Evaluating Snapping-to-Photos Virtual Travel Interfaces for 3D Reconstructed Visual Reality

Benjamin Nuernberger
University of California, Santa Barbara
bnuernberger@cs.ucsb.edu

Matthew Turk
University of California, Santa Barbara
mturk@cs.ucsb.edu

Tobias Höllerer
University of California, Santa Barbara
holl@cs.ucsb.edu

(a) Before POI snapping. (b) Photo after POI snapping. (c) Before POV snapping. (d) Photo after POV snapping.

Figure 1: Point-of-interest (POI) snapping (a-b) allows the user to snap the viewpoint to a particular photo in the scene based on both the current view and the user's mouse cursor or finger touch position; point-of-view (POV) snapping (c-d) allows users to snap the viewpoint to the photo that is most similar to the current viewpoint, either automatically or upon clicking or touching. The white border visualizations (a,c) indicate the extent of the photo to be snapped to.

ABSTRACT

Navigating through a virtual, 3D reconstructed scene has recently become very important in many applications. A popular approach is to virtually travel to the photos used in reconstructing the scene; such an approach may be generally termed a "snapping-to-photos" virtual travel interface. While previous work has either used fully constrained interfaces (always at the photos) or minimally constrained interfaces (free-flight navigation), in this paper we introduce new snapping-to-photos interfaces that lie in between these two extremes. Our snapping-to-photos interfaces snap the view to a photo in 3D based on viewpoint similarity and optionally the user's mouse cursor or finger-tap position. Experimental results, with both indoor and outdoor scene reconstructions, found that our snapping-to-photos interfaces are preferred over the baseline fully constrained-to-photos interface, that there exist differences between indoor and outdoor scenes, and that users preferred and were able to reach target photos better with click-to-snap point-of-interest snapping compared to automatic point-of-view snapping.

CCS CONCEPTS

• **Human-centered computing** → **Interaction paradigms**;

KEYWORDS

Virtual navigation, 3D user interfaces, user experiments

VRST '17, November 8–10, 2017, Gothenburg, Sweden

ACM ISBN 978-1-4503-5548-3/17/11…$15.00
https://doi.org/10.1145/3139131.3139138

ACM Reference Format:
Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2017. Evaluating Snapping-to-Photos Virtual Travel Interfaces for 3D Reconstructed Visual Reality. In *Proceedings of VRST '17*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3139131.3139138

1 INTRODUCTION

Virtual navigation of photo-captured visual reality has been a goal for many years [Chen 1995; Lippman 1980], with applications including virtual tourism [Snavely et al. 2006], street-level exploration [Anguelov et al. 2010; Kopf et al. 2010], and augmented reality (AR) remote collaboration [Gauglitz et al. 2014b]. Such applications also demonstrate the variety of possible ways to capture visual reality, including crowd-sourced approaches [Snavely et al. 2006; Tuite et al. 2011], driving expensive camera rigs around [Anguelov et al. 2010], and using handheld camera devices for one-on-one remote collaboration [Gauglitz et al. 2014b]. While much research has focused on how to create 3D models from captured imagery, less attention has been devoted to the user interface question of how to optimally virtually navigate through such captured scenes.

Of particular interest is how to support virtual navigation through *emerging* reconstructed scenes. At the beginning, 3D reconstructed models of particular areas may be sparse and very incomplete, yet over time the density and completeness of the model will grow. In other cases, users may want to virtually travel between densely captured areas and ones that have only partially been captured. Having the ability to virtually navigate through any of the models will enable many applications. For example, remote users will now have the ability to virtually navigate and place AR annotations in the world for in-situ users with AR devices to utilize. Situations where such emerging models are especially important include emergency response (*e.g.*, search & rescue, or coordination to extinguish a wildfire) and 1-on-1 remote collaboration. The long-term question

to enable this interaction is: Is there an interface that can richly support such emerging 3D reconstructions of visual reality?

In the popular virtual navigation of captured visual reality systems today, the user's viewpoint is constrained to be at the high resolution input photographs that were used to build the underlying 3D reconstruction [Anguelov et al. 2010; Brivio et al. 2013; Kushal et al. 2012; Snavely et al. 2006]. To move between photos, image-based rendering transitions can be used [Goesele et al. 2010; Kushal et al. 2012; Snavely et al. 2008, 2006; Tompkin et al. 2012], helping to prevent spatial disorientation [Tompkin et al. 2012]. Popular examples of this *fully constrained-to-photos* approach include Google Street View [Anguelov et al. 2010] and Matterport[1]. In both of these examples, the scenes usually have evenly spaced 360° panoramas that are fairly simple to move between and the movement is usually as expected.

The major downside is that creating scenes for these systems, however, typically require expensive camera rigs, advanced user training, and/or extensive acquisition times. Thus, some (*e.g.*, Matterport) are beginning to use cameras on mobile devices (*e.g.*, smartphones, wearables, etc.) which are cheap, require little training, and are quick to use. Furthermore, in order to realize remote virtual navigation of captured visual reality in a large scale fashion (both spatially and temporally, and in emerging models), mobile cameras will inevitably be used to create navigable 3D reconstructions of visual reality (*e.g.*, from crowd-sourcing [Snavely et al. 2006; Tompkin et al. 2012; Tuite et al. 2011], micro aerial vehicles, and even AR remote collaboration [Gauglitz et al. 2014b]).

However, this latter type of scene capture is typically done in a more unstructured fashion, in which case users of the aforementioned constrained-to-photos interfaces have less confidence and more difficulty virtually navigating (see Sec. 4 and Fig. 4). Specifically, because photos are now arranged in an unstructured way, in 6 degrees-of-freedom (DoF), constrained-to-photos travel can easily cause movement in unexpected, non-intuitive ways.

In this paper, we investigate less constrained approaches to visit photos in 3D. We term this general virtual travel paradigm of visiting photos in 3D as "snapping-to-photos." Our less constrained snapping-to-photos interfaces can use any basic virtual travel interface as their basis and then attempt to snap the view to a photo in 3D based on viewpoint similarity and optionally the user's mouse cursor or finger-tap position. Point-of-interest (POI) snapping allows the user to snap the viewpoint to a particular photo in the scene based on both the current view and the user's mouse cursor or finger touch position; point-of-view (POV) snapping allows users to snap the viewpoint to the photo that is most similar to the current viewpoint, either automatically or upon clicking or touching (see Figure 1). While our snapping-to-photos interfaces are a general technique usable for any type of 3D reconstruction, they are especially suitable for scenes captured in an unstructured manner. Our contributions include:

(1) A set of novel snapping-to-photos virtual navigation interfaces for exploring photos in 3D image-based reconstructed scenes, with both POI and POV snapping.
(2) Two experimental evaluations of snapping-to-photos interfaces, using both indoor and outdoor datasets.

[1]https://matterport.com/

2 RELATED WORK

2.1 General Virtual Navigation

Virtual navigation has been categorized in different ways over the years [Bowman et al. 2004; Christie et al. 2008; Jankowski and Hachet 2015]. Generally, it is composed of travel and wayfinding, the motor and cognitive components of navigation. It can also be categorized based on the task: exploration, search, and maneuvering [Bowman et al. 2004]. While most research has focused on virtually navigating synthetic scenes, Jankowski and Hachet refer to the exploration of photos in 3D image-based reconstructions as the "Exploration of joint 2D and 3D data" [Jankowski and Hachet 2015]; our work falls into this category.

Snapping-to-photos approaches can be related both to virtual navigation with potential fields [Beckhaus et al. 2001; Tanaka et al. 2016] and 3D viewpoint bookmarks [Benedetto et al. 2014; Elvins et al. 2001; Forgione et al. 2016]. Potential fields offer a way to guide users in virtual navigation [Beckhaus et al. 2001; Tanaka et al. 2016]; snapping-to-photos can be thought of as applying a potential field that attracts the viewpoint to photos in the scene. Moving between discrete 3D viewpoint bookmarks is very similar to visiting photos via snapping; in synthetic scenes, such 3D viewpoint bookmarks have been shown to be beneficial for both system and user navigation performance [Benedetto et al. 2014; Elvins et al. 2001; Forgione et al. 2016].

