The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

A Deep Ranking Model for Spatio-Temporal Highlight Detection from a 360° Video

**Youngjae Yu, Sangho Lee, Joonil Na,
Jaeyun Kang, Gunhee Kim**
Seoul National University
{yj.yu, sangho.lee, joonil}@vision.snu.ac.kr, {kjy13411}@gmail.com, gunhee@snu.ac.kr

Abstract

We address the problem of highlight detection from a 360° video by summarizing it both spatially and temporally. Given a long 360° video, we spatially select pleasantly-looking normal field-of-view (NFOV) segments from unlimited field of views (FOV) of the 360° video, and temporally summarize it into a concise and informative highlight as a selected subset of subshots. We propose a novel deep ranking model named as *Composition View Score* (CVS) model, which produces a spherical score map of composition per video segment, and determines which view is suitable for highlight via a sliding window kernel at inference. To evaluate the proposed framework, we perform experiments on the Pano2Vid benchmark dataset (Su, Jayaraman, and Grauman 2016) and our newly collected 360° video highlight dataset from YouTube and Vimeo. Through evaluation using both quantitative summarization metrics and user studies via Amazon Mechanical Turk, we demonstrate that our approach outperforms several state-of-the-art highlight detection methods. We also show that our model is 16 times faster at inference than AutoCam (Su, Jayaraman, and Grauman 2016), which is one of the first summarization algorithms of 360° videos.

Introduction

User-generated 360° videos are flooding online, with emergence of virtual reality and active support by major social network platforms such as Facebook and YouTube. Since 360° videos provide panoramic views of the entire scene, they free viewers not to get caught up in the intent of the videographer. However, without proper guidance, viewer experience can be severely handicapped, due to difficulty of understanding the entire content with limited human's field-of-view. Therefore, like normal user videos, the highlight detection is also highly necessitated for much of online 360° content, to quickly browse the overview of the content. One important difference of the 360° video highlight detection is that the video summarization should be achieved both *spatially* and *temporally*. The spatial summarization selects pleasantly-looking normal field-of-view (NFOV) segments from unlimited field of views (FOV) of 360° videos. Next, the temporal summarization generates a concise and informative highlight of a long video by selecting a subset of subshots.

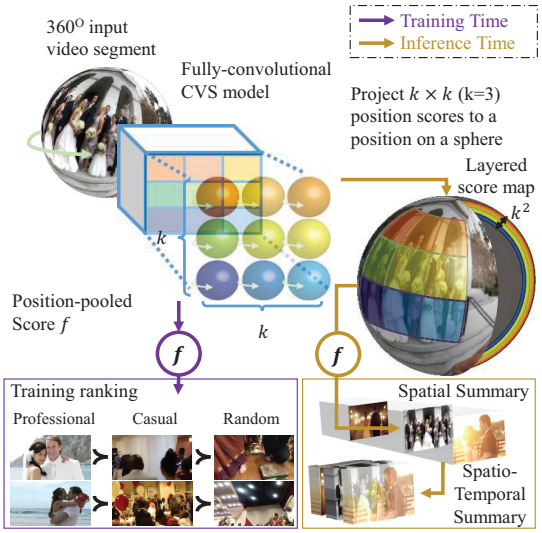

Figure 1: The intuition of the proposed *Composition View Score* (CVS) model for highlight detection from a 360° video. For each 360°video segment, fully convolutional CVS model generates a layered spherical score map to determine which view is suitable for highlight. It learns the fidelity of views from the professional videos of the same topic (*e.g.* wedding) as reference, and at inference, a sliding window kernel computes the composition scores of views.

We address the problem of highlight detection from a long 360° video by summarizing the video spatially and temporally. That is, we aim at selecting a pleasantly-looking NFOV within a 360° FOV, and at the same time producing a concise highlight. To this end, we propose a novel deep ranking neural network, named as *Composition View Score* (CVS) model. Given a 360° video, the CVS model produces a spherical score map of composition. Here we use the term *composition* to refer to a unified index to determine which view is suitable for highlight, considering all relevant properties such as the presence or absence of key objects, and the framing of objects. Based on the composition score map, we first perform *spatial* summarization by finding out a best NFOV subshot per 360° video segment, and then *temporal* summarization by selecting N top-ranked NFOV subshots

as a highlight for the entire 360° video.

Our approach has several noteworthy characteristics. First, to learn the notion of good visual composition for highlight, we collect the online NFOV videos that are taken and edited by professional videographers or general users as positive reference. Second, to quantify the difference between good and bad views, our deep ranking model learns the fidelity of views in the preference order of the professional NFOV video shots, normal users' NFOV shots, and randomly sampled NFOV shots from a 360° video. Third, to reduce the time complexity of the algorithm, we directly obtain a spherical score map for an entire 360° video segment. It can substantially reduce redundant score computations of highly overlapping adjacent NFOV candidates, compared to other state-of-the-art spatial summarization methods of 360° videos (*e.g*. (Su, Jayaraman, and Grauman 2016; Su and Grauman 2017)).

For evaluation, we use the existing Pano2Vid benchmark dataset (Su, Jayaraman, and Grauman 2016) for spatial summarization. We also collect a novel dataset of 360° videos from YouTube and Vimeo for spatio-temporal highlight detection. Our experiments show that our approach outperforms several state-of-the-art methods (Su, Jayaraman, and Grauman 2016; Gygli, Song, and Cao 2016; Yao, Mei, and Rui 2016) in terms of both quantitative summarization metrics (*e.g*. mean cosine similarity, mean overlap, and mAP) and user studies via Amazon Mechanical Turk.

The major contributions of this work are as follows.

(1) To the best of our knowledge, our work is the first attempt to summarize 360° videos both spatially and temporally for highlight detection. To this end, we develop a novel deep ranking model and collect a new dataset of 360°videos from YouTube and Vimeo.

(2) We propose *Composition View Score* (CVS) model, which produces a spherical composition score map of composition per video segment of 360°videos, and determines which view is suitable for highlight via a sliding window kernel at inference. Our framework is significantly faster at inference by reducing the redundant computation of adjacent NFOV candidates, which has been a serious problem in existing 360°video summarization models.

