Route Tapestries: Navigating 360° Virtual Tour Videos Using Slit-Scan Visualizations

Jiannan Li
jiannanli@dgp.toronto.edu
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada

Jiahe Lyu
jiahe.lyu@mail.utoronto.ca
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada

Maurício Sousa
mauricio@dgp.toronto.edu
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada

Ravin Balakrishnan
ravin@dgp.toronto.edu
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada

Anthony Tang
tonytang@utoronto.ca
Faculty of Information, University of Toronto
Toronto, Ontario, Canada

Tovi Grossman
tovi@dgp.toronto.edu
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada

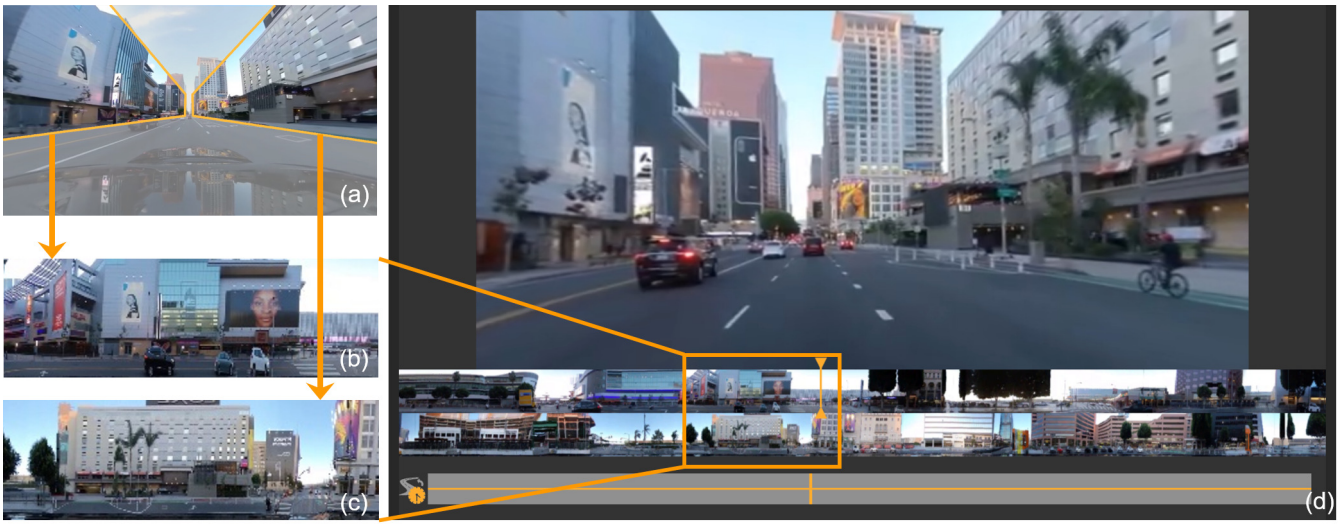

Figure 1: (*a, b, c*) Route Tapestries are continuous orthographic-perspective projections of the scenes along the camera route of a 360°virtual tour video. (*d*) Tapestry Player uses Route Tapestries as its timeline for efficient navigation of 360° virtual tour videos. Video source: https://youtu.be/kdGlselFto0

ABSTRACT

An increasingly popular way of experiencing remote places is by viewing 360° virtual tour videos, which show the surrounding view while traveling through an environment. However, finding particular locations in these videos can be difficult because current interfaces rely on distorted frame previews for navigation. To alleviate this usability issue, we propose Route Tapestries, continuous orthographic-perspective projection of scenes along camera routes. We first introduce an algorithm for automatically constructing Route Tapestries from a 360° video, inspired by the slit-scan photography technique. We then present a desktop video player interface using a Route Tapestry timeline for navigation. An online evaluation using a target-seeking task showed that Route Tapestries allowed users to locate targets 22% faster than with YouTube-style equirectangular previews and reduced the failure rate by 75% compared to a more conventional row-of-thumbnail strip preview. Our results highlight the value of reducing visual distortion and providing continuous visual contexts in previews for navigating 360° virtual tour videos.

UIST '21, October 10–14, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8635-7/21/10…$15.00
https://doi.org/10.1145/3472749.3474746

CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**.

KEYWORDS

360° Video, Navigation, Virtual Tour

ACM Reference Format:

Jiannan Li, Jiahe Lyu, Maurício Sousa, Ravin Balakrishnan, Anthony Tang, and Tovi Grossman. 2021. Route Tapestries: Navigating 360° Virtual Tour Videos Using Slit-Scan Visualizations. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3472749.3474746

1 INTRODUCTION

360° virtual tour videos show the surrounding view while traveling through an environment and allow users to look around freely. They convey a strong sense of immersion and have become a popular way for people to experience and explore remote places. For example, families may use them to compare vacation destinations; students may visit prospective college campuses, or office workers may seek a brief covert respite from their workplace. Online video platforms such as YouTube[1] have dedicated virtual tour channels, where tours of urban landscapes, museums, and college campuses are common; these channels[2] include both normal field-of-view (NFOV) and 360° virtual tours, some of which have accumulated several million views. While this emerging media is popular, it also creates new usability challenges for viewers and content creators: current 360° video playback interfaces are still primarily based on designs for NFOV videos, without considerations for the affordances of 360° content.

Navigating to specific scenes is a common task people perform when consuming NFOV videos [25, 35]. However, contemporary interfaces for quickly navigating video content are poorly suited for 360° videos. Many tour videos are long in duration. For example, the top-five most watched 360° videos on ProWalks[3], a popular virtual tour Youtube channel, have an average length of 84 minutes. For these long videos, people might want to quickly skip over some parts of them and resume watching once a particular scene of interest is in sight. Because the viewer can only see a portion of the entire 360° scenes at a time, simply clicking on points along the timeline to jump to a later or earlier frame may cause them to miss relevant visual information. With a few exceptions [33, 51], current systems primarily rely on planar thumbnails as an overview of an entire frame to facilitate temporal navigation. The thumbnail is displayed when the user selects a frame by placing the pointer over the player timeline or dragging the timeline slider. The previews are created using projection methods to warp the spherical 360° images into 2D visualizations (e.g. equirectangular projection [28], and stereographic—a.k.a. Little Planet—projection [30]). While these projection techniques have their strengths [5, 30], they distort the landscapes and architectural features in 360° virtual tour videos (Figure 2) and make finding particular targets more challenging.

To reduce the visual distortion in timeline previews, we introduce *Route Tapestries*, which supports video navigation through strip-shaped 'tapestries' constructed with a continuously captured scene along the route. We drew inspiration from Video Tapestry [4],

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

[2]e.g. J Utah, Prowalk Tours, Virtual Japan, VR World 360°

[3]https://www.youtube.com/c/ProWalks/featured

Figure 2: A 360° virtual tour video frame projected through (*a*) perspective projection (NFOV) (*b*) equirectangular projection (*c*) Little Planet projection. Video source https://youtu.be/_lA6svC-v6k

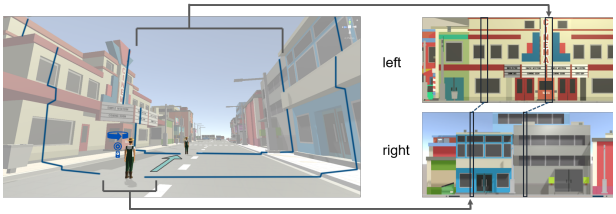

Figure 3: Using the slit-scan imaging technique, a moving 360° camera captures the scenes along the route as long strips by 'scanning' them. At short intervals, pixels from the part of the scene marked by the blue lines are captured and concatenated to form the eventual Route Tapestries.

an NFOV video navigation interface using strip-shaped content summaries consisting of keyframe mosaics. We make the critical observation that in a large number of virtual tour videos, the viewer's scenes of interest lie on mostly continuous boundaries along the camera route (e.g. buildings along a street, walls of a gallery, etc.). In Route Tapestries, we capture these boundaries (typically to the left and right of the path of travel) as continuous 'strips' extending along the camera path (Figure 3) and leverage them for video navigation. Our visualization technique uses slit-scan imaging, where a scene is captured one slice at a time while a moving camera 'scans' the scene (as illustrated in Figure 3). As a result, the generated Tapestries are formed through orthographic-perspective projection and appear in general much less distorted to human eyes than equirectangular or stereographic images (compare Figure 1 and Figure 2). Visual summaries for video skimming and navigation have a long history of success in the research literature [4, 12, 31, 50]. As a visual summary, Route Tapestries provide continuous and low-distortion presentations of virtual tour video content to facilitate navigation. Instead of going through a video frame by frame, users can look at the captured environments in their entirety and make more contextualized navigation decisions.

In this paper we present the design process, technical implementation, and evaluation of Route Tapestries. We first analyze several designs for removing visual distortion and explain our rationale for choosing a visually continuous design over a discrete one, such as two rows of NFOV thumbnails showing the left and right sides of the camera path. We then introduce an automated algorithm that generates Route Tapestries from a 360° virtual tour video. We built Tapestry Player, a desktop-based 360° video player prototype using Route Tapestries for timeline navigation. We then conducted

a controlled experiment where 12 participants completed target-finding tasks using Tapestry Player and two baseline techniques (equirectangular previews as on YouTube, and row-of-thumbnail strip previews as a discrete alternative to Route Tapestries). The study results show that with Route Tapestries, the participants completed the tasks 22% faster than equirectangular previews and missed 55% and 75% fewer targets than equirectangular and row-of-thumbnail strip previews, respectively.