2.2 Snapping-to-Photos Virtual Navigation

Image-based reconstructions present several challenges compared to synthetic scenes for virtual navigation. Unlike synthetic scenes, 3D reconstructed scenes are typically incomplete and/or inaccurate, due to unobserved space and observed but poorly reconstructed space. For example, a scene that has much occlusion will not be fully captured unless it has been observed from many viewpoints. In addition, for dynamic scenes or scenes captured at different times (*e.g.*, crowd-sourced over many days), it is not clear how to represent the scene with a single 3D model. For these reasons, directly applying interfaces made for navigating synthetic scenes may not be sufficient [Bowman et al. 2004; Christie et al. 2008; Jankowski and Hachet 2015]. One straightforward and promising approach to alleviate these issues is to show the photos that were used for the reconstruction. Showing a photo inherently allows for a ground truth rendering of the scene from that particular viewpoint. Such snapping-to-photos approaches have been used both for scenes captured in structured ways and unstructured ways.

2.2.1 Structured Captures. Early work, such as the Aspen Movie-Maps [Lippman 1980] and QuickTime VR [Chen 1995], essentially used a very structured capture process and thus had a very structured virtual navigation experience, moving between photos of the scene. More recently, Google Street View [Anguelov et al. 2010] and Matterport have also used structured captures of visual reality. A recent "technical preview" edition of Microsoft Photosynth[2] introduced several types of constrained navigation where the scene was captured in a specific manner in order to achieve a particular constrained navigation experience (*e.g.*, orbits, linear trajectories, etc.).

[2]Photosynth development & support ended on February 6, 2017, see https://blogs.msdn.microsoft.com/photosynth/2017/02/06/microsoft-photosynth-has-been-shut-down/.

In this paper, we focus on the more general case of unstructured captures, where a single user may capture the scene with a handheld device, or multiple users may capture the scene in a crowd-sourced fashion [Tuite et al. 2011]; virtually navigating through such unstructured captured scenes is becoming very important, especially for emerging scenes.

2.2.2 Unstructured Captures. A major step forward in navigating unstructured captures of visual reality was the Photo Tourism work [Snavely et al. 2008, 2006], in which users could either move freely and click on camera frusta visualizations to visit photos, or be constrained to move between photos of the scene. Microsoft's Photosynth product was a direct result of Snavely *et al.*'s work, and several others have made similar interfaces using either a *free-flight plus clicking on camera frusta visualizations to visit photos* interface or a *constrained-to-photos* interface [Brivio et al. 2013; Furukawa et al. 2009; Gauglitz et al. 2014a,b; Nuernberger et al. 2016; Xiao and Furukawa 2014][3]. Photosynth is fully constrained-to-photos, whereas our interfaces are more general, less constrained, and compatible with any generic virtual travel interface. The baseline interface in Sec. 4 is similar to Photosynth.

However, these initial interfaces for visiting photos in unstructured captures are not always easy to use. Existing approaches either (a) fully constrain movement to be between photos ("constrained-to-photos") or (b) allow free-flight movement and require clicking on camera frusta visualizations to move to photos (see Figure 2). The problem with the former approach (a) is that usability issues occur when the virtual movement is disorienting and unexpected, as is common with limiting movement to always be at photos in unstructured scene captures (see Section 4). The problem with the latter approach (b) is that camera frusta visualizations are sometimes hard to understand and they visually clutter the scene. Brivio *et al.* [Brivio et al. 2013] specifically noted these problems and used a panel of thumbnail images as one approach to mitigate the clutter.

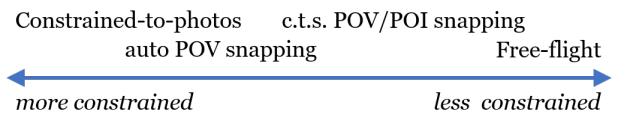


Figure 2: Snapping-to-photos interfaces can be more or less constrained. Previous methods lie at the extremities of this spectrum, whereas our click-to-snap (c.t.s) POV/POI snapping and automatic POV snapping lie in between.

2.2.3 Experimental Evaluations. For structured scene captures, Kopf *et al.* [Kopf et al. 2010] investigated street-side search tasks with a novel constrained interface called Street Slide. For unstructured scene captures, to our knowledge, only Brivio *et al.* [Brivio et al. 2013] have conducted a comparative evaluation of snapping-to-photos interfaces, comparing their PhotoCloud method against Photosynth. PhotoCloud can be considered a minimally constrained free-flight plus clicking on frusta visualizations approach. While they used only one outdoor dataset, our experimental evaluations used three indoor and four outdoor datasets. Furthermore, they used only move-to-photo tasks, whereas our experiments additionally included exploration and spatial knowledge tasks. Empirical testing determined that PhotoCloud's approach would introduce too much visual clutter (a problem Brivio *et al.* acknowledged) on indoor datasets and thus instead we focused on comparing our snapping-to-photos interfaces against a baseline closer to Photosynth's approach.

[3]In this paper we do not consider the case of navigating through videos of events [Ballan et al. 2010; Tompkin et al. 2012].

3 SNAPPING-TO-PHOTOS INTERFACES

Our snapping-to-photos interfaces allow users to quickly move (snap) to a particular photo in 3D. They take any virtual travel interface as a foundation and add the additional capability of moving to photos in 3D. For example, a walking or flying travel interface can be used for basic controls to move throughout the scene, and on top of this basic travel interface, snapping-to-photos allows easy and intuitive movement to photos placed in the scene. We implemented three base interfaces as described later in Sections 4, 5, and 6.

The main two questions for designing a snapping-to-photos interface are: (1) whether the desired movement is towards a point-of-interest (POI) or a point-of-view (POV); and (2) how to choose the desired photo to move to.

POI-based movement is focused on a scene entity (*e.g.*, a landmark of interest), whereas POV-based movement is focused on positioning and orienting the viewpoint. We present both of these types of movement as two flavors of snapping-to-photos virtual navigation. For example, POI snapping could help when the user is interested in a certain object in the scene, whereas POV snapping could help if the user wants to (virtually) stand at a certain point in the scene and look in a specific direction.

POI snapping occurs upon a single click or tap, while POV snapping can occur either automatically or upon a single click or tap. For automatic snapping, users need to first pause movement with the basic travel interface. In our experiments, we used 0.5 seconds as the pausing threshold, determined empirically since shorter times can cause too much snapping and longer times can be unnecessary. For both POI and POV snapping, we use linear interpolation to adjust the position of the virtual camera and spherical linear interpolation for its orientation; snapping occurs in one second.

Choosing the photo to move to is based upon a viewpoint similarity cost function C. We first discuss POV snapping and its cost function C_{POV}, and then POI snapping and its cost function C_{POI}.

3.1 Point-of-view (POV) Snapping

POV snapping attempts to snap the viewpoint to the photo $\hat{p}$ that is most similar to the current viewpoint, chosen as follows:

$$\hat{p} = \underset{p \in \mathcal{P}_v}{\arg\min}\, C_{POV}(v, p) \quad (1)$$

where v is the current view and p is a photo in the set of photos $\mathcal{P}_v$ compatible with v. We used simple frustum overlap checks to determine compatibility; additional filtering can be applied as desired (*e.g.*, photos within a certain Euclidean distance are compatible).

We first implemented several existing viewpoint similarity cost functions for 3D image-based reconstructions[4] [Gauglitz et al. 2014b;

[4]In this work, we did not consider general synthetic view similarity cost functions that were created for synthetic scenes [Zhao and Ooi 2016; Zhu et al. 2011].