(3) For both Pano2Vid benchmark dataset (Su, Jayaraman, and Grauman 2016) and our newly collected 360°highlight video dataset, our approach outperforms several state-of-the-art methods in terms of both quantitative metrics and user evaluation via Amazon Mechanical Turk.

Related Work

360° video summarization. There has been very few studies for summarization of 360° user videos, except the *AutoCam* framework proposed by Su *et al*. (Su, Jayaraman, and Grauman 2016; Su and Grauman 2017) and the deep 360 pilot proposed by Hu *et al*. (Hu et al. 2017). Compared to the AutoCam and deep 360 pilot method, our model has the following novelties in three respects. First, in terms of problem definition, our work aims to summarize a 360-degree video both spatially and temporally, whereas the AutoCam and deep 360 pilot performs spatial summarization only. Second, in the algorithmic aspect, the AutoCam uses a logistic regression classifier and the deep 360 pilot exploits recurrent neural networks to determine where to look at each segment; on the other hand, our model employs a deep fully-convolutional network. Third, our CVS model outperforms the AutoCam in terms of performance and computation time, which will be more elaborated in our experiments. For instance, our model greatly reduces the number of rectilinear projections and subsequent convolutional operations per score computation from 198 to 12 at the inference time.

Temporal video summarization. Temporal video summarization provides a compressed abstraction of the original video while maintaining its key content (Truong and Venkatesh 2007). There are many criteria for determining key content, such as visual attention (Ejaz, Mehmood, and Baik 2013; Borji and Itti 2013), importance or interestingness (Gygli et al. 2014; 2013; Fu et al. 2014) and diversity/non-redundancy (Liu, Hua, and Chen 2010; Zhao and Xing 2014). Recently, many approaches use web-image priors for selecting informative content, assuming that images of the same topic often capture key events in high quality (Kim, Sigal, and Xing 2014; Khosla et al. 2013; Song et al. 2015). As another direction, several methods use supervised-learning dataset including human-annotated summaries (Gong et al. 2014; Gygli, Grabner, and Van Gool 2015) to learn balanced score functions between interestingness and diversity.

Video highlights. Rather than capturing a variety of events, video highlight models mainly focus on the importance or interestingness to summarize videos. To measure the interestingness, several methods use hand-crafted features from various low-level image features, such as aesthetic quality, camera following, and close-ups of faces (Gygli et al. 2014; Lee, Ghosh, and Grauman 2012). Another group of approaches exploit category-specific information to better define highlights (Sun, Farhadi, and Seitz 2014; Potapov et al. 2014). However, the scalability of domain-specific models is limited by the difficulty of collecting raw footages and their corresponding annotated videos. Some methods, meanwhile, attempt to deal with inherent noise in the web crawling dataset. Gygli *et al*. (Gygli, Song, and Cao 2016) train a model to learn the difference between a good and a bad frame. Sun *et al*. (Sun, Farhadi, and Seitz 2014) train a latent linear ranking model to generate a summary of a raw video using its corresponding edited video available online.

YouTube/Vimeo Dataset

We build a new dataset for the 360° highlight detection task from YouTube and Vimeo. Its key statistics are outlined in Table 1. The *professional* and *casual* categories indicate NFOV videos by professional videographers and normal users, respectively. We select two popular topics: *wedding* and *music video*, which involve a large volume of both 360° videos and NFOV videos that share the common storylines. Furthermore, the videos of these topics include multiple concurrent events, and thus are more interesting for spatio-temporal summarization.

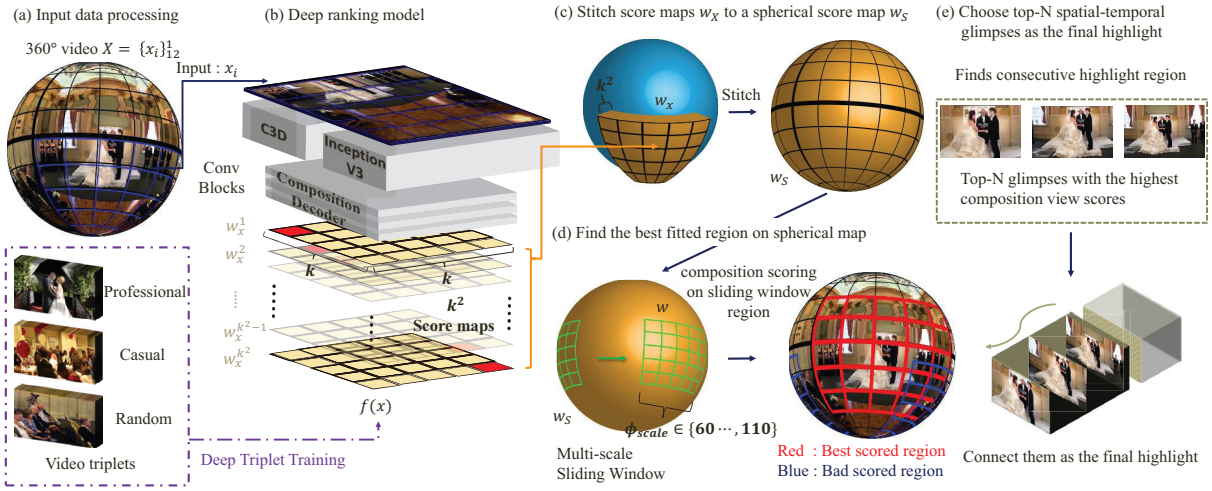

Figure 2: The overall framework for generating a spatio-temporal highlight using the Composition View Score (CVS) model.

Table 1: Statistics of our new 360° video highlight dataset.

Video	Topic	Type	# video	Total (hour)	mean (minute)
360 ° video	wedding	360°	62	54.8	53.1
	MV		53	17.2	19.5
NFOV video	wedding	professional	755	87.1	6.9
		casual	664	104.5	9.4
	MV	professional	333	3.3	4.4
		casual	654	47.5	0.6

Video Format

We sample every video into 5 frames per second (fps). We then segment every video into a sequence of subshots (or segments) by using the content-based subshot boundary detection of the `PySceneDetect`[1] library.

