2017 IEEE Conference on Computer Vision and Pattern Recognition

Deep 360 Pilot: Learning a Deep Agent for Piloting through 360° Sports Videos

Hou-Ning Hu[1]*, Yen-Chen Lin[1]*, Ming-Yu Liu[2], Hsien-Tzu Cheng[1], Yung-Ju Chang[3], Min Sun[1]
[1]National Tsing Hua University [2]NVIDIA research [3]National Chiao Tung University
{eborboihuc, hsientzucheng}@gapp.nthu.edu.tw armuro@cs.nctu.edu.tw,
{yenchenlin1994, sean.mingyu.liu}@gmail.com sunmin@ee.nthu.edu.tw

Abstract

Watching a 360° sports video requires a viewer to continuously select a viewing angle, either through a sequence of mouse clicks or head movements. To relieve the viewer from this "360 piloting" task, we propose "deep 360 pilot" – a deep learning-based agent for piloting through 360° sports videos automatically. At each frame, the agent observes a panoramic image and has the knowledge of previously selected viewing angles. The task of the agent is to shift the current viewing angle (i.e. action) to the next preferred one (i.e., goal). We propose to directly learn an online policy of the agent from data. Specifically, we leverage a state-of-the-art object detector to propose a few candidate objects of interest (yellow boxes in Fig. 1). Then, a recurrent neural network is used to select the main object (green dash boxes in Fig. 1). Given the main object and previously selected viewing angles, our method regresses a shift in viewing angle to move to the next one. We use the policy gradient technique to jointly train our pipeline, by minimizing: (1) a regression loss measuring the distance between the selected and ground truth viewing angles, (2) a smoothness loss encouraging smooth transition in viewing angle, and (3) maximizing an expected reward of focusing on a foreground object. To evaluate our method, we built a new 360-Sports video dataset consisting of five sports domains. We trained domain-specific agents and achieved the best performance on viewing angle selection accuracy and users' preference compared to [53] and other baselines.

1. Introduction

360° video gives a viewer immersive experiences through displaying full surroundings of a camera in a spherical canvas, which differentiates itself from traditional multimedia. As consumer- and production-grade 360° cameras become readily available, 360° videos are captured every minute. Moreover, the promotion of 360° videos by social media giants including YouTube and Facebook further boosts their fast adoption. It is expected that 360° videos will become a major video format in the near future. Studying how to display 360° videos to a human viewer, who has a limited field of visual attention, emerges as an increasingly important problem.

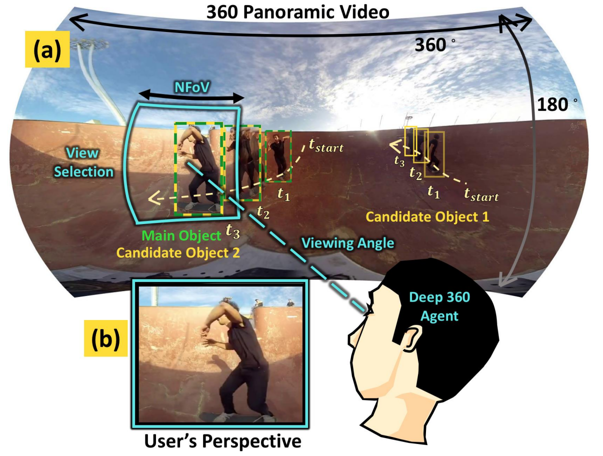

Figure 1. Panel (a) overlaps three panoramic frames sampled from a 360° skateboarding video with two skateboarders. One skateboarder is more active than the other in this example. For each frame, the proposed "deep 360 pilot" selects a view – a viewing angle, where a Natural Field of View (NFoV) (cyan box) is centered at. It first extracts candidate objects (yellow boxes), and then selects a main object (green dash boxes) in order to determine a view (just like a human agent). Panel (b) shows the NFoV from a viewer's perspective.

Hand Manipulation (HM) and Virtual Reality (VR) are two main ways for displaying 360° videos on a device with a Natural Field of View (NFoV) (typically a 60° to 110° FoV as shown in Fig. 1). In HM, a viewer navigates a 360° video via a sequence of mouse clicks; whereas, in VR, a viewer uses embedded motion sensors in a VR headset for navigation. Note that both HM and VR require a viewer to select a viewing angle at each frame, while the FoV is defined by the device. For sports videos, such a selection mechanism could be cumbersome because "foreground objects" of interest change their locations continuously. In fact, a recent study [32] showed that both HM

*indicates equal contribution

1063-6919/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPR.2017.153

and VR can cause a viewer to feel discomfort. Just imagine how hard it is to follow an X-game skateboarder in a 360° video. Hence, a mechanism to automatically navigate a 360° video in a way that captures most of the interest events for a viewer would be beneficial.

Conceptually, a 360°-video viewer is a human agent: at each frame, the agent observes a panoramic image (i.e., the observed state) and steers the viewing angle (i.e. the action) to cover the next preferred viewing angle (i.e., the goal). We refer to this process as *360 piloting*. Based on this analogy and, more importantly, to relieve the viewer from constantly steering the viewing angle while watching 360° videos, we argue for an intelligent agent that can automatically piloting through 360° sports videos.

Using an automatic mechanism for displaying video contents is not a new idea. For example, video summarization–condensing a long video into a short summary video [58]–has been used in reviewing hourly long surveillance videos. However, while a video summarization algorithm makes binary decisions on whether to select a frame or not, an agent for 360 piloting needs to operate on a spatial space to steer the viewing angle to consider events of interest in a 360° video. On the other hand, in virtual cinematography, most camera manipulation tasks are performed within relatively simpler virtual environments [8, 22, 12, 40] and there is no need to deal with viewers' perception difficulty because 3-D positions and poses of all entities are known. However, a practical agent for 360 piloting needs to directly work with raw 360° videos. For displaying 360° videos, Su et al. [53] proposed firstly detecting candidate events of interest in the entire video, and then applying dynamic programming to link detected events. However, as this method requires observing an entire video, it is non-suited for video streaming applications such as foveated rendering [45]. We argue that being able to make a selection based on the current and previous frames (like a human agent does) is critical for 360 piloting. Finally, both [53] and recent virtual cinematography works [7, 6] aim for smooth viewing angle transition. Such transition should also be enforced for 360° piloting.

