Video-Story Composition via Plot Analysis

Jinsoo Choi Tae-Hyun Oh In So Kweon
KAIST, Republic of Korea
jschoi@rcv.kaist.ac.kr, thoh.kaist.ac.kr@gmail.com, iskweon@kaist.ac.kr

Abstract

We address the problem of composing a story out of multiple short video clips taken by a person during an activity or experience. Inspired by plot analysis of written stories, our method generates a sequence of video clips ordered in such a way that it reflects plot dynamics and content coherency. That is, given a set of multiple video clips, our method composes a video which we call a video-story. We define metrics on scene dynamics and coherency by dense optical flow features and a patch matching algorithm. Using these metrics, we define an objective function for the video-story. To efficiently search for the best video-story, we introduce a novel Branch-and-Bound algorithm which guarantees the global optimum. We collect the dataset consisting of 23 video sets from the web, resulting in a total of 236 individual video clips. With the acquired dataset, we perform extensive user studies involving 30 human subjects by which the effectiveness of our approach is quantitatively and qualitatively verified.

1. Introduction

People have the natural desire to capture and store personal experiences and memories. Today, we are able to record our activities more easily with decreasing cost of cameras and media storages. Moreover, with the success of smart phones and applications, photos and videos have become omnipresent in our daily lives. Consequently, people tend to capture photos and record videos without a *limited storage burden* and process them later. Unfortunately, manual post-processing of these contents is usually tedious, and thus a need for an automatic summarization of contents has arisen leading to many research on this topic [11, 17, 20]. This need has arisen due to the people's tendency to preserve only meaningful contents. More recently, rather than simply extracting summaries, many people choose to generate meaningful *stories* out of photos [15] and videos [20], and many applications (*e.g.* 1 Second Everyday, Roadmovies, Snapmovie) attempt to provide this type of media. Basically, works on photo story generation deal with aligning multiple photos supposedly into a temporal order. Similar works such as photo sequencing [3, 7] take a photo sequence input of the same scene or action and produce a temporally ordered photo sequence. In video summarization, a long video is summarized into a shorter version while maintaining the overall story [20]. Mobile applications recently gaining much popularity like 1 Second Everyday and Roadmovies allow the user to capture short video clips with a mobile camera and then concatenate the short video clips taken one after another to produce a single video.

Figure 1: **Story plot analysis.** A written story consists of an *exposition*, *rising action*, *climax*, and *resolution*. Given a collection of independently captured video clips, we aim to build a video sequence with a story plot. This figure shows our results with *River-Surfing* short video clips. The sequence starts the exposition with a clip showing a stationary scene of the environment. The following scenes portray consecutive rising action and eventually show the actual river surfing at the climax leading to similar scenes until resolution. Notice the adjacent scenes are visually coherent as well which do not interrupt the flow of the story.

We address the problem of composing an ordered video clip sequence which we call *video-stories* out of multiple video clips taken by a person during an activity or experience. When a person captures separate video clips later to be concatenated, that person may not be so much concerned about the overall structure of the resulting video. For example, let's say a person captures separate video clips while on a *surfing trip*. The person may start off recording multiple

clips of the actual surfing. Then, the person may decide to appreciate the environment of the scene and record the surrounding environment. Next, the person may choose to capture family and friends. Then, the person may again resume capturing the actual surfing. The video clips captured may well contain all of the aspects the person intended to capture during the surfing trip. However, when the videos are concatenated together in temporal order, it may not reflect a sense of structure, but rather simply a series of individual experiences on the same day. Thus, our goal is to take separate video clips and produce a video reflecting plot structure and sense of story, namely a video-story.

Our work is inspired by the notion of plot analysis for written stories. Fig. 1 shows the notion of a plot diagram and its component notions along with actual results obtained by our method. A typical story contains an *exposition* where it introduces the beginning and setting of the story. The *rising action* phase represents the intermediate events between the beginning and the climax of the story which typically involves building up action and dynamics. The *climax* represents the main event, and *resolution* marks the ending of the story. Simply, a story gradually increases in action dynamics and reaches its peak at the climax and gradually reaches the end of the story. Although the dynamics present in the resolution varies among stories, most will have an ending with more activity than its beginning. Also, stories will tend to be coherent in its contents, meaning it will not have abrupt changes in scenes nor abrupt revisitation of scenes. Our goal is to structure the individual video clips into a video-story following this general plot structure while maintaining coherent story transitions.

An overview of our approach is as follows. First, given multiple short video clips, we measure the amount of activity in the individual clips via a dynamicity measure which we define. We also measure coherency between the clips based on a patch matching algorithm. Next, we design an objective function that scores video sequences depending on how well it represents a story plot structure and how smooth and coherent the clip transitions are. The dynamicity measure is used to evaluate how well the candidate video sequence follows the story plot structure. Similarly, the dissimilarity measure is utilized in evaluating the overall smoothness in clip transitions. Finally, we find the optimal solution via Branch-and-Bound algorithm.

Our main contribution is the idea of composing a structured video out of multiple short videos that has story-like qualities. We propose a general framework that achieves this goal. To the best of our knowledge, our work is the first to address the problem of automatically composing a story sequence with multiple video clips separately captured. In order to accomplish this, (1) we define a dynamicity metric based on optical flow features to reflect activity in video clips. (2) We introduce a reliable bidirectional patch matching algorithm to measure the dissimilarity between clips. (3) We design an objective function that returns the best chain of clips representing sense of story and coherency. (4) We introduce a Branch-and-Bound algorithm to efficiently find the optimal solution. (5) We construct a dataset of 23 video sets (total of 236 individual clips) and conduct extensive experiments involving 30 subjects.

2. Related Work

Recent research on generating a *story* form of media is mainly dealt in video summarization and image sequencing which we review in this section.