In a 360° video, there can be infinitely many view points at each time t, each of which can be determined from the latitude and longitude coordinates (θ, ϕ) in the spherical coordinates. Once we define a pair of (θ, ϕ), we can generate an NFOV subshot from the 360° video by a rectilinear projection with the viewpoint as a center. We set the size of each subshot to span a horizontal angle of 90° with a 4:3 aspect ratio. The default display format for a 360° video is usually obtained by equirectangular projection (*e.g.* 2D world maps from the spherical Earth). This projection maps the meridians to vertical straight lines of constant spacing, and circles of latitude to horizontal straight lines of constant spacing. The resulting $\mathbf{x}, \mathbf{y}$ coordinates in the equirectangular format are: $\mathbf{x} = (\phi - \phi_0)\cos\theta_1, \mathbf{y} = (\theta - \theta_1)$ where θ is the latitude, θ_1 is the standard parallels, ϕ is the longitude, and ϕ_0 is the central meridian of the map.

Following (Su, Jayaraman, and Grauman 2016), we use the term *spatio-temporal glimpses* (or simply *glimpses*) as a five-second NFOV video clip sampled from a 360° video, from the camera principal axis direction (θ, ϕ) at time t. Therefore, the spatial summarization reduces to selecting a best ST-glimpses for a sequence of 360° video segments.

[1]https://github.com/Breakthrough/PySceneDetect.

Approach

Figure 2 illustrates the overall framework to generate a spatio-temporal highlight for a long 360° video. The input of our framework is a long 360° video in a form of a sequence of video segments $V = \{v_1, \cdots, v_T\}$, where each segment v_t consists of spherical frames of 5 seconds. The output is a sequence of highlight NFOV video subshots $S = \{s_1, \cdots, s_N\}$, where each video subshot s_i consists of video frames of 5 seconds, and N is a user parameter to set the length of the final highlight video.

The Composition View Score Model

Model architecture. The *Composition View Score* (CVS) model is the key component of our framework. It computes a composition score for any NFOV spatio-temporal glimpse $\mathbf{x}$ sampled from a 360° video. As shown in Figure 2(b), the model is fully convolutional; it consists of feature extractors followed by a learnable deep convolutional decoder.

We represent a spatio-temporal glimpse $\mathbf{x}$ using two feature extractors, C3D (Tran et al. 2015) pretrained on the UCF-101 (Soomro, Zamir, and Shah 2012) for motion description, and Inception-v3 (Szegedy et al. 2016) pretrained on the ImageNet dataset (Russakovsky et al. 2015) for frame description. For C3D features, we obtain the conv4b feature map $\mathbf{x}_m \in \mathbb{R}^{14\times14\times512}$ using a 112×112 resized glimpse. For Inception-v3 features, we use the mixed_6c feature map $\mathbf{x}_f \in \mathbb{R}^{14\times14\times768}$ from a 260×260 sized glimpse. Finally, we stack $\mathbf{x}_m$ and $\mathbf{x}_f$ to be $\mathbf{x}_s \in \mathbb{R}^{14\times14\times1280}$, which is the input of our deep convolutional decoder.

The decoder consists of five convolutional layers, as shown in Table 2. It produces a set of k^2 score maps $\mathbf{w}_x = \{\mathbf{w}_x^1, \mathbf{w}_x^2 \cdots, \mathbf{w}_x^{k^2}\} \in \mathbb{R}^{k\times k\times k^2}$, whose $(k \times i + j)$ channel $\mathbf{w}_x^{k\times i+j}$ encodes the composition for the (i, j) position in a $k \times k$ grid. We divide $\mathbf{x}$ into $k \times k$ bins by a regular grid,

Table 2: The architecture of the composition decoder. We set k to 5, and use stride 1 and no zero-padding for all layers.

layer name	output size	kernel size / # of channels
conv1	12×12×512	3 × 3 / 512
conv2	10×10×512	3 × 3 / 512
conv3	8×8×1024	3 × 3 / 1024
conv4	5×5×2048	4 × 4 / 2048
pos_map	5×5×25	1 × 1 / 25

i.e. , $\mathbf{x} = \{x_{1,1}, \cdots, x_{i,j}, \cdots, x_{k,k}\}$, and aggregate all of position-wise scores using a Gaussian position-pooling:

$$f(\mathbf{x}) = \sum_{i,j} \sum_{l,m} \kappa(l-i)\kappa(m-j)\, \mathbf{w}_x^{c(k,l,m)}(i,j|\mathcal{M}), \quad (1)$$

$$\text{where } \kappa(u) = \frac{\exp(-u^2/2h^2)}{\sqrt{2\pi}h}, c(k,l,m) = k \times l + m,$$

for $i, j, l, m \in \{0, 1, \cdots, k-1\}$. Here, κ is a Gaussian kernel, h is the kernel bandwidth, and $\mathcal{M}$ denotes all the CVS model parameters. We set default k to 5.

It is partly inspired by the position-pooling of R-FCN (Li et al. 2016). However, our position-pooling softly encodes all score maps according to their relative position in the $k \times k$ grid using the Gaussian kernel, while R-FCN pools only over the (i, j)-th score map. This enhances the scale invariance of our model, and produces a better score by considering the surrounding context within the regular grid.

Training with video triplets. We pose the 360° highlight detection as a ranking problem, in which the CVS model selects the NFOV glimpse $\mathbf{x}$ with the highest composition view score $f(\mathbf{x})$ of Eq.(1), from many possible NFOV glimpses in a 360° frame. It is different from the AutoCam (Su, Jayaraman, and Grauman 2016) formulating as a binary classification problem using logistic regression, which is limited to rank a large number of candidates finely.