We propose "deep 360 pilot"—a deep learning-based agent that navigates a 360° sports video in a way that smoothly captures interesting moments in the video. Our "deep 360 pilot" agent not only follows foreground objects of interest but also steers the viewing angle smoothly to increase viewers' comfort. We propose the following online pipeline to learn an online policy from human agents to model how a human agent takes actions in watching sports videos. First, because in sports videos foreground objects are those of viewers' interest, we leverage a state-of-the-art object detector [50] to identify candidate objects of interest. Then, a Recurrent Neural Network (RNN) is used to select the main object among candidate objects. Given the main object and previously selected viewing angles, our method predicts how to steer the viewing angle to the preferred one by learning a regressor. In addition, our pipeline is jointly trained with the following functions: (1) a regression loss measuring the distance between the selected and ground truth viewing angles, (2) a smoothness loss to encourage smooth transition in viewing angle, and (3) an expected reward of focusing on a foreground object. We used the policy gradient technique [62] to train the pipeline since it involves making an intermediate discrete decision corresponding to selecting the main object. To evaluate our method, we collected a new 360° sports video dataset consisting of five domains and trained an agent for each domain (referred to as 360-Sports). These domain-specific agents achieve the best performance in regression accuracy and transition smoothness in viewing angle.

Our main contributions are as follows: (1) We develop the first human-like online agent for automatically navigating 360° videos for viewers. The online processing nature suits the agent for streaming videos and predicting views for foveated VR rendering. (2) We propose a jointly trainable pipeline for learning the agent. Since the main object selection objective is non-differentiable, we employ a policy gradient technique to optimize the pipeline. (3) Our agent considers both viewing angle selection accuracy and transition smoothness. (4) We build the first 360° sports videos dataset to train and evaluate our "deep 360 pilot" agent.

2. Related Work

We review related works in video summarization, saliency detection, and virtual cinematography.

2.1. Video Summarization

We selectively review several most relevant video summarization works from a large body of literature [58].

Important frame sampling. [33, 19, 43, 27] proposed to sample a few important frames as the summary of a video. [47, 54, 63] focused on sampling domain-specific highlights. [67, 54, 18] proposed weakly-supervised methods to select important frames. Recently, a few deep learning-based methods [66, 66, 65] have shown impressive performance. [48, 49, 55] focused on extracting highlights and generating synopses which showed several spatially non-overlapping actions from different times of a video. Several methods [17, 25] involving user interaction have also been proposed in the graphics and the HCI communities.

Ego-centric video summarization. In ego-centric videos, cues from hands and objects become easier to extract compare to third-person videos. [30] proposed video summarization based on the interestingness and diverseness of objects and faces. [36] further proposed tracking objects and measuring the influence of individual frames. [28] proposed a novel approach to speed-up ego-centric videos while removing unpleasant camera movements.

In contrary to most video summarization methods which concern whether to select a frame or not, a method for 360 piloting concerns which viewing angle to select for each panoramic frame in a 360° video.

2.2. Saliency Detection

Many methods have been proposed to detect salient regions typically measured by human gaze. [35, 21, 1, 46, 59, 64, 46] focused on detecting salient regions on images. Recently, [34, 24, 9, 44, 5, 61, 60, 57] leveraged deep learning and achieved significant performance gain. For videos, [10, 20, 37, 52, 41, 29] relied on low-level appearance and motion cues as inputs. In addition, [26, 16, 51, 39, 13] included information about face, people, objects, or other contexts. Note that saliency detection methods do not select views directly, but output a saliency score map. Our method is also different to visual attention methods for object detection [38, 3, 42] in that it considers view transition smoothness as selecting views, which is crucial for video watching experience.

Ranking foreground objects of interest. Since regions of interest in sports videos are typically foreground objects, [55] proposed to use an object detector [4] to extract candidate objects of interest, then rank the saliency of these candidate objects. For 360 piloting, we propose a similar baseline which first detects objects using RCNN [50], then select the viewing angle focusing on the most salient object according to a saliency detector [64].

2.3. Virtual Cinematography

Finally, existing virtual cinematography works focused on camera manipulation in simple virtual environments/video games [8, 22, 12, 40] and did not deal with the perception difficulty problem. [14, 56, 7, 6] relaxed the assumption and controlled virtual cameras within restricted static wide field-of-view video of a classroom, video conference, or basketball court, where objects of interest could be easily extracted. In contrast, our method handles raw 360° sports videos downloaded from YouTube[1] in five domains (e.g., basketball, parkour, etc.). Recently, Su et al. [53] also proposed handling raw 360° videos download from YouTube. They referred to this problem as Pano2Vid – automatic cinematography in 360° videos – and proposed an offline method. In contrast, we propose an online human-like agent acting based on both present and previous observations. We argue that for handling streaming videos and other human-in-the-loop applications (e.g., foveated rendering[45]) a human-like online agent is necessary in order to provide more effective video-watching support.

3. Our Approach

We first define the 360 piloting problem in details (Sec. 3.1). Then, we introduce our deep 360 pilot approach (Sec. 3.2–Sec. 3.6). Finally, we describe the training procedure of our model (Sec. 3.7).

3.1. Definitions

We formulate the 360 piloting task as the following online viewing angle selection task.

Observation. At time t, the agent observes a new frame v_t, which is the t-th frame of the 360° video. The sequence of frames that the agent has observed up to this time is referred to as $\mathbf{V}_t = \{v_1, ..., v_t\}$.