Video summarization. Works on generating a summary of a long video can take different representations. Keyframe-based methods represent the video summary as a sequence of keyframes selected from the video. Wolf *et al*. [28] used optical flow features and Liu *et al*. [19] used object tracks to select the set of keyframes. Methods including mosaic-based representation [1] have been explored to efficiently cluster scenes into physical settings, and user interaction based approaches [10] have been proposed to render action summary layouts. Lee *et al*. [17] proposed to find important people and objects from regional cues for egocentric video summarization. Also, some optimization approaches include works that represent a video as a high dimensional trajectory curve and analyze via binary curve splitting algorithm [6]. Apart from keyframe representation, some works represent summaries via skims or subshots. Naturally, some works address spatio-temporal features [16, 25] for subshot selection. Ngo *et al*. [21] proposed a motion attention model based on human perception to compute subshot quality. Feldman *et al*. [8] proposed a novel core-set algorithm for k-segmentation of streaming data. On the other hand, a supervised learning approach [11] has also been conducted for selecting appropriate subshots. In addition to feature based approaches, Lu *et al*. [20] proposed to measure influence between subshots based on visual objects in egocentric videos. In robotics, Volkov *et al*. [26] proposed a feature-based core-set algorithm for summarizing video data. Video summarization focuses on representing the summary of a single long video, whereas our approach addresses generating a video-story from multiple video clips.

Image sequencing. Many works on image sequencing have attempted to align the order of images that lack temporal ordering. Basha *et al*. [3] proposed to detect static and dynamic features and then conduct an epipolar geometry based approach to find temporally ordered image sequences. Feature based methods including motion signature based synchronization [7] have been explored as well. Wang *et al*. [27] jointly utilized image and text information to generate image storylines. Also, works have addressed

using geolocation and path cues [5, 13] to generate image sequences illustrating the tourist's experience in temporal order. Apart from temporally aligning photo sequences, Averbuch-Elor *et al*. [2] introduced a spectral technique for recovering the spatial order of photos taken by a group of people around the same event. Works on storyline graphs have been introduced for large-scale web images [15], personal photos [22], and outdoor activity classes [14]. Image sequencing and storyline graphs aim to identify the temporal ordering of images. We aim to instill a sense of story into our proposed video-story generation method.

3. Approach - Composing the Video-Story

Our approach addresses the problem of making a video-story out of multiple video clips. It is our job to find the best way to order these clips such that the resulting video-story (1) follows a story plot structure, and (2) is coherent in presenting the subsequent clips one-by-one.

Consider we are given N video clips denoted as $C = \{c_1, ..., c_N\}$. Let $\boldsymbol{s} \subset P$ denote an ordered sequence, where P denotes the set of all possible permutations of C. Our goal is to find the optimally ordered sequence:

$$\boldsymbol{s}^* = \arg\min_{\boldsymbol{s} \subset P} Q(\boldsymbol{s}), \tag{1}$$

where $Q(\boldsymbol{s})$ is an objective function:

$$Q(\boldsymbol{s}) = \alpha \mathcal{P}(\boldsymbol{s}) + (1 - \alpha)\mathcal{D}(\boldsymbol{s}). \tag{2}$$

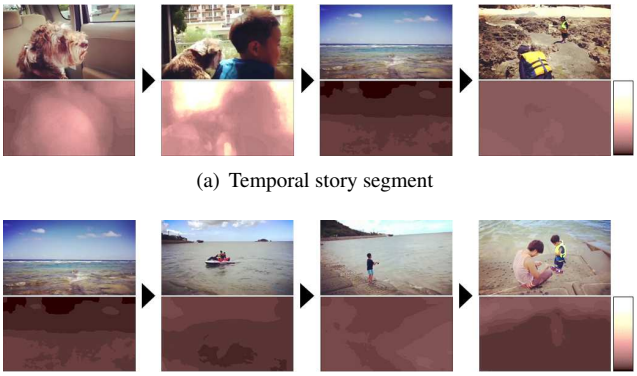
The *Plot Penalty* term $\mathcal{P}(\boldsymbol{s})$ denotes the penalty given to a candidate sequence of clips depending on how poorly it is structured as a story. The *Dissimilarity* term $\mathcal{D}(\boldsymbol{s})$ denotes the dissimilarity present in adjacent clips given to a candidate sequence of clips. We provide detailed explanations of these terms in Sec. 3.1.

Directly selecting the best permutation of clips through an exhaustive search is NP-hard. Thus, we need an efficient algorithm to find the optimal solution. We introduce a novel Branch-and-Bound algorithm that efficiently finds the best video-story while guaranteeing global optimum. Detailed explanations are provided in Sec. 3.2.

3.1. Story Scores for Candidate Video-Stories

We provide detailed illustrations on the terms introduced in Eq. (2) and their importance in evaluating the story quality of a candidate video-story sequence.

Plot dynamics of the overall video. The *Plot Penalty* term $\mathcal{P}(\boldsymbol{s})$ indicates how much a candidate sequence of video clips represents a poor story structure and is crucial to the novelty of our approach. Specifically, this term penalizes for not following the general story plot as illustrated in Fig. 1. Thus, a good video-story will in general contain the essential aspects of a story plot such as the presence of a gradual rising action phase, a climax and a resolution. Consequently, a video-story that follows the general plot structure will return a small *Plot Penalty* score.



(a) Temporal story segment

(b) Proposed story segment

Figure 2: **Plot dynamics.** This figure shows story segments taken from (a) the original temporal sequence and (b) the video-story generated by our proposed method. In these video-stories, clips (shown as snapshots in the top rows) transition from left to right. The heat map (shown below each snapshot) shows the accumulation of dense optical flow magnitudes. Our proposed story reflects smooth rising dynamics with coherent scenes whereas the original sequence shows orderless dynamics with inconsistent scenes.