Kopf et al. 2014; Kushal et al. 2012; Snavely et al. 2008, 2006] and through empirical testing, we found that using an intersection-over-union (IOU) cost for scene overlap between v and p is extremely intuitive and helpful[5]. To our knowledge we are the first to use the IOU cost for scene overlap in a viewpoint similarity metric for virtual navigation. The cost function for POV snapping is:

$$C_{POV}(v,p) = \alpha_{pos}C_{pos}(v,p) + \alpha_{rot}C_{rot}(v,p) + \alpha_{IOU}C_{IOU}(v,p) \tag{2}$$

(1) **Translational Motion**. The first term represents a weighted and clamped Euclidean distance between the current view v and the photo p; let V and P be the 3D positions of the view and photo, respectively:

$$C_{pos}(v,p) = \min\left(1, \frac{||V-P||_2}{\beta_{pos}}\right) \tag{3}$$

where β_{pos} depends on the scene. We used β_{pos} = 5m for indoor scenes and = 50m for outdoor scenes.

(2) **Rotational Motion**. The second term is a weighted rotational difference between the view v and the photo p:

$$C_{rot}(v,p) = \frac{angDiff(v,p)}{\beta_{ang}} \tag{4}$$

where β_{ang} = 180°, and the $angDiff$ returns the angular orientation difference in degrees between a view and photo.

(3) **Scene Overlap**. The third term is a weighted and clamped intersection-over-union cost of the respective depth maps of the view and photo, D_v and D_p:

$$C_{IOU}(v,p) = \min\left(1, \max\left(0, 1 - \frac{D_v \cap reproj(D_p, v)}{D_v \cup reproj(D_p, v)}\right)\right) \tag{5}$$

where the function $reproj$ reprojects a depth map into the view v, taking occlusion into account. To favor photos that can see the center of the view v and to account for differences in the extrinsics and instrinics of the view and photo, each reprojected pixel x is reweighted as follows; let X be the 3D point corresponding to the reprojected pixel x:

$$x' = w(x)\frac{||X-V||}{||X-P||}\frac{p_{fov}}{v_{fov}} \tag{6}$$

where p_{fov} and v_{fov} are the fields of views for p and v, respectively, and $w(x)$ weights the contribution of this reprojected pixel to favor the center of the view; in our implementation, we used a simple 3x3 grid cell Gaussian-based weighting for w. In practice, we subsample the depth maps in order for the evaluation of C_{POV} to run in real-time; we sampled every 30px in all our experiments.

In all experiments, we used $\alpha_{pos} = 1$, $\alpha_{rot} = 1$, and $\alpha_{IOU} = 3$. If a photo was chosen as $\hat{p}$ in the previous frame, we re-weighted its cost in the evaluation of Eq. 1 in order to favor temporal consistency; in all experiments, the cost was multiplied by 0.5.

[5]The IOU cost, a.k.a. the Jaccard distance, is commonly used in evaluating image segmentation algorithms [Everingham et al. 2010].

3.2 Point-of-interest (POI) Snapping

POI snapping not only takes into account the current viewpoint v, but it also factors in the user's mouse cursor or finger-tap position on the screen u. Thus, the photo $\hat{p}$ to move to is:

$$\hat{p} = \arg\min_{p \in \mathcal{P}_v} C_{POI}(v,u,p) \tag{7}$$

where

$$\begin{aligned} C_{POI}(v,u,p) = \alpha_{pos}C_{pos}(v,p) + \alpha_{rot}C_{rot}(v,p) + \\ \alpha_{IOU}C'_{IOU}(v,u,p) + \alpha_u C_u(p,u) \end{aligned} \tag{8}$$

where C'_{IOU} is identical to C_{IOU} except its weighting function w' depends on whether the reprojected pixel x is inside a rectangular window R around the cursor position u:

$$w'(x) = \begin{cases} \beta_w \frac{Area(D_v)}{Area(R)} & \text{if } x \text{ is in R} \\ 0 & \text{if } x \text{ is not in R} \end{cases} \tag{9}$$

where β_w = 0.7 and R is 100x100 in our implementation.

To favor a point-of-interest, we factor in the distance to the 3D point U behind the 2D mouse cursor or finger-tap position u:

$$C_u(p,u) = \min\left(1, \frac{||P-U||_2}{\beta_u}\right) \tag{10}$$

where β_u = 5m for indoor scenes and = 50m for outdoor scenes.

In all experiments, $\alpha_u = 5$. As in POV snapping, POI snapping uses the same temporal consistency re-weighting for photo costs.

3.3 Visualizations

Similar to existing fully constrained-to-photos methods [Anguelov et al. 2010; Brivio et al. 2013], we use a simple white border visualization to represent the approximate extent of what a chosen photo $\hat{p}$ could see (see Figure 1). Borders appear instantly and are faded out over of a period of one second. We project the photo's border onto the 3D model or a proxy plane, depending on the scene. For dense indoor scenes, we visualize this white border directly on the 3D model. For outdoor scenes that include sky and ground that have not been modeled, we robustly fit a plane to the photo's depth map and show the white border on this plane. To account for any vanishing lines from the fitted plane, we visualize any missing part of the white border on a plane at infinity[6]. If the fitted plane is more than 60° away from the current viewing direction, we simply show the white border on the plane at infinity to avoid white border visualizations with large distortions. We used 4px as the border width in our experiments.

We also experimented with showing a virtual semi-transparent DSLR camera placed and oriented according to the chosen photo $\hat{p}$ (see example in supplemental material[7]). To indicate to the user that the photo is outside the field of view, we changed its color from black to red. Empirical testing showed that having both the white border visualizations and the DSLR cameras would likely cause too much visual clutter and distraction in many situations. Thus, for our experiments we only used the white border visualization.

[6]We used a stencil buffer to implement this.

[7]http://cs.ucsb.edu/~bnuernberger/vrst2017/

3.4 Implementation Results

In all experiments we used an Intel Core i7-4790 CPU running at 3.60GHz with 32GB of RAM, and an NVIDIA GeForce GTX 980 Ti GPU. Typical timings for snapping-to-photos are shown in Table 1.

Table 1: Typical average timings (ms) for evaluating viewpoint costs per photo (Eq. {2, 8}) & overall (Eq. {1, 7}), # photos evaluated per frame, photo resolution, & frames per second.

UI & Dataset	Photo	Overall	$\lvert\mathcal{P}_v\rvert$	Resolution	FPS
POV outdoor	0.044	3.85	86.09	2736x1824	30.56
POV indoor	0.064	25.39	398.07	640x480	27.40
POI outdoor	0.047	4.16	89.52	2736x1824	30.91
POI indoor	0.073	29.21	398.13	640x480	22.28

4 EXPERIMENT I

We compared a fully constrained-to-photos baseline interface with our less constrained, snapping-to-photos interfaces in both densely captured indoor and sparsely captured outdoor environments. We did not include free-flight navigation with clicking on camera frusta visualizations [Brivio et al. 2013; Snavely et al. 2006] as an interface condition, because empirical testing determined that the visual clutter from the frusta visualizations would make such an interface almost unusable.

4.1 Conditions

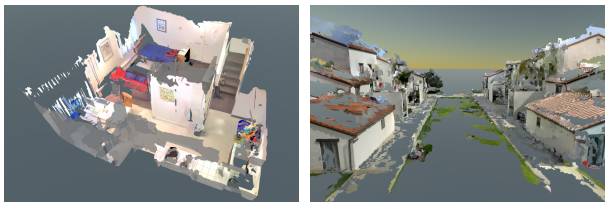
We had two main independent variables: scene and interface. Scene was a between-subjects factor with two levels of indoor and outdoor. Interface was a within-subjects factor with the following three levels (order counterbalanced):

Interface A: Fully constrained-to-photos (baseline).
Interface B: Click-to-snap POI Snapping.
Interface C: Click-to-snap POV Snapping.