Our goal is to learn the CVS architecture that assigns a higher score to a view with good composition suitable for highlight. Rather than defining the goodness of composition with hand-crafted heuristics based on cinematography rules (Arev et al. 2014; Heck, Wallick, and Gleicher 2000; Gleicher, Heck, and Wallick 2002; Heck, Wallick, and Gleicher 2007), we leverage a data-driven approach; we collect professional NFOV videos and normal users' casual NFOV videos for the same topic, and use them as positive references of view selection. Since there are not many professional videos, we also exploit normal users' videos, although they are not as good as professional ones. During data collection, we observe that the quality gaps between professional videos and normal users' casual videos are significant. Thus, as highlight exemplars, we rank professional NFOV videos higher than the casual ones to correctly quantify the quality differences among the positive samples. Assuming that a randomly selected view is likely to be framed badly, we regard a randomly cropped glimpse from a 360° video as negative samples. As a result, we define the ranking constraints over the training dataset $\mathcal{D}$, which consists of video triplets of three different classes as shown in Figure 3 as follows:

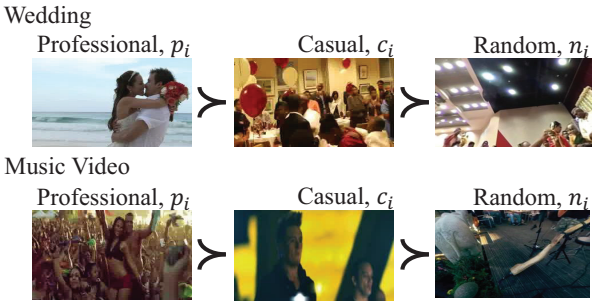

Figure 3: Examples of ranking order for professional, casual, and random NFOV glimpses about *Wedding* and *MV* topics.

$$f(p_i) \succ f(c_i) \succ f(n_i), \quad \forall\, (p_i, c_i, n_i) \in \mathcal{D}, \quad (2)$$

where p_i, c_i, n_i indicate a professional, casual, and negative sample (*i.e.* a video segment), respectively.

To impose the ranking constraints, we train the CVS model using the following loss. In particular, we assign different weights to different type pairs of video classes:

$$\begin{aligned}\mathcal{L}_i = \alpha \max(0, f(c_i) - f(p_i) + 1) \\ + (1-\alpha)\max(0, f(n_i) - f(c_i) + 1),\end{aligned} \quad (3)$$

where $\alpha \in [0, 1]$ is a hyperparameter; we set $\alpha = 0.3$ in our implementation. The final objective is the total loss over the dataset $\mathcal{D}$ combined with a l_2 regularization term:

$$\mathcal{L} = \sum_i \mathcal{L}_i + \lambda ||\mathcal{M}||_F^2. \quad (4)$$

where λ is a regularization hyperparameter.

Inference

The goal of inference is to select a best NFOV glimpse for a given 360° video segment. By repeating this process and connecting the selected glimpses, we can construct a highlight summary. For efficient inference, we first compute a 360°composition score map for each 360°video segment using the learned CVS model. We then perform sliding window search over the 360°score map to select a highlight view.

360°composition score maps. Note that there has been proposed no CNN architecture that takes a 360° video segment as input and produces a 360° score map as output. Therefore, we first propose an approximate procedure to obtain a spherical score map for a 360°video segment v_t at time t as follows. We divide v_t into 12 spatio-temporal glimpses $\mathbf{X}_t$ by discretizing viewpoints at longitudes $\phi \in \Phi = \{0°, 90°, 180°, 270°\}$ and at latitudes $\theta \in \Theta = \{0°, \pm 67.5°\}$. Since each glimpse spans 90° in the latitude axis with a 4:3 aspect ratio, $\mathbf{X}_t$ does not nearly overlap with one another while covering the entire spherical view.

Next, we transform every spatio-temporal glimpse $\mathbf{x} \in \mathbf{X}_t$ to an NFOV using rectilinear projection, and feed it into

the learned CVS model to compute its score map. However, one issue here is that since the CVS is convolutional, the information loss occurs near the boundaries of adjacent glimpses. Likewise padding in the CNN, we adopt the following simple trick; when transforming every glimpse $\mathbf{x}$ to an NFOV using rectilinear projection, we enlarge it by $1/k$ over the original size, in order to create a score map $\mathbf{w}_{x'} \in \mathbb{R}^{(k+2)\times(k+2)\times k^2}$. Later when we stitch these 12 score maps $\mathbf{w}_{X,t}$ to make a single 360° sphere score map $\mathbf{w}_{S,t}$, we cut out the padded boundary scores and use only $\mathbf{w}_x \in \mathbb{R}^{k\times k\times k^2}$ in the center. In this way, all the position scores are seamlessly wrapped in a sphere score map and without information loss around glimpse boundaries.

Sliding window search. After generating the spherical score map $\mathbf{w}_{S,t}$, we find the best fitted highlight position using a flexible sliding window kernel. As the sliding window scans on the score map, the composition score at the window location is calculated by cropping the score map $\mathbf{w}$ of the area from $\mathbf{w}_{S,t}$. With a fixed aspect ratio of 4:3, the sliding window can change its scale by varying its horizontal size, whose can be any $\theta_{\mathbf{scale}} \in [60°, 110°]$. In our experiments, we test three scales of sliding windows with the horizontal sizes of $(65.5°, 90°, 110°)$, and find the maximally scored position and scale.

Execution time comparison. One critical issue of existing models for spatial summarization is time complexity at the inference stage. For example, AutoCam (Su, Jayaraman, and Grauman 2016) computes capture-worthiness scores for a fixed number of overlapping 198 glimpses per 360° video segment, and chooses the best glimpse with the highest score. In this process, AutoCam performs severely overlapped convolutional operations and too many rectilinear projections. This exhaustive approach requires high computation time; for example, 3 hours per 1 minute 360° video as reported in (Su, Jayaraman, and Grauman 2016). To overcome this issue, our approach is to first precompute a non-overlapping 360° score map, and then use a flexible sliding window to find a highlight NFOV at inference. This approach greatly reduces the number of iterative rectilinear projections and CNN computations from 198 to 12, which subsequently curtails the processing time from 178 min to 11 min. The comparison on actual execution time will be reported in Table 6 in our experiments.

