Look-From Camera Control for 3D Terrain Maps

Kurtis Danyluk
University of Calgary
Calgary, AB, Canada
ktdanylu@ucalgary.ca

Bernhard Jenny
Monash University
Melbourne, VIC, Australia
bernie.jenny@monash.edu

Wesley Willett
University of Calgary
Calgary, AB, Canada
wj@wjwillett.net

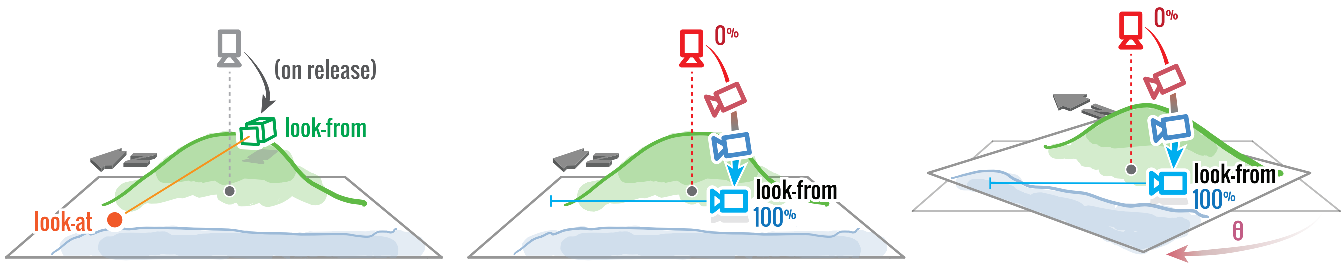


Figure 1: We examine three *look-from* camera control variations: *Discrete Look-From-At* (left), *Continuous Look-From-Forward* (center), and *Continuous Look-From-Towards* (right).

ABSTRACT

We introduce three lightweight interactive camera control techniques for 3D terrain maps on touch devices based on a *look-from* metaphor (*Discrete Look-From-At*, *Continuous Look-From-Forward*, and *Continuous Look-From-Towards*). These techniques complement traditional touch screen pan, zoom, rotate, and pitch controls and allow viewers to quickly transition between top-down, oblique, and ground-level views. We present the results of a study in which we asked participants to perform elevation comparison and line-of-sight determination tasks using each technique. Our results highlight how *look-from* techniques can be integrated on top of current direct manipulation navigation approaches by combining several direct manipulation operations into a single *look-from* operation. Additionally, they show how *look-from* techniques help viewers complete a variety of common and challenging map-based tasks.

CHI 2019, May 4–9, 2019, Glasgow, Scotland Uk

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300594

CCS CONCEPTS

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

KEYWORDS

Map interaction; look-from camera control; touch; terrain.

ACM Reference Format:
Kurtis Danyluk, Bernhard Jenny, and Wesley Willett. 2019. Look-From Camera Control for 3D Terrain Maps. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland Uk.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300594

1 INTRODUCTION

Digital terrain tools like Google Earth are useful because they allow viewers to smoothly navigate complex virtual terrain representations. They do so by transitioning dynamically between more traditional top-down map views and oblique perspective views that highlight the shape of the underlying terrain. These systems normally allow viewers to control their view of the map using mouse or touch-based direct manipulation controls that pan, zoom, rotate, and pitch a camera within the virtual environment. Using these controls, viewers can fluidly adjust their point of view, exploring the map and viewing terrain from a variety of different perspectives.

However, many common views—particularly those from ground-level and from specific vantage points in the scene—can be difficult to recreate using direct manipulation navigation techniques. In particular, navigating from a top-down view to an oblique or ground-level view often requires a complicated chain of direct manipulation operations that can be difficult to execute precisely and challenging to reverse. Good camera controls can greatly affect how users perceive a system [36] and making these sequences of interactions easier for viewers is an important challenge.

To address this challenge, we show how these common camera movements can instead be described as *look-from* operations. *Look-from* camera controls reformulate camera movements as declarations from the user that they want to look from one point in a scene at or towards another.

We present three variants—*Discrete Look-From-At*, *Continuous Look-From-Forward*, and *Continuous Look-From-Towards*—which each allow viewers to specify ground-level views using a single touch gesture and can be used in conjunction with existing direct manipulation controls. While these techniques vary in terms of their constraints, all three complement the existing vocabulary of direct manipulation techniques for digital maps (pan, zoom, rotate, pitch).