4.1.1 Constrained-to-Photos (Baseline) Interface. We designed a constrained-to-photos interface as a baseline, inspired by Photosynth. In an attempt to be more fair in our comparison, we utilized the same equations from Sec. 3 to implement the baseline interface while restricting movement to always be at the photos. Right-click dragging causes rotation about the camera's optical center[8]. During rotation, we evaluate Eq. 1, and upon release of dragging, the view is automatically animated to the chosen photo $\hat{p}$. Mouse hovering causes Eq. 7 to be evaluated; a single left-click animates to the chosen photo $\hat{p}$. Finally, the scroll wheel is used to evaluate Eq. 1 at a potential future viewpoint v' that moves and orients the current view v in the direction of the ray cast by the mouse cursor position, forward or backward, depending on the direction of the scroll.

4.1.2 Basic Free-flight Interface for Interfaces B & C. For indoor scenes, left-click dragging causes the view to move parallel to the image plane while keeping the initial 3D point U always behind the cursor. Right-click dragging causes the view to rotate about the camera's optical center, and the mouse scroll wheel moves forward or backward along the camera's optical axis at a fixed speed.

[8]For all interfaces, only pitch and yaw rotation is allowed for usability purposes. Users could reverse the rotation direction as desired.



(a) An indoor dataset (571 photos). (b) An outdoor dataset (128 photos).

Figure 3: Two datasets used in the experiments.

For outdoor scenes, left-click dragging moves parallel to the ground such that dragging downward to the bottom of the screen moves the viewpoint to the 3D point first clicked on (cf. Drag'n Go [Moerman et al. 2012]), and dragging with both mouse buttons held is identical to left-click dragging for indoor scenes. We made this change in the basic interface for outdoor scenes because we realized that forward scrolling at a fixed speed put the POV snapping interface at a disadvantage for some tasks using large distances.

4.2 Procedure

Participants first completed pre-study questionnaires. For each condition, they were trained on the interface for at least three minutes; before the first condition, participants were also trained on the tasks. We utilized three reconstructed indoor scenes from BundleFusion [Dai et al. 2017] and textured with MVE [Fuhrmann et al. 2015; Waechter et al. 2014], and three outdoor scenes reconstructed with MVE [Fuhrmann et al. 2015; Waechter et al. 2014]. Figure 3 and the supplemental material give examples of the scenes. For the indoor scenes, we subsampled video frames by choosing the least blurry [Pertuz et al. 2013] of every 15 frames; for outdoor scenes, all photos were available for navigation. For all experiments, participants were seated in a quiet room and used a 2560x1440 resolution monitor; we limited the virtual navigation resolution to 1600x900 to run efficiently. The virtual view's field of view was 60°, while the indoor and outdoor photos' fields of view were approximately 44.76° and 52.46°.

Users completed the following three tasks for each interface, using three different scenes for each interface in order to avoid any bias in acquiring spatial knowledge:

(1) 1 Exploration task: five minutes of general exploration.
(2) 5 Direction Estimation tasks: given a prompted location and facing direction, users had to point to a target object in the scene, using an on-screen GUI.
(3) 4 Move-to-Photo tasks: users had to move to a particular photo as quickly and as accurately as possible, with a 30 second timeout. They could press the space-bar if they were satisfied with where they reached to stop the task.

Please see the supplemental video[9] for examples of the tasks. After each condition and after the entire study, additional questionnaires (including NASA-TLX [Hart and Staveland 1988]) were completed. The entire procedure took around 75 minutes and participants were compensated by $10.

[9]https://youtu.be/mL6pVCDoFQM

4.3 Results

4.3.1 Participants. 24 participants completed the study (18 female, 6 male; avg. 21.38 years old, min. 18, max. 54). 8 used a computer mouse every day, 9 at least several days a week, and 7 almost never. 21 participants were right-handed, and all chose to use the mouse with their right hand. 22 reported being barely or not familiar with interactive 3D software, 1 somewhat familiar, and 1 very familiar. They rated their skills with 3D software as 1.67 on average on a 7-point Likert scale, where 7 is expert and 1 is novice.

4.3.2 Exploration Task Results. We recorded several different measures to assess the exploration task, including how much certain buttons were pressed, total distance traveled, and unique photos visited. We used mixed factorial ANOVAs to assess the results, with factors scene and interface, with pairwise comparisons using Tukey's method. For non-normally distributed data, we applied a log transform to bring it closer to a normal distribution.

Left-click usage was statistically significant (log applied, $F_{2,44}$ = 97.808, p < 0.001); pairwise comparisons showed that left-click was used less with interface A (2.191% of the time) compared to both B and C (avg. 10.752% and 11.891% of the time, respectively; p < 0.001 for both pairwise comparisons). Right-click usage was not statistically significant (log applied, $F_{2,44}$ = 3.088, p = 0.056). Right-clicks were used more for indoor scenes than outdoor scenes ($F_{1,22}$ = 12.183, p = 0.002; 25.932% of the time for indoor vs. 15.17% for outdoor). Scroll usage was statistically significant for factor interface (log applied, $F_{2,44}$ = 5.971, p = 0.005). However, pairwise comparisons did not reveal any statistical significance (average non-log usage percentages for A, B, C: 0.773%, 1.196%, 1.463%).

After applying a log transform, there was a main effect on total distance traveled by factor interface ($F_{2,44}$ = 6.304, p = 0.004; average non-log values 312.7m, 420.8m, 457.9m, for A, B, C, respectively). Pairwise comparisons revealed A < B (p = 0.006) and A < C (p = 0.008). In outdoor scenes, not surprisingly, users traveled further ($F_{1,22}$ = 225.581, p < 0.001; average non-log values 91.64m and 702.7m, for indoors and outdoors, respectively). We observed that at least half of the participants took on a bird's eye viewpoint of indoor scenes while using interfaces B or C (it was impossible to do so with interface A since all photos were at human height level). The number of unique photos visited was statistically significant ($F_{2,44}$ = 55.536, p < 0.001); pairwise comparisons found that there were more unique photos visited with interface A (avg. 58.38) compared to both B and C (avg. 38.83, 33.58 respectively; p < 0.001 for both).

4.3.3 Direction Estimation Task Results. Using a metric of absolute angular error, there was a statistically significant difference between indoor and outdoor scenes (log applied, $F_{1,22}$ = 7.184, p = 0.014; indoors avg. 33.952°, outdoor avg. 63.467°). There was no main effect for factor interface ($F_{2,44}$ = 1.308, p = 0.281).

4.3.4 Move-to-Photos Task Results. A mixed factorial logistic regression found that users achieved target photos more for indoor than outdoor scenes (odds ratio 1.342, p = 0.041; 63 achieved photos for indoor vs. 44 for outdoor). For indoor scenes, 86.11% of the time users reached within a threshold of the target photo (45° and 0.5m for indoors and 5m for outdoors), and 76.39% for outdoor scenes. There was no significant effect found for factor interface.

4.3.5 Post-Interface Questionnaire Results. We used mixed factorial Aligned Rank Transform (ART) ANOVAs [Wobbrock et al. 2011] to analyze the responses for the after-interface questionnaire Likert statements shown in Figure 4. All statements except for the one regarding the visualizations being helpful were statistically significant for factor interface. There were no significant effects found for scene, and NASA-TLX responses were non-significant.

4.3.6 Post-Study Questionnaire Results. Participants ranked the interfaces in order of preference for exploration tasks, move-to-photo tasks, and overall; see Figure 5. Mixed factorial ART ANOVAs indicated statistically significances for factor interface in all three cases (for exploration, move-to-photo, and overall, respectively: $F_{2,44}$ = 8.157, p < 0.001; $F_{2,44}$ = 8.986, p < 0.001; $F_{2,44}$ = 8.484, p < 0.001;). Pairwise comparisons showed B and C being ranked higher than A (all p < 0.01). Factor scene did not give any significant results.