Goal. The goal of the agent is to select a viewing angle l_t at time t so that the sequence of viewing angles $\mathbf{L}_t = \{l_1, ..., l_t\}$ smoothly capture events of interest in the 360° video. Note that $l_t = (\theta_t, \phi_t)$ is a point on the 360° viewing sphere, parameterized by the azimuth angle $\theta_t \in [0°, 360°]$ and elevation angle $\phi_t \in [-90°, 90°]$

Action. In order to achieve the goal, the agent takes the action of steering the viewing angle by Δ_t at time t. Given the previous viewing angle l_{t-1} and current action Δ_t, the current viewing angle l_t is computed as follows,

$$l_t = \Delta_t + l_{t-1}. \tag{1}$$

Online policy. We assume that the agent takes an action Δ_t at frame t according to an online policy π as follows,

$$\Delta_t = \pi(\mathbf{V}_t, \mathbf{L}_{t-1}), \tag{2}$$

where the online policy depends on both the current and previous observation $\mathbf{V}_t$ and previous viewing angles $\mathbf{L}_{t-1}$. This implies that the previous viewing angles affect the current action similar to what a human viewer acts when viewing a 360° sports video. Hence, the main task of 360 piloting is about learning the online policy from data.

In the following, we discuss various design choices of our proposed deep 360 pilot where the online policy in Eq. 2 is modeled as a deep neural network.

3.2. Observing in Object Level

Instead of extracting information from the whole 360° panoramic frame at each time instance, we propose to focus on foreground objects (Fig. 2(b)) for two reasons. Firstly, in sports videos, foreground objects are typically the targets to be followed. Moreover, the relative size of foreground objects is small compared to the whole panoramic image. If processing is done at the frame level, information of object fine details would be diluted. Working with object-based observations help our method extract subtle appearance and motion cues to take an action. We define object-level observation $\mathbf{V}_t^O$ as,

$$\mathbf{V}_t^O = \{v_1^O, ..., v_t^O\} \tag{3}$$

$$\text{where } v_t^O \text{ is given by } v_t^O = \text{con}_V(O_t, P_t, M_t). \tag{4}$$

$$\text{and } O_t = \text{con}_H(\{o_t^i\}), P_t = \text{con}_H(\{p_t^i\}), \tag{5}$$

$$M_t = \text{con}_H(\{m_t^i\}). \tag{6}$$

Note that $\text{con}_H()$ and $\text{con}_V()$ denote horizontal and vertical concatenation of vectors, respectively. The vector

[1] https://www.youtube.com/

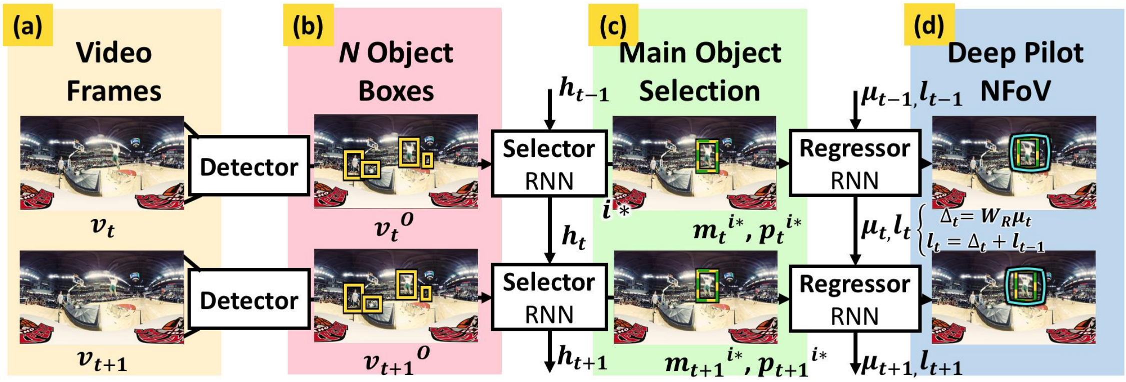

Figure 2. Visualization of our deep 360 pilot model. Panel (a) shows two consecutive frames. Panel (b) shows the top-N confident object bounding boxes (yellow boxes) given by the detector. Panel (c) shows the selected main object (green dash box) given by the RNN-based Selector. Panel (d) shows the final NFoV centered at the viewing angle (cyan box) predicted by the RNN-based regressor.

$o_t^i \in R^d$ denotes the i-th object appearance feature, the vector $p_t^i \in R^2$ denotes the i-th object location (the same parameterization as l_t) on the view sphere at frame t and the vector $m_t^i \in R^k$ denotes the i-th object motion feature. If there are N objects, the dimension of O_t, P_t, and M_t are $d \times N$, $2 \times N$, and $k \times N$, respectively. Then the dimension of concatenated object feature v_t^O is $(d + 2 + k) \times N$. Note that our agent is invariant to the order of objects. More explanation is shown in technical report [23]. In the online policy (Eq. 2), we replace $\mathbf{V}_t$ with $\mathbf{V}_t^O$ which consists of object appearance, motion, and location.

3.3. Focusing on the Main Object

We know that as watching a sports video a human agent gazes at the main object of interest. Assuming the location of the main object of interest, p_t^{i*}, is known, a naive policy for 360 piloting would be a policy that closely follows the main object and the action taken at each time instance is

$$\hat{\Delta}_t = p_t^{i*} - l_{t-1}. \tag{7}$$

Since a machine agent does not know which object is the main one, we propose the following method to estimate the index $i*$ of the main object. We treat this task as a classification task and predict the probability $S_t(i)$ that the object i is the main object as follows,

$$S_t = \pi(\mathbf{V}_t^O) \in [0, 1]^N, \tag{8}$$

where $\sum_i S_t(i) = 1$. Given S_t,

$$i* = \arg\max_i S_t(i). \tag{9}$$

In this case, the agent's task becomes *discretely* selecting one main object (Fig. 2(c)). We will need to handle this discrete selecting while introducing policy gradient [62].

We note that the size of $\mathbf{V}_t^O$ grows with the number of observed frames, which increase the computation cost. We propose to aggregate object previous information via a Recurrent Neural Network (RNN).