We examine the impact of these *look-from* techniques via a study in which we asked participants to perform line-of-sight and elevation comparison tasks using each *look-from* variation and using *direct manipulation* controls. Our results show how these camera control variations complement the existing direct manipulation techniques for navigating digital maps, enabling new kinds of navigation strategies. They also show how participants found *look-from* techniques to be easier to use and more enjoyable than *Direct Manipulation* controls. Additionally, our two continuous *look-from* techniques highlight promising new opportunities for precise, interactive refinement of camera views selected using a *look-from* metaphor – something not possible with previous techniques. Finally, based on our experience with these initial variants and feedback collected in the study, we provide design considerations for future *look-from* navigation tools for digital maps.

2 RELATED WORK

Our work builds on past research on interactive camera control and techniques for digital terrain navigation. However, our work is the first to explore single-touch camera control gestures for navigating digital terrain models.

Interactive Camera Control

Past research has considered *look-from* camera control broadly in the context of "through-the-lens" camera control [12], a subset of targeted movement [17]. However, the existing literature provides little explicit guidance about how to integrate these techniques into new systems, or about the practical usability of these controls—particularly for digital terrain applications. While through-the-lens control metaphors cover a wide variety of camera transformations [12] we focus primarily on variants that use touch interactions [10, 30] to specify the camera's position and orientation relative to a distinct location in the environment.

Cockburn et al. [7] provide a thorough overview of zooming and panning interfaces that allow users to view information, such as maps, at varying levels of detail. While these solutions can improve viewer experience with top-down views, they do not allow viewers to change to oblique or ground level views.

Early work by Ware and Osborne [32] also explored and evaluated several basic metaphors to allow viewers to freely explore 3D space with full six degrees of freedom. Later work presented alternative controls schemes that allow viewers to move through 3D space using minimal gestures and control interfaces, but provide no evaluation of their effectiveness [27, 37]. Martinet et al.'s DS3 [21] also examined object-manipulation camera control using separate translation and rotation controls. Other interactive camera control techniques relevant to our research include those first formalized by Mackinlay et al. [20] where the user indicates a target object and the camera transitions towards the target along a linear path. While these approaches provide an alternative to *direct manipulation* they are optimized for small object scenes and not large-scale virtual environments.

Interaction Techniques for Terrain Navigation

Camera control for digital globes and large terrain requires multiscale navigation and camera control techniques. McCrae et al. [22] proposed a combination of various camera control techniques for navigating from a globe view to a microscopic scale. The use of constraints applied to camera positioning, orientation and animation has also been explored for the navigation of terrain. Buchholz et al. [3] adapt physically-based navigation by Turner et al. [29] for navigation in terrain models. With physically-based navigation, the camera is modeled as a rigid body [28], or as a mass-spring system [3], with user input exerting forces on that body. Additional constraints for moving the camera are often essential for navigation in terrain [14] or 3D city models [15]. Occlusion avoidance is particularly relevant for camera navigation in ragged terrain. In response, force fields [34] or distance fields [31] have been proposed for physically-based navigation.

Alternatives to these traditional camera control techniques also exist. Hagedorn and Döllner [13] proposed sketch-based navigation in 3D virtual environments, particularly geospatial visualization. The user draws sketches on a touch display, which are interpreted to control the camera. For example, a curve drawn on a street is converted to an animated drive along that street. Pierce and Pausch [24] place 3D representations of distant landmarks and places in the currently visible view. The representations are interactive and trigger a ride to the corresponding location.

Other work has attempted to aid viewers in understanding terrain maps by adding new interaction techniques to 2D digital maps without allowing for full six degrees of freedom navigation. Lightweight Relief Shearing [33] and Elastic Terrain [4] both use simple touch gestures to expose the shape of the terrain in ways that are compatible with existing pan-zoom interactions, but because they reveal shape by obliquely shearing the entire map, it is not possible to create ground-level views with them. In spite of this, participants in initial evaluations of these techniques [33] often tried to stretch the terrain in order to examine the silhouettes of terrain features and recreate specific ground-level views.

Finally, we also build on findings from Abend et al. [1] who show that users of Google Earth most often retain the north orientation of maps and tend to quickly return to a north orientation after a rotation.

