of a total of 48 pre-defined actions). See Fig. 6(a) for an illustration. To remove the redundancy in action labels, we grouped similar types of actions into 24 categories, *e.g.*, $\{$*arm rotating*, *rotate arms*, *right arm rotation*, *raise arms*, *both arm rotation*$\}$ are categorized as *arm raise*, $\{$*jump*, *jump on left leg*, *jump on right leg*$\}$ as *jump*, and so on. Each action was represented by root position, orientation and relative joint angles, resulting in a 30-D feature vector. We represented each frame using a 20-word dictionary (built by k-means) and soft-clustering. The shot-level feature was used as in Sec. 3.1.

Competitive methods: We compared our MBF method against three baselines: k-means, ACA [41] (temporal clustering), and co-clustering (COC) [10]. We performed k-means and ACA on a sequence concatenated by two input sequences, because the two methods do not consider video source information. For ACA, we set the parameter of maximal shot length to 60. Because ACA [41] performs a temporal pre-segmentation, we rounded the clustering results to the closest ground truth boundary. Except for our MBF method, we assigned the same number of initial clusters as the number of ground truth actions. Note that MBF does not require setting the initial number of clusters; it uses the quality function (6) to automatically determine the optimal number of bicliques.

Metric: To provide a quantitative evaluation on the quality of summaries, we introduce a metric similar to standard *precision*, *recall* and *F1 score*. Suppose we are given two sequences A and B that each contains a number of shots, and K retrieved clusters/bicliques $\mathcal{C} = \{\mathcal{C}_k\}_{k=1}^{K}$. Let ℓ_i be the label of the i-th shot, and $\mathcal{C}_k^{\mathsf{A}} = \{\mathcal{C}_k \cap \mathsf{A}\}$ the set of shots in both $\mathcal{C}_k$ and A (similarly for $\mathcal{C}_k^{\mathsf{B}}$). We define the *precision* for each cluster $\mathcal{C}_k$ as:

$$p(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k^{\mathsf{A}}| \cdot |\mathcal{C}_k^{\mathsf{B}}|} \sum_{i \in \mathcal{C}_k^{\mathsf{A}}, j \in \mathcal{C}_k^{\mathsf{B}}} I(\ell_i = \ell_j), \tag{7}$$

where $|\mathcal{C}_k|$ is the cluster size. Precision measures the ratio of the number of correctly discovered shot pairs to the number of total shot pairs in one cluster, resulting in a value within $[0, 1]$. A higher value of precision indicates a "purer" cluster, implying more pairs belonging to the same action. To measure the performance over all clusters, we compute the averaged precision (AP) defined as $AP(\mathcal{C}) = \frac{1}{K} \sum_{k=1}^{K} p(\mathcal{C}_k)$. Similarly, we compute *recall* for all retrieved clusters/bicliques:

$$r(\mathcal{C}) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k^{\mathsf{A}}, j \in \mathcal{C}_k^{\mathsf{B}}} I(\ell_i = \ell_j), \tag{8}$$

where $N = \sum_{i \in \mathsf{A}, j \in \mathsf{B}} I(\ell_i = \ell_j)$ is the number of true shot-pairs. A higher recall indicates a higher accuracy of retrieving similar shots shared between two videos. Given the definitions, we compute the F1 score as $F1(\mathcal{C}) = \frac{2AP(\mathcal{C})r(\mathcal{C})}{AP(\mathcal{C}) + r(\mathcal{C})}$.

We use an illustrative example to explain our metric. Suppose we have a pair of three-shot sequences $\mathsf{A} = [1_{\mathsf{A}}, 2_{\mathsf{A}}, 1_{\mathsf{A}}]$ and $\mathsf{B} = [2_{\mathsf{B}}, 1_{\mathsf{B}}, 1_{\mathsf{B}}]$, where each contains two shots labeled as "class 1" and one shot labeled as "class 2". An ideal clustering result should be $\mathcal{C}^{\star} = \{\{1_{\mathsf{A}}, 1_{\mathsf{A}}, 1_{\mathsf{B}}, 1_{\mathsf{B}}\}, \{2_{\mathsf{A}}, 2_{\mathsf{B}}\}\}$ with $AP(\mathcal{C}^{\star}) = 1$ and $r(\mathcal{C}^{\star}) = 1$. Now, suppose an algorithm produced the result as $\mathcal{C} = \{\{1_{\mathsf{A}}, 1_{\mathsf{B}}\}, \{1_{\mathsf{A}}, 1_{\mathsf{B}}\}, \{2_{\mathsf{A}}, 2_{\mathsf{B}}\}\}$ that divides class 1 into two clusters (*e.g.*, k-means with $k = 3$); our metric values will be $AP(\mathcal{C}) = 1$ and $r(\mathcal{C}) = 0.6$. In this way, precision measures an *intra-cluster* purity, while recall measures an *inter-cluster* purity, *i.e.*, it tells us the sensitivity on whether relevant shots are grouped in the same cluster.