3.4. Aggregating Object Information

Our online policy is implemented as a selector network as shown in Fig. 2(b)). It consists of a RNN followed by a softmax layer. The RNN aggregates information from the current frame and past state to update its current state, while the softmax layer maps the current state of the RNN into a probability distribution via W_s.

$$\begin{aligned} h_t &= RNN_S(v_t^O, h_{t-1}), \\ S_t &= \text{softmax}(W_s h_t) \end{aligned} \tag{10}$$

3.5. Learning Smooth Transition

So far our model dose not take care of the smooth transition in viewing angle. Hence, we propose to refine the action from the selector network, $\hat{\Delta}_t = p_t^{i*} - l_{t-1}$, with the motion feature, m_t^{i*} (Fig. 2(d)), as follows,

$$\begin{aligned} \mu_t &= RNN_R(\text{con}_V(m_t^{i*}, \hat{\Delta}_t), \mu_{t-1}). \\ \Delta_t &= W_R \, \mu_t, \end{aligned} \tag{11}$$

Here, we concatenate the motion feature and the proposed action from the selection network to form the input at time t to the regressor network RNN_R. The RNN_R then updates its state from μ_{t-1} to μ_t. While RNN_S focuses on main object selection, RNN_R focuses on action refinement. The state of RNN_R is then mapped to the final steering action vector Δ_t via W_R. The resulting viewing angle is then given by $l_t = \Delta_t + l_{t-1}$.

3.6. Our Final Model

As shown in Fig. 2, our model has three main blocks. The *detector* block extracts object-based observation v_t^o as described in Eq. 4. The *selector* block selects the main object index $i*$ following Eq. 10 and Eq. 9. The *regressor* block regresses the viewing angle l_t given main object location p_t^{i*} and motion m_t^{i*} following Eq. 7, Eq. 11, and Eq. 1.

3.7. Training

We will first discuss the training of the regressor network and then discuss the training of the selector network.

Finally, we show how to train these two networks jointly. Note that we use the viewing angle l_t^{gt} at each time instance provided by human annotators as the ground truth.

Regressor network. We train the regressor network by minimizing the Euclidean distance between the predicted viewing angle and the ground truth viewing angle at each time instance. For enforcing a smooth steering, we also regularize the training with a smoothness term, which penalizes a large rate of change in viewing angles between two consecutive frames. Let $v_t = l_t - l_{t-1}$ be the viewing angle velocities at time t. The loss function is then given by

$$\sum_{t=1}^{T} \|l_t - l_t^{gt}\|_2 + \lambda \|v_t - v_{t-1}\|_2 \tag{12}$$

where λ is a hyper-parameter balancing the two terms and T is the number of frames in a video.

Selector network. As the ground truth annotation for each frame is provided as the human viewing angle, the main object $i*^{gt}$ to be focused on at each frame is unknown. Therefore, we resort to the approximated policy gradient technique proposed in [62] to train the selector network.

Let $l(i)$ be a viewing angle associated with object i that is computed by the regressor network. We define the reward of selecting object i (steering the viewing angle to $l(i)$) to be $r(l(i))$ where the reward function r is defined based on the overlapping ratio between the NFOV centering at l_t^{i*} and the NFOV centering at $l(i)$. The details of the reward function design is shown in technical report [23]. We then train the selector network by maximizing the expected reward

$$\mathcal{E}(\theta) = E_{i \sim S(i,\theta)}[r(l(i))], \tag{13}$$

using the policy gradient

$$\begin{aligned} \nabla_\theta \mathcal{E}(\theta) &= \nabla_\theta E_{i \sim S(i,\theta)}[r(l(i))] & (14) \\ &= E_{i \sim S(i,\theta)}[r(l(i)) \nabla_\theta \log S(i,\theta)], & (15) \end{aligned}$$

where θ is the model parameter of the selector network.

We further approximate $\nabla_\theta \mathcal{E}(\theta)$ using sampling as,

$$\nabla_\theta \mathcal{E}(\theta) \simeq \frac{1}{Q} \sum_{q=1}^{Q} r(l(i_q)) \nabla_\theta \log S(i_q, \theta), \tag{16}$$

where q is the index of sampled main object, Q is the number of samples, and the approximated gradient is referred to as the policy gradient.

Joint training. Since the location of the object selected by the selector network is fed into the regressor network for computing the final viewing angle and the reward function for training the selector network is based on the regressor network's output, the two networks are trained jointly. Specifically, we joint update the trainable parameters in both networks similar to [42], which hybrids the gradients from the reinforcement signal and supervised signal.

	SB	Park.	BMX	Dance	BB	Total
#Video	56	92	53	56	85	342
#Frame	59K	27K	16K	56K	22K	180K

Table 1. Statistics of our Sports-360 dataset. SB, Park., BMX, and BB stand for skateboarding, parkour, bicycle motocross, and basketball, respectively. K stands for thousand.

4. Sports-360 Dataset

We have collected a new dataset called Sports-360[2], which consists of 342 360° videos downloaded from YouTube in five sports domains: basketball, parkour, BMX, skateboarding, and dance (Fig. 3). Domains were selected according to the following criteria: (i) high availability of such videos on YouTube, (ii) the retrieved videos contain dynamic activities rather than static scenes, and (iii) containing a clear human-identifiable object of interest in most of the video frames. The third criterion is required to obtain unambiguous ground truth viewing angle in our videos.

In each domain, we downloaded the top 200 videos sorted by relevance. Then, we removed videos that were either in poor resolution or stitching quality. Next, we sampled and extracted a continuous video clip from each video where a scene transition is absent (many 360° videos are edited and contain scene transitions). Finally, we recruited 5 human annotators, and 3 were asked to "label the most salient object for VR viewers" in each frame in a set of video segments containing human-identifiable objects. Each video segment was annotated by 1 annotator in the panorama view (see Fig. 4a). The annotation results were verified and corrected by the other 2 annotators.

We show example panoramic frames and NFoV images centered at ground truth viewing angles in Fig. 3. Our dataset includes both video segments and their annotated ground truth viewing angles. The statistics of our dataset (i.e., number of videos and frames per domain) is shown in Table. 1. We split them by assigning 80% of the videos for training, and 20% for testing.

5. Experiments

We evaluate deep 360 pilot on the Sports-360 dataset. We show that our model outperforms baselines by a large margin both quantitatively and qualitatively. In addition, we also conduct a user preference study. In the following, we first define the evaluation metric. Then, we describe the implementation details and baseline methods. Finally, we report the quantitative, qualitative, and human study results.