An interaction effect existed between scene and interface for move-to-photo tasks ($F_{2,44}$ = 5.417, p = 0.008). For indoor scenes, there was no effect of interface on preference ($F_{2,22}$ = 0.843, p = 0.444). However, for outdoor scenes, there was an effect of interface on preference ($F_{2,22}$ = 20.054, p < 0.001); pairwise analysis showed both B and C being ranked higher than A (both p < 0.001).

4.3.7 Experiment I Discussion. As seen in the post-study preference results, users preferred our less constrained snapping-to-photos interfaces over the fully constrained-to-photos baseline interface. Users were more confident with our interfaces than the baseline; in fact, the majority of participants did not agree with the "confident" statement for the baseline interface A. Most users did not agree that they felt they could go where they wanted to go for the baseline, whereas most agreed regarding that statement for our snapping-to-photos interfaces. In the open-ended comments, several participants appreciated that interfaces B and C gave them more control; *e.g.*, "In interfaces B and C, I could look from above to get an overall feel for the environment. Interface A was more difficult because of the lack of that option." Four participants also appreciated the more fluid feeling of interface C compared to B; in "Experiment II" we took a closer look at the white border visualizations for both types of snapping interfaces.

It is interesting that users ranked our interfaces higher than the baseline for outdoor move-to-photo tasks but not indoor move-to-photo tasks. This may suggest that our less constrained interfaces are needed more for outdoor scenes (or, generally, larger more incomplete scenes) than for indoor scenes. Also, outdoor scenes may be harder to navigate in general and to acquire spatial knowledge as seen from the direction and move-to-photo tasks' results.

Finally, users preferred our interfaces even though the baseline allowed more unique photos to be visited. Perhaps an ideal interface would give users more control than A but still utilize the positive aspects of A that allow more photos to be visited. In Experiment II, we investigated auto POV snapping which can be considered less constrained than the baseline but more constrained than B and C.

5 EXPERIMENT II

Experiment II focused on investigating differences between click-to-snap POI snapping and automatic POV snapping, in indoor and outdoor scenes. We allowed users to experience both interfaces with

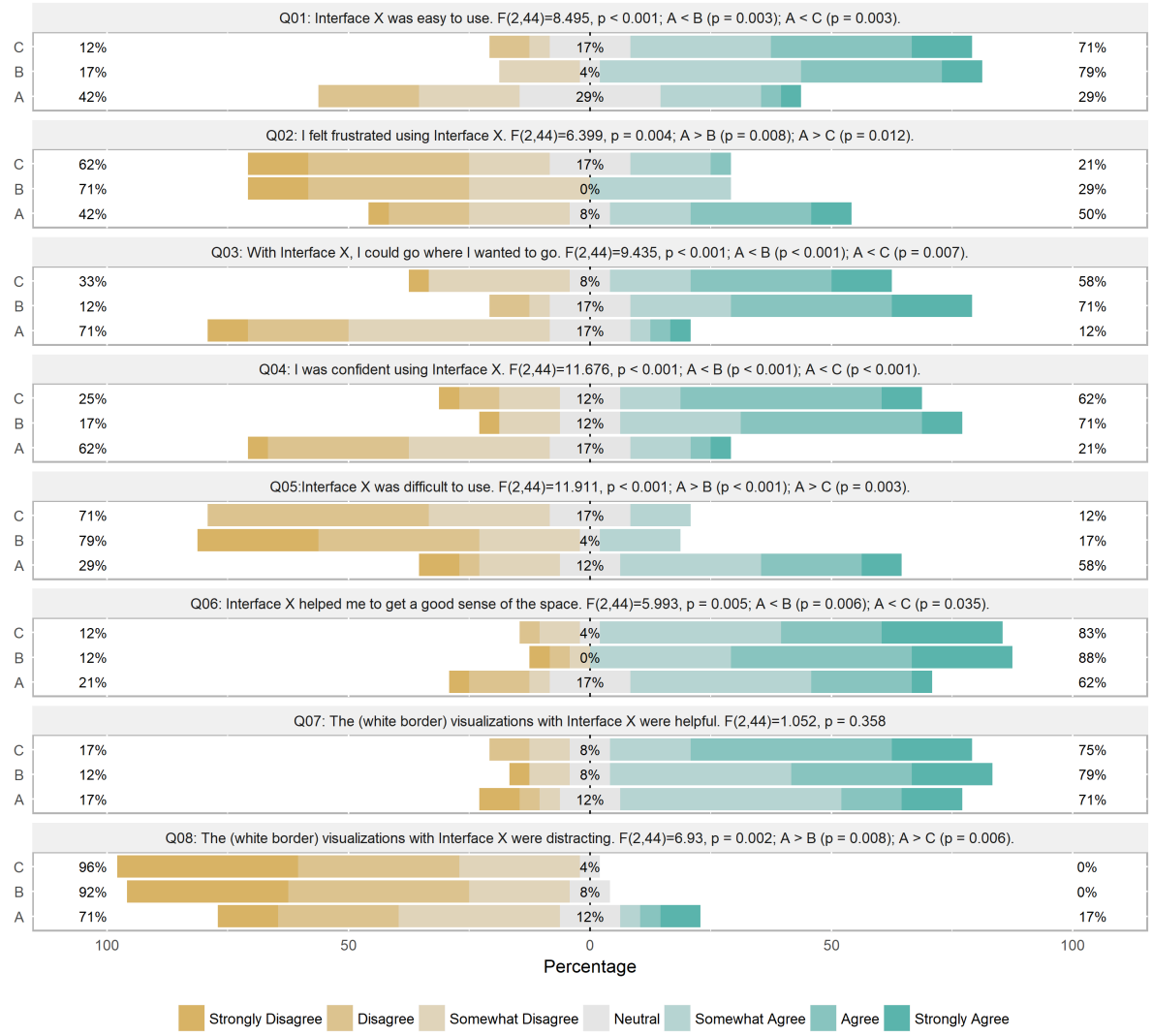


Figure 4: Experiment I Interface Questionnaire Results (best viewed in color). The three interfaces were the baseline fully constrained-to-photos (A), our click-to-snap POI snapping (B), and our click-to-snap POV snapping (C).

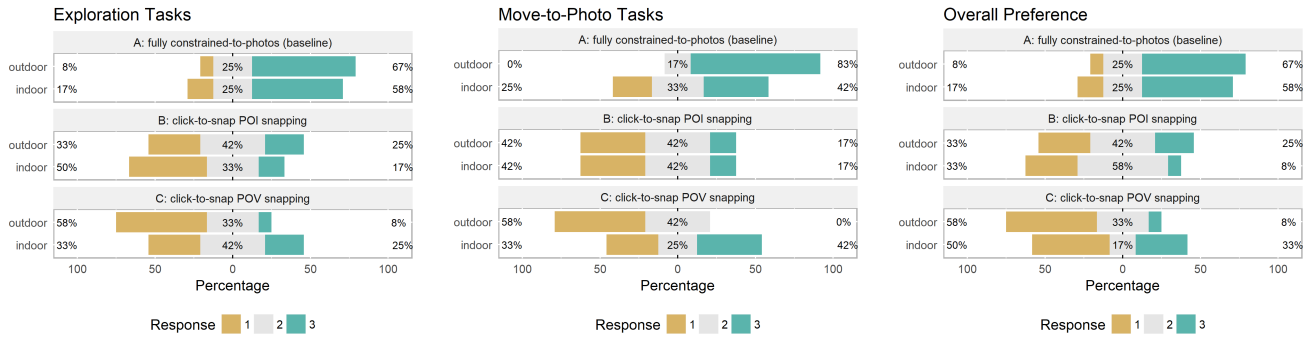


Figure 5: Experiment I Preference Results, for both indoor and outdoor datasets (best viewed in color).

and without the white border visualizations. The basic interface used for Experiment II was the outdoor basic interface in Sec. 4.1.2.