Now, the components of a plot (*i.e*. exposition, rising action, climax, and resolution) are based on the notion of amount of activity. In other words, plot components are structured depending on how dynamic each scenes are. We define a *dynamicity* measure based on dense optical flow features [18] to represent the amount of activity present in a video clip. Recent methods using dense optical flow have shown efficient video representation for action recognition tasks and have achieved state-of-the-art results. Given a video clip c_l with length $L(c_l)$ and displacement vectors $\Delta \boldsymbol{T}_t = \boldsymbol{T}_{t+1} - \boldsymbol{T}_t$ at time t, we define the dynamicity as

$$D(c_l) = \frac{\sum_{j=1}^{L(c_l)} \|\Delta \boldsymbol{T}_j\|_2}{L(c_l)}. \tag{3}$$

This measure represents the amount of activity contained in clip c_l normalized by the clip length $L(c_l)$.

Camera motion caused by the user however cannot be thought of as a dynamic component of the clip, because it usually has little to do with the actual dynamics of the scene depicted. Thus, we preprocess the displacement vectors $\Delta \boldsymbol{T}_t$ by taking the camera motion into account. To estimate camera motion, we extract SURF [4] descriptors in each frame and compute homographies with RANSAC [9] for consecutive frames. We use the homographies to cancel out the camera motion to produce displacement vectors $\Delta \boldsymbol{T}_t$ containing pure dynamics of the scene.

Based on the dynamicity measure, given a sequence $\boldsymbol{s}$ with N video clips, we define the *Plot Penalty* term $\mathcal{P}(\boldsymbol{s})$ as

$$\mathcal{P}(\boldsymbol{s}) = \sum_{i=1}^{N-1} \tilde{P}(\boldsymbol{s}_i, \boldsymbol{s}_{i+1}), \tag{4}$$

where

$$\tilde{P}(\boldsymbol{s}_i, \boldsymbol{s}_{i+1}) = \begin{cases} D(\boldsymbol{s}_i) - D(\boldsymbol{s}_{i+1}) & \text{if } D(\boldsymbol{s}_{i+1}) < D(\boldsymbol{s}_i), \\ 0 & \text{if } D(\boldsymbol{s}_{i+1}) \geq D(\boldsymbol{s}_i). \end{cases} \tag{5}$$

For clarification, the $\mathcal{P}(\boldsymbol{s})$ term penalizes candidate video-stories on decreasing dynamicity of adjacent clip pairs. This simple formulation in fact provides an elegant representation of all component properties of a story plot. First of all, the exposition (first clip of the video-story) would tend to start off with low dynamics as suspected. Also, the overall video-story would most likely follow the rising action phase due to how the penalty measure $\tilde{P}(\boldsymbol{s}_i, \boldsymbol{s}_{i+1})$ is defined. Consequently, the climax may come after the rising action phase. Since most stories typically end with higher dynamics than its exposition, this formulation implicitly allows the resolution to end with high dynamics.

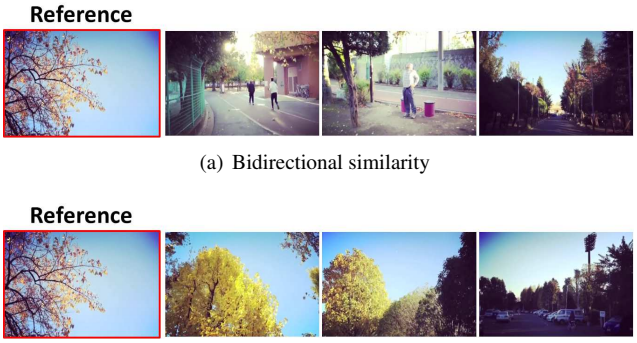
Furthermore, notice that the magnitude of penalization is equal to the dynamicity difference. This is backed with the intuition that a significant drop in dynamics must be penalized more than a subtle drop in dynamics. Fig. 2 shows a comparison of plot dynamics of a temporally ordered video-story segment and our proposed video-story segment.

Coherency of story contents. The *Dissimilarity* term $\mathcal{D}(\boldsymbol{s})$ indicates how much a candidate sequence of video clips contains dissimilar clips adjacent to each other. A good story usually reflects smooth transitions of events. For instance, a scene of the ocean would likely be followed by other scenes showing the ocean rather than a cascade of unrelated scenes. In this sense, a video-story that presents better coherency in contents with smoother transitions between clips will return a smaller *Dissimilarity* score.

We first introduce how to measure dissimilarity between two clips by modifying the bidirectional similarity [24] measure which is based on a patch matching algorithm. We choose to define our dissimilarity measure in this way in order to take advantage of its property: robustness to local changes between video frames. Given two clips c_1 and c_2, let G and H each denote a set of patches from c_1 and c_2, respectively. The dissimilarity measure is defined as

$$\begin{aligned} d(c_1, c_2) = & \frac{1}{n_1} \sum_{G \in c_1} \min_{H \in c_2} Dist(G, H) + \\ & \frac{1}{n_2} \sum_{H \in c_2} \min_{G \in c_1} Dist(G, H), \end{aligned} \tag{6}$$

where $Dist(G, H)$ is obtained by the Sum of Squared Distance (SSD), measured in CIE $L^*a^*b^*$ color space and normalized by the patch size. The original implementation of the dissimilarity measure allows more than one patch from a clip to correspond to the same patch in the other clip. However, upon computing the dissimilarity measure, we enforce a one-to-one correspondence requirement on sets G and H. That is, each and every patch taken from clips c_1 and c_2 must have exclusive correspondences. Consequently, this imposes an emphasis on global dissimilarity between clips as a whole. The original bidirectional similarity is mainly used in spacial and/or temporal summarization of images or videos. That is, bidirectional similarity is used to make smaller and/or shorter versions of an input image or video, which naturally emphasizes the need to find local similarities. Since our approach requires to find the overall similarity (or dissimilarity) across clips, an emphasis on finding the global similarity better fits our needs, and thus the one-to-one correspondence requirement is enforced. An illustration of this aspect is shown in Fig. 3.