5.1. Evaluation Metrics.

To quantify our results, we report both Mean Overlap (MO) and Mean Velocity Difference (MVD). **MO** measures how much the NFoV centered at the predicted viewing angle overlaps (i.e., Intersection over Union (IoU)) with that of the ground truth one at each frame. A prediction is pre-

[2] Our dataset and code can be downloaded from https://aliensunmin.github.io/project/360video

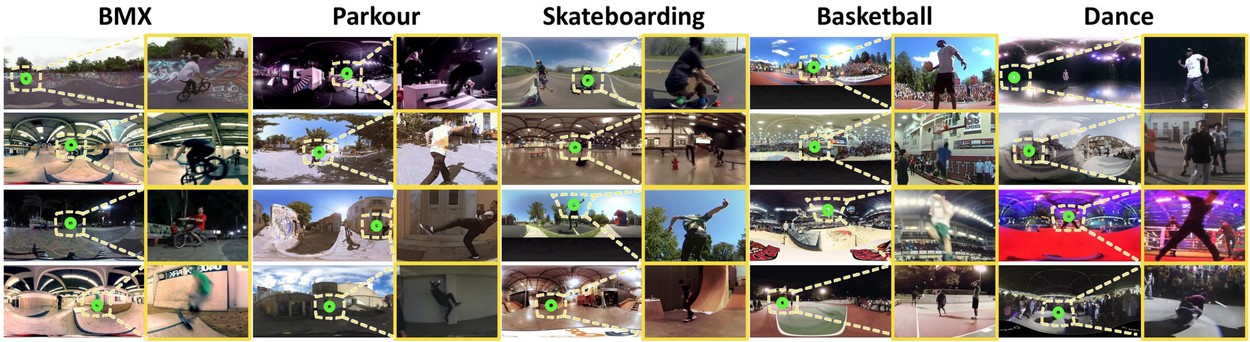

Figure 3. Our Sports-360 dataset. We show example pairs of panoramic and NFoV images in five domains: BMX, parkour, skateboarding, basketball, and dance. In each example, a panoramic frame with ground truth viewing angle (green circle) is shown on the left. The zoomed in NFoV (yellow box) centered at the ground truth viewing angle is shown on the right. The NFoV illustrates the viewers perspective.

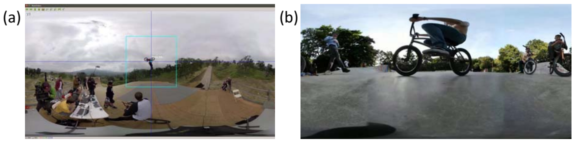

Figure 4. (a) Annotators mark main objects in 360° videos with a mouse. The blue cross helps annotators locate cursor position, and the cyan box indicates NFoV. Main reason to label in panorama is shown in the technical report [23]. (b) Example of bmx bike.

cise if the IoU is 1. **MVD** evaluates the curvature of the predicted viewing angle trajectories. It is given by the norm of the difference of viewing angle velocities in two consecutive frames given by $\|v_t - v_{t-1}\|_2$. Note that, in average, the trajectory is smoother if its MVD at each frame is low.

5.2. Implementation Details

Detector. We use the Faster R-CNN [50] model pre-trained on 2014 COCO detection dataset [31] to generate about 400 bounding boxes for each frame. Then, we apply the tracking-by-detection algorithm [2] to increase the recall of the object detection. Finally, we apply detection-by-tracking [2] to select reliable detection linked into long tracklets. Given these tracklets, we select top $N = 16$ reliable boxes per frame as our object-based observation. Detailed sensitivity experiment results can be found in the technical report [23]. We found it is beneficial to use general object detectors. In the sport video domains studied, non-human objects such as skateboard, basketball, or bmx bike (Fig. 4b) provides strong cues for main objects. For each object, we extract mean pooling of the Conv5 feature $\in R^{512}$ in the network of R-CNN as the appearance feature o_t^i, and Histogram of Optical Flow [11] of boxes with 12 orientation bins as the motion representation $m_t^i \in R^{12}$.

Selector. The hidden representation of RNN_S is set to 256 and it processes input $v_t^O \in R^{(d+2+k)\times N}$ in sequences of 50 frames.

Regressor. The hidden representation of RNN_R is set to 8. We set λ to 10.

Learning. We optimize our model using stochastic gradients with batch size = 10 and maximum epochs = 400. The learning rate is decayed by a factor of 0.9 from the initial learning rate of $1e^{-5}$ every 50 epochs.

5.3. Methods to be Compared

We compared the proposed deep 360 pilot with a number of methods, including the state-of-the-art method AUTOCAM [53], two baseline methods combining saliency detection with the object detector [50], and a variant of deep 360 pilot without a regressor.

AUTOCAM [53]: Since their model is not publicly available, we use the ground truth viewing angles to generate NFoV videos from our dataset. These NFoV videos are used to discriminatively assign interestingness on a set of pre-defined viewing angles at each frame in a testing video. Then, AUTOCAM uses dynamic programming to select optimal sequence of viewing angles. Finally, the sequence of viewing angles is smoothed in a post-processing step. Note that since AUTOCAM proposes multiple paths for each video, we use ground truth in testing data to select top ranked sequence of viewing angles as the system's final output. This creates a strong "offline" baseline.

RCNN+Motion: We first extract detected boxes' optical flow. Then, we use a simple motion saliency proposed by [15], median flow, and HoF [11] as features to train a gradient boosting classifier to select the box that is most likely to contain the main object. Finally, we use center of the box selected sequentially by the classifier as predictions.

RCNN+BMS: We leverage the saliency detector proposed by Zhang et al. [64] to detect the most salient region in a frame. With the knowledge of saliency map, we can extract the max saliency scores in each box as a score. Then we emit the most salient box center sequentially as our optimal viewing angle trajectories.

Ours w/o Regressor: We test the performance of our deep 360 pilot without regressor. It emits box center of the selected main object as prediction at each frame.