5.1 Procedure

The overall procedure was similar to Experiment I. Users experienced two interfaces and two scenes (indoor and outdoors, order counterbalanced), four conditions overall. The indoor scene was reused from Experiment I and a separate outdoor scene was created from Theia [Sweeney et al. 2015] and MVE [Fuhrmann et al. 2015; Waechter et al. 2014]; the outdoor scene's photos had approximately 51.56° fields of view. Users were trained on the basic interface and then experienced each condition consisting of an exploration task and 4 move-to-photo tasks. Due to time limitations, users could only experience two scenes, so we did not include a direction estimation task which would require different scenes for each condition. During exploration, visualizations were shown for 90s, then not shown for 90s (order counterbalanced), and then users were allowed to toggle showing them however they wished.

5.2 Results

5.2.1 Participants. 12 participants completed the study (4 female, 8 male; avg. 24.95 years old, min. 18, max. 51). 8 used a computer mouse every day, 2 at least several days a week, and 2 almost never. 10 participants were right-handed, and all preferred using the computer mouse right-handed or with either hand. 7 reported being barely or not familiar with interactive 3D software, 2 somewhat familiar, and 3 very familiar (no main effects were found based on splitting these users between at least somewhat familiar). They rated their skills with 3D software as 3.583 on average on a 7-point Likert scale, where 7 is expert and 1 is novice.

5.2.2 Exploration Task Results. We used 2-way repeated measures ANOVAs to assess the results with factors interface and scene. In terms of interaction, the main notable difference was the amount of right-clicking (16.94% of the time for POI snapping, 22.057% for POV snapping; $F_{1,11} = 5.702$, $p = 0.036$). For the last two minutes during which users could toggle visualizations, on average, users kept visualizations on 73.00% of the time for POV snapping and 72.47% of the time for POI snapping.

5.2.3 Move-to-Photos Task Results. . We analyzed the data using repeated measures factorial logistic regression. Participants achieved the target photo more with POI snapping compared to POV snapping (odds ratio 2.850, $p = 0.001$; 58.33% of the time, 56 / 96, with POI snapping; 35.42%, 34 / 96, with POV snapping). Significantly more target photos were reached for outdoor scenes (odds ratio 3.597, $p < 0.001$; 33.33% for indoor scenes; 60.42% for outdoor scenes). For both POV snapping and indoor scenes, 85.42% of the time users reached within a threshold of the target photo (45° and 0.5m for indoors and 5m for outdoors), and 93.75% for both POI snapping and outdoor scenes. A 2-way repeated measures ANOVA found no main effects on non-timeout time by interface (log applied; avg. 16.932s for POI and 17.6s for POV) nor scene (avg. 15.95s indoor, 18.25s outdoor). Training effects were minimal as users achieved photos 44.79% of the time when first experiencing a dataset and 48.96% of the time after that (near photos, 71.86% and 78.13% of the time, respectively).

5.2.4 Post-Condition Questionnaire Results. . We used 2-way repeated measures ART ANOVAs [Wobbrock et al. 2011] to analyze the responses for after-condition questionnaire statements; Figure 6 shows results for factor "Interface." Statistically significant results are shown in the figure for questions Q01, Q02, Q04, and Q05, regarding ease of use, frustration, confidence, and difficulty. There were no statistically significant results for the "scene" factor.

5.2.5 Post-Scene and Post-Study Questionnaire Results. We asked participants to choose which interface they prefer for exploration tasks, move-to-photo tasks, and overall; see Figure 7. For after-scene questionnaires, one-sample median tests indicated statistically significant differences for the preferences: exploration ($p = 0.004$), move-to-photo ($p < 0.001$), and overall ($p < 0.001$). There were no statistically significant results between indoor and outdoor scenes using a repeated measures ART ANOVA. For the post-study, one-sample median tests indicated a statistically significant difference only regarding move-to-photo tasks preferences ($p = 0.006$).

5.2.6 Experiment II Discussion. Automatic POV snapping can be considered closer to a fully constrained-to-photos interface (cf. Figure 2) and thus it may not be surprising that users generally preferred the click-to-snap POI snapping that gave users more control. Regarding the move-to-photo tasks, users also performed better with POI-based snapping. This seems to indicate that POI snapping allows users to more easily maneuver to specific photos.

The fact that not a small number of users reported focusing on the white border visualizations more than the scene is interesting. One user noted that the system "...must be able to toggle on and off" the visualizations, while another said, "The borders detract from the realism but make it easier to find a specific picture." Around a third of users felt that the white border visualizations should be off by default (see Figure 6). This further emphasizes the fact that users appreciate more control over the experience.

At least three participants noted that the auto POV snapping in indoor scenes was more useful than for outdoor scenes.

6 MULTI-TOUCH INTERFACE

We also implemented a multi-touch version of our snapping-to-photos interfaces. One finger taps and dragging are equivalent to left-clicks and dragging. Rotation is performed by dragging two fingers, using the average position of both fingers. A three finger drag gesture allows movement parallel to the image plane while one finger dragging is parallel to the ground. Please see the supplemental video for an illustration of our multi-touch interface.

We allowed a variety of people to experience this interface at our organization's May 2017 open house event. Around 10 people of various ages used the multi-touch automatic POV snapping interface for extended periods of time. Most commented that the interface was easy to use and intuitive. Several visitors stated explicitly that it was very fun to use.

7 DISCUSSION

While constrained 3D navigation interfaces can be very helpful [Stuerzlinger and Wingrave 2011], the proper balance between control and constraints must be made. Based on our results for mouse interaction, overall we recommend click-to-snap POI-snapping and