Spatio-Temporal Summarization

Since our framework is a ranking model, it is straightforward to extend our CVS model to spatio-temporal summarization. That is, rather than selecting a glimpse at every time step, we can rank all the glimpses over the entire video. We first build a candidate set of glimpses, each of which is selected from every video segment of 5 seconds by using the composition view score and the smooth-motion constraint (Su, Jayaraman, and Grauman 2016), which enforces that the latitude and longitude difference between consecutive glimpses must be less or equal than 30°: $|\theta_t - \theta_{t-1}|, |\phi_t - \phi_{t-1}| \leq 30°$. We then choose top-$N$ glimpses with the highest composition scores, and connect them as a final highlight. N is a user parameter for the highlight length. Since the professional videos that our CVS model uses as training reference are often highlights edited by professional videographers, high-ranked glimpses become excellent highlight candidates.

Implementation Details

We initialize all training parameters using the Xavier initialization (Glorot and Bengio 2010) and insert the batch normalization (Ioffe and Szegedy 2015) prior to all convolutional layers. We optimize the objective in Eq.(4) using vanilla stochastic gradient descent with a mini-batch size of 16. Experimentally, we use leaky ReLU (Maas, Hannun, and Ng 2013) as non-linear activation, and set our initial learning rate as 0.001 and divide it by 2 at every 8 epochs.

Experiments

We evaluate the performance of the proposed CVS model with two datasets. First, using the Pano2Vid dataset (Su, Jayaraman, and Grauman 2016), we show that the CVS model improves the spatial summarization performance compared to several baseline methods. Second, using our novel 360° video highlight dataset, we demonstrate that our framework achieves the state-of-the-art performance of generating spatio-temporal video highlights.

Evaluation Metrics

In the Pano2Vid dataset, human annotators provide multiple edited videos per 360° video, in which they label the center coordinates of the selected glimpses in the camera principal axes (*i.e*. latitude and longitude) at each video segment. Using the labeled coordinates as groundtruth, we compare the similarity between the human-made view trajectories and predicted trajectories. We use the metrics of *mean cosine similarity* and *mean overlap* as proposed in the Pano2Vid benchmark.

To quantify the highlight detection performance (*i.e*. spatio-temporal summarization) in our dataset, we compute the *mean Average Precision* (mAP) (Sun, Farhadi, and Seitz 2014). As groundtruths, we add three different highlight annotations to each of 25 randomly sampled 360°test videos per topic. Four human annotators watch full 360° videos and select top-N salient glimpses as video highlight subshots, with at least a distance of 5 seconds between choices.

Baselines

For performance comparison with our CVS model, we select (i) three simple baselines of spatial summarization used in (Su, Jayaraman, and Grauman 2016), (ii) AutoCam (Su, Jayaraman, and Grauman 2016) proposed for the original Pano2Vid task, and (iii) two state-of-the-art pairwise ranking model of deep neural networks for normal video summarization, RankNet (Gygli, Song, and Cao 2016) and TS-DCNN (Yao, Mei, and Rui 2016). All baselines share the same video feature representation by the C3D model (Tran et al. 2015) pretrained on the UCF-101 (Soomro, Zamir, and Shah 2012), except the TS-DCNN as described below.

We do not consider the deep 360 pilot method (Hu et al. 2017) as a baseline, because it is a supervised method that requires a center viewpoint per frame as a label for training.

Table 3: Experimental results of spatial summarization on the Pano2Vid (Su, Jayaraman, and Grauman 2016) dataset. Higher values represent better performance in both metrics.

Methods	Frame cosine sim	Frame overlap
Center	0.572	0.336
Eye-Level	0.575	0.392
Saliency	0.387	0.188
AutoCam (w/o stitching)	0.541	0.354
AutoCam-stitch	0.581	0.389
RankNet	0.562	0.398
TS-DCNN	0.578	0.441
CVS-C3D	0.656	0.554
CVS-Inception	0.642	0.545
CVS-Fusion	0.701	0.590
CVS-C3D-stitch	0.774	0.646
CVS-Inception-stitch	0.768	0.666
CVS-Fusion-stitch	**0.800**	**0.677**

Table 4: Comparison of trajectory similarity after applying the stitch algorithm in (Su, Jayaraman, and Grauman 2016).

	Cosine sim		Overlap	
	Trajectory	Frame	Trajectory	Frame
AutoCam-stitch	0.304	0.581	0.255	0.389
CVS-C3D-stitch	0.524	0.774	0.503	0.646
CVS-Fusion	**0.563**	**0.800**	**0.530**	**0.677**

Center, Eye-level (Su, Jayaraman, and Grauman 2016). These two baselines are stochastic models tested in the Pano2Vid benchmark, that select spatio-temporal glimpses at each segment with the following preferences. The Center method samples the glimpses at $\theta = 0, \phi = 0$, and then performs a random motion with a Gaussian distribution. The Eye-Level method samples the glimpses at $\theta = 0$ with a fixed set of $\phi \in \{0°, 20°, 40°, \cdots, 340°\}$.

Saliency (Su, Jayaraman, and Grauman 2016). This baseline uses the graph-based saliency score (Harel et al. 2006) to classify capture-worthy glimpses, and join the selected glimpses using the motion constraint that AutoCam uses.

AutoCam (Su, Jayaraman, and Grauman 2016). With C3D video features (Tran et al. 2015), AutoCam uses the logistic regression to classify which glimpses are worth to capture. It generates a summary video in the following two steps: (i) sampling the best scored glimpses at each video segment, and (ii) stitching the glimpses based on both the capture-worthiness scores and a smooth motion constraint. We denote the AutoCam using the both steps as AutoCam-stitch, while the one with the first step only as AutoCam.

RankNet (Gygli, Song, and Cao 2016). RankNet is a deep pairwise ranking model used in Video2GIF task (Gygli, Song, and Cao 2016). Unlike our fully convolutional model, RankNet predicts a single score from the C3D features through additional consecutive fully connected layers. We train the RankNet with the adaptive Huber loss as proposed in (Gygli, Song, and Cao 2016).