(a) Bidirectional similarity

(b) Bidirectional similarity with one-to-one correspondence

Figure 3: **Effect of one-to-one correspondence requirement.** This figure shows top three similar (smallest dissimilarity) clips of a reference clip (leftmost column) using (a) the original bidirectional similarity and (b) the bidirectional similarity with one-to-one correspondence requirement. Notice that the addition of this requirement emphasizes global similarity between clips as a whole rather than local similarity.

Given N video clips and a clip sequence $\boldsymbol{s}$, we define the *Dissimilarity* term $\mathcal{D}(\boldsymbol{s})$ as follows.

$$\mathcal{D}(\boldsymbol{s}) = \sum_{i=1}^{N-1} d(\boldsymbol{s}_i, \boldsymbol{s}_{i+1}), \tag{7}$$

which is simply the sum of all dissimilarity measures between adjacent clips. This term accounts for the smooth scene transitions illustrated in Fig. 1 and Fig. 2(b).

3.2. Searching for the Optimal Video-Story

Exhaustively searching for the optimal story sequence from all possible permutations of video clips is NP-hard. In this section, we provide an efficient way to find the global optimum based on the Branch-and-Bound algorithm [12] with breadth-first-search. We start by a brief introduction

Algorithm 1 Lower-bound score of a subspace

Input: Subspace $I^n \in \mathbb{N}^N$
Compute $\hat{\mathcal{P}} = \sum_{i=1}^{n-1} \tilde{P}(I_i^n, I_{i+1}^n)$
Compute $\hat{\mathcal{D}} = \sum_{i=1}^{n-1} d(I_i^n, I_{i+1}^n)$
if $n < N$ **then**
    Assign previously used elements of $\tilde{\boldsymbol{D}}$ as infinity
    $m = \min(\tilde{\boldsymbol{D}})$
    $\hat{\mathcal{D}} = \hat{\mathcal{D}} + (N - n)m$
end if
Lower-bound score: $L^b = \alpha\hat{\mathcal{P}} + (1 - \alpha)\hat{\mathcal{D}}$

Algorithm 2 Upper-bound score of a subspace

Input: Subspace $I^n \in \mathbb{N}^N$
Let $\hat{\boldsymbol{s}}$ denote a sequence in subspace I^n where $\hat{\boldsymbol{s}}_{1:n} = I^n_{1:n}$
if $n < N$ **then**
    for $j = n + 1 : N$ **do**
        $\tilde{\boldsymbol{P}}^j = [\tilde{P}(\hat{\boldsymbol{s}}_j, c_1), \cdots, \tilde{P}(\hat{\boldsymbol{s}}_j, c_N)]$
        $\boldsymbol{d}^j = [d(\hat{\boldsymbol{s}}_j, c_1), \cdots, d(\hat{\boldsymbol{s}}_j, c_N)]$
        Assign elements at clip indices already taken as infinity
        $\boldsymbol{q} = \alpha\tilde{\boldsymbol{P}}^j + (1 - \alpha)\boldsymbol{d}^j$
        $\hat{\boldsymbol{s}}_{j+1} = \arg\min_i \boldsymbol{q}_i$
    end for
end if
Upper-bound score: $U^b = \alpha\mathcal{P}(\hat{\boldsymbol{s}}) + (1 - \alpha)\mathcal{D}(\hat{\boldsymbol{s}})$

to Branch-and-Bound, then illustrate how the bounds are defined, and finally present our search procedure.

Introduction to Branch-and-Bound. The basic idea of Branch-and-Bound (BnB) is to divide the search space into smaller subspaces and discard subspaces that cannot contain a better solution than the current one. The discard decision is made by a rejection test based on the bounds of the subspace. If a subspace passes the rejection test, then it is again partitioned into smaller subspaces. The size of the subspaces iteratively decreases and the current solution converges to the global optimum. It is important to define tight intervals between the lower and upper-bounds because it affects the algorithm speed. If the intervals are tighter, early rejections of subspaces will occur more frequently. Thus, the number of branches (subspace subdivisions) are reduced, leading to a faster search procedure. In order to apply BnB to our problem, we specify the subspace subdivision scheme (*i.e.* how to divide subspaces into smaller subspaces), and define how to obtain the lower and upper-bounds of a subspace. This leads us to develop the first video-story search algorithm whose global optimality is guaranteed.

Defining lower and upper bounds. Defining how the bounds are computed for a subspace is important since it forms the basis of the rejection test which affects the overall algorithm efficiency. Before we explain how the bounds are defined, we briefly describe how the subspace branches are defined. Given N video clips, we define the search subspace $I^n \in \mathbb{N}^N$ in the form of a sequence of natural numbers with length N. Let the first n entries of subspace I^n be specified by clip indices, then the remaining entries are left blank to define the search area of the subspace. With each branching, the next entry (*i.e.* $(n+1)$-th entry) of the preceding subspace branch I^n is specified by a clip index (*i.e.* generating $I^{(n+1)}$), and consequently converges iteratively to a single optimum sequence of clips (*i.e.* our best video-story sequence).

