Supplementary Material: 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

1. Reward Function

Let $l_t(i)$ be the viewing angle associated with object i that is computed by the regressor network, and l_t^{gt} be the ground truth viewing angle at frame t. We define the reward function r as follow,

$$r(l_t(i), l_t^{gt}) = \begin{cases} 1 - \frac{\|l_t(i) - l_t^{gt}\|_2}{\eta}, & \text{if } \|l_t(i) - l_t^{gt}\|_2 <= \eta \\ -1, & \text{otherwise} \end{cases} \tag{1}$$

where η equals the distance from the center of a viewing angle to the corner of its corresponding NFoV, i.e., $\sqrt{32.75^2 + 24.56^2} = 40.9$ if we define NFOV as spanning a horizontal angle of 65.5° with a $4:3$ aspect ratio. When $l_t == l_t^{gt}$, the reward is 1, which is the maximum reward. When $\|l_t(i) - l_t^{gt}\|_2 > \eta$, i.e., the predicted viewing angle is not covered by ground truth viewing angle's NFoV, the reward is -1.

2. Sensitivity Analysis

In order to see if the number of candidate objects affects the performance of our system significantly, we conduct a sensitivity experiment on the number of candidate objects N. We evaluate our deep 360 pilot with $N = \{8, 16, 32\}$ in each domain. Experiment results in Table. 2 suggests that deep 360 pilot is not sensitive to the number of candidate objects N. Also, for all the three values of N, deep 360 pilot still outperforms other baselines.

3. Typical Examples

We compare our "deep 360 pilot" method with several baselines: AUTOCAM, Our without Regressor in Fig. 1, and RCNN + Motion, RCNN + BMS in Fig. 2. Each method will generate a series of NFoV predictions on 3 videos: (a) a BMX video with a fast moving foreground object, (b) a skateboarding video with 2 main skateboarders, and (c) video of basketball players with relatively small movement. In Fig. 1 we can see that AUTOCAM almost stay at the same position, which is hard to follow the quick moving object in video (a) and (b), but our methods successfully capture the main foreground objects in each frame. In Fig. 2, RCNN with either BMS saliency method or optical flow based method fail to stay focus on the main foreground objects in video (a) and (b), but our method is significantly outperforming these baselines. In the example of video (c) in both Figures, all predictions of different methods seem to be similar because of the smaller movement. However, our method still captures the main basketball player more precisely, where the player running from right to left to finish the slam dunk.

4. Human Evaluation Videos

We upload a demo video which contains 3 examples selected from videos used for our human evaluation study, each of them comes from different domains. In each example, we demonstrate 4 methods, human label, AUTOCAM, ours, ours w/o Regressor, concurrently to make a clear comparison. This video can be found from https://aliensunmin.github.io/project/360video

5. Reviewers' Comments

We address critical comments from the reviewers below.

5.1. Why not annotate in the Natural Field of View (NFoV)?

We found that annotating directly in the NFoV is very inefficient because the annotators have to watch a video many times with different NFoVs. This involves a number of back-and-forth operations and makes the annotation process extremely tedious. In contrast, it is more efficient to compare two NFoV trajectories. Hence, we conducted a user study in NFoV, which matched the setting of our targeted use case.

*indicates equal contribution

		AUTOCAM	Ours (20 videos)
Similarity	Trajectory	0.304	**0.426**
	Frame	0.581	**0.764**
Overlap	Trajectory	0.255	**0.355**
	Frame	0.389	**0.560**

Table 1. Performance on [1].

5.2. Apply our model on the dataset in AUTOCAM [1].

We cannot train our domain-specific agents using the dataset in [1] because the training videos are given in the NFOV format instead of the 360° format. Hence, we applied our model (trained for skateboarding) to all the 20 testing videos (downloaded on Jan. 30, 2017) provided with ground truths in the project page of [1]. Note that all the 20 testing videos are given in the 360° format but are in different domains (hiking, mountain climbing, parade, and soccer). We reported the results using metrics adopted by [1] in Table. 1. Typical and failure example videos are available at https://aliensunmin.github.io/project/360video. We found that our method achieved a 140% performance boost of [1] in both similarity and overlap trajectory metric.

5.3. How order consistency are handled when objects disappear/reappear?

For each frame, the feature vectors of the objects are concatenated as a vector based on the order of their scores (See Eq.5 in the main paper). When an object disappears or reappears, the concatenated vector does change. However, we empirically found that RNN seems to embed different vectors of similar scenes into similar points in the embedded space and did not suffer from this problem.