5.4. Benchmark Experiments

We compare our method with our variant and baseline methods in Table. 2. In the following, we summarize our findings. AUTOCAM achieves the best MO among three baseline methods in 4 out of 5 domains. Our method significantly outperforms AUTOCAM in MO (at most 22%

Method	Skateboarding		Parkour		BMX		Dance		Basketball	
	MO	MVD	MO	MVD	MO	MVD	MO	MVD	MO	MVD
Ours **w/o Regressor.**	**0.71**	6.03	**0.74**	4.72	**0.71**	10.73	**0.79**	4.32	**0.67**	8.62
Ours	0.68	**3.06**	**0.74**	**4.41**	0.69	**8.36**	0.76	**2.45**	0.66	**6.50**
AUTOCAM [53]	*0.56*	*0.25*	*0.56*	*0.71*	*0.47*	*0.55*	*0.73*	*0.15*	0.51	*0.66*
RCNN+BMS.	0.25	37.5	0.2	30.8	0.22	32.4	0.24	40.5	0.2	25.27
RCNN+Motion.	0.56	34.8	0.47	26.2	0.42	25.2	0.72	31.4	*0.54*	25.2

Table 2. Benchmark experiment results. Except "AUTOCAM" achieving a very low MVD through an offline process, "Ours w/o Regressor" achieves the best MO (the higher the better) and "Ours" achieves the best MVD (the lower the better). Most importantly, "Ours" strikes a good balance between MO and MVD.

	Skateboarding	Parkour	BMX	Dance	Basketball
Comparison	win / loss	win / loss	win / loss	win / loss	win / loss
vs AUTOCAM	34 / 2	35 / 1	31 / 5	34 / 2	36 / 0
vs Ours **w/o Regressor**	28 / 8	29 / 7	26 / 10	31 / 5	34 / 2
vs human	15 / 21	10 / 26	7 / 29	14 / 22	7 / 29

Table 3. User study results. For all of the five sports domains, our method is significantly preferred over AUTOCAM and Our **w/o Regressor**. Also, it is comparable to expert human in skateboarding and dance.

gain in BMX and at least 3% gain in Dance). Although AUTOCAM achieves significantly lower MVD compared to our method, we argue that its lower MO will critically affect its viewing quality, since the majority of our videos typically contain fast moving main objects. Since we do not know how to trade MVD over MO and vice versa, we resort to a user study to compare AUTOCAM with our method. Our comparison with ours w/o regressor is the other way around. Both methods achieve similar MO while our method achieves lower MVD. These results show that with regressor, the agent steers the viewing angle more smoothly. Fig. 5 shows the trajectories of viewing angles predicted by both methods for a testing video. From this visual inspection, we verify that the smoothness term results in a less jittering trajectory.

5.5. User Study

We conduct a user study mainly to compare our method with AUTOCAM and ours w/o regressor. The following is the experimental setting. For each domain, we sample two videos where all three methods achieve MO larger than 0.6 and MVD smaller than 10. This is to prevent users from comparing bad quality results, which makes identifying a better method difficult. For each video, we ask 18 users to compare two methods. In each comparison, we show videos piloted by two methods with random order via a 360° video player. The number of times that our method wins or loses is shown in Table 3. Based on a two-tailed binomial test, our method is statistically superior to AUTOCAM with p-value<0.001. This implies that users consider MO more important in this comparison. Base on the same test, our method is statistically superior to our w/o regressor with p-value<0.05. This implies that when MOs are similarly good, a small advantage of MVD results in a strong preference for our method. We also conduct a comparison between our method with the human labeled ground truth viewing angles. Base on the same test, our method is indistinguishable to human on skateboarding with p-value<0.405 and on dance with p-value<0.242.

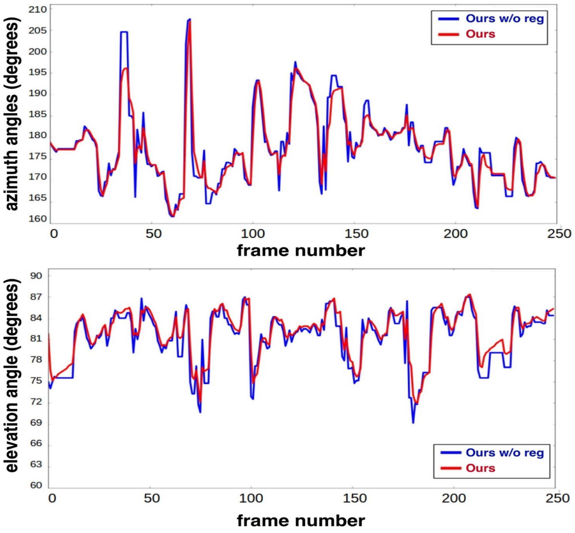

Figure 5. Comparison of Ours and Ours w/o Regressor. These two methods yields similar MO, while Ours predicts smoother viewing angles in both principal axes.

5.6. Typical Examples

We compare our "deep 360 pilot" with AUTOCAM in Fig. 6. In the first example, both our method and AUTOCAM work well since the main object in dancing does not move globally. Hence, the ground truth viewing angle is not constantly moving. In the next three examples, our method produces smooth trajectories while maintaining adequate view selection without any post-processing step. In contrast, AUTOCAM struggles on capturing fast-moving objects since Su et al. [53] constrains every glimpses' length up to 5 seconds. Moreover, the pre-defined 198 views force many actions to be cut in half by the rendered NFoV. We further compare our method on a subset of publicly avail-