As a first step in exploring the concept of Route Tapestries, in this work we focus on its application specifically for virtual tour videos on a desktop environment. We conclude the paper with a discussion on adapting the current approach for head-mounted displays and a broader range of 360° video content types. We make three contributions in this work:

- The concept of Route Tapestries as an efficient way to navigate 360° virtual tour videos, and a method for generating them automatically;
- The design and implementation of Tapestry player;
- A user study that demonstrates the benefits of reducing distortion and maintain visual continuity in timeline previews for 360° virtual tour video navigation.

2 RELATED WORK

This work is situated within the growing area of research exploring new ways of interacting with 360° videos and is built on prior work in video navigation interfaces. Our technical approach is inspired by the slit-scan techniques applied in both photography and computer science. We also review research in multi-perspective panoramas, which are visually similar visualizations to ours.

2.1 Interacting with 360° Videos

The full panoramic view of 360° videos enables strong immersion but also brings about new usability challenges. Several recent projects aimed to assist *spatial navigation* in 360° video players to help viewers locate important characters or events out of their field-of-view. Pavel et al. [36] proposed two techniques, pre-aligning the points of interest with the viewing direction at each cut and actively reorienting the shot to reveal relevant content upon user input. Lin et al. [22] introduced Outside-In, which signals off-screen points of interest through picture-in-picture previews. Liu et al. [23] took a different perspective, generating video textures that will keep looping seamlessly until the viewer turns to them. Other projects aimed to automate spatial navigation [7, 18, 46, 47] by finding points of interesting using computational measures, such as saliency [45]. Pavel et al. [36] also discussed similar approaches to automate the shot reorientation techniques.

Temporal navigation of 360° videos is also a challenge. Some producer-oriented tools [13, 29, 30, 51, 52] offer temporal navigation interface specifically tailored for 360° videos. Nguyen et al. [30] presented Vremiere, a system for editing 360° videos directly in HMDs. Vremiere displayed a 'Little Planet' thumbnail for timeline navigation and highlighted its benefits in promoting spatial awareness. ConvCut [51] used content analysis to provide support for efficiently editing long 360° conversation footage into short highlights. It augmented raw footage with conversation transcripts and other semantic information to aid temporal navigation. Neng and Chambell [28] presented a desktop 360° video player that showed rectangular thumbnails for selected frames in the videos. Hand gestures have been used for 360° video navigational input [38, 43] but were limited to linear controls such as play/stop or fast-forward.

Our work focuses on temporal navigation for a specific type of 360° videos which deliver virtual tour experiences. Geo-tags along the camera routes [20, 33] can facilitate navigating such videos. However, camera position information is not always available for arbitrary videos. In comparison to temporal navigation interfaces for generic 360° videos, such as Little Planet [30] or equirectangular previews, our technique aims to improve efficiency by creating visual summaries that leverages the camera motion and scene characteristics of 360° virtual tour videos.

2.2 Video Navigation Interfaces

Research on NFOV video navigation has a long and rich history. Earlier approaches included improving the timeline slider to allow for more fine-grained adjustments [14, 40]. Some later research explored aiding navigation with extra information derived either from video content analysis or external annotations. Low-level visual features [44], scene boundaries [3], and salient frames [39] can help viewers leverage extra visual information, but they usually do not support seeking arbitrary scenes. SceneSkim [35] enabled searching and browsing movies through synchronized captions and plot summaries. Kim et al. [19] enhanced MOOC video timelines with user interaction traces to help learners find essential parts. These approaches typically rely on clear semantic structures, which are less common in virtual tour videos. Direct manipulation of objects in the videos [10, 32] offers an intuitive navigation method, but they are not suitable for virtual tour videos where scenes change frequently. The Swift technique [24], which displays pre-cached low-resolution frames during scrubbing, has been shown to improve scene-finding performance. Similar features can also be found on online video platforms such as YouTube and are incorporated into our system.

Also relevant to our approach is video navigation through content summaries. One kind of summaries consists of a selection of keyframes. The early Hierarchical Video Magnifier [26] marked the video timeline with thumbnails for evenly sampled frames and supported recursive zooming. Later research expanded their method with more sophisticated thumbnail selection and clustering schemes [9]. Thumbnails can also be presented in a grid layout to provide an overview of either local [25] or global [15] content. In contrast, other summaries integrate relevant visual elements to compose a coherent narrative. Goldman et al. [12] turn video input into storyboard-style images with arrows that illustrate character motion and can be dragged along for video scrubbing. Video Tapestry [4] merged visually similar video keyframes to form a navigation timeline. Although their user study did not find an efficiency improvement in navigation, we are inspired by the concept and visual style. Video Summagator [31] transforms a 2D video into a 3D volume, in which navigation can be achieved through moving along the extra dimension. While these techniques are not broadly applicable to generic videos, they exploit scene or character continuity in source videos to create visually appealing summaries and practical navigation tools. Our method is inspired by prior

NFOV video navigation interfaces based on content summarization, leveraging scene continuity to produce a compact visualization for 360° virtual tour videos.

2.3 Slit-Scan Visualizations

Slit-scan imaging has a long history in photography and has inspired both visual artists and computer scientists alike [41, 49, 54]. In slit-scan photography, a thin slit is placed between the camera and the scene, blocking incoming light rays except those passing through the slit. In some applications, as objects and entities move past the slit, slit-scan photography creates a visual timeline of objects passing by the slit (e.g., [49]). In other applications, the camera itself moves about the scene and generate panoramas. Images formed thorough this method are typically orthographic-perspective, i.e. orthographic along the object or camera motion path but perspective along the slit [54]. Zheng [54] introduced the Route Panorama system. It produced strip-shaped panoramas for street views using a GPS-tracked camera to scan the scenes. For such cases, Peleg et al. [37] has shown that the optimal slit shape should be orthogonal to the scene optical flow.

Prior research in HCI has also explored using slit-scan visualization for analyzing recorded events. Nunes et al. [34] introduced the TimeLine system, which incorporated slit-scan visualizations for discovering temporal patterns in video history. Tang et al. [49] presented a video slicing approach consisting of joining pixels from user-placed marks on the video, called slit-tears, for revealing various visual patterns in video scenes. We extend prior work on slit-scan visualization [37, 54] by introducing an automatic generation process applicable to user-generated 360° videos. We also show the potential of slit-scan visualizations for navigating 360° virtual tour videos.

2.4 Multi-perspective Panorama

Multi-perspective panoramas provide visually appealing visualizations that aggregate location- or motion-based information by combining different perspectives from multiple pictures or videos. They are especially suited for illustrating planar scenes such as landscapes or street imagery. Roman et al. [41] introduced an interactive system for generating multi-perspective urban landscape images composed of serially blended cross-slits images from video frames captured by a moving vehicle. Agarwala et al. [1] employed an automatic Markov Random Field optimization approach to generate composited panoramas of street imagery. On a different note, Street Slide [21] improves the flat multi-perspective panoramas produced by the previous approaches by adding parallax effects to create immersive panoramas. Our approach builds on visualizations similar to multi-perspective panoramas. However, the approaches above prioritize visual quality and usually requires information not available in user-generated videos, such as manual labels [1, 41] or precise camera position [21] from external sensors.

3 NAVIGATING 360° VIRTUAL TOUR VIDEOS: DESIGN PROCESS

To reduce distortion in current navigation previews for 360° virtual tour videos, we have explored a range of design options. Our design process started with the observation that the scenes along the left and right sides of the camera routes can provide key information for navigating 360° virtual tour videos. We first considered single-thumbnail previews (equirectangular, Little Planet, and NFOV) and strip previews consisted of multiple discrete thumbnails. We further noted that for 360° virtual tour videos where scenes of interest lie on largely continuous boundaries (e.g. buildings and walls) along the tour path, visually continuous previews, including Route Tapestries and stitched panoramas, can be created and they hold potentials for stronger navigation support. Below we introduce our analysis of these options and the design of Route Tapestries.