Table 5: Results of highlight detection on our 360° video highlight dataset. Higher mAPs indicate better performance.

Methods	Wedding	MV
Center	7.88	5.90
RankNet	11.98	11.65
TS-DCNN	13.23	12.28
CVS-C3D	16.32	12.15
CVS-Inception	16.13	12.38
CVS-Fusion (pairwise)	14.34	12.56
CVS-Fusion	**17.96**	**14.92**

Table 6: Comparison of computational costs between our CVS model and AutoCam (Su, Jayaraman, and Grauman 2016). The projected area is the total area of glimpses, expressed as multiples of the sphere area of a 360°frame.

	Processing time	# of ST-glimpses	Projected area
AutoCam	178 min	198	$\times$ 4.5479
CVS	11 min	12	$\times$ 1.96

Two-Stream DCNN (Yao, Mei, and Rui 2016)). The TS-DCNN is a recent pairwise ranking model of spatio-temporal deep CNNs for ego-centric video highlight detection. The spatial component using AlexNet (Krizhevsky, Sutskever, and Hinton 2012) represents scenes and objects in the video by frame appearance, while the temporal counterpart using C3D (Tran et al. 2015) conveys the motion dynamics. Overall TS-DCNN is similar to the RankNet, although they use both spatial and temporal representation. Thus, it can be good comparison to our CVS-Fusion model.

Evaluation of the Pano2Vid Task

For training of our CVS model, we use HumanCam positive samples p_i and randomly cropped negative samples n_i from panoramic videos in the Pano2Vid dataset. We use a simple max-margin loss $\mathcal{L}_i = \max(0, f(n_i) - f(p_i) + 1)$ instead of Eq.(3), because the training data are divided into two classes only (*i.e.* positive and negative). As an ablation study, we test our CVS model with three different configurations of video representation: (i) C3D only, (ii) Inception-v3 only, and (iii) both of C3D and Inception-v3. We also evaluate the variants of our method that use the smooth motion constraint, denoted by (*)-stitch.

Table 3 shows the experimental results in terms of frame cosine similarity and overlap region metrics, which are official measures of the Pano2Vid task. Our method CVS-Fusion-stitch outperforms all the baseline methods by a substantial margin in both metrics. The smooth motion constraint helps better summarization as the variants denoted by (*)-stitch outperforms those without the constraint. Even without the smooth motion constraint, the CVS models perform better than any baseline, regardless of which features the model uses.

Computational cost. Table 6 compares the computation costs between our CVS and AutoCam. Due to the redundant computation of overlapping glimpses as shown in Fig-

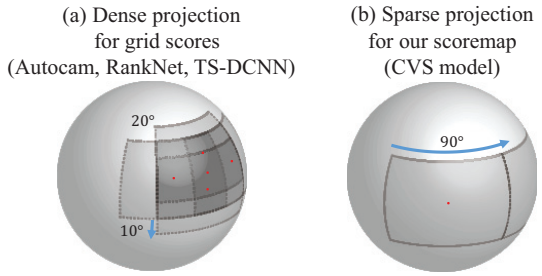

Figure 4: The visualization of (a) dense projection (198 glimpse) and (b) sparse projection (12 glimpse) by CVS score maps.

CVS-Fusion vs	Wedding	MV
Center	**68.0** % (117/150)	**57.3** % (86/150)
RankNet	**67.3** % (101/150)	**65.3** % (98/150)
TS-DCNN	**64.0** % (96/150)	**58.0** % (87/150)

Table 7: AMT results for 360° highlight detection. We show the percentages of turkers' votes for our method (CVS-Fusion) over baselines.

ure 4, AutoCam performs rectilinear projection on $\times 4.55$ of the actual area of the view sphere, while our framework projects $\times 1.96$ of the sphere area. Since the projection time is the bottleneck in practice, the computation time of our framework is significantly lower than that of AutoCam. We compare the average of processing time for one-minute 360° video: 178 min of AutoCam and 11 min of CVS. We experiment on a machine with one Intel Xeon processor E5 2695 v4 (18 core) and GTX Titan X Pascal GPU.

Evaluation of Highlight Detection

Quantitative results. Table 5 shows the results of highlight detection (*i.e*. spatio-temporal summarization) on our 360° video highlight dataset. We set $N = 15$ as the highlight length for all algorithms. We do not test the AutoCam method, because it has no mechanism to generate a temporal summary; instead, we compare with pairwise ranking models, such as RankNet and TS-DCNN. Our CVS-Fusion achieves the best performance in terms of mAP. As with observed performance drops, every key element of the CVS-Fusion model (*i.e*. two different features and the triplet ranking loss) is critical to the performance. Specifically, the triplet ranking loss significantly improves the performance of our model than the pairwise ranking loss, by providing a clearer guideline for a better composition. For example, in terms of mAP for Wedding and MV datasets, our CVS model learned by the triplet ranking loss shows performance improvements by (5.98, 3.27), (4.73, 2.64), (3.62, 2.36), compared to RankNet, TS-DCNN, and CVS-Fusion-pairwise learned by the pairwise loss (*i.e*. *professional* and *casual* as positives and *random* as negatives), respectively.

User studies via Amazon Mechanical Turk. We perform AMT tests to observe general users' preferences on the highlights detected by different algorithms. We randomly sample 20 test videos per topic from our 360° video highlight dataset. At test, we show an original video and two sequences of highlight subshots generated by our model and one baseline method in a random order. Turkers are asked to pick a better one without knowing which comes from which method. We collect answers from three different turkers per test example. We compare our best method (CVS-Fusion) with three baselines: Center, RankNet, and TS-DCNN.

Table 7 shows the results of AMT tests. It validates that general turkers prefer the output of our approach to those of baselines. Note that our method using the triplet ranking loss is more preferred than the models using the pairwise ranking loss, RankNet and TS-DCNN. These results coincide with quantitative results in Table 5.