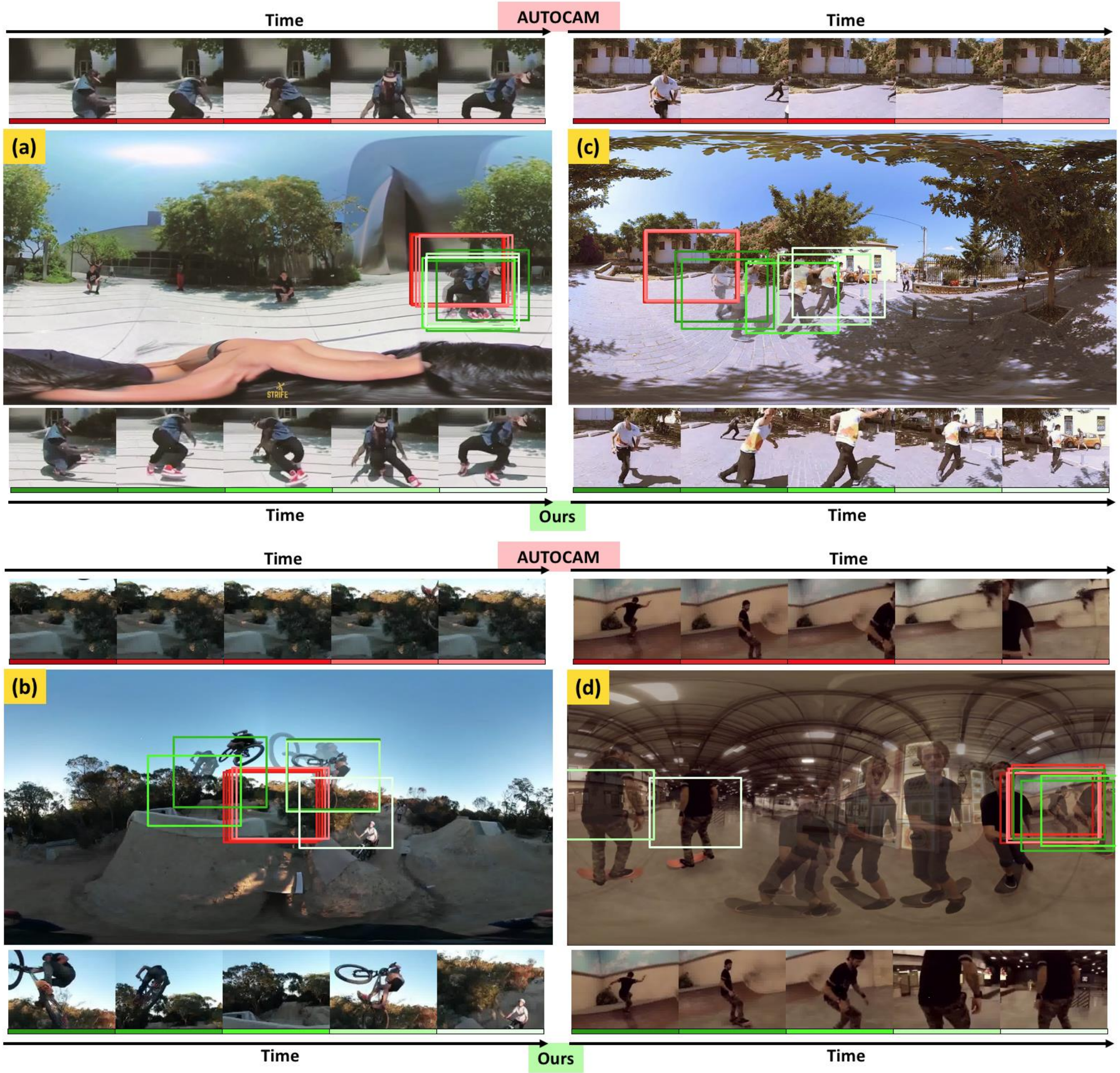

Figure 6. Typical examples from four domains: (a) dance, (b) BMX, (c) parkour, and (d) skateboarding. For each example, the middle panel shows a panoramic image with motaged foreground objects. The top and bottom panels show zoomed in NFoV centered at viewing angles generated by AUTOCAM and our method, respectively. We further overlaid the NFov from AUTOCAM and our method in red and green boxes, respectively, in the middle panoramic image.

able videos from dataset of [53]. We get a 140% performance boost in quantitative metrics of [53]. Similar comparisons to other baseline methods and more results on dataset of [53] are shown in the technical report [23].

6. Conclusion

We developed the first online agent for automatic 360° video piloting. The agent was trained and evaluated using a newly composed Sport-360 dataset. We aimed at developing a domain-specific agent for the domain where the definition of a most salient object is clear (e.g., skateboarder). The experiment results showed that our agent achieved much better performance as compared to the baseline methods including [53]. However, our algorithm would suffer in the domains where our assumption is violated (containing equally salient objects or no objects at all). In the future, we would like to reduce the amount of ground truth annotation needed for training our agent.

Acknowledgements

We thank NOVATEK, MEDIATEK and NVIDIA for their support.

References

[1] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Ssstrunk. Frequency-tuned salient region detection. In CVPR, 2009. 3

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In CVPR, 2008. 6

[3] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. In ICLR'15. 2015. 3

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In ICCV, 2009. 3

[5] N. D. B. Bruce, C. Catton, and S. Janjic. A deeper look at saliency: Feature contrast, semantics, and beyond. In CVPR, June 2016. 3

[6] J. Chen and P. Carr. Mimicking human camera operators. In WACV, pages 215–222. IEEE, 2015. 2, 3

[7] J. Chen, H. M. Le, P. Carr, Y. Yue, and J. J. Little. Learning online smooth predictors for realtime camera planning using recurrent decision trees. In CVPR, 2016. 2, 3

[8] D. B. Christianson, S. E. Anderson, L. wei He, D. Salesin, D. S. Weld, and M. F. Cohen. Declarative camera control for automatic cinematography. In AAAI, 1996. 2, 3

[9] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. A deep multi-level network for saliency prediction. In ICPR, 2016. 3

[10] X. Cui, Q. Liu, and D. Metaxas. Temporal spectral residual: fast motion saliency detection. In ACM Multimedia, 2009. 3

[11] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In ECCV, 2006. 6

[12] D. K. Elson and M. O. Riedl. A Lightweight Intelligent Virtual Cinematography System for Machinima Production. In AIIDE, 2007. 2, 3

[13] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In ECCV, 2012. 3

[14] J. Foote and D. Kimber. Flycam: Practical panoramic video and automatic camera control. In ICME, 2000. 3