3.1 Equirectangular, Little Planet, and NFOV Images

Equirectangular and Little Planet thumbnails are preview formats that have already been adopted in commercial 360° video players (e.g. YouTube) and VR video editors [30]. Both of them introduces strong visual distortion through non-perspective projection (Figure 4.a and 4.b). Converting them to NFOV images which show the left and right sides of the camera paths (Figure 4.c) can completely remove projection distortion. To gauge the navigation performance of the three preview formats, we conducted serveral initial pilot studies comparing equirectangular with Little Planet, and equirectangular with a pair of NFOV images (one for the left side and one for the right) using a target-seeking task. We found that equirectangular previews tended to be more efficient than Little Planet for navigating 360° virtual tour videos, likely because the Little Planet projection overly compresses the scenes at the ground level. We also found that NFOV image pairs did not appear to improve navigation performance over equirectangular previews. This appeared to be a result of the limited FOV range of NFOV image (90° horizontal and 60° vertical) making the surrounding environments difficult to understand. While it is possible to further increase the image FOV, the visual quality deteriorates quickly on the edges of the

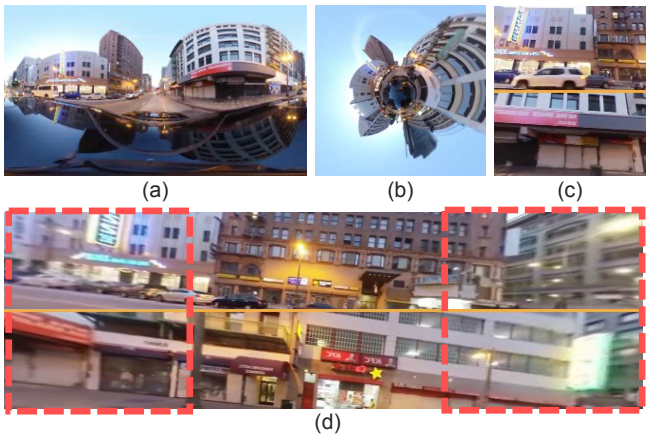

Figure 4: Single-thumbnail preview designs for 360° virtual tour videos. (*a*) equirectangular (*b*) Little Planet (*c*) NFOV images (90° horizontal FOV) (*d*) wide-FOV images (150° horizontal FOV), note the visual quality in the highlighted areas. Video source https://youtu.be/kdGlselFto0

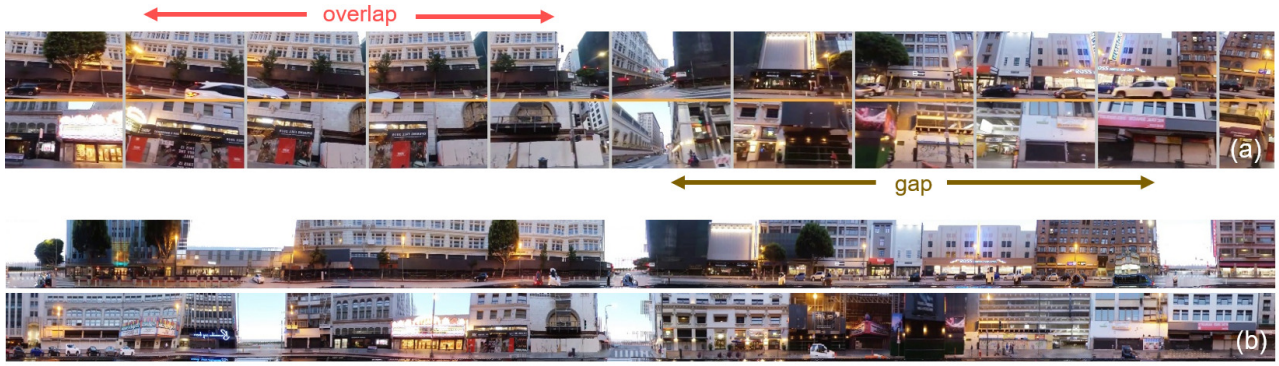

Figure 5: (*a*) Row-of-thumbnail strips and (*b*) Route Tapestries for the left and right sides along the camera route in a 360° virtual tour video. (*a*) and (*b*) show approximately the same part of a street. Video source https://youtu.be/kdGlselFto0

converted NFOV images, due to the spherical projection model of the source frame (Figure 4.d).

3.2 Row-of-Thumbnail Strips

Displaying multiple thumbnails in a row is a timeline preview design employed in commercial video editors (e.g. iMovie, Adobe Premiere Pro) and research prototypes [25, 26] to provide contexts and facilitate navigation. Similarly, two strips of NFOV thumbnails, one for the left and one for the right, can be used as previews for navigating virtual tour videos (Figure 5.a). However, depending on the sampling rate, there could be overlaps or gaps between consecutive thumbnails in the strip. For example, overlaps can be found between the thumbnails in the left part of Figure 5.a and gaps can be found in the right part. Overlaps limit the range of the visible contexts and gaps could potentially cause difficulty in parsing video scenes from thumbnails. Therefore, we further explored visually continuous preview designs and later conducted user study to investigate whether visual continuity can combat these issues.

3.3 Route Tapestries

We make the observation that in a large number of 360° virtual tour videos, the scenes of interest for viewers form largely continuous boundaries along the camera travel paths. Such videos can be indexed with Route tapestries, which are long, continuous image strips depicting environments on one particular side of the camera route (Figure 5.b). Route Tapestries can be created through orthographic-perspective projection using the slit-scan technique. Specifically, the image strips are composed of vertical pixel slices from many individual NFOV images, each of which is transformed from a 360° video frame through a virtual NFOV camera pointing at the specified side of the camera path. The pixel slices are selected through thin virtual 'slits' placed in front of the virtual NFOV cameras. To properly determine the positions and the sizes of these slits, we use camera trajectories and rough scene geometries obtained from simultaneous localization and mapping (SLAM). Since this method leverages the scene geometric information from the full 360° frames and the full duration of the videos, it is robust to the complexities in real world videos, such as motion blur and moving objects. In Sec. 4, we will introduce the procedure for automatically generating Route Tapestries from 360° virtual tour videos.

Figure 6: Examples of successful and unsuccessful image stitching results. The video frames were taken from approximately the same part of the video as in Fig. 5.

3.4 Stitched Panoramas

Another potential option to create visually continuous previews for 360° virtual tour videos is to merge NFOV images showing one side of the route to form a panorama using image stitching techniques (e.g. [6, 56]). However, the contents and visual quality of common 360° virtual tour videos pose significant challenges for this approach. First, popular image stitching methods assume planar scenes or pure rotation as the camera motion [48], both of which are commonly violated in user-generated virtual tour videos. Second, image stitching algorithms rely on precise feature matching [6] or reliable pixel similarity metrics [56] between image pairs. Rolling shutter effects, motion blur, illumination change, and moving objects, all common in virtual tour videos, make both the requirements hard to satisfy and lead to poor stitching results or failure. We experimented with the widely-used stitching algorithm by Brown and Lowe [6], using a popular open-source implementation[4]. We found that it was hardly feasible to produce long panoramas (stitched from more than ten images) from NFOV images densely sampled (every 1 second) from 360° virtual tour videos. Furthermore, failures and poor quality results were still common for shorter sequences (5-8) of images. See Figure 6 for examples of successfully and unsuccessfully stitched panoramas from NFOV images.

[4]https://github.com/ppwwyyxx/OpenPano

4 ROUTE TAPESTRY GENERATION

Previous methods that exploit the slit-scan technique for image mosaic generation rely on camera motion data from external sensors [54] or accurate optical flow [37], both of which are hard to obtain for user-generated 360° virtual tour videos. Furthermore, these methods assumes that the captured scene lies roughly on a single plane. This assumption is frequently violated in virtual tour videos, where the cameras often travel from wide streets to narrow alleys, or vice versa. We propose a method for automatically generating Route Tapestries for a large portion of user-generated 360° virtual tour videos. On a high level, it extracts the camera trajectory and low-resolution scene geometry from the input video and then uses them to select the desirable video frame and slit parameters for covering each small part of the scene. All the pixels through the slits are then concatenated together to form a composite image.

4.1 Assumptions

Our method assumes that the objects of interest for video navigation form largely continuous boundaries along the camera path, and they take a significant portion of the viewport when the viewer turns to look at them. The boundaries can be buildings along a road, walls of a room, or trees along a path in the woods. These assumptions are met in a large number of 360° virtual tour videos. We performed searches on YouTube for 360 videos uploaded from May 2020 to May 2021 with the keyword 'virtual tour'. The results showed that 49% and 81% of the top 100 most viewed videos between 4-20 minutes and over 20 minutes respectively show tours in indoor or outdoor urban environments, or similar places like historical city sites. These videos largely consist of moving shots that show streets, corridors, and other environments featuring clear boundaries on the left and right sides of camera paths. Some parts of the videos do contain static shots (e.g. narrator speaking) but these parts could be isolated by algorithms or video creators, which will be further discussed in Sec. 8.2. For simplicity, we also assumes that there is no jump cut or large camera altitude change in the video.

4.2 Camera Tracking and Scene Geometry Acquisition

We first use visual simultaneous localization and mapping (SLAM) to detect camera trajectories as well as sparse scene feature points from the input video. We also compute the average speed v of the camera throughout the video as a distance measure. Our central goal is to recover the the shape of the boundaries of the camera path from the sparse point cloud without expensive 3D mesh reconstruction [42]. In the remaining algorithm description, we use the Route Tapestry for the scenes on the right side of the camera path as an example. At every keyframe identified in the SLAM process, we find out feature points on the right boundary by first selecting the points that are visible in this frame and further filtering out those that are too high, too low, or not visible when the viewer turns to the right side of the route. Specifically, any feature point at the position (x, y, z) in the camera reference frame (Figure 7.a) is removed if any of the following conditions is violated

$$\alpha_1 v < y < \alpha_2 v, \quad \left|\frac{y}{x}\right| < \tan\frac{\theta_v}{2}, \quad \left|\frac{z}{x}\right| < \tan\frac{\theta_h}{2}$$

α_1 and α_2 are constants for setting the lower and upper bounds of feature point altitude, and θ_v and θ_h are constants that determines the vertical and horizontal FOV of the virtual perspective camera for generating the right-side Tapestry. We compute the median location of the remaining points as an *anchor* on the right boundary. All the anchors and the camera trajectory are then projected onto the ground plane. A moving median filter is applied to 2D locations of the anchors to remove outliers. The resulting 2D anchors are connected to form a piecewise linear curve as the right boundary (Figure 7.b). This curve is linearly interpolated between the anchors to yield a sequence of N points $\{\mathbf{p_i}, i = 1 \dots N\}$ densely sampled from the boundary. Similarly, the 2D camera trajectory can be represented as a sequence of M points $\{\mathbf{c_j}, j = 1 \dots M\}$, where M is the total number of video frames. In our implementation, we used the open-source ORB-SLAM [27] for localization and mapping with an omnidirectional camera model. For feature point filtering, we used $\alpha_1 = -3$, $\alpha_2 = 3$, $\theta_v = \pi/2$, and $\theta_h = \pi/2$, a window size of 10 points for the moving median filter.