3 NAVIGATION VIA DIRECT MANIPULATION

Current state of the art digital map systems are most commonly controlled using several direct manipulation interactions: pan, zoom, rotate, pitch. In all these controls the most important feature is that they preserve the location that the viewer grabs in the scene, either using touch or a mouse.

Pan controls allow the viewer to move the camera view parallel to the current view plane. **Zoom** controls allow the viewer to move the camera towards or away from a zoom point. This causes objects to become bigger or smaller as per Thales' theorem. **Rotate** controls allow the viewer to roll the camera, rotating it about the local horizontal axis. **Pitch**, or tilt, controls allow the viewer to pitch the camera, rotating it about the local vertical axis.

To identify common pain points associated with direct manipulation camera control in digital terrain maps, we conducted a pilot study in which we asked three participants to navigate virtual environments and recreate ground-level viewpoints in Google Earth. In the pilot, we gave participants an image of a scene taken at an oblique angle. Alongside that image, we presented participants with an instance of Google Earth initialized to a top down view of the same region. We then asked the participant to recreate that oblique view using Google Earth's pan, zoom, rotate, and pitch interactions.

When recreating views, participants routinely struggled during the transition between the top-down, oblique, and ground-level perspectives. We observed several recurring problems that make these kinds of transitions challenging:

Chaining interactions. Moving between top-down and ground-level views typically requires a viewer to chain together multiple pan, zoom, rotate, and pitch operations. This requires a degree of foresight as interactions made early in the sequence—such as moving to ground level too early—influence the effect of subsequent interactions.

Disorientation. Sequences of pan, rotate, and pitch interactions compound one another and can make it difficult to locate and recreate desired viewpoints. In our pilot, participants often became disoriented, requiring a brief pause between steps to reorient themselves. These brief pauses slowed down the overall process.

Reversibility. Undoing or reversing sequences containing multiple direct manipulations interactions can become very difficult, making it challenging to backtrack to previous camera positions. In our pilot, participants who had made a mistake often chose to completely reset the scene and start over rather than attempt to reverse their prior interactions.

Interestingly, we observed that when explaining the tasks participants would describe them in terms of two points. The point they wanted to look from and the point they wanted to look at. This inspired our *look-from* approach.

4 *LOOK-FROM* CAMERA CONTROL

User-defined *look-from* and *look-at* locations constrain the space of possible camera paths that are relevant to the viewer. Given these constraints, a map application can animate the camera directly to the desired view, seamlessly panning, zooming, rotating, and pitching the camera as needed.

Reframing camera control as a *look-from-at* interaction also has the potential to considerably simplify viewer interaction, reducing the sequence of 4+ independent pan, zoom, rotate, and pitch interactions necessary to create a ground-level view down to just two selection operations. Moreover, selecting these *look-from* and *look-at* points can be accomplished easily in a single gesture by mapping them to *touch-down* and *touch-up* interactions respectively. This kind of straightforward *look-from-at* interaction also remains compatible with direct manipulation operations, since viewers can easily use additional pan, zoom, rotate, and pitch interactions to adjust the resulting view.

Unlike direct manipulation operations, which allow viewers to continuously refine and reverse an action, this rudimentary *look-from-at* interaction provides little opportunity for adjustment during the interaction itself. However, by relaxing the requirement for the *look-at* point, we can create a variety of alternative *look-from* interactions which increase the interactivity of the camera control possible within a single touch.

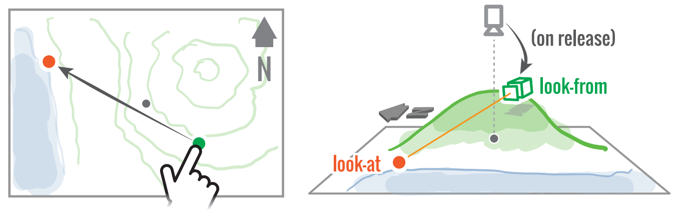
Card et al. [5], Jacob et al. [16] and Mackinlay et al. [19] describe many different ways to map user input controls, however we limit our examination to *look-from* camera control techniques that can be executed using single-touch gestures. This constraint makes it possible for any of the techniques to be executed as a *quasi-mode* [26] initiated via a long press, double-tap, or other explicit interaction. As a result, these *look-from* techniques can be included in environments that also support traditional direct-manipulation camera controls—allowing the two approaches to complement one another.