[15] G. Gkioxari and J. Malik. Finding action tubes. 2015. 6

[16] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. TPAMI, 34(10):1915–1926, 2012. 3

[17] D. Goldman, B. Curless, D. Salesin, and S. Seitz. Schematic storyboarding for video visualization and editing. In SIGGRAPH, 2006. 2

[18] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In NIPS, 2014. 2

[19] Y. Gong and X. Liu. Video summarization using singular value decomposition. In CVPR, 2000. 2

[20] C. Guo, Q. Ma, and L. Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In CVPR, 2008. 3

[21] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In NIPS, 2006. 3

[22] L.-w. He, M. F. Cohen, and D. H. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In ACM CGI, 1996. 2, 3

[23] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Technical report of deep 360 pilot. 2017. https://aliensunmin.github.io/project/360video. 4, 5, 6, 8

[24] S. Jetley, N. Murray, and E. Vig. End-to-end saliency mapping via probability distribution prediction. In CVPR, 2016. 3

[25] N. Joshi, S. Metha, S. Drucker, E. Stollnitz, H. Hoppe, M. Uyttendaele, and M. F. Cohen. Cliplets: Juxtaposing still and dynamic imagery. In UIST, 2012. 2

[26] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In ICCV, 2009. 3

[27] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In CVPR, 2013. 2

[28] J. Kopf, M. F. Cohen, and R. Szeliski. First-person hyper-lapse videos. ACM Trans. Graph., 33(4), July 2014. 2

[29] T. Lee, M. Hwangbo, T. Alan, O. Tickoo, and R. Iyer. Low-complexity hog for efficient video saliency. In ICIP, pages 3749–3752. IEEE, 2015. 3

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

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014. 6

[32] Y.-C. Lin, Y.-J. Chang, H.-N. Hu, H.-T. Cheng, C.-W. Huang, and M. Sun. Tell me where to look: Investigating ways for assisting focus in 360 video. In CHI, 2017. 1

[33] D. Liu, G. Hua, and T. Chen. A hierarchical visual model for video object summarization. TPAMI, 32(12):2178–2190, 2010. 2

[34] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In CVPR, 2016. 3

[35] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. TPAMI, 33(2):353–367, 2011. 3

[36] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In CVPR, 2013. 2

[37] V. Mahadevan and N. Vasconcelos. Spatiotemporal saliency in dynamic scenes. TPAMI, 32(1):171–177, 2010. 3

[38] S. Mathe, A. Pirinen, and C. Sminchisescu. Reinforcement Learning for Visual Object Detection. In CVPR, June 2016. 3

[39] S. Mathe and C. Sminchisescu. Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. TPAMI, 37, 2015. 3

[40] P. Mindek, L. Čmolík, I. Viola, E. Gröller, and S. Bruckner. Automatized summarization of multiplayer games. In ACM CCG, 2015. 2, 3

[41] P. Mital, T. Smith, R. Hill, and J. Henderson. Clustering of gaze during dynamic scene viewing is predicted by motion. Cognitive Computation, 3(1):5–24, 2011. 3

[42] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu. Recurrent models of visual attention. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, NIPS. 2014. 3, 5

[43] C. Ngo, Y. Ma, and H. Zhan. Video summarization and scene detection by graph modeling. In CSVT, 2005. 2

[44] J. Pan, K. McGuinness, E. Sayrol, N. O'Connor, and X. Giro-i Nieto. Shallow and deep convolutional networks for saliency prediction. In CVPR, 2016. 3

[45] A. Patney, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and D. Luebke. Perceptually-based foveated virtual reality. In SIGGRAPH, pages 17:1–17:2, 2016. 2, 3

[46] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In CVPR, 2012. 3

[47] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In ECCV, 2014. 2

[48] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. In ICCV, 2007. 2

[49] A. Rav-Acha, Y. Pritch, and S. Peleg. Making a long video short. In CVPR, 2006. 2

[50] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015. 2, 3, 6

[51] D. Rudoy, D. B. Goldman, E. Shechtman, and L. Zelnik-Manor. Learning video saliency from human gaze using candidate selection. In CVPR, pages 1147–1154, 2013. 3

[52] H. Seo and P. Milanfar. Static and space-time visual saliency detection by self-resemblance. Journal of Vision, 2009. 3

[53] Y.-C. Su, D. Jayaraman, and K. Grauman. Pano2vid: Automatic cinematography for watching 360 videos. In ACCV, 2016. 1, 2, 3, 6, 7, 8

[54] M. Sun, A. Farhadi, and S. Seitz. Ranking domain-specific highlights by analyzing edited videos. In ECCV, 2014. 2

[55] M. Sun, A. Farhadi, B. Taskar, and S. Seitz. Summarizing unconstrained videos using salient montages. In ECCV, 2014. 2, 3

[56] X. Sun, J. Foote, D. Kimber, and B. Manjunath. Region of interest extraction and virtual camera control based on panoramic video capturing. TMM, 7(5):981–990, 2005. 3

[57] Y. Tang and X. Wu. Saliency detection via combining region-level and pixel-level predictions with cnns. In ECCV, 2016. 3

[58] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. TMCCA, 3(1), Feb. 2007. 2

[59] J. Wang, A. Borji, C.-C. J. Kuo, and L. Itti. Learning a combined model of visual saliency for fixation prediction. TIP, 25(4):1566–1579, 2016. 3

[60] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In ECCV, 2016. 3

[61] Q. Wang, W. Zheng, and R. Piramuthu. Grab: Visual saliency via novel graph model and background priors. In CVPR, June 2016. 3

[62] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3):229–256, 1992. 2, 4, 5

[63] T. Yao, T. Mei, and Y. Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In CVPR, 2016. 2

[64] J. Zhang and S. Sclaroff. Exploiting surroundedness for saliency detection: a boolean map approach. TPAMI, 38(5):889–902, 2016. 3, 6

[65] K. Zhang, W. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In ECCV, 2016. 2

[66] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In CVPR, June 2016. 2

[67] B. Zhao and E. Xing. Quasi real-time summarization for consumer videos. In CVPR, June 2014. 2