4.3 Slit Parameters and Image Formation

In this step, we seek to find a video frame and an ideal slit size and slit location for capturing each small part of the environment on the route boundary. For every small segment s_i between two consecutive points $(\mathbf{p_i}, \mathbf{p_{i+1}}), i = 1 \dots N-1$ on the boundary, we first find the closest camera position $\mathbf{c_j}$ to the center of the segment $\mathbf{q_i} = (\mathbf{p_i} + \mathbf{p_{i+1}})/2$. Choosing the closest camera position ensures the segment is captured at a high resolution and encourages visual continuity by assigning segments close to each other to camera positions that are also close to each other. After all segments on the boundary have been assigned to a camera position, we iterate over all camera positions to compute slit sizes and slit locations for the corresponding video frames. For a video frame captured at the position $\mathbf{c_j}$, we get the boundary segment set S_j containing all the segments that this frame has been assigned to. We compute

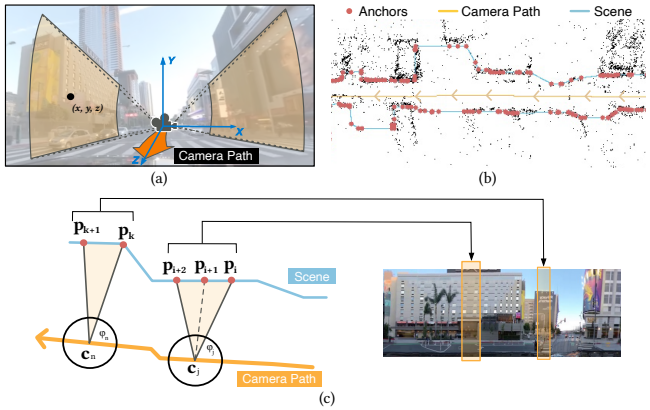

Figure 7: The Route Tapestry generation procedure. (*a*) Feature points that belongs to the boundary are selected (*b*) The reconstructed boundary: feature points (black), boundary anchors (red), the full boundary after interpolation (blue), the camera path (yellow) (*c*) calculating the slit position and slit size.

the minimum angle ϕ_j that can cover all segments $s_k \in S_j$ as the slit FOV. This 360° frame is then converted to an NFOV image facing towards the segments to be captured. That is, the effective perspective camera for this NFOV image should align with the direction of the internal bisector of angle ϕ_j (Figure 7.c). Assuming the effective focal length is f, the virtual slit should be placed at a distance of f in front of the camera lens, and the exact slit size d_j can then be computed through

$$d_j = 2f \tan \frac{\phi_j}{2}$$

Finally, all pixels through the slits are joined together following the order of the video frames.

In our implementation, we constructed a KD tree for the camera trajectory for fast closest-point queries. However, this global search can occasionally be problematic as the closest camera position found may be associated with a video frame at a very different point in time (e.g. when the cameraperson comes back to the same location later in the video). We locate these astray search results and replace them with the positions found through a local search performed on the part of the trajectory (5% of the trajectory in our implementation) where the boundary segment is first captured.

4.4 Combining Left and Right Tapestries

The lengths of the left and right Tapestries generated with the same effective focal length f can be different as the environments on the two sides and their distances to the camera are usually not the same. For tasks such as quickly skimming through the video content or searching for an known target, it is desirable that the lengths of the left and the right Route Tapestries are equal so that a user can browse them as one combined image (more details in Sec 5.1). The length matching can be done through scaling the effective focal length for one of the Tapestries. The total length l of a Tapestry can be written as

$$l = \sum_{j=1}^{M} 2f \tan \frac{\phi_j}{2}$$

where M is total number of points on the camera trajectory. For simplicity, we set $\phi_j = 0$ if the camera position c_j has not been assigned to any boundary segments. To match the lengths of Tapestry a with Tapestry b with a fixed focal length f^b, f^a can be set as

$$f^a = \frac{\sum_{j=1}^{M} \tan \frac{\phi_j^b}{2}}{\sum_{j=1}^{M} \tan \frac{\phi_j^a}{2}} f^b$$

A decrease in effective focal length will apply a virtual 'zoom-out' effect on the resulting Tapestry and thus shorten its length.

4.5 Post-processing

Standard image processing techniques can be applied to resulting Tapestries to further enhance its quality, such as increasing contrast or removing unnecessary content. For example, in Tapestries for driving videos, the roof of the car may be captured as a rectangular color blob spanning the bottom of the Tapestry. It can be removed by identify the clear boundary between the color blob and the rest of the Tapestry. We used a vertical Sobel operator to compute the image gradient and locate this boundary by finding the image row with the maximum average gradient.

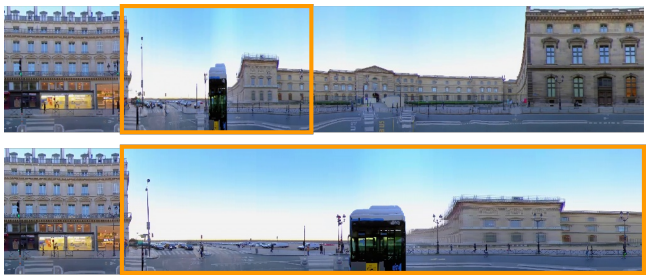

Figure 8: The Route Tapestries generated by our method (*top*) and the method of Zheng [54] (*bottom*). The distant intersection and building are stretched in the bottom result, which uses camera velocities from SLAM to simulate GPS-based velocities and a fixed scene depth that is reasonably accurate before the intersection. Video source https://youtu.be/AUZbkYwFY4M

4.6 Results

Our method does not require external sensors and automatically adapts to varying camera velocities and scene depths (see the distant intersection and building in Figure 8, top). This improves on previous methods, such as the algorithm of Zheng [54], which uses camera velocities from GPS but still need accurate scene depths that are hard to obtain for user-generated videos. Assuming a fixed scene depth could work for some parts of the video but might lead to clear visual distortion in other parts (see Figure 8, bottom). Our algorithm can generate Route Tapestries for a wide range of 360° virtual tour videos. See Appendix A for more sample Route Tapestries. We generated these examples using a Python implementation of our algorithm on a Windows desktop computer with 2.2GHz Intel i7-8750 CPU and 16 GB memory. With a pre-computed map and camera trajectory from ORB-SLAM, it took less than 1 minute to compute the slit parameters and 30-40 minutes to generate two Tapestries for a 15-minute 360° virtual tour video. Most of the computing time was spent on converting 360° frames to NFOV frames; this process can be sped up with a GPU-based conversion algorithm in place of the CPU-based one we used.

5 TAPESTRY PLAYER: A 360° VIDEO PLAYER WITH ROUTE TAPESTRIES

To prototype interactions and conduct performance evaluation for Route Tapestries, we built Tapestry Player, a desktop 360° video player using Route Tapestries as its timeline. We now describe its design and implementation.

5.1 Interface Overview

The overall layout of Tapestry Player is similar to consumer desktop 360° video players with a timeline below the video window (Figure 9). Users can drag on the video window to reorient their viewing direction in the 360° space. The Tapestry timeline consists of a progress bar separated into two parts by a horizontal line and

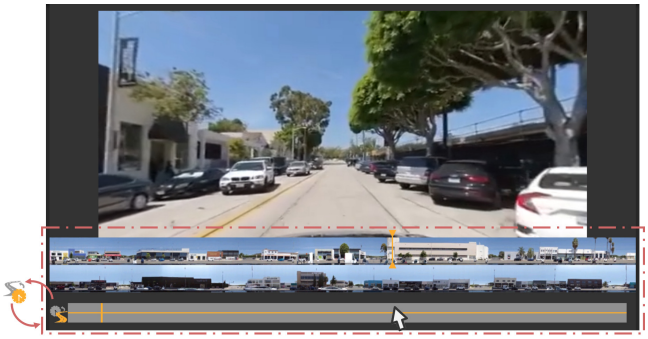

Figure 9: The Tapestry Player interface. The Tapestry timeline and previews are at the bottom. Users can click on the icon at the bottom-left corner to switch between the spatial mode and the temporal mode. Video source https://youtu.be/rIkV_bKLvSE

a playhead slider in the shape of a vertical yellow bar (Figure 9). The player uses two equal-length Route Tapestries for the left and right sides along the camera route as the video previews. When the pointer hovers over the progress bar, the Tapestry previews are displayed right above it. Since the full tapestries are usually much longer than the window width, only the parts that correspond to the current cursor position are visible in the player window (Figure 9). The user can further move the cursor between the upper and lower part of the progress bar to select the Tapestry for the left or right side, where a pair of arrows highlight the exact Tapestry position the cursor currently focuses on. Clicking on the progress bar reorients the user towards the corresponding side of the camera path in the video. The mapping from the points on the progress bar to the horizontal positions on the Tapestries and the video frames depends on the timeline mode, described below.