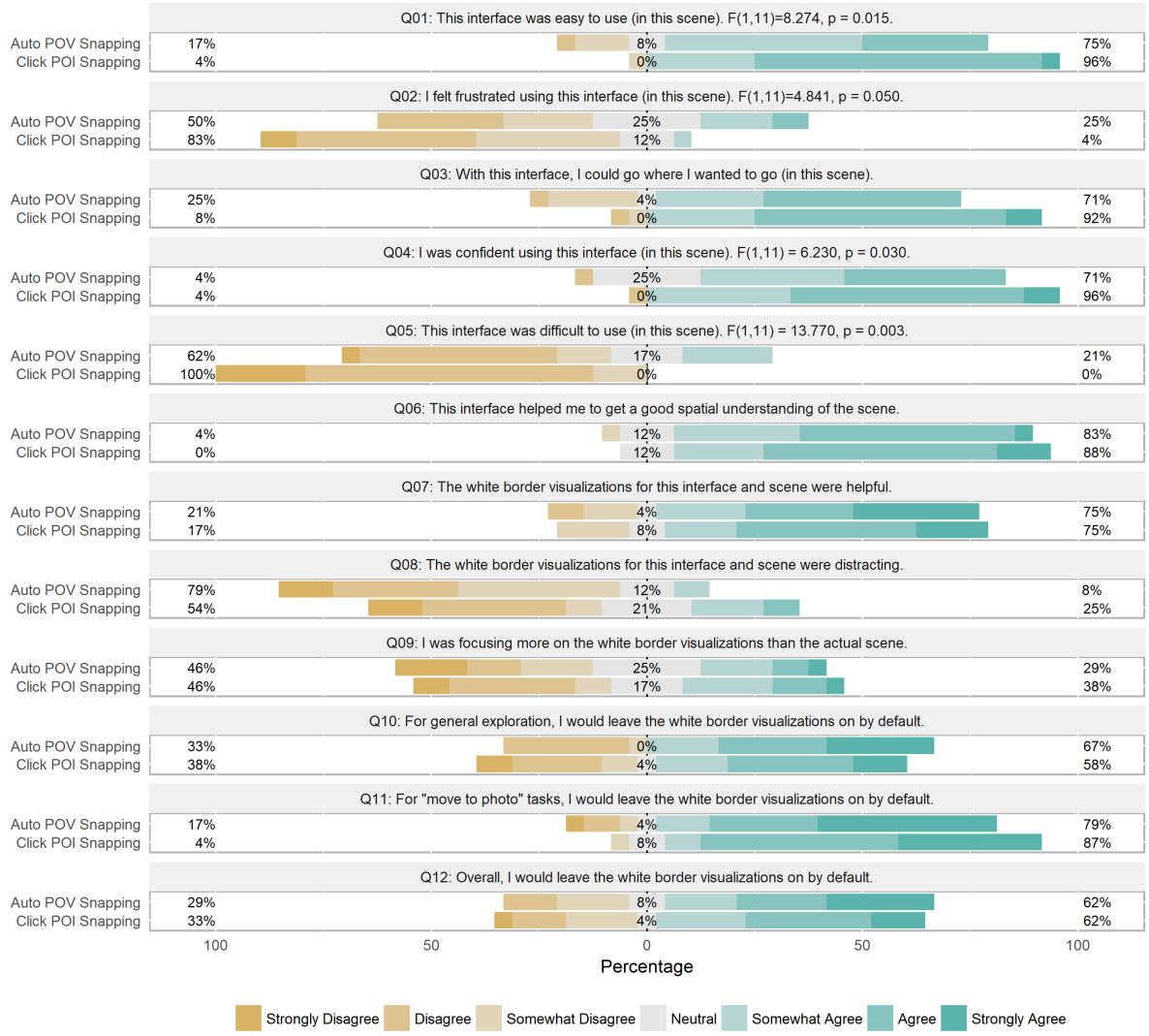


Figure 6: Experiment II Interface Questionnaire Results (best viewed in color).

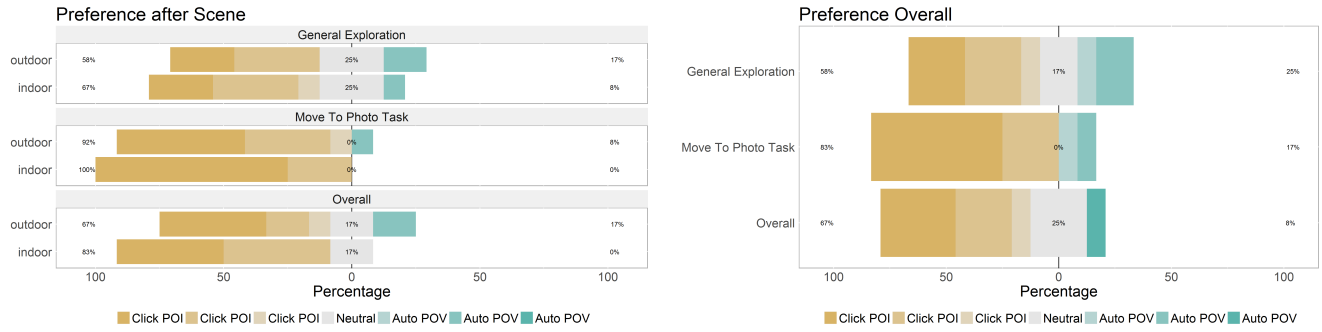


Figure 7: Experiment II Preference Results (best viewed in color).

the ability for users to toggle visualizations on and off. In multi-touch interaction, the concept of a hovering (without clicking / touching) cursor does not exist as it does with mouse interaction; in this case, auto POV snapping may be more usable, but further studies should be done to confirm this.

We focused on evaluating snapping-to-photos interfaces for unstructured captured scenes in situations that have not been evaluated in prior work. The only comparative study known to us [Brivio et al. 2013] used only a single outdoor dataset. Ours may be the first user evaluation that uses both indoor, dense datasets and outdoor, sparse datasets, with our results showing differences between the two. We hope our results will encourage more research in this area.

In this paper, the focus was not on designing a new rendering algorithm or using novel display technology. Thus, for simplicity and to avoid any possible bias in rendering differences for each of the interfaces, we used only the underlying 3D model during transitions between photos. We leave the investigation of using immersive head-worn displays with snapping-to-photos exploration of emerging image-based reconstructions to future work.

Another avenue for future work is to incorporate scene and user movement semantics into the viewpoint similarity cost function. For example, perceptual models of viewpoint preference and visual saliency [Freitag et al. 2015; Secord et al. 2011] may aid in creating a better POI snapping interface. User movement semantics may improve POV snapping by favoring snapping to photos in front when the user is moving forward and vice versa. Finally, orbiting movement is another area ripe for future exploration for snapping-to-photos interfaces.

8 CONCLUSION

Navigating through emerging 3D image-based reconstructions of visual reality is a challenging and important problem. In this paper, we introduced two new snapping-to-photos virtual navigation interfaces that loosen the constraints compared to previous methods and that are especially useful for navigating scenes captured in an unstructured way. Experiment I results included that users prefer our snapping-to-photos interfaces over the baseline fully constrained-to-photos interface and that for move-to-photo tasks, this preference appeared for outdoor scenes but not indoor scenes. Experiment II, comparing click-to-snap POI snapping and automatic POV snapping, revealed that users perform better with click-to-snap POI snapping for move-to-photo tasks and prefer click-to-snap POI snapping over automatic POV snapping. Richly supporting virtual navigation of emerging scene models, with interfaces such as snapping-to-photos, will ultimately enable more usable applications of virtual navigation of visual reality.

ACKNOWLEDGMENTS

We thank Ehsan Sayyad for help with Unity, Kelly Bielaski for help with the base travel interfaces, Prof. Mary Hegarty for suggestions on user study design, and Zach Terner for help with the statistical analysis. This work was supported by NSF grant IIS-1423676 and ONR grant N00014-16-1-3002. Sponsorship to present this work at the ACM Virtual Reality Software and Technology Conference was provided by the Jet Propulsion Laboratory, California Institute of Technology, under contract to NASA, the National Aeronautics and Space Administration.

REFERENCES

Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. 2010. Google street view: Capturing the world at street level. *Computer* 43, 6 (2010), 32–38. https://doi.org/10.1109/MC.2010.170

Luca Ballan, Gabriel J. Brostow, Jens Puwein, and Marc Pollefeys. 2010. Unstructured video-based rendering. *ACM Transactions on Graphics* 29, 4 (July 2010), 1. https://doi.org/10.1145/1778765.1778824

Steffi Beckhaus, Felix Ritter, and Thomas Strothotte. 2001. Guided exploration with dynamic potential fields: The CubicalPath system. *Computer Graphics Forum* 20, 4 (Dec. 2001), 201–210. https://doi.org/10.1111/1467-8659.00549

Marco Di Benedetto, Fabio Ganovelli, Marcos Balsa Rodriguez, Alberto Jaspe Villanueva, Roberto Scopigno, and Enrico Gobbetti. 2014. ExploreMaps: Efficient construction and ubiquitous exploration of panoramic view graphs of complex 3D environments. *Computer Graphics Forum* 33, 2 (May 2014), 459–468. https://doi.org/10.1111/cgf.12334

Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2004. *3D User Interfaces: Theory and Practice.* Addison Wesley Longman Publishing Co., Inc.