References

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

Our Method	Skateboarding		Parkour		BMX		Dance		Basketball	
	MO	MVD	MO	MVD	MO	MVD	MO	MVD	MO	MVD
N=8	0.68	2.99	0.69	3.71	0.65	8.58	0.74	2.53	0.67	5.36
N=16	0.68	3.06	0.74	4.41	0.69	8.36	0.76	2.45	0.66	6.50
N=32	0.68	3.22	0.65	3.28	0.70	7.94	0.73	2.48	0.69	5.04

Table 2. Sensitivity analysis of number of candidate objects N on all five domains.

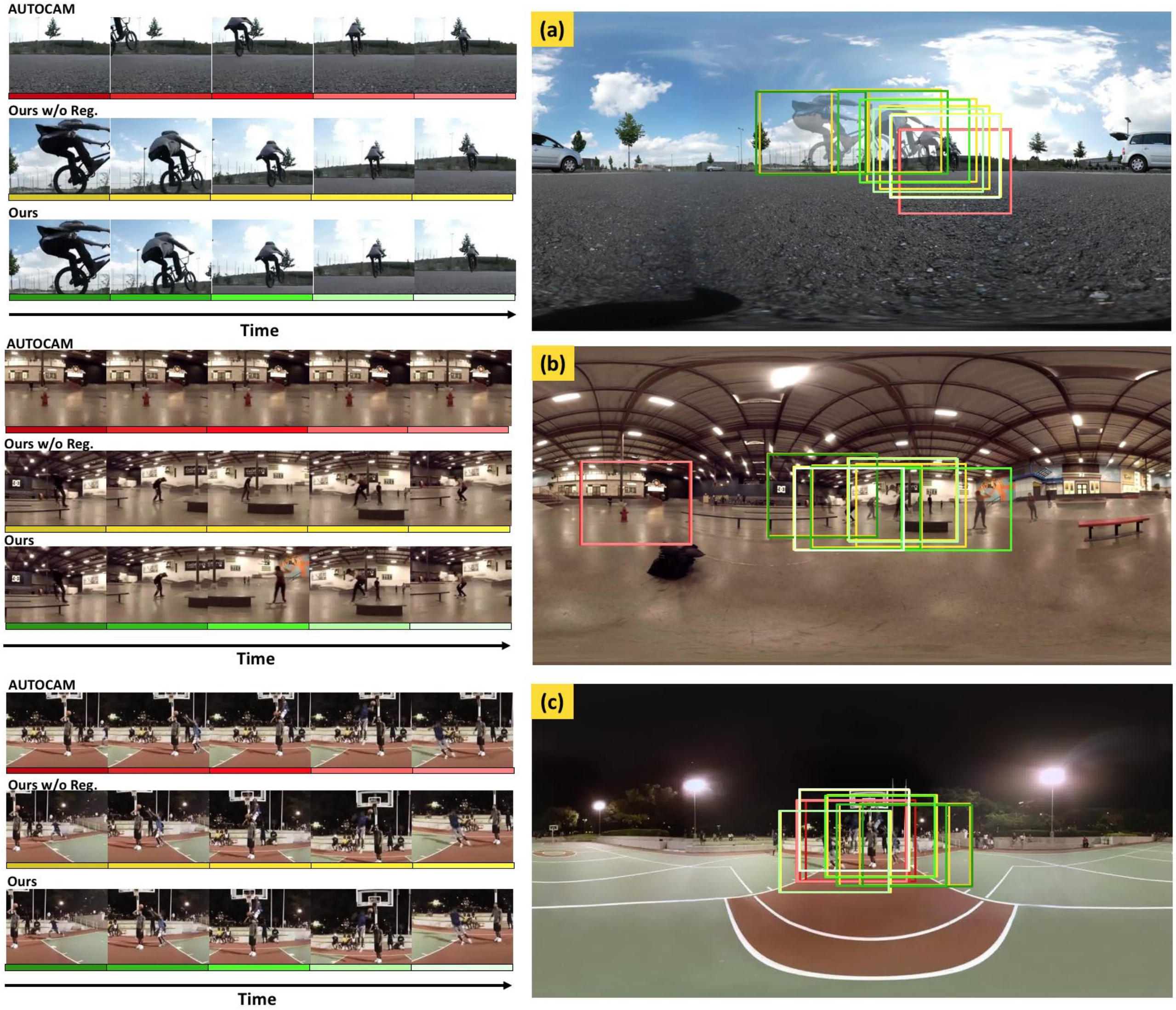

Figure 1. Typical examples of three methods: AUTOCAM, Our method, and Our method without Regressor, from three domains: (a) BMX, (b) skateboarding, and (c) basketball. For each example, the right panel shows a panoramic image with montaged foreground objects. The left panel shows zoomed in NFoV centered at viewing angles generated by each method, respectively. We further overlaid the NFoV from AUTOCAM, Our method, and Ours without Regressor in red, green and yellow boxes, respectively, in the left panoramic image.