The Tapestry timeline can be toggled between two modes: *spatial* and *temporal* (Figure 9). While the latter works similarly to conventional timelines and supports browsing the video linearly in time, the former enables content navigation that is approximately linear in space. In the spatial mode, the horizontal positions along the progress bar are linearly mapped to the horizontal positions along the full length of the Tapestries. The positions along the Tapestries are in turn mapped to the individual video frames that the Tapestry slices at those positions are taken from. In the temporal mode, the timeline behaves similarly to conventional video timelines, where the horizontal positions along the progress bar are linearly mapped to the frames in the video. A video frame can in turn be mapped to the Tapestry slice that it has contributed to. A video frame that does not contribute to any slice is mapped to the same slice as its closest slice-contributing neighbouring frame. As an example, if the 360° camera stops for 10 seconds in a tour video, this period will be skipped on the spatial mode timeline but available on the temporal mode timeline, with all frames mapped to the same slice on the Tapestry. When the cursor moves on the progress bar, in the spatial mode the two Tapestries will move at the same rate while in the temporal mode they will move independently to maintain temporal synchronization at the cursor position. We expect these two modes to work complementary to each other: the spatial mode for quickly scanning the video content and locate relevant scenes and the temporal mode for understanding the complete scene around a particular frame. Users can use the temporal mode to navigate to the frames skipped in the spatial mode.

In both modes, frames between keyframes are mapped to pixel positions on Tapestries using linear interpolation. Clicking a point on the progress bar immediately makes the playback jump to the scene shown at the corresponding Tapestry position, and reorients the viewer towards that scene. Clicking the scene on the Tapestry directly can achieve the same effect. Users can also drag the slider to scrub through the video. During scrubbing, the user can use the Tapestries, as well as a *Swift*-style [24] full-screen low-resolution preview to check the scene of the current frame. The low-resolution preview is in the equirectangular format for covering the entire 360° field of view. Note in the spatial mode the slider moves according to the playback progress through the first Tapestry. We choose to use a single-slider timeline design for both modes to maintain interface consistency.

5.2 Implementation Details

We implemented Tapestry Player using Unity 3D. The player window is 1920 pixels in width and 1080 pixels in height. The Tapestry timeline has a length of 1830 pixels and a height of 64 pixels. Each Route Tapestry is 90 pixels in height and varies in length depending on the video content. The effective focal length of the originally longer Tapestry is scaled to match the lengths of the Tapestries for the two sides. The 360° video is rendered with a 16:9 aspect ratio, in a window of 1600 pixels by 900 pixels. All Route Tapestries were pre-rendered using the algorithm introduced in Section 4.

6 EVALUATION

We designed a controlled experiment to compare the efficiency of navigating 360° virtual tour videos using Route Tapestries against two baseline conditions. Our focus in this study was to understand which interface would allow participants to identify and locate specific scenes faster. Further, we were interested in how participants used the interfaces—for example, whether they choose to scrub through the video. Following prior studies on video navigation performances [24, 25], our evaluation tasks asked participants to navigate through the 360° video to find the target scenes given to them using the previews and the timeline provided by each interface.

We chose the two baseline interfaces to compare Route Tapestries against–the equirectangular player and NFOV-strip player. The equirectangular player models a typical 360° video player as found on YouTube, and we consider this to be a "standard" player. The NFOV-strip player employs the row-of-thumbnail strip previews, as described in 3.2. Similar previews have been widely adopted in video editing software, and prior work [25, 26] has demonstrated its effectiveness for aiding video navigation. We selected this technique as we are interested in whether the visual continuity of Route Tapestries, at the cost of additional processing, can help users avoid the issues that come with discrete thumbnail sampling, such as gaps and overlaps, and achieve better performance.

6.1 Design

The study was a repeated measures within-subject design, with the video player *interface* as the independent variable. The three interfaces were *Route Tapestry*, *equirectangular*, and *NFOV-strip*. Participants completed 14 trials of a target-finding task on a single 360° virtual tour video. We gave participants a different 360° virtual tour video for each interface to reduce learning effects between videos. We chose these three videos and the targets to be comparable in duration, visual complexity, and style. To control the order effect of *interfaces*, we used the same video presentation order but fully counterbalanced the *interface* presentation order across participants. The order of presentation of the target locations was randomized.

6.2 Interfaces

Route Tapestry. The interface follows the Tapestry Player design as described in Section 5. It uses a left- and right-side Route Tapestries of the tour video as the timeline. The lengths of the Tapestries ranged from 37181 to 40342 pixels, depending on the content of the video. All Tapestries had a height of 90 pixels. All the right-side Tapestries have been matched to the corresponding left-side Tapestries in length. All the left side Tapestries used an effective focal length of 90°. The effective focal lengths of the right side Tapestries have been adjusted to 100°, 106°, and 122°, respectively. The pixels from the car roof has been removed using the technique described in Sec 4.5.

Equirectangular. We modeled this standard player around a YouTube desktop 360° player interface (Figure 10, top), where a timeline appears underneath the video window. When the user hovers over the timeline, an equirectangular frame thumbnail corresponding to the pointer position on the timeline is displayed above the pointer. When the user scrubs the playhead slider, a low-resolution equirectangular preview for the currently selected frame is overlaid atop the video window.

In a 1920×1080 window, the timeline is 1830 pixels in length and 64 pixels in height, therefore of the same size as the Tapestry timeline. We chose not to strictly follow the YouTube player timeline dimensions to keep dragging and pointing input difficulty consistent across conditions. The equirectangular frame preview has a size of 320 × 180 in pixels, consistent with the YouTube player, as of January 2021.

NFOV-strip. Employing the row-of-thumbnail design introduced in Sec 3.2, this interface is identical to the temporal mode of Tapestry Player with one exception: when the user's pointer hovers on the timeline, two rows of evenly-sampled NFOV thumbnails, one for the left side of the camera route and one for the right side, are displayed instead of Route Tapestries (Figure 10, bottom). The individual thumbnails were obtained by evenly sampling the source videos and converting the sampled 360° frames to perspective images through two virtual NFOV cameras. One camera was turned 90° counterclockwise from the camera forward direction, and the other turned 90° clockwise. The two virtual cameras were further rotated 15° upwards around the pitch axes to avoid capturing the roof of the car. The NFOV strips were of the same dimension as the Route Tapestries for each individual video. The aspect ratio and horizontal FOV of the individual thumbnails resembles the NFOV image as seen in the video window (16:9, 90°). The NFOV thumbnails for the three study video are sampled every 3.57s, 3.87s, and 3.88s, respectively.

Figure 10: The two baseline interfaces used in the study: *equirectangular* on the top, and *NFOV-strip* on the bottom. Both sub-figures show the interfaces during scrubbing.

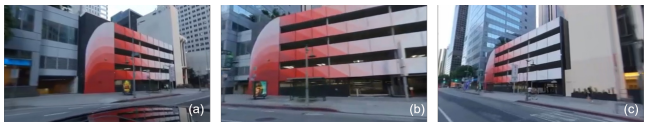

Figure 11: Each target scene was presented to participants via three NFOV screenshots from the task video. These pictures showed the target as seen from the its left (*a*), center (*b*), and right (*c*) side for an observer facing it.

6.3 Procedure and Task

Due to COVID-19 safety measures, we conducted this study as a remote, online study. Participants completed the study on their personal computers with the experimenter connected via a video chat connection. The participants downloaded the study software from an online repository, and instructions were delivered via pre-recorded video tutorials to ensure instruction quality.

The study began with the participants watching an overview tutorial explaining study tasks. They then completed a demographics questionnaire. In the main body of the study, participants completed two practice and 14 timed trials of the study task for each of the three *interface* conditions. Participants used the practice trials to familiarize themselves with the interface on a separate video not used for any timed trials. In each trial of the target-finding task, participants were to locate a target within the 360° video using the given player interface. Each target is a segment of the urban landscapes, i.e., one or more buildings, that appeared in the video. In each trial, they were first presented with three pictures depicting the target, as shown in Figure. 11. They were instructed to take

time studying these pictures before clicking a 'start' button to begin the timed trial. During the trial, they could see the target pictures in a sidebar. The participant needed to navigate to the part of the video where the camera passed by the target buildings, adjust the viewing direction to face them and press the space key to confirm the selection. When the trial was completed, or a 3-minute timeout counter was reached, the software would show the next target.

We recorded each trial's task completion time, the number of timeouts, pointer traces, and viewing direction changes within the 360° videos.

6.4 Video and Target Selection

We chose three similar 360° virtual tour videos as the study materials from YouTube (plus one for training)[5]. The videos were captured from a car driving through an urban landscape. Each was edited to be 15 minutes long.

We carefully selected 14 target scenes from each video to include a variety of targets while keeping the target sets comparable across the three videos in terms of the scene lengths and positions in the videos. All videos have an equal number of targets (7) chosen on the left and right sides of the camera travel path. We identified a "valid segment" of the video for each target for when we would consider the task to be completed. These would vary in temporal duration but would begin when the closer edge of the target appears 45° to the left or right of the camera moving direction, and then ends when the nearer edge of the target is at about 135° to the left or right of the camera forward direction. We chose the positions of the target in the video and the length of their valid segments to cover a wide range of target types meanwhile following comparable distributions across the three videos (Figure 12). All targets are at least partially visible in the NFOV-strip thumbnails.

