Pano2Vid: Automatic cinematography for watching 360° videos

Yu-Chuan Su, Dinesh Jayaraman, and Kristen Grauman

The University of Texas at Austin

The supplementary materials consist of:

- Capture-worthiness analysis
- How to generate multiple trajectories
- HumanEdit interface
- HumanCam video selection details
- Spatio-temporal glimpses sampling details

Please see `http://vision.cs.utexas.edu/projects/Pano2Vid` for example AutoCam output, HumanEdit annotation, and the annotation interface.

A Capture-worthiness Analysis

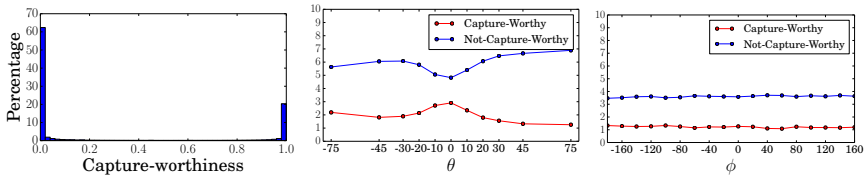
In this section, we perform further analysis of the first stage of our pipeline, discriminative capture-worthiness assessment. We first show the score distribution in Fig 1a. It turns out to be almost binary, indicating there is a clear distinction between "natural" and "un-natural" content and the problem of whether a NFOV video looks natural is well defined in the feature space. The distribution also shows that there are some ST-glimpses indeed look natural, and using them as the basis of AutoCam is reasonable.

We next show the score distribution with respect to latitude in Fig 1b and longitude in Fig 1c. We define "capture-worthy" glimpses as those with the score ≥ 0.95 and "non-capture-worthy" as those with score ≤ 0.05. The y-axis is the percentage of glimpses with respect to all candidate glimpses.

The capture-worthiness score does not have center preference but does have a weak eye-level preference. This difference is caused by the nature of the two preferences. The center preference is induced by the camera recorders that try to align content of interest in 360° videos to the center and is only present in some videos. On the other hand, the eye-level prior is induced by video framing and is universal to all videos. This is consistent with the experiment results in Sec 4.2 where Eye-level performs better, suggesting it is more universal and therefore stronger. The preference is nonetheless weak and ST-glimpses not at eye-level still have a good chance to be natural.

B How to Generate Multiple Trajectories

Because there may be multiple valid trajectories in each 360° video, instead of considering a single output, we generate $K = 20$ different trajectories by each



(a) Value distribution. (b) wrt latitude. (c) wrt longitude.

Fig. 1: Capture-worthiness score distribution analysis. Some glimpses indeed look "natural," i.e. having capture-worthy score ≥ 0.95. Also, the score demonstrates weak eye-level preference but no center preference.

method and consider the joint performance of these top-K trajectories. In this section, we describe how to generate multiple outputs in detail.

- AutoCam—We use the algorithm described in the main text to generate one optimal trajectory for every glimpse location in the last frame, i.e. $\Omega_{T,\theta,\phi}\ \forall\ (\theta,\phi)$. We then select K trajectories with maximum accumulated capture-worthiness scores. Note the accumulated score in intermediate frames can be reused, so all these trajectories can be constructed in one pass of the dynamic programming algorithm. The same method is used to generate K trajectories for Saliency.
- AutoCam w/o Stitching—This is a stochastic method that samples a ST-glimpse at each time step independently of others. The probability distribution of ST-glimpses within each frame is obtained by taking the softmax function on the capture-worthiness scores. We sample the glimpses K times independently to generate K trajectories.
- Center—This is also a stochastic method where the trajectory starts at $\theta = 0, \phi = 0$ and then performs random motion in the following time steps. Similar to AutoCam w/o Stitching, we generate multiple trajectories by performing K independent samples.
- Eye-level—We generate trajectories with static ST-glimpse location $\Omega_{\theta,\phi}$ in every frame where $\theta = 0°$ and $\phi \in \{0°, 20°, 40°, \ldots, 340°\}$. The ϕ samples are the same as our ST-glimpses and results in $K = \frac{360}{20} = 18$ trajectories instead of 20.

In the **Distinguishablility** and **Transferability** metrics in HumanCam-based evaluation, the K trajectories are treated as K independent samples for classification. In **HumanCam-likeness** metric, these K trajectories are ranked jointly, and the average/median ranking is used to evaluate the method on a particular 360° video. Similar to **HumanCam-likeness**, the K trajectories are evaluated independently in HumanEdit-based evaluation and their average similarity/overlap is used as the performance measurement on the 360° video.