Qualitative Results. Figure 5 shows qualitative examples of spatio-temporal glimpses that our CVS model chooses as highlights. We also depict the composition score map, computed by Gaussian position-pooling, over the input video of the equirectangular projection (ERP) format. We observe that high scores are often distributed to main characters, while relatively low scores are assigned to the regions with little saliency (*e.g*. sky in the background).

Figure 6 illustrates the position score maps for (a) a professional, (b) a casual, and (c) a randomly sampled glimpses. In the example (a), our framework correctly assigns high scores to actual highlights of a music video characterized by the group dance with a good framing. In the example (b), our model attaches flat scores to the frames that capture proper content of the input video, but show a mediocre framing, shifted to the right. This may be due to low scores (depicted in blue) in the left vacant parts of the example $\{w^0, w^5, w^{10}, w^{15}, w^{20}\}$. This tendency is an incentive to move the view selection to the right to increase the score. In the negative sample (c), all fitness scores are very low for the incorrect camera view.

Figure 7 shows an example of highlight detection on a 360° test video of our dataset. Our model successfully detects the main events labeled by human annotators, especially in top-6 scored glimpses. Compared to TS-DCNN (Yao, Mei, and Rui 2016), the CVS model can successfully assign higher scores to the frames containing central events. By using both spatial and temporal features, our model can discover dynamic movements of main characters as highlights like a guitarist head-banging in MV, or important moments such as a couple kissing in Wedding.

Conclusion

We addressed a problem of 360° video highlight detection via both spatial and temporal summarization. We proposed a novel deep ranking model named Composition View Score (CVS), which produces a spherical score map of composition per video segment to determine which view is suitable for highlight. Using the spherical position score maps, our model is much faster at inference than existing methods. In our experiments, we showed that the CVS model outperformed state-of-the-art methods not only for spatial summarization in the Pano2Vid dataset, but also for highlight detection task in our newly collected video highlight dataset.

Acknowledgments. We thank Jinyoung Sung for the

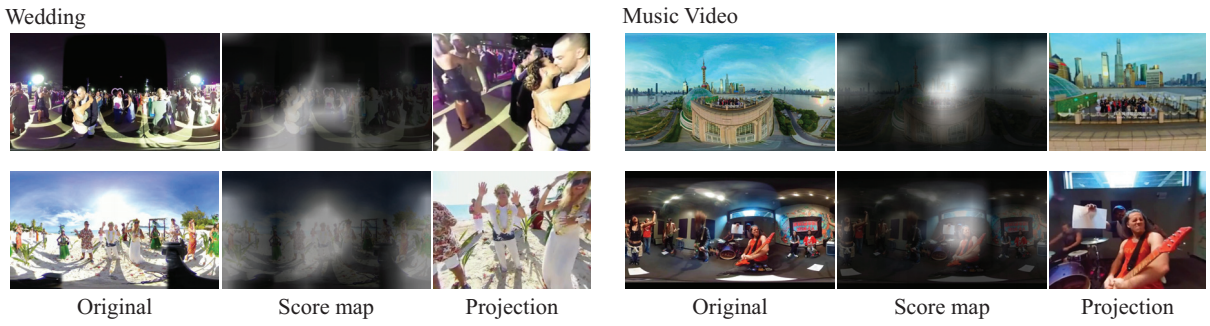

Figure 5: Examples of view selection. For a given spatio-temporal glimpse (left), we show the composition view score map (middle), and the projected NFOV with the highest score (right). The higher the view score is, the whiter it appears on the map.

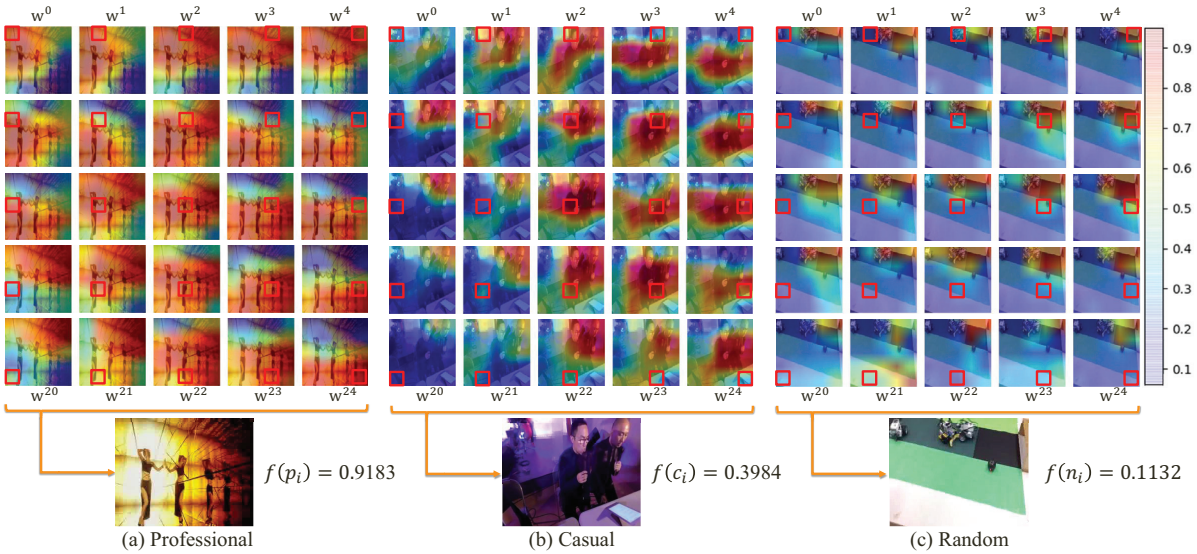

Figure 6: Examples of 25 position score maps $\mathbf{w}_x$ for (a) a professional, (b) a casual, and (c) a random glimpse in our Music Video (MV) highlight dataset. Our model successfully assigns higher scores to better views for the highlight.