Figure 12: Distribution of targets in the videos and sample targets used in the study. The lengths of the markers corresponds to the duration when the targets are close enough to the camera. The colors of the markers denote whether they are to the left or the right of the camera path. The right two columns show the average valid segment length and temporal position of the targets for each video used in the study. See the footnote in Sec. 6.4 for the video sources.

[5]Video sources: https://youtu.be/kdGlselFto0 (video 1 and 2), https://youtu.be/rIkV_bKLvSE (video 3), https://youtu.be/2Lq86MKesG4 (training)

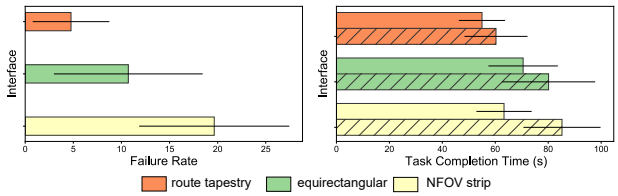

Figure 13: Average task failure rate (*left*) and task completion time (*right*) by interface. The solid bars show the average time if failure trials are excluded. The hatched bars show the average time if the completion time for all failure trials are considered to be the trial time limit (180s). Error bars represent 0.95 confidence interval.

6.5 Participants

We recruited 12 paid participants (3 females, 9 males, M_{age} = 25.1, SD_{age} = 2.6) to take part in our online study. Each participated in the study individually from their homes using their computers. The majority of the participants had no prior experience with virtual reality (9 out of 12) or 360° videos (8 out of 12). Two participants were experienced 360° video and virtual reality content consumers.

6.6 Apparatus

The participants completed the study with their personal computers and used an external mouse as the input device. To control the effect of display size on performance, we asked participants to run the study software in the full-screen mode on a 13"-16" display (M_{size} = 14.5, SD_{size} = 1.1). All the computers that the participants used for the study met the requirements of the study software.

7 EVALUATION RESULTS AND ANALYSIS

We compared the task performance of the three *interfaces* in terms of failure rate and completion time. Additionally, we analyzed the participants' input traces to understand their usage patterns and the potential causes of the performance differences.

7.1 Task Performance

We computed average failure rate and task completion time to compare the participants' performance using the three *interfaces*. We computed two types of task completion time: one with only the successful trials, and one with all the trails in which the failure trails were assigned with the maximum time allowed (180s). Overall, there were 59 out of 504 (11.7%) failure trials.

7.1.1 Failure Rate. Overall, *NFOV-strip* had the most failure trials (33, 19.6%), followed by *equirectangular* (18, 10.7%) and then *Route Tapestry* (8, 4.8%). A Friedman test on failure rate found a significant difference between *interfaces* (χ^2 = 10.7, $p < 0.01$). Post-hoc pairwise comparison using paired Wilcoxon tests showed that *Route Tapestry* (M = 4.8%, SD = 7.0%) had significantly lower failure rate than *NFOV-strip* (M = 19.6%, SD = 13.7%). On average, the failure rate of *Route Tapestry* was about the half as *equirectangular*, and the *equirectangular* condition had about the half of the failure rate of *NFOV-strip* (Figure 13, left).

7.1.2 Task Completion Time. A repeated measures ANOVA (RM-ANOVA) on task completion time using only success trials showed a significant difference between the three *interfaces* ($F_{2,22} = 6.81, p < 0.01$). Post-hoc tests[6] showed that *Route Tapestry* was significantly faster than *equirectangular*. The participants took 22.0% less time completing the tasks using *Route Tapestry* ($M = 55.0s$, $SD = 15.4s$) than *equirectangular* ($M = 70.5s$, $SD = 23.2s$). No significant differences were found between *NFOV-strip* ($M = 63.3s$, $SD = 18.4s$) and the other two *interfaces* (Figure 13, right, the solid bars). Note that the task completion time calculated without failure trials underestimates the *actual* time needed to find the targets. Therefore we assign the maximum time allowed (180s) to failure trials and recompute the average completion time to get a conservative estimation for the actual time needed for the tasks. For this metric, a RM-ANOVA showed a significant difference between the three *interfaces* ($F_{2,22} = 10.25, p < 0.01$). Post-hoc tests found that *Route Tapestry* would take significantly less time than both *NFOV-strip* and *equirectangular*. Overall, the lower bound of the actual task completion time for *Route Tapestry* ($M = 60.3s$, $SD = 21.1s$) is 29.3% and 24.8% lower than *NFOV-strip* ($M = 85.2s$, $SD = 25.7s$) and *equirectangular* ($M = 80.1s$, $SD = 31.1s$), respectively (Figure 13, right, the hatched bars).

7.2 Analysis of Participants' Interaction Traces

Using pointer movement and click records, we further explored how participants reoriented the viewing angles, and how often they chose to use the smaller preview by hovering over the timeline versus to use the large preview by dragging the slider. This helps us to understand the usage patterns and potential performance-influencing factors.

7.2.1 Viewing Direction Change. As the participants needed to have the target scene in their field-of-view to complete a task, the difference in time spent on viewing direction manipulation might have contributed to the performance gap. The direction snapping feature of *Route Tapestry* and *NFOV-strip*—immediate reorientation when the user clicked on the timeline—might give them an advantage over *equirectangular*. To better understand this factor, we calculated the total time spent on camera manipulation in successful trials with the three *interfaces*.

We found that although the average viewing direction manipulation time per trial was shorter with *Route Tapestry* ($M = 2.05s$, $SD = 2.92s$) and *NFOV-strip* ($M = 3.10s$, $SD = 4.54s$) than with *equirectangular* ($M = 3.31s$, $SD = 3.84s$), it was not a major component in the full task completion time, and the proportions were similar for all three *interfaces* (3%-4%). RM-ANOVA and post-hoc pairwise tests still showed the same trends on task completion time with viewing direction manipulation time removed (RM-ANOVA: $F_{2,22} = 6.16, p < 0.01$). This suggests that the performance gaps were more likely due to differences in the time spent on temporal navigation between the *interfaces*.

7.2.2 Pointer Input Modality: Drag/Hover. With all three *interfaces*, the participants had the choice of hovering the pointer over the timeline to see the small frame preview or dragging the slider to see the large preview. We tracked pointer hovering and dragging traces on the timelines separately every 100ms and used this data to study which modality the participants decided to employ with different *interfaces*. Our analysis here explores all the data collected in the study, including trials that ended with a timeout.

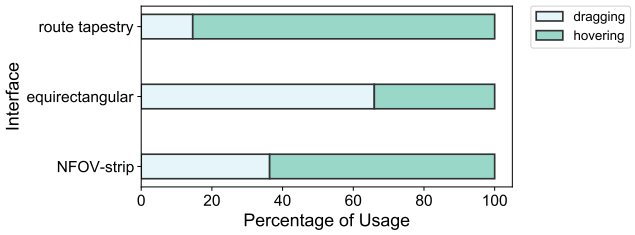

Figure 14: Percentage of time of pointer hovering vs pointer dragging on the timelines, by interface.

We found very different patterns between the three *interfaces* (Figure 14): hovering was dominating with *Route Tapestry* (85.4%), less common with *NFOV-strip* (63.7%), but only used about one third of the time for *equirectangular* (34.0%). During the study, some participants commented on the physical fatigue due to dragging and their stronger preference for hovering. The modality choices suggested that in comparison to *Route Tapestry*, the hover-triggered previews were not as useful for the participants when using *NFOV-strip* and even less so with *equirectangular*, driving them to use dragging, a potentially more physically-demanding modality [16].

8 DISCUSSION

Based on the analysis of the user performance and interaction patterns of Route Tapestries and the two baselines, we discuss the further implication of the study results, limitations of this work, and a few promising future directions.

8.1 Study Results

Overall, the study results show that for target-finding tasks in 360° virtual tour videos, participants were faster and completed more tasks using *Route Tapestry* compared to the two baselines that used equirectangular and NFOV-strip previews. Notably, in comparison to Route Tapestry, the participants took on average 28.2% more time completing the tasks with equirectangular and missed three times more targets with NFOV-strip. The results confirmed the benefits of reducing visual distortion and maintaining visual continuity when creating previews for 360° virtual tour videos.

8.1.1 The Benefits of Less Distorted Previews. For around two-thirds of their task time in the equirectangular condition, the participants dragged the timeline slider to activate the large preview, in contrast to only 14.6% with Route Tapestries and 36.3% with NFOV strips. Since dragging is in general considered as a more physically demanding input mode [16], we believe that the participants made the conscious choice to use it instead of hovering in the equirectangular condition, likely because small equirectangular previews were insufficient for target finding. Even with the large equirectangular previews, which were four times larger than the visible part of the Route Tapestries, the participants still spent on average 28.2%

[6] All post-hoc tests for RM-ANOVA used paired t-test with Bonferroni-Holm correction.

longer time to complete the tasks than with Route Tapestries and missed twice as many targets. This significant performance gap highlighted that the less distorted Route Tapestry previews could help users scan scenes and recognize targets more efficiently and reliably in 360° virtual tour videos.