For a subspace I^n, where the first n entries are fixed, the lower-bound of a subspace is acquired by first obtaining the lower-bound *Plot Penalty* score $\hat{\mathcal{P}}$ by Eq. (4) only up to the n-th entry. The lower-bound *Dissimilarity* score $\hat{\mathcal{D}}$ is partially obtained by Eq. (7) up to the n-th entry, and then must be completed as described as follows. Here, let us define the dissimilarity matrix $\tilde{\boldsymbol{D}}$ as the symmetric matrix where its elements are the dissimilarity measures between clip pairs as

$$\tilde{\boldsymbol{D}} = \begin{bmatrix} d(c_1, c_1) & d(c_1, c_2) & \cdots & d(c_1, c_N) \\ d(c_2, c_1) & d(c_2, c_2) & \cdots & d(c_2, c_N) \\ \vdots & \vdots & \ddots & \vdots \\ d(c_N, c_1) & d(c_N, c_2) & \cdots & d(c_N, c_N) \end{bmatrix}. \quad (8)$$

For each of the remaining entries after the n-th entry in I_n, we add the minimum element of $\tilde{\boldsymbol{D}}$ excluding the elements already used to partially calculate $\tilde{\boldsymbol{D}}$. The lower-bound score is obtained by a weighted sum of $\hat{\mathcal{P}}$ and $\hat{\mathcal{D}}$ in the same way as described in Eq. (2). A detailed algorithm for computing the lower-bound score is shown in Alg. 1. Notice that no sequence in the subspace I^n can possibly have a score lower than the lower-bound of its subspace, which indicates that the lower-bound definition is suitable.

The upper-bound score of a subspace can be obtained by finding an arbitrary sequence within the subspace and taking its score. We can define the upper-bound score as such because the score of any sequence within a subspace is always greater than or equal to the lowest score possible in that subspace. Although selecting any sequence at random would suffice as the upper-bound, we would like to make a tight interval. Thus, we would like to find a sequence with a low score, but would also like to find it fast (for sufficient algorithm speed). We define our upper-bound as the score of a sequence in the subspace found by a sequential search method. For a subspace I^n where the first n entries are fixed, we define a sequence $\hat{\boldsymbol{s}}$ with the same n entries. We assign the next entry (*i.e.* $(n+1)$-th entry) with a clip index returning the lowest score, and repeat until the sequence $\hat{\boldsymbol{s}}$ is complete. This is done by defining vectors $\tilde{\boldsymbol{P}}^j$ and $\boldsymbol{d}^j$ whose elements represent all pairwise penalty measures and dissimilarity measures with $\hat{\boldsymbol{s}}_j$ respectively, where

Algorithm 3 Branch-and-Bound for optimal video-story

Input: Search space $I^0 \in \mathbb{N}^N$ (*i.e.* Initial search space with no fixed entry)
for $n = 0 : N-1$ **do**
  Subdivide I^n by assigning remaining clip indices to the $(n+1)$-th entry
  Store subdivided subspaces in $\mathcal{L}_I$
  Compute lower and upper-bound scores: L^b and U^b, and store in $\mathcal{L}_b$
  $U^{b*} = \min U^b \in \mathcal{L}_b$
  Remove all subdivided subspaces from $\mathcal{L}_I$ whose $L^b > U^{b*}$
end for
Return: I^N (*i.e.* The subspace I^N last to remain in $\mathcal{L}_I$ is the only remaining subspace and represents the global optimal video-story sequence $\boldsymbol{s}^*$)

$\hat{\boldsymbol{s}}_j$ is the j-th element of $\hat{\boldsymbol{s}}$. Obviously, already selected clip indices cannot be selected again. The upper-bound score is obtained by a weighted sum of $\mathcal{P}(\hat{\boldsymbol{s}})$ and $\mathcal{D}(\hat{\boldsymbol{s}})$ in the same way as Eq. (2). A detailed algorithm is shown in Alg. 2.

Search procedure. An example illustration of the search procedure is provided in Fig. 4. Ultimately, the goal is to find the global optimal story sequence $\boldsymbol{s}^* \in \mathbb{N}^N$ from a search space $I^0 \in \mathbb{N}^N$ (which has no entries fixed with video clip indices). The BnB algorithm iteratively subdivides the search space by fixing the entries of I^0 with clip indices one-by-one. The subdivided subspaces are stored in the subspace list $\mathcal{L}_I$. Also in each iteration, the associated bounds are computed and stored in list $\mathcal{L}_b$. Since our problem is a minimization problem, the rejection test decides to discard a subspace when its corresponding lower bound is greater than the current minimum upper bound. We are safe to remove these subspaces from $\mathcal{L}_I$ since it signifies that the best solutions drawn from these subspaces are worse than any solution drawn from the current best subspace, and thus the optimal solution cannot be within these subspaces. The procedure stops when the last entry (*i.e.* N-th entry) is assigned leaving only one subspace I^N which represents the global optimal story sequence $\boldsymbol{s}^*$. Notice that the lower and upper-bounds of I^N is equal to each other and thus signifies the algorithm's convergence to the optimal solution. The detailed algorithm of the optimal video-story search procedure is shown in Alg. 3.

4. Experimental Results

We now analyze and evaluate our method. Since evaluating the quality of video-stories is a subjective task, we conduct extensive user studies to quantitatively evaluate our method. We provide detailed explanations on the evaluation settings, evaluation tasks via user studies, and quantitative and qualitative results.