Additionally, we follow the work of Christianson et al. [6], which examines declarative camera controls from the perspective of film, to help formulate how *look-from* camera controls

Table 1: Input-response mappings for the three *look-from* techniques.

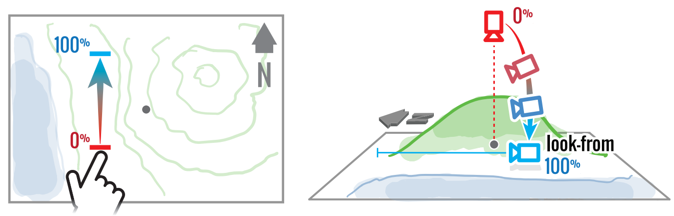
<u>**Discrete Look-From-At**</u>

Touch **down** ⟶ specify *look-from* location
Touch **up** ⟶ specify *look-at* location
initiate animated transition



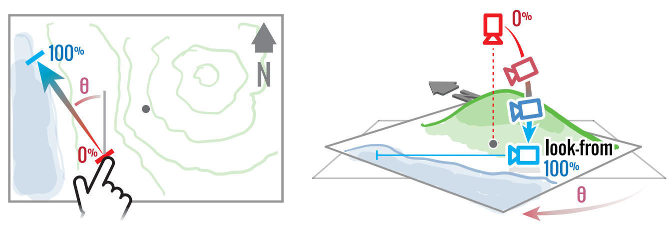
<u>**Continuous Look-From-Forward**</u>

Touch **down** ⟶ specify *look-from* location
Touch **move** ⟶ advance/reverse interactive transition
[0% ⟷100%] = distance from touch down
Touch **up** ⟶ stop transition



<u>**Continuous Look-From-Towards**</u>

Touch **down** ⟶ specify *look-from* location
Touch **move** ⟶ advance/reverse interactive transition
[0% ⟷100%] = distance from touch down
⟶ rotate camera path
[0°⟷360°] = angle relative to touch down
Touch **up** ⟶ animated transition back to original view



should act to best engage a user, motivating some of the details of our implementation.

We explore three distinct look-from variants (*Discrete Look-From-At*, *Continuous Look-From-Forward*, and *Continuous Look-From-Towards*) which showcase the diversity of camera manipulations possible in a single gesture.

*Discrete Look-From-**At*** (Table 1–top) is a straightforward and simple execution of the *look-from-at* metaphor. A viewer selects *look-from* and *look-at* locations on the surface of the terrain. From those points, the system generates an automated camera transition which pans, zooms, rotates, and pitches the camera to position it just behind the *look-from* location, pointed towards the *look-at* location.

*Continuous Look-From-**Forward*** (Table 1–center) generalizes the *look-from-at* pattern by assuming that the *look-at* location is always located in the direction the map is facing, in front of the *look-from* location. This frees the viewer to specify the *look-from* location on touch-down, and then use the rest of the gesture to dynamically move the camera along a path between its initial position and the *look-from* point. Sliding upward advances the camera further along the path—adjusting its position, zoom, and pitch as it progresses. Sliding back reverses that action. This allows the viewer to dynamically refine the camera view, making it possible to smoothly navigate to oblique views above the selected location as well as to locations at ground level. Because the technique is compatible with existing direct manipulation gestures, such as two-finger rotation, viewers can use these to rotate the map either before or after a *look-from-forward* interaction.

*Continuous Look-From-**Towards*** (Table 1–bottom) increases the expressiveness (but also the complexity) of *look-from-forward* by allowing viewers to interactively rotate the map and advance the camera simultaneously. As with the previous techniques, the viewer specifies the *look-from* location on touch-down. Dragging away from the initial touch point in any direction advances the camera towards the *look-from* location, changing its position, zoom, and pitch. Meanwhile, the camera path is rotated based on the on-screen angle between the vector formed by the viewer's initial touch point and the current touch location and a vector pointing straight up from the initial touch point. This greatly increases the number of possible ground-level and oblique views that can be reached in a single interaction. However, it also increases the potential for disorientation as viewers control their camera relative to the starting camera position, not the current camera position, and they are free to break the line of interest [6]. By default, we automatically transition the camera back to its initial view at the end of an interaction to allow for quicker exploration.