Paolo Brivio, Luca Benedetti, Marco Tarini, Federico Ponchio, Paolo Cignoni, and Roberto Scopigno. 2013. PhotoCloud: Interactive remote exploration of joint 2D and 3D datasets. *IEEE CG&A* 33, 2 (2013), 86–97. https://doi.org/10.1109/MCG.2012.92

Shenchang Eric Chen. 1995. QuickTime VR: An Image-based Approach to Virtual Environment Navigation. In *ACM SIGGRAPH.* ACM, New York, NY, USA, 29–38. https://doi.org/10.1145/218380.218395

Marc Christie, Patrick Olivier, and Jean Marie Normand. 2008. Camera control in computer graphics. *Computer Graphics Forum* 27, 8 (Dec. 2008), 2197–2218. https://doi.org/10.1111/j.1467-8659.2008.01181.x

Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration. *ACM Trans. Graph.* 36, 3, Article 24 (May 2017), 18 pages. https://doi.org/10.1145/3054739

T. Todd Elvins, David R. Nadeau, Rina Schul, and David Kirsh. 2001. Worldlets: 3-D Thumbnails for Wayfinding in Large Virtual Worlds. *Presence: Teleoper. Virtual Environ.* 10, 6 (Dec. 2001), 565–582. https://doi.org/10.1162/105474601753272835

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88, 2 (2010), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Thomas Forgione, Axel Carlier, Géraldine Morin, Wei Tsang Ooi, and Vincent Charvillat. 2016. Impact of 3D Bookmarks on Navigation and Streaming in a Networked Virtual Environment. In *ACM MMSys.* ACM, New York, NY, USA, Article 9, 10 pages. https://doi.org/10.1145/2910017.2910607

Sebastian Freitag, Benjamin Weyers, Andrea Bönsch, and Torsten W. Kuhlen. 2015. Comparison and Evaluation of Viewpoint Quality Estimation Algorithms for Immersive Virtual Environments. In *ICAT - EGVE '15.* Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 53–60. https://doi.org/10.2312/egve.20151310

Simon Fuhrmann, Fabian Langguth, Nils Moehrle, Michael Waechter, and Michael Goesele. 2015. MVE—An image-based reconstruction environment. *Computers & Graphics* 53, Part A (Dec. 2015), 44–53. https://doi.org/10.1016/j.cag.2015.09.003

Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. 2009. Reconstructing building interiors from images. In *2009 IEEE 12th International Conference on Computer Vision.* 80–87. https://doi.org/10.1109/ICCV.2009.5459145

Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014a. In Touch with the Remote World: Remote Collaboration with Augmented Reality Drawings and Virtual Navigation. In *ACM VRST.* ACM, New York, NY, USA, 197–205. https://doi.org/10.1145/2671015.2671016

Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014b. World-stabilized Annotations and Virtual Scene Navigation for Remote Collaboration. In *ACM UIST.* ACM, New York, NY, USA, 449–459. https://doi.org/10.1145/2642918.2647372

Michael Goesele, Jens Ackermann, Simon Fuhrmann, Carsten Haubold, Ronny Klowsky, Drew Steedly, and Richard Szeliski. 2010. Ambient Point Clouds for View Interpolation. *ACM Trans. Graph.* 29, 4, Article 95 (July 2010), 6 pages. https://doi.org/10.1145/1778765.1778832

Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Adv. in Psyc.* 52 (1988), 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

Jacek Jankowski and Martin Hachet. 2015. Advances in interaction with 3D environments. *Computer Graphics Forum* 34, 1 (2015), 152–190. https://doi.org/10.1111/cgf.12466

Johannes Kopf, Billy Chen, Richard Szeliski, and Michael Cohen. 2010. Street Slide: Browsing Street Level Imagery. *ACM Trans. Graph.* 29, 4, Article 96 (July 2010), 8 pages. https://doi.org/10.1145/1778765.1778833

Johannes Kopf, Michael F. Cohen, and Richard Szeliski. 2014. First-person Hyperlapse Videos. *ACM Trans. Graph.* 33, 4, Article 78 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601195

Avanish Kushal, Ben Self, Yasutaka Furukawa, David Gallup, Carlos Hernandez, Brian Curless, and Steven M. Seitz. 2012. Photo tours. *3DIM/3DPVT Conf.* (Oct. 2012), 57–64. https://doi.org/10.1109/3DIMPVT.2012.62

Andrew Lippman. 1980. Movie-maps: An Application of the Optical Videodisc to Computer Graphics. *SIGGRAPH Comput. Graph.* 14, 3 (July 1980), 32–42. https://doi.org/10.1145/965105.807465

Clement Moerman, Damien Marchal, and Laurent Grisoni. 2012. Drag'n Go: Simple and fast navigation in virtual environment. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. 15–18. https://doi.org/10.1109/3DUI.2012.6184178

Benjamin Nuernberger, Kuo-Chin Lien, Lennon Grinta, Chris Sweeney, Matthew Turk, and Tobias Höllerer. 2016. Multi-view Gesture Annotations in Image-based 3D Reconstructed Scenes. In *Proceedings of the 22nd ACM VRST*. ACM, New York, NY, USA, 129–138. https://doi.org/10.1145/2993369.2993371

Said Pertuz, Domenec Puig, and Miguel Angel Garcia. 2013. Analysis of Focus Measure Operators for Shape-from-focus. *Pattern Recogn.* 46, 5 (May 2013), 1415–1432. https://doi.org/10.1016/j.patcog.2012.11.011

Adrian Secord, Jingwan Lu, Adam Finkelstein, Manish Singh, and Andrew Nealen. 2011. Perceptual Models of Viewpoint Preference. *ACM Trans. Graph.* 30, 5, Article 109 (Oct. 2011), 12 pages. https://doi.org/10.1145/2019627.2019628

Noah Snavely, Rahul Garg, Steven M. Seitz, and Richard Szeliski. 2008. Finding Paths Through the World's Photos. *ACM Trans. Graph.* 27, 3, Article 15 (Aug. 2008), 11 pages. https://doi.org/10.1145/1360612.1360614

Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics* 25, 3 (July 2006), 835. https://doi.org/10.1145/1141911.1141964

Wolfgang Stuerzlinger and Chadwick A. Wingrave. 2011. *The Value of Constraints for 3D User Interfaces.* Springer Vienna, Vienna, 203–223. https://doi.org/10.1007/978-3-211-99178-7_11

Christopher Sweeney, Tobias Höllerer, and Matthew Turk. 2015. Theia: A Fast and Scalable Structure-from-Motion Library. *Proceedings of the 23rd ACM International Conference on Multimedia* (2015), 693–696. https://doi.org/10.1145/2733373.2807405

Ryohei Tanaka, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. 2016. Guidance field: Potential field to guide users to target locations in virtual environments. In *IEEE Symposium on 3D User Interfaces*. 39–48. https://doi.org/10.1109/3DUI.2016.7460029

James Tompkin, Kwang In Kim, Jan Kautz, and Christian Theobalt. 2012. Videoscapes: Exploring Sparse, Unstructured Video Collections. *ACM Trans. Graph.* 31, 4, Article 68 (July 2012), 12 pages. https://doi.org/10.1145/2185520.2185564

Kathleen Tuite, Noah Snavely, and Nadine Tabing. 2011. PhotoCity : Training Experts at Large-scale Image Acquisition Through a Competitive Game. *ACM SIGCHI* (2011), 1383–1392. https://doi.org/10.1145/1978942.1979146

Michael Waechter, Nils Moehrle, and Michael Goesele. 2014. *Let There Be Color! Large-Scale Texturing of 3D Reconstructions.* Springer International Publishing, Cham, 836–850. https://doi.org/10.1007/978-3-319-10602-1_54

Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *ACM SIGCHI*. ACM, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

Jianxiong Xiao and Yasutaka Furukawa. 2014. Reconstructing the World's Museums. *IJCV* 110, 3 (2014), 243–258. https://doi.org/10.1007/s11263-014-0711-y

Shanghong Zhao and Wei Tsang Ooi. 2016. Modeling 3D synthetic view dissimilarity. *The Visual Computer* 32, 4 (01 Apr 2016), 429–443. https://doi.org/10.1007/s00371-015-1069-z

Minhui Zhu, Sebastien Mondet, Géraldine Morin, Wei Tsang Ooi, and Wei Cheng. 2011. Towards Peer-assisted Rendering in Networked Virtual Environments. In *Proceedings of the 19th ACM International Conference on Multimedia (MM '11)*. ACM, New York, NY, USA, 183–192. https://doi.org/10.1145/2072298.2072324