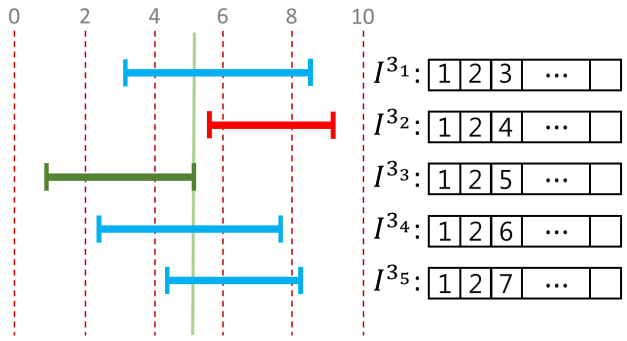


Figure 4: **Branch-and-Bound search procedure.** This shows an example illustration of BnB regarding subspaces with 3 fixed entries. The subspace I^{3_2} has a lower-bound larger than the minimum upper-bound, thus the subspace I^{3_2} is removed from the subspace list $\mathcal{L}_I$. This means that any story sequence starting with the clip indices $(1, 2, 4)$ cannot be the optimal story sequence.

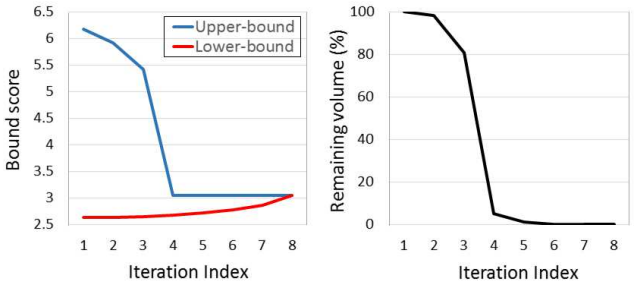


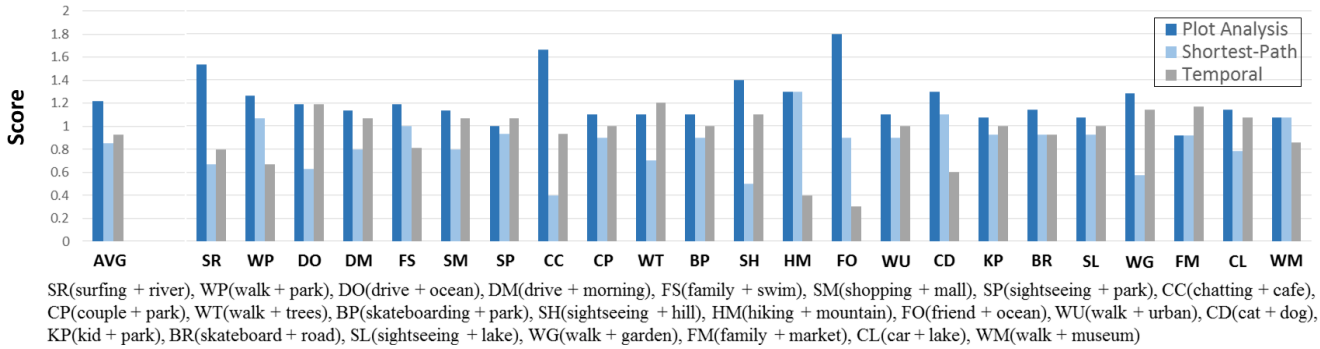
Figure 5: Convergence of bounds and search space volume.

Dataset. We collect 23 user-made video-stories from YouTube. Each video-story consists of 8-12 video clips which are 2-3 seconds long. This gives a total of 236 video clips for our dataset. The contents of our dataset include many activities (*e.g.* sightseeing, skateboarding, walking, surfing, shopping, driving, swimming, *etc.*) at various locations (*e.g.* river, park, ocean, streets, mall, landmarks, museum, marketplace, garden, beach, *etc.*).

Methods for comparison. We provide evaluation results on the following methods.

• *Plot Analysis* refers to our method described in Sec. 3. The weight parameter α in Eq. (2) is set to 0.5 in order to equally emphasize plot dynamics and coherency of story contents. We stress that this parameter was not tuned, although tuning this parameter via cross-validation could improve the results. The resulting video-story is found by the BnB algorithm and Fig. 5 shows an illustration of the lower and upper-bound convergence and search space volume convergence to prove that the bounds are valid.

• *Shortest-path*: We construct a graph connecting all pairs of clips weighted by the dissimilarity measures. We select the shortest-path sequence from the graph. It is natural to think that smooth transitions with similar clips next to each are enough to make a sufficient video-story. There-



SR(surfing + river), WP(walk + park), DO(drive + ocean), DM(drive + morning), FS(family + swim), SM(shopping + mall), SP(sightseeing + park), CC(chatting + cafe), CP(couple + park), WT(walk + trees), BP(skateboarding + park), SH(sightseeing + hill), HM(hiking + mountain), FO(friend + ocean), WU(walk + urban), CD(cat + dog), KP(kid + park), BR(skateboard + road), SL(sightseeing + lake), WG(walk + garden), FM(family + market), CL(car + lake), WM(walk + museum)

Figure 6: **Pairwise preference scores.** The scores represent pairwise preference scores normalized by the number of subjects. The leftmost bar set shows the average preference scores. The labels indicate the contents of the video set. Since the preferences are recorded in a pairwise manner, the score should be at least higher than 1 to validate that our method is superior to other baselines.

fore, this baseline provides a comparison for our method to verify whether coherency is enough for sufficient video-story composition or if plot analysis (*i.e*. coherency as well as plot dynamics) is indeed superior.

• *Temporal*: This baseline is the original video-story composed by the actual user. The separate video clips are taken in temporal order and sequenced together to make a video-story. This baseline is used to verify whether simply ordering clips temporally is indeed the best way to compose a video-story.

4.1. Evaluation on Overall Video-Story Quality

This task involves showing subjects the video-stories each generated from the aforementioned baselines. For each set of video-stories, the contents are identical, but the sense of story differs according to each baseline. We ask the subjects to evaluate the sense of story present in the video-stories. The evaluation is done in the form of selecting the better story in a pairwise manner. The selected video-story in the pairwise comparison is given 1 point, thus the maximum score a video-story can get is 2 in a video set. We do not reveal which is which, and the presentation order of the video-stories is random.