5 IMPLEMENTATION

To explore the space of possible single-touch *look-from* interactions, we iteratively implemented and examined instances of each of these techniques as well as numerous hybrids and variants. Our implementation uses Unity 2017/8 with C# and relies on elevation data from Amazon Web Services[1] and satellite imagery from Bing Maps[2]. The prototype supports multitouch direct manipulation interactions modeled on those used in the mobile versions of Google Earth, including two-finger gestures for zoom, rotate, and pitch. Pan is performed with a one finger sliding gesture. Zoom is performed with a two-finger pinch gesture. Pitch is performed with a two-finger gesture where viewers slide both fingers up to pitch the camera up, and both fingers down to pitch down. Finally, rotate is performed with a two-finger twisting gesture.

Discrete Look-From-At (Table 1–top) is implemented as a single discrete transition that begins with a touch-down event. Using that touch point we cast a ray from the camera to find the *look-from* point of the interaction. The interaction ends with a touch-up event. As with the touch-down, we cast a ray from the camera to find where that touch up occurred in world space and mark it as the *look-at* point. Once those points are identified we begin the camera transition. If we cannot cast an unobstructed line between the *look-from* and *look-at* points, we raise the *look-from* and *look-at* positions at a rate of 2 to 1 until an unobstructed line can be cast between them. This improves the visibility of the camera in the direction of the the *look-at* point while keeping the camera position relatively close to the selected *look-from* point. The camera is then smoothly transitioned using a linear interpolation function such that (a) its position is equal to the adjusted *look-from* point and (b) it is facing the adjusted *look-at* point.

Continuous Look-From-Forward (Table 1–center) is a continuous interaction technique that begins with a touch-down event. Using that touch point we cast a ray from the camera to find the *look-from* point of the interaction. To find the *look-at* point in front of the *look-from* point we then cast another ray from the camera at an angle of 20° above the first ray. At the end of the transition, the camera is located at the *look-from* position and looking at the *look-at* location. If the second ray fails to intersect the terrain (for example if 20° overshoots the edge of the map or the horizon) we reduce the angle in steps of 1° until an intersection is found.

The interaction continues with a touch move event. The vertical distance in screen space between the starting point and the current touch point is measured. That distance is then mapped to a range between 0 and 1. If the current point is more than 50% of the screen away from the starting point then it is mapped to 1, if it is 0% away then it is mapped to 0. Between 0% and 50% we map the distance to a value between 0 and 1 using a smooth-step function. We then linearly interpolate the camera position between the starting position and end position using the mapped distance. We also spherically linearly interpolate the camera rotation between the start and end rotation by the square of the mapped distance. Finally, the camera position is moved back slightly along the vector between the *look-from* and *look-at* point, keeping the *look-from* point in view to reduce viewer confusion. While the finger remains on the screen this interaction can be advanced or reversed by moving further or closer to the starting point, respectively. The interaction ends with a touch-up event and the camera remains in its last position.

Continuous Look-From-Towards (Table 1–bottom) is a continuous interaction technique that works very similarly to *Continuous Look-From-Forward*. To do this we try to first rotate the camera's starting position to replicate the orientation that we assume in a *Look-From-Forward* interaction scheme. The viewer interaction begins with a touch down event. Using that touch point we cast a ray from the camera to where that touch was in world space, this defines the *look-from* location.

The interaction continues with a touch move event. We cast a ray from the near plane of the starting camera frustum from the moved touch point to find our *look-at* position. This sets *look-from* and *look-at* points such that the projection of the vector between them on the near plane of the starting camera is the same as the vector between the starting and moved touch points. Using that vector, we immediately rotate the camera's starting position such that that vector is up. This avoids camera rolls and odd camera transitions where the horizon appears to be skewed, which can confuse viewers. We then measure the Euclidean distance in screen space between the starting point and the current touch point and use it to interpolate the camera position and rotation using the exactly the same method as for *Continuous Look-From-Forward*.

While the finger remains on the screen this interaction can be advanced or reversed by moving further from or closer to the starting point, respectively. At any point the viewer can lock the camera position by using another finger to tap a lock icon located at the starting point. The viewer interaction ends with a touch-up event. Unless the camera position was locked, the camera animates back to the starting location and rotation. If locked, the camera remains in place and the viewer can continue to adjust the view using direct manipulation or additional *Continuous Look-From-Forward* interactions.