C HumanEdit Interface

In this section, we illustrate the HumanEdit interface by breaking down all the components. **Please see the project webpage for the interface in action.**

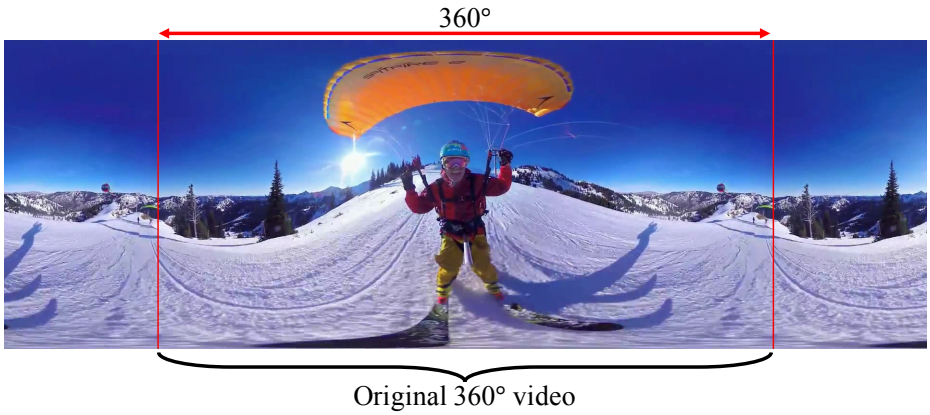


Fig. 2: We display the entire 360° video in equirectangular projection.

We display the entire 360° video in equirectangular projection so the annotator can see all video content at once. See Fig. 2. While the goal is to capture FOV video by the virtual NFOV camera, content outside the camera FOV provides important information for directing the camera. This is similar to the situation when a videographer is capturing a video, where the videographer may look at the surrounding environment without moving the camera.

Fig. 3: We extend the video horizontally by 90° on both side.

If we display only 360° horizontally, the $\phi = \pm 180°$ boundaries will appear on the opposite side of the screen and be discontinous, even though they correspond to the same direction physically. This discontinuity may cause difficulty for perception and annotation. To remedy this problem, we extend the video by 90° on both left and right (Fig. 3) so $\phi \pm \Delta\phi$ are adjacent to ϕ on the screen $\forall\phi$. The $360° + 2 \times 90°$ video will span the width of the screen.

Fig. 4: The annotator controls the camera center using the mouse. The camera FOV is backprojected onto the 360° video during annotation to show the annotator what it captures.

The annotators are asked to control the virtual camera direction within the video using mouse while the video is playing, and we record the mouse location throughout the video as HumanEdit. The camera center, i.e. mouse location, is highlighted by a red circle. To help the annotator understand what content does the virtual camera really captures, we backproject the camera FOV on the video (in cyan) in real time. See Fig. 4.