Since our dataset consists of 23 videos, there are a total of 23 video sets. A total of 30 subjects (age range from 21-55 years old, and about half have no background in computer vision) participated in this task and were asked to evaluate at least 10 video sets. This gives *at least* 30 (subjects) $\times$ 10 (video sets) $\times$ 3 (pairwise comparisons) $=$ 900 tasks done by our subjects. We estimate each pairwise comparison to take 3 minutes to complete, resulting in *at least* 45 hours of user study. To the best of our knowledge, this is one of the most extensive user studies carried out in video composition evaluation. We would like to note that similar evaluation structures are performed in the data mining community (*e.g*. text analysis [23]) and the computer vision community [15, 20] implying that the evaluation we perform is not designed to conveniently return desired results.

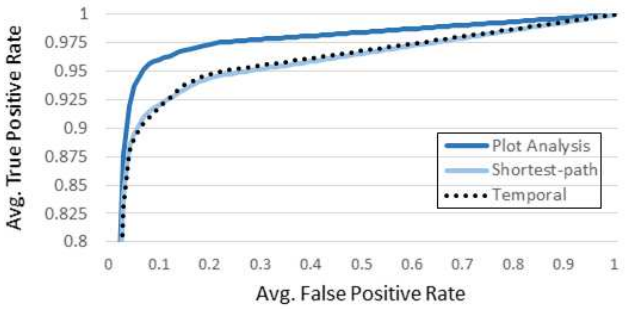
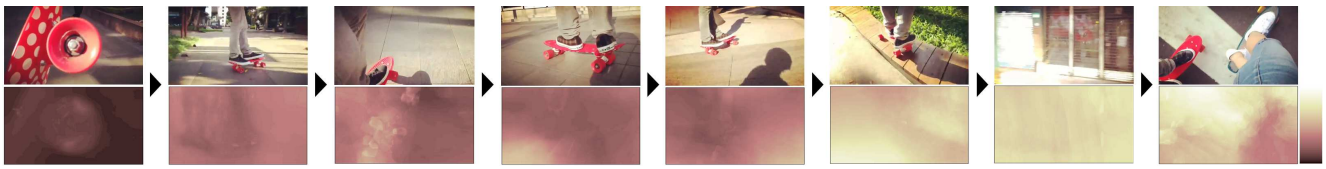


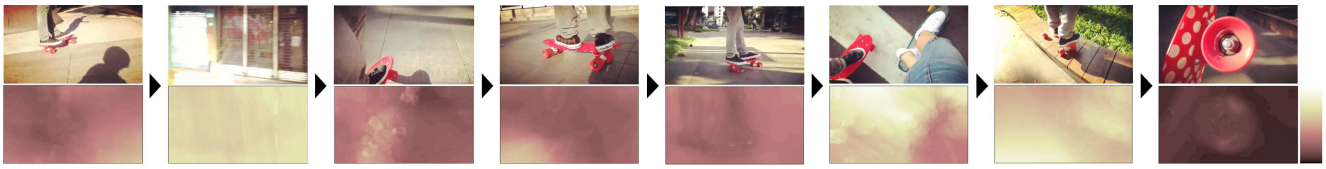
Figure 7: **Component-wise comparison results.** The curves represent average ROC curves for our method and baselines on story component evaluation.

Fig. 6 shows the results of the pairwise preference test of our method and baselines.

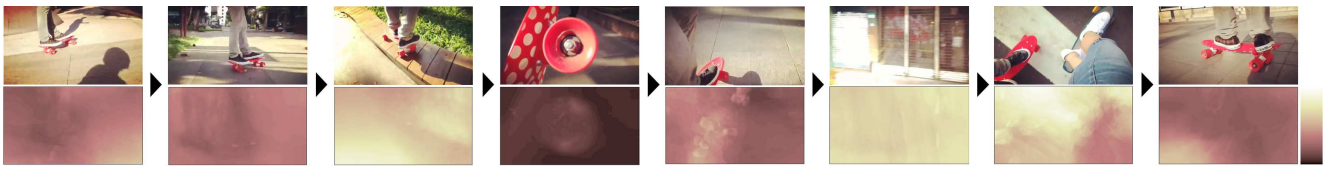
We find that our method composes video-stories with better sense of story when the contents have at least the slightest relevance between constituent video clips. For example, the video set WU involves walking in the city. Each video clip seems to be taken independently without any consideration of a story structure, but still contains the slightest relevance among them due to the same urban context. Our plot analysis approach manages to group similar clips into smooth story transitions and arrange the clip dynamics into a plot structure. The shortest-path method on the other hand, sufficiently links individual video clips with similar contents, however fails to compose a story plot. Lastly, the temporal baseline shows inconsistency in quality of video-stories. This suggests that composing video-stories in temporal order does not always guarantee high quality stories. In some cases involving uneventful or uncorrelated contents (*e.g*. video set FM: uncorrelated scenes and activities in a marketplace), our method shows less advantage over other baselines. Since the order of presenting uneventful or uncorrelated video clips does not greatly affect the story, it is reflected in our evaluation results.



(a) Plot Analysis video-story



(b) Shortest-path video-story



(c) Temporal video-story

Figure 8: **Video-story example result.** (a) Our Plot Analysis, (b) Shortest-path, and (c) Temporal baseline. The heat map shown below each clip snapshot is the accumulation of dense optical flow magnitudes representing the overall dynamics of each clip.

4.2. Evaluation via Story Components

In this task we evaluate video-stories by its constituent components. Since we have shown the overall evaluation of video-stories as a whole in Sec. 4.1, we now evaluate on a smaller scale: constituent story components in a video-story. The idea is to obtain ground truth video-story sequences and evaluate our method and baselines with them in a component-wise manner. However, it is difficult to obtain ground truth sequences since it is highly subjective. To avoid this difficulty, we perform an experiment as follows.