Results: As an illustration purpose, we first performed experiments using only a pair of sequences *86_03* and

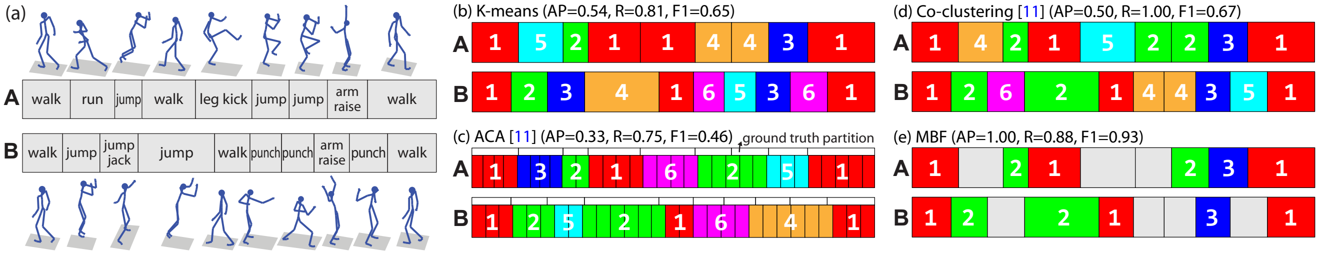


Figure 6. Discovering common shots between sequences (**A**) 86_03 and (**B**) 86_05 of the CMU-Mocap dataset. (a) the ground truth actions. (b)∼(e) the discovered results using k-means, ACA [41], co-clustering [10] and MBF (our method), respectively. (AP, R, F1) denotes the averaged precision, recall and F1 score, respectively. Shots indicated by the same numbers belong to the same cluster. White rectangles in (c) indicate the ground truth shot boundaries, in comparison with segmentation results of [41].Note that, compared to other approaches, our method can "skip" shots that do not co-occur between two sequences.

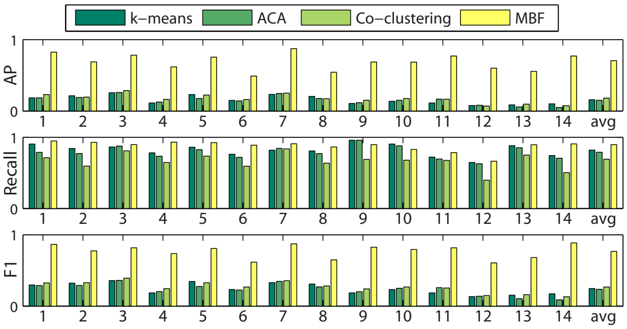


Figure 7. Performance comparison on the CMU-Mocap dataset. Three bar charts show AP, recall and F1 score for all competitive methods; x-axis indicates the number of sequences and the average of all sequences (denoted as avg).

86_05. Fig. 6 shows detailed results from this experiment. As can be seen, MBF achieved the best F1 score by identifying almost all common actions in each cluster, showing the effectiveness of discovering co-occurrences *between* video sequences. For k-means, ACA and co-clustering, we set the number of clusters as the number of ground truth actions ($K=6$). As shown in (b), k-means failed to group the same actions in one cluster, *e.g.*, the *jump* action was separated into two clusters 2 and 4. Both k-means and ACA clustered shots without considering the sources of shots (*i.e.*, corresponding video sequence). As a result, they were unable to discover co-occurring shots between sequences, *e.g.*, cluster 6 in (b), and clusters 3,4 in (c), which are undesirable to our objective. On the other hand, co-clustering in (d) considered pairwise clustering, and thus better discovered the shared content *between* sequences. Unlike all competitive methods, MBF in (e) relaxes the requirement of assigning each shot to a cluster, allowing our approach to discard irrelevant shots that appear only in a single video.

Next, we conducted an experiment on all pairs of sequences of *Subject 86*. For k-means, ACA and co-clustering, the number of clusters was set as the number of ground truth actions among two sequences. We report the averaged precision, recall and F1 score for each sequence pair, and evaluate the performance on a sequence $\mathbf{s}_i$ by averaging the metrics with all sequences $\{\mathbf{s}_j\}_{j\neq i}$. Fig. 7 shows comparison across different methods. As can be seen, MBF consistently achieved the highest AP across all sequences. We believe this is because MBF relaxed the requirement of assigning each shot a cluster label, and thus better targeted at finding relevant shots. However, for some sequences, MBF performed worse in recall, because MBF has a more strict quality control that may exclude a shot that was dissimilar to other shots in a cluster. Overall, MBF attained a significantly higher F1 score than other methods, validating its usage for discovering visual co-occurrences.

In addition to the results reported, for a complementary comparison, we also evaluated the performance for both COC and MBF on the shots that are selected by MBF. We used the metrics described above, and computed the averaged (AP, R, F1) over all pairs, resulting in (0.33,0.46,0.40) for COC and (0.66,0.81,0.70) for MBF, as in Fig. 7. This shows the capability of MBF in selecting a subset of shots that preserves visual similarity, where COC attempts to match all shots simultaneously. The two results together show more clearly how MBF achieves more accurate matches by ignoring a majority of dissimilar shots.