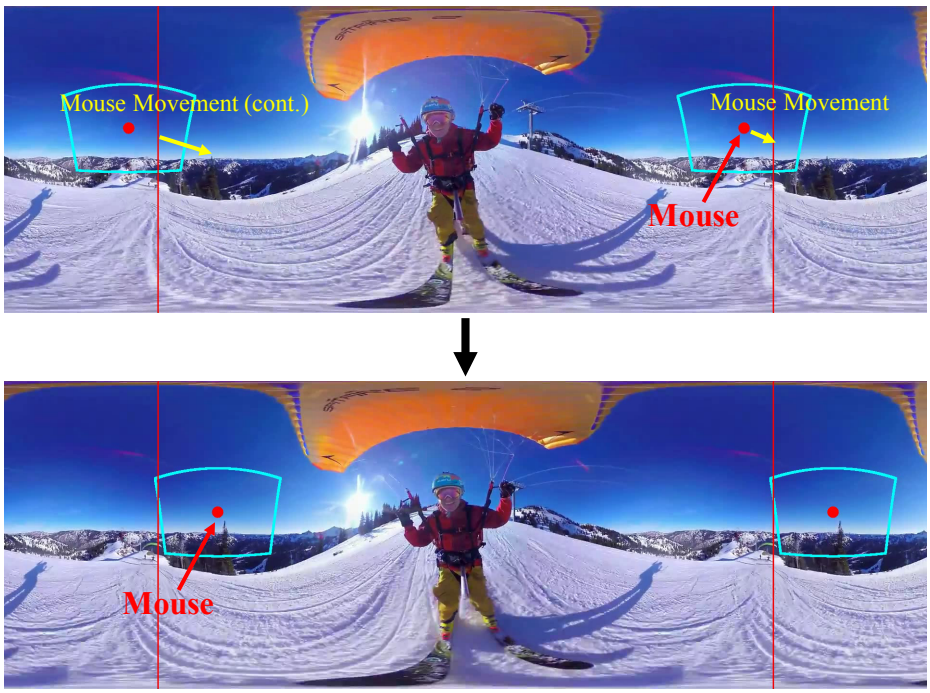


Fig. 5: The mouse is restricted in the original 360° horizontally. If the mouse move over the 360° edge, it will be repositioned to the duplicate position within 360° boundary.

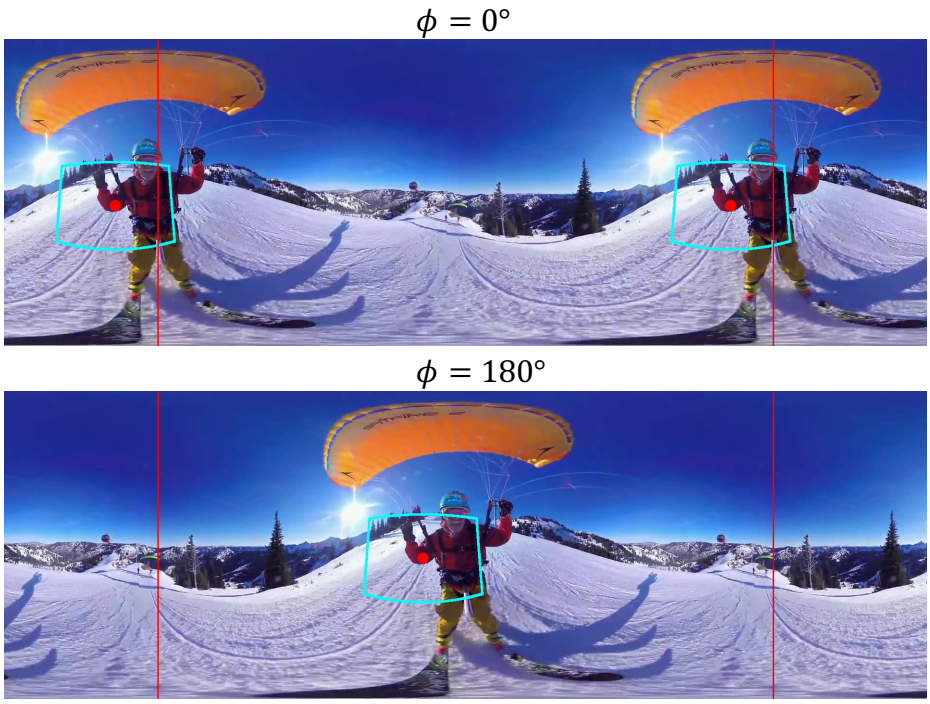


Fig. 6: The annotator can select ϕ that corresponds to the display center.

To simulate the smooth camera motion in the real world, we restrict the mouse to move within the original 360° video and reposition the mouse to the duplicate position within 360° boundaries when it passes through the $\phi = \pm 180°$ edges, as shown in Fig. 5. Because the camera center and FOV are also duplicated, they will not disappear but behave as if the mouse remains at the original location after it is repositioned.

We also ask the annotator to choose the longitude for the center of display before the video starts playing. See Fig. 6. This allows the annotator to place the content of interest in the middle of screen and makes annotation easier.

D HumanCam Videos Collection

We use the Youtube api with the "videoDuration" filter set to short (i.e. <4 minutes,) to search for NFOV videos. Because the Youtube api only returns a limited number (about 500) of results for each query, we perform 4 queries for each term with the "publish time" filter set to {2016, 2015, 2014, None} respectively. Duplicate results are removed (by Youtube ID).

E ST Glimpses Sampling

We perform dense sampling for ST glimpses both spatially and temporally. Spatial samples are drawn from spherical coordinates following the camera model [1]. The samples are chosen to be as dense as possible to enable fine camera control, but sparse enough to meet computation resource constraints. We sample time and longitude ϕ uniformly, but more densely around the equator for latitude θ, because adjacent glimpses with the same polar angle will have larger overlap at high latitude.

References

1. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: CVPR. (2012)