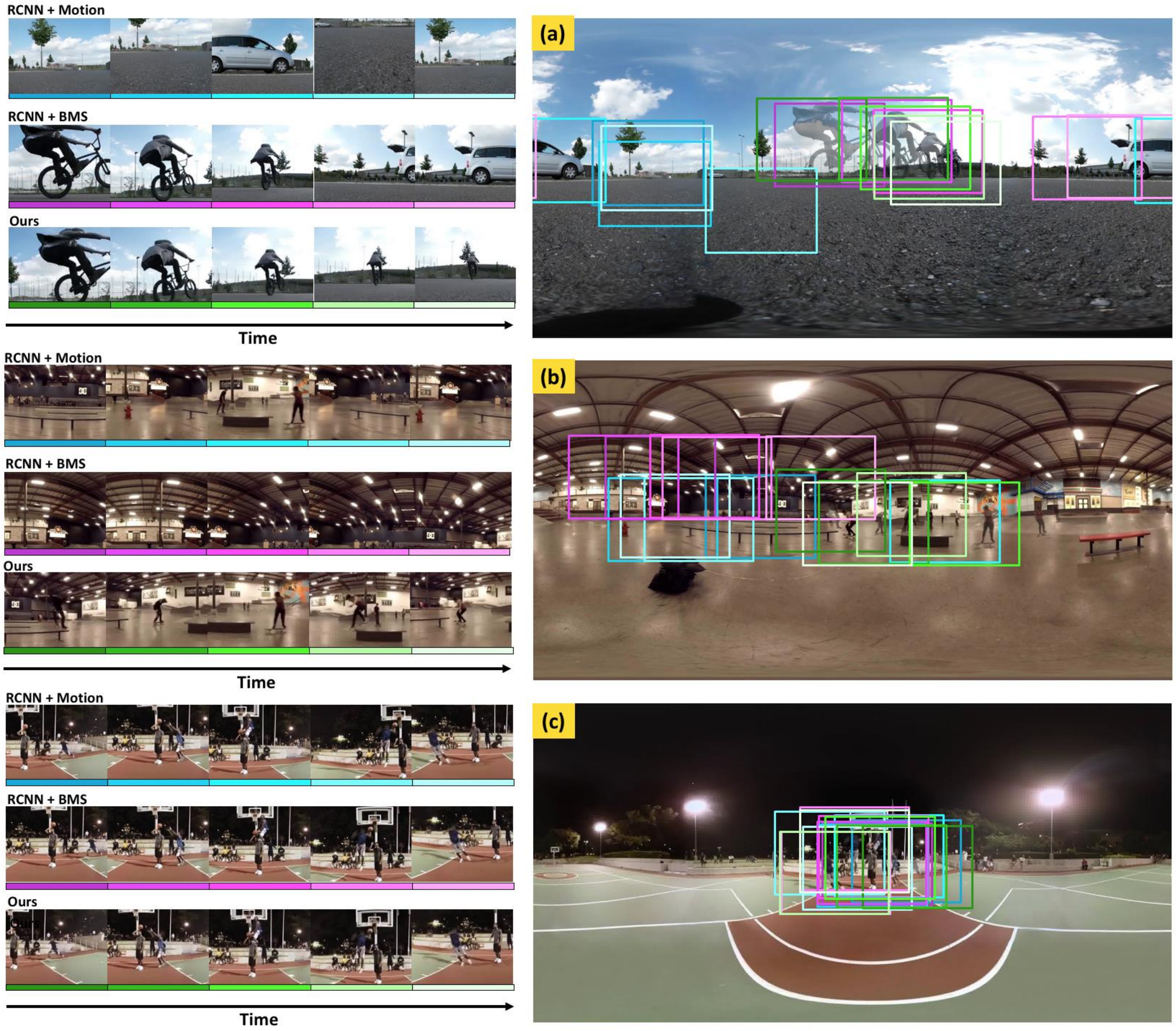

Figure 2. Typical examples of three methods: RCNN + Motion, RCNN + BMS, and Our method, from three domains: (a) BMX, (b) skateboarding, and (c) basketball. Here we illustrate different results by the same way as in Fig. 1, but overlaid the NFoV from RCNN + Motion, RCNN + BMS, and Our method in cyan, pink and green boxes.