8.1.2 The Benefits of Continuous Previews. As the results suggested, converting equirectangular previews to other less distorted formats, such as NFOV thumbnail strips, does not guarantee improvement in navigation performance. Despite visual distortion, equirectangular previews are projected from the full 360° video frame and contain a considerable amount of continuous visual context around the camera position. The high failure rate (around 20%) of NFOV thumbnail strips suggests that discrete visual contexts, albeit distortion-free, may impede rather than assist navigating 360° virtual tour videos. Since all the targets were at least partially visible in the thumbnails, the high failure rate indicated a difficulty in recognizing the targets from the thumbnails. The discontinuity could reduce the effectiveness of the previews in two aspects. First, users need to mentally connect the discrete images and imagine the missing parts to make sense. This additional effort slows down browsing and makes recognizing targets, especially those without distinct colors or patterns, more challenging. Second, distinctive parts of the target may be left out due to the gaps between the thumbnails.

8.2 Limitations and Opportunities

Our design, implementation, and evaluation of Route Tapestries were subject to some limitations.

Navigation through Route Tapestries requires that the scenes of interest form a visually dominant and mostly continuous boundary along the path of a moving camera. While the content of many popular 360° virtual tour videos meet these assumptions, they do not hold for static shots, e.g. narrator speaking, or moving shots in large open spaces, e.g., a soccer field. Future work could explore means to detect whether and where Route Tapestries are applicable to a video. Algorithmic methods could use optical flow algorithms on 360° images (e.g. [2]) to locate static shots and use scene recognition algorithms [53, 55] to identify open spaces. Other video summarization techniques, such as those visualizing changes [8, 12] or intelligently selecting thumbnails [46] for static shots, could then be applied to these isolated segments. Alternatively, video uploading interfaces could include an annotation tool for video creators to specify where to apply Route Tapestries and other techniques.

We noted that browsing Route Tapestries generated from very long videos throughout for interesting scenes may still be time-consuming. Inspired by prior work on video navigation through maps [33], we believe an opportunity to improve further the efficiency of 360° virtual tour video navigation is to annotate the Tapestry timeline with semantic labels such as building boundaries or types to enable efficient focus+context search [11].

In this work we specifically focused on the task of navigating 360° virtual tour videos. While we expect this would be a task that users commonly perform, as they do with NFOV videos [25, 35], we do not have empirical data about the prominence of navigation in interactions with 360° virtual tour videos. We also do not yet know how our novel interface design could change users' navigation patterns. Our future work would study these behaviors in more realistic settings, such as observing users' navigation behaviors when they watch tour videos to prepare for future trips. Additionally, we did not access the capacity of Route Tapestries for video skimming, i.e. efficiently learning the gist of long videos.

In the evaluation, we chose videos with comparable content and target location distributions, but the differences between them cannot be precisely controlled due to the nature of real-world virtual tour videos. Future work can apply computer-generated, high-resolution videos for more precise control over the video content.

8.3 Future Work

Our exploration of Route Tapestries suggests several exciting new design opportunities.

8.3.1 Route Tapestries for HMDs. Watching 360° virtual tour videos in head-mounted displays (HMDs) offers a strong sense of immersion and presence. The current 360° video player interfaces for HMDs are similar to their desktop versions and thus could have similar navigation issues. The design of the desktop Tapestry player can be applied to HMD with a few small modifications. To better match the immersive viewing experience, the Route Tapestries can be rendered in a circular rather than linear manner, and possibly surrounding the viewer. Alternatively, the Tapestries can be positioned to match spatial layout of the video scenes and promote spatial awareness.

8.3.2 Route Tapestries of Other Types of Videos. We are interested in extending the applicability of Route Tapestries to a wider range of 360° videos, such as drone videos, virtual tour videos shot in more complex environments, and videos with both static and moving shots.

Route Tapestries for 360° drone videos could be created by adding a downward-pointing virtual NFOV camera to our current algorithm. These Tapestries can potentially be used for both video navigation and exploring vast terrains.

We are particularly interested in using richer 3D and semantic information to enable the efficient navigation of 360° virtual tour videos that capture complex environments beyond those with clear boundaries along the path. Such videos include not only a wider range of videos for leisure purposes but also recordings from body-worn 360° cameras equipped by first responders exploring the fields [17]. With more accurate depth and semantic information, objects which the camera passes by can be segmented and clustered based on their distances to the camera. Each object cluster can be rendered as a single Route Tapestry, then stacked together according to their depths to form a composite Tapestry. When the user explore this Tapestry, individual layers can move with parallax in response to cursor movements, resembling the parallax scrolling effect seen in 2D video games. Further, with more efficient and accurate 3D mesh reconstruction methods, Route Tapestries can be directly created with a moving orthographic camera that scans the environment.

Route Tapestries can be combined with other types of video content summaries to support navigating 360° videos with both static and moving shots. With salient objects or characters identified, static shots can also be visualized in a strip format, where

the background is shown as a wide-angle image and the character movement traces are visualized with arrows [12] or stroboscopic effects [8]. These strip visualizations can then be joined with Route Tapestries created from moving shots to form a linear narrative of the video.

9 CONCLUSION

In this paper, we proposed browsing and navigating 360° virtual tour videos through a continuous orthographic-perspective projection of the scenes along the camera route, which we call Route Tapestries. We presented an algorithm for generating Route Tapestries using the slit-scan imaging technique and the design and evaluation of Tapestry Player, a desktop 360° video player incorporating Route Tapestries timelines. We conducted a user study comparing Tapestry Player with two alternative baseline designs, which used equirectangular and row-of-thumbnail strip previews, with a target-finding task. The study results showed that the participants were more efficient and found more targets when using Tapestry Player, highlighting the benefits of reducing visual distortion and maintaining continuous visual contexts for navigating 360° virtual tour videos.

We hope Route Tapestries can inspire 360° video player designs that make virtual tours more enjoyable for people who want to visit remote places but choose not to do so physically because of time, health, cost, or other reasons. In our future work, we plan to extend the Route Tapestry approach for navigating a wider variety of 360° videos and explore Route Tapestries for HMDs.

ACKNOWLEDGMENTS

This research was supported in part by the National Sciences and Engineering Research Council of Canada (NSERC) under Grant IRCPJ 545100-18, RGPIN-2017-06415, and DGDND-2017-00093. We thank Rahul Arora and Wenzheng Chen for valuable discussions.

REFERENCES

[1] Aseem Agarwala, Maneesh Agrawala, Michael Cohen, David Salesin, and Richard Szeliski. 2006. Photographing long scenes with multi-viewpoint panoramas. In *ACM SIGGRAPH 2006 Papers*. 853–861.

[2] Luigi Bagnato, Pascal Frossard, and Pierre Vandergheynst. 2009. Optical flow and depth from motion for omnidirectional images using a tv-l1 variational framework on graphs. In *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 1469–1472.

[3] Werner Bailer, Christian Schober, and Georg Thallinger. 2006. Video Content Browsing Based on Iterative Feature Clustering for Rushes Exploitation.. In *TRECVID*. Citeseer.

[4] Connelly Barnes, Dan B Goldman, Eli Shechtman, and Adam Finkelstein. 2010. Video tapestries with continuous temporal zoom. In *ACM SIGGRAPH 2010 papers*. 1–9.

[5] Wutthigrai Boonsuk, Stephen Gilbert, and Jonathan Kelly. 2012. The impact of three interfaces for 360-degree video on spatial cognition. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2579–2588.

[6] Matthew Brown and David G Lowe. 2007. Automatic panoramic image stitching using invariant features. *International journal of computer vision* 74, 1 (2007), 59–73.

[7] Seunghoon Cha, Jungjin Lee, Seunghwa Jeong, Younghui Kim, and Junyong Noh. 2020. Enhanced Interactive 360° Viewing via Automatic Guidance. *ACM Transactions on Graphics (TOG)* 39, 5 (2020), 1–15.

[8] Carlos D. Correa and Kwan-Liu Ma. 2010. Dynamic Video Narratives. *ACM Trans. Graph.* 29, 4, Article 88 (July 2010), 9 pages. https://doi.org/10.1145/1778765.1778825

[9] Anastasios D Doulamis and Nikolaos D Doulamis. 2004. Optimal content-based video decomposition for interactive video navigation. *IEEE Transactions on Circuits and Systems for Video Technology* 14, 6 (2004), 757–775.

[10] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. 2008. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 237–246.

[11] Susan Dumais, Edward Cutrell, and Hao Chen. 2001. Optimizing search by showing results in context. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 277–284.

[12] Dan B Goldman, Brian Curless, David Salesin, and Steven M Seitz. 2006. Schematic storyboarding for video visualization and editing. *Acm transactions on graphics (tog)* 25, 3 (2006), 862–871.

[13] Jeremy Hartmann, Stephen Diverdi, Cuong Nguyen, and Daniel Vogel. 2020. View-Dependent Effects for 360° Virtual Reality Video. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology - UIST '20*. ACM. https://doi.org/10.1145/3379337.3415846

[14] Wolfgang Hürst. 2006. Interactive audio-visual video browsing. In *Proceedings of the 14th ACM international conference on Multimedia*. 675–678.

[15] Dan Jackson, James Nicholson, Gerrit Stoeckigt, Rebecca Wrobel, Anja Thieme, and Patrick Olivier. 2013. Panopticon: A parallel video overview system. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 123–130.

[16] Peter W Johnson, Steven L Lehman, and David M Rempel. 1996. Measuring muscle fatigue during computer mouse use. In *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vol. 4. IEEE, 1454–1455.