We show 4 workers all of the clips in each of the 23 video sets (total of 236 clips) and have them identify which two clips should be close together when composing a video-story for every video set. Instead of asking the workers to indicate a whole story sequence, asking to identify separate pairs of clips have several advantages. (1) It lessens the task burden on the workers. (2) Identifying pairs is less prone to subjectiveness and sets of pairs contain more concentrated information than a whole sequence. (3) It returns reliable ground truth information easier to statistically analyze.

We take the union of the output returned by the workers as ground truth. By thresholding the distance between clips in the sequence, we obtain average ROC curves for our method and baselines shown in Fig. 7. Once again, we would like to point out that similar experiments are performed on various works in the data mining [23] and computer vision community [15, 20] to emphasize that the evaluations are fair.

Qualitative result example. Fig. 8 shows example video-stories composed by our method and other baselines. Notice how our video-story starts with a low dynamic clip as the exposition. The clips that follow represent the gradual rising action phase leading to a climax. The video-story ends with a dynamic clip as the resolution. The contents of our result show coherency as well, grouping similar scenes together. On the contrary, the video-story composed via the shortest-path method lacks structure in plot dynamics. The original temporally ordered video-story not only lacks plot structure, but also content coherency of adjacent clips.

5. Conclusion

Our work deals with composing a story out of multiple short videos, namely a video-story. For this goal, we have defined and developed the plot analysis approach. Specifically, we have shown how to incorporate plot dynamics into a sequence of video clips, while also preserving content coherency. This was done by developing a novel Branch-and-Bound algorithm guaranteeing the globally optimal solution. Our extensive user study verifies the effectiveness of our approach. In the future, it would be interesting to take semantic information of the video clips into account for story composition. Now people can take video clips when they feel like it, without thinking of content order, and still expect a well-structured video-story in the end.

Acknowledgements. This work was supported by the Technology Innovation Program (No. 10048320), funded by the Korea government (MOTIE).

References

[1] A. Aner-Wolf and J. R. Kender. Video summaries and cross-referencing through mosaic-based representation. *Computer Vision and Image Understanding (CVIU)*, 2004. 2

[2] H. Averbuch-Elor and D. Cohen-Or. Ringit: Ring-ordering casual photos of a temporal event. *ACM TOG*, 2015. 3

[3] T. Basha, Y. Moses, and S. Avidan. Photo sequencing. In *ECCV*, 2012. 1, 2

[4] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 3

[5] C.-Y. Chen and K. Grauman. Clues from the beaten path: Location estimation with bursty sequences of tourist photos. In *IEEE CVPR*, 2011. 3

[6] D. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *Proceedings of ACM International Conference on Multimedia (MM)*, 1998. 2

[7] E. Dexter, P. Prez, and I. Laptev. Multi-view synchronization of human actions and dynamic scenes. In *BMVC*, 2009. 1, 2

[8] D. Feldman, G. Rossman, M. Volkov, and D. Rus. Coresets for k-segmentation of streaming data. In *NIPS*, 2014. 2

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, 1981. 3

[10] D. B. Goldman, B. Curless, and S. M. Seitz. Schematic storyboarding for video visualization and editing. In *ACM TOG(SIGGRAPH)*, 2006. 2

[11] M. Gygli, H. Grabner, and L. V. Gool. Video summarization by learning submodular mixtures of objectives. In *IEEE CVPR*, 2015. 1, 2

[12] R. Horst and H. Tuy. *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013. 4

[13] E. Kalogerakis, O. Vesselova, J. Hays, A. Efros, and A. Hertzmann. Image sequence geolocation with human travel priors. In *IEEE ICCV*, 2009. 3

[14] G. Kim and E. P. Xing. Jointly aligning and segmenting multiple web photo streams for the inference of collective photo storylines. In *IEEE CVPR*, 2013. 3

[15] G. Kim and E. P. Xing. Reconstructing storyline graphs for image recommendation from web community photos. In *IEEE CVPR*, 2014. 1, 3, 7, 8

[16] R. Laganire, R. Bacco, Hocevar, L. A., P. P., G., and B. E. Ionescu. Video summarization from spatio-temporal features. In *In Proc. of ACM TRECVid Video Summarization Workshop*, 2008. 2

[17] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *IEEE CVPR*, 2012. 1, 2

[18] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *IEEE CVPR*, 2008. 3

[19] D. Liu, G. Hua, and T. Chen. A hierarchical visual model for video object summarization. In *IEEE TPAMI*, 2010. 2

[20] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *IEEE CVPR*, 2013. 1, 2, 7, 8

[21] C. Ngo, Y. Ma, and H. Zhang. Automatic video summarization by graph modeling. In *IEEE ICCV*, 2003. 2

[22] P. Obrador, R. De Oliveira, and N. Oliver. Supporting personal photo storytelling for social albums. In *Proceedings of ACM International Conference on Multimedia (MM)*, 2010. 3

[23] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *ACM SIGKDD*, 2010. 7, 8

[24] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *IEEE CVPR*, 2008. 4

[25] N. Vasconcelos and A. Lippman. A spatiotemporal motion model for video summarization. In *IEEE CVPR*, 1998. 2

[26] M. Volkov, G. Rosman, D. Feldman, J. W. Fischer III, and D. Rus. Coresets for visual summarization with applications to loop closure. In *ICRA*, 2015. 2

[27] D. Wang, T. Li, and M. Ogihara. Generating pictorial storylines via minimum-weight connected dominating set approximation in multi-view graphs. In *AAAI*, 2012. 2

[28] W. Wolf. Key frame selection by motion analysis. In *IEEE ICASSP*, 1996. 2