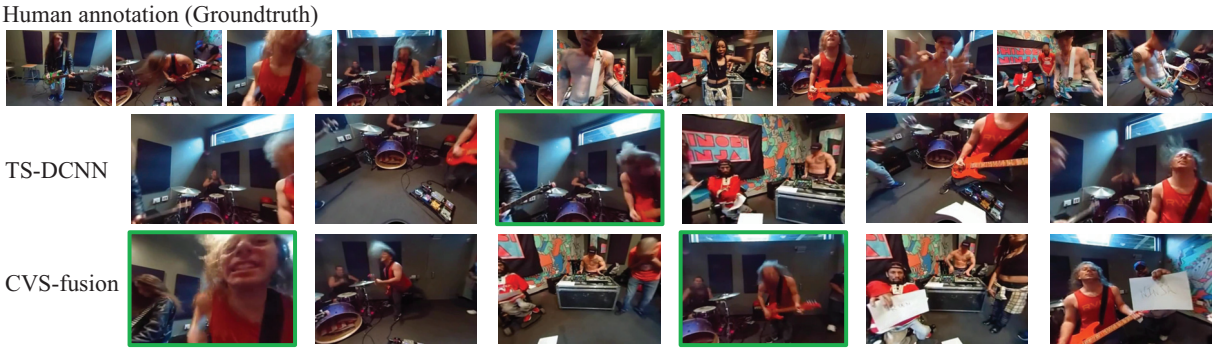

Figure 7: An example of highlight detection on a 360° test video. We show human-annotated groundtruth highlight (top), and compare top-6 scored glimpses by TS-DCNN (middle) and our CVS model (bottom). Green boxes indicate true-positives.

helpful discussion about the model. This work was supported by the Visual Display Business (RAK0117ZZ-21RF) of Samsung Electronics. Gunhee Kim is the corresponding author.

References

Arev, I.; Park, H. S.; Sheikh, Y.; Hodgins, J.; and Shamir, A. 2014. Automatic Editing of Footage from Multiple Social Cameras. In *SIGGRAPH*.

Borji, A., and Itti, L. 2013. State-of-the-Art in Visual Attention Modeling. *IEEE TPAMI* 35(1):185–207.

Ejaz, N.; Mehmood, I.; and Baik, S. W. 2013. Efficient Visual Attention Based Framework for Extracting Key Frames from Videos. *Signal Processing: Image Communication* 28(1):34–44.

Fu, Y.; Hospedales, T. M.; Xiang, T.; Gong, S.; and Yao, Y. 2014. Interestingness Prediction by Robust Learning to Rank. In *ECCV*.

Gleicher, M. L.; Heck, R. M.; and Wallick, M. N. 2002. A Framework for Virtual Videography. In *Smart Graphics*.

Glorot, X., and Bengio, Y. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *AISTATS*.

Gong, B.; Chao, W.-L.; Grauman, K.; and Sha, F. 2014. Diverse Sequential Subset Selection for Supervised Video Summarization. In *NIPS*.

Gygli, M.; Grabner, H.; Riemenschneider, H.; Nater, F.; and Van Gool, L. 2013. The Interestingness of Images. In *CVPR*.

Gygli, M.; Grabner, H.; Riemenschneider, H.; and Van Gool, L. 2014. Creating Summaries from User Videos. In *ECCV*.

Gygli, M.; Grabner, H.; and Van Gool, L. 2015. Video Summarization by Learning Submodular Mixtures of Objectives. In *CVPR*.

Gygli, M.; Song, Y.; and Cao, L. 2016. Video2GIF: Automatic Generation of Animated GIFs from Video. In *CVPR*.

Harel, J.; Koch, C.; Perona, P.; et al. 2006. Graph-Based Visual Saliency. In *NIPS*.

Heck, R.; Wallick, M.; and Gleicher, M. 2000. Towards Virtual Videography. In *ACM MM*.

Heck, R.; Wallick, M.; and Gleicher, M. 2007. Virtual Videography. *ACM TOMM* 3(1):4.

Hu, H.-N.; Lin, Y.-C.; Liu, M.-Y.; Cheng, H.-T.; Chang, Y.-J.; and Sun, M. 2017. Deep 360 pilot: Learning a deep agent for piloting through 360 sports video. In *CVPR*.

Ioffe, S., and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167*.

Khosla, A.; Hamid, R.; Lin, C.-J.; and Sundaresan, N. 2013. Large-Scale Video Summarization Using Web-image Priors. In *CVPR*.

Kim, G.; Sigal, L.; and Xing, E. P. 2014. Joint Summarization of Large-scale Collections of Web Images and Videos for Storyline Reconstruction. In *CVPR*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In *NIPS*.

Lee, Y. J.; Ghosh, J.; and Grauman, K. 2012. Discovering Important People and Objects for Egocentric Video Summarization. In *CVPR*.

Li, Y.; He, K.; Sun, J.; et al. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *NIPS*.

Liu, D.; Hua, G.; and Chen, T. 2010. A Hierarchical Visual Model for Video Object Summarization. *IEEE TPAMI* 32(12):2178–2190.

Maas, A. L.; Hannun, A. Y.; and Ng, A. Y. 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *ICML*.

Potapov, D.; Douze, M.; Harchaoui, Z.; and Schmid, C. 2014. Category-specific Video Summarization. In *ECCV*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* 115(3):211–252.

Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. Tvsum: Summarizing Web Videos Using Titles. In *CVPR*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv:1212.0402*.

Su, Y.-C., and Grauman, K. 2017. Making 360° Video Watchable in 2D: Learning Videography for Click Free Viewing. In *CVPR*.

Su, Y.-C.; Jayaraman, D.; and Grauman, K. 2016. Pano2Vid: Automatic Cinematography for Watching 360° Videos. In *ACCV*.

Sun, M.; Farhadi, A.; and Seitz, S. 2014. Ranking Domain-specific Highlights by Analyzing Edited Videos. In *ECCV*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*.

Truong, B. T., and Venkatesh, S. 2007. Video Abstraction: A Systematic Review and Classification. *ACM TOMM* 3(1):3.

Yao, T.; Mei, T.; and Rui, Y. 2016. Highlight Detection with Pairwise Deep Ranking for First-Person Video Summarization. In *CVPR*.

Zhao, B., and Xing, E. P. 2014. Quasi Real-Time Summarization for Consumer Videos. In *CVPR*.