[17] Brennan Jones, Anthony Tang, Carman Neustaedter, Alissa N Antle, and Elgin-Skye McLaren. 2020. Designing Technology for Shared Communication and Awareness in Wilderness Search and Rescue. In *HCI Outdoors: Theory, Design, Methods and Applications*. Springer, 175–194.

[18] Kyoungkook Kang and Sunghyun Cho. 2019. Interactive and automatic navigation for 360° video playback. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–11.

[19] Juho Kim, Philip J Guo, Carrie J Cai, Shang-Wen Li, Krzysztof Z Gajos, and Robert C Miller. 2014. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 563–572.

[20] Don Kimber, Jonathan Foote, and Surapong Lertsithichai. 2001. Flyabout: spatially indexed panoramic video. In *Proceedings of the ninth ACM international conference on Multimedia*. 339–347.

[21] Johannes Kopf, Billy Chen, Richard Szeliski, and Michael Cohen. 2010. Street slide: browsing street level imagery. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 1–8.

[22] Yung-Ta Lin, Yi-Chi Liao, Shan-Yuan Teng, Yi-Ju Chung, Liwei Chan, and Bing-Yu Chen. 2017. Outside-in: Visualizing out-of-sight regions-of-interest in a 360 video using spatial picture-in-picture previews. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 255–265.

[23] Sean J. Liu, Maneesh Agrawala, Stephen DiVerdi, and Aaron Hertzmann. 2019. View-Dependent Video Textures for 360° Video. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 249–262. https://doi.org/10.1145/3332165.3347887

[24] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2012. Swift: reducing the effects of latency in online video scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 637–646.

[25] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2013. Swifter: improved online video scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1159–1168.

[26] Michael Mills, Jonathan Cohen, and Yin Yin Wong. 1992. A magnifier tool for video data. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 93–98.

[27] Raul Mur-Artal and Juan D Tardós. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* 33, 5 (2017), 1255–1262.

[28] Luís AR Neng and Teresa Chambel. 2010. Get around 360 hypervideo. In *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*. 119–122.

[29] Cuong Nguyen, Stephen DiVerdi, Aaron Hertzmann, and Feng Liu. 2017. CollaVR: Collaborative in-headset review for VR video. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 267–277.

[30] Cuong Nguyen, Stephen DiVerdi, Aaron Hertzmann, and Feng Liu. 2017. Vremiere: in-headset virtual reality video editing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 5428–5438.

[31] Cuong Nguyen, Yuzhen Niu, and Feng Liu. 2012. Video summagator: an interface for video summarization and navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 647–650.

[32] Cuong Nguyen, Yuzhen Niu, and Feng Liu. 2013. Direct manipulation video navigation in 3D. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1169–1172.

[33] Gonçalo Noronha, Carlos Álvares, and Teresa Chambel. 2012. Sight surfers: 360° videos and maps navigation. In *Proceedings of the ACM multimedia 2012 workshop on Geotagging and its applications in multimedia*. 19–22.

[34] Michael Nunes, Saul Greenberg, Sheelagh Carpendale, and Carl Gutwin. 2007. What did I miss? Visualizing the past through video traces. In *ECSCW 2007*. Springer, 1–20.

[35] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2015. SceneSkim: Searching and Browsing Movies Using Synchronized Captions, Scripts and Plot Summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface and Software Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/2807442.2807502

[36] Amy Pavel, Björn Hartmann, and Maneesh Agrawala. 2017. Shot orientation controls for interactive cinematography with 360 video. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 289–297.

[37] Shmuel Peleg, Benny Rousso, Alex Rav-Acha, and Assaf Zomet. 2000. Mosaicing on adaptive manifolds. *IEEE Transactions on pattern analysis and machine intelligence* 22, 10 (2000), 1144–1154.

[38] Benjamin Petry and Jochen Huber. 2015. Towards effective interaction with omnidirectional videos using immersive virtual reality headsets. In *Proceedings of the 6th Augmented Human International Conference*. 217–218.

[39] Suporn Pongnumkul, Jue Wang, Gonzalo Ramos, and Michael Cohen. 2010. Content-aware dynamic timeline for video browsing. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 139–142.

[40] Gonzalo Ramos and Ravin Balakrishnan. 2003. Fluid interaction techniques for the control and annotation of digital video. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*. 105–114.

[41] Augusto Roman, Gaurav Garg, and Marc Levoy. 2004. Interactive design of multi-perspective images for visualizing urban landscapes. In *IEEE visualization 2004*. IEEE, 537–544.

[42] A. Romanoni and M. Matteucci. 2015. Incremental reconstruction of urban environments by Edge-Points Delaunay triangulation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4473–4479. https://doi.org/10.1109/IROS.2015.7354012

[43] Gustavo Alberto Rovelo Ruiz, Davy Vanacken, Kris Luyten, Francisco Abad, and Emilio Camahort. 2014. Multi-viewer gesture-based interaction for omni-directional video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 4077–4086.

[44] Klaus Schoeffmann, Mario Taschwer, and Laszlo Boeszoermenyi. 2010. The video explorer: a tool for navigation and searching within a single video based on fast content analysis. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*. 247–258.

[45] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. 2018. Saliency in VR: How do people explore virtual environments? *IEEE transactions on visualization and computer graphics* 24, 4 (2018), 1633–1642.

[46] Yu-Chuan Su and Kristen Grauman. 2017. Making 360 video watchable in 2d: Learning videography for click free viewing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1368–1376.

[47] Yu-Chuan Su, Dinesh Jayaraman, and Kristen Grauman. 2016. Pano2Vid: Automatic Cinematography for Watching 360 Videos. In *Asian Conference on Computer Vision*. Springer, 154–171.

[48] Richard Szeliski. 2006. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision* 2, 1 (2006), 1–104.

[49] Anthony Tang, Saul Greenberg, and Sidney Fels. 2008. Exploring video streams using slit-tear visualizations. In *Proceedings of the working conference on Advanced visual interfaces*. 191–198.

[50] Yukinobu Taniguchi, Akihito Akutsu, and Yoshinobu Tonomura. 1997. PanoramaExcerpts: extracting and packing panoramas for video browsing. In *Proceedings of the fifth ACM international conference on Multimedia*. 427–436.

[51] Anh Truong and Maneesh Agrawala. 2019. A Tool for Navigating and Editing 360 Video of Social Conversations into Shareable Highlights.. In *Graphics Interface*. 14–1.

[52] Anh Truong, Sara Chen, Ersin Yumer, David Salesin, and Wilmot Li. 2018. Extracting regular fov shots from 360 event footage. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.

[53] Lin Xie, Feifei Lee, Li Liu, Koji Kotani, and Qiu Chen. 2020. Scene recognition: A comprehensive survey. *Pattern Recognition* 102 (2020), 107205.

[54] Jiang Yu Zheng. 2003. Digital route panoramas. *IEEE MultiMedia* 10, 3 (2003), 57–67.

[55] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning Deep Features for Scene Recognition using Places Database. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/3fe94a002317b5f9259f82690aeea4cd-Paper.pdf

[56] Assaf Zomet, Anat Levin, Shmuel Peleg, and Yair Weiss. 2006. Seamless image stitching by minimizing false edges. *IEEE transactions on image processing* 15, 4 (2006), 969–977.

A ROUTE TAPESTRY EXAMPLES

Please see the next two pages.

Figure 15: Tour of a Moebius exhibition in a gallery. From https://youtu.be/YkvLXkVZtlc

Figure 16: Driving tour in Los Angeles 1. From https://youtu.be/rIkV_bKLvSE

Figure 17: Driving tour in Los Angeles 2. From https://youtu.be/kdGlselFto0

Figure 18: Cycling tour in the alleys in Harajuku, Tokyo. From https://youtu.be/ZBTXyiTrARQ

Figure 19: Campus tour of the Harvard Student Center. From https://youtu.be/nFn2_a10O3o

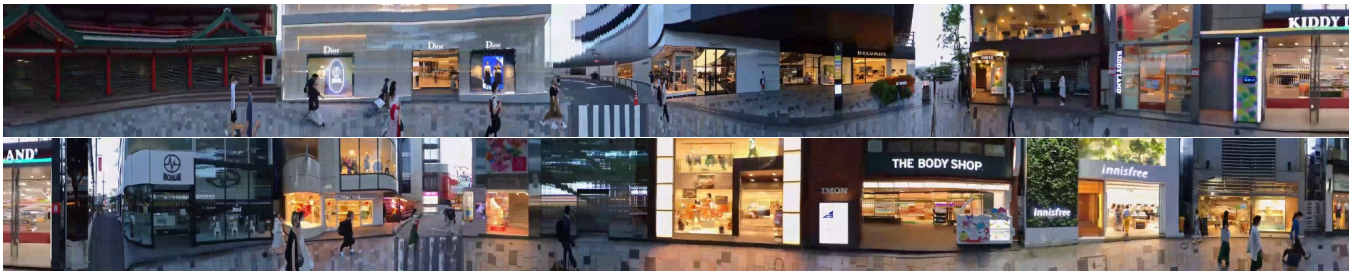

Figure 20: Cycling tour in Omotesandō, Tokyo. From https://youtu.be/ZBTXyiTrARQ

Figure 21: Driving tour in Los Angeles 3. From https://youtu.be/kdGlselFto0