5. Conclusion

We presented *video co-summarization*, a novel perspective to video summarization that summarizes one, or multiple, videos by identifying visual co-occurrences among a video collection. To deal with the sparsity of co-occurring shots, we developed a Maximal Biclique Finding (MBF) algorithm. The advantages of MBF include: It is optimized to find shots that appear *jointly* across multiple videos, even if they are sparse; it discards patterns that are only specific to a single video, thus are less relevant to the main topic; it is parallelizable with closed-form updates, and thus is scalable. We showed the effectiveness of our approach compared to several popular unsupervised techniques via both qualitative and quantitative experiments. Moving forward, we plan to improve our method using active learning or weakly-supervised learning, providing a more principled way to weigh nodes in the bipartite graph.

References

[1] CMU Mocap. http://mocap.cs.cmu.edu/.

[2] G. Alexe, S. Alexe, Y. Crama, S. Foldes, P. L. Hammer, and B. Simeone. Consensus algorithms for the generation of all maximal bicliques. *Discrete Applied Mathematics*, 145(1):11–21, 2004.

[3] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. Segmenting motion capture data into distinct behaviors. In *Graphics Interface Conference*, 2004.

[4] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

[5] F. Chen and C. De Vleeschouwer. Formulating team-sport video summarization as a resource allocation problem. *TCSVT*, 21(2):193–205, 2011.

[6] W.-S. Chu, C.-P. Chen, and C.-S. Chen. MOMI-cosegmentation: Simultaneous segmentation of multiple objects among multiple images. In *ACCV*, 2010.

[7] W.-S. Chu, F. Zhou, and F. De la Torre. Unsupervised temporal commonality discovery. In *ECCV*, 2012.

[8] Y. Cong, J. Yuan, and J. Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *TMM*, 14(1):66–75, 2012.

[9] K. Dale, E. Shechtman, S. Avidan, and H. Pfister. Multi-video browsing and summarization. In *CVPRW*, 2012.

[10] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *SIGKDD*, 2001.

[11] S. Feng, Z. Lei, D. Yi, and S. Z. Li. Online content-aware video condensation. In *CVPR*, 2012.

[12] M. Fleischman, B. Roy, and D. Roy. Temporal feature induction for baseball highlight classification. In *ACM MM*, 2007.

[13] C. A. Floudas and V. Visweswaran. A global optimization algorithm for certain classes of nonconvex nlps. *Computers & chemical engineering*, 14(12):1397–1417, 1990.

[14] G. H. Golub and C. F. Van Loan. *Matrix computations*. JHU Press, 2012.

[15] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[16] M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *CVPR*, 2011.

[17] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *ECCV*. 2014.

[18] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, 2013.

[19] G. Kim, L. Sigal, and E. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014.

[20] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[21] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.

[22] D. Liu, G. Hua, and T. Chen. A hierarchical visual model for video object summarization. *TPAMI*, 32(12):2178–2190, 2010.

[23] Z. Liu, A. Basso, D. C. Gibbon, B. Shahraray, and E. M. Zavesky. Brief and high-interest video summary generation, 2012. US Patent 8,195,038.

[24] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013.

[25] K. M. Mahmoud, N. M. Ghanem, and M. A. Ismail. Unsupervised video summarization via dynamic modeling-based hierarchical clustering. In *ICMLA*, 2013.

[26] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19–60, 2010.

[27] A. G. Money and H. Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Jrnl. of Visual Comm. and Image Repres.*, 19(2):121–143, 2008.

[28] N. Nagarajan and C. Kingsford. Uncovering genomic reassortments among influenza strains by enumerating maximal bicliques. In *Int'l. Conf. on Bioinfo. and Biomedicine*, 2008.

[29] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[30] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang. Video summarization and scene detection by graph modeling. *TCSVT*, 15(2):296–305, 2005.

[31] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014.

[32] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *TPAMI*, 30(11):1971–1984, 2008.

[33] M. Sun, A. Farhadi, and S. Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014.

[34] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. on Multimedia Computing, Comm., and Apps*, 3(1):3, 2007.

[35] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *IJCV*, 88(2):284–302, 2010.

[36] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *ACM MM*, 2010.

[37] F. Wang and B. Merialdo. Multi-document video summarization. In *ICME*, 2009.

[38] J. Wu and J. M. Rehg. Centrist: A visual descriptor for scene categorization. *TPAMI*, 33(8):1489–1501, 2011.

[39] X. Wu, C.-W. Ngo, and Q. Li. Threading and autodocumenting news videos: a promising solution to rapidly browse news topics. *IEEE Signal Proc. Mag.*, 23(2):59–68, 2006.

[40] B. Zhao and E. Xing. Quasi real-time summarization for consumer videos. In *CVPR*, 2014.

[41] F. Zhou, F. De la Torre, and J. K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *TPAMI*, 35(3):582–596, 2013.

[42] G. Zhu, Q. Huang, C. Xu, Y. Rui, S. Jiang, W. Gao, and H. Yao. Trajectory based event tactics analysis in broadcast sports video. In *ACM MM*, 2007.