6 STUDY

To explore the use of these three *look-from* variations we conducted a within-subjects lab study in which we asked participants to perform two common and difficult map reading tasks, line-of-sight assessment and elevation comparison,

[1]https://aws.amazon.com/public-datasets/terrain/
[2]https://msdn.microsoft.com/en-us/library/bb259689.aspx

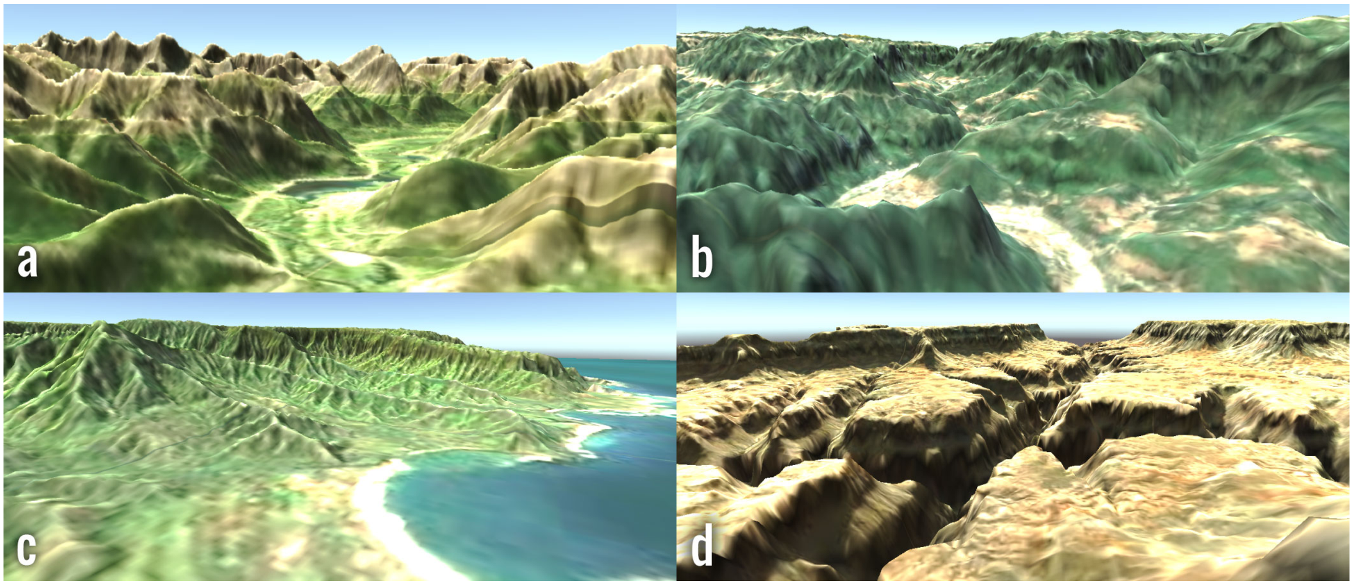


Figure 2: Regions used in our study: (a) Rocky Mountains, (b) Mount Wutai, (c) Mount Waialeale, (d) Grand Canyon.

across four different types of terrain. In addition, we asked participants to provide feedback about the difficulty of using the techniques and about which techniques they enjoyed using. Our goal was to understand how the addition of *look-from* controls changed participants' navigation strategies and examine any impact these interaction techniques had on participants' speed, accuracy, and overall experience.

Test Environment

We conducted the experiment using a Microsoft Surface 4 tablet. Participants interacted using the touch screen with no keyboard or other peripherals attached. Using our interactive prototype, we rendered four large virtual environments (Figure 2) at ground resolutions between 1:121 and 1:32 pixels per meter. We chose these locations to represent a variety of environments and scales, including: mountainous terrain (a 124km^2 section of the Canadian Rocky Mountains centered on Canmore, Alberta), rolling flat-topped peaks (a 32km^2 region around Mount Wutai in China's Shanxi Province), a volcanic plateau (a 64km^2 region surrounding Mount Waialeale on the Hawaiian island of Kauai), and deep canyons (a 32km^2 region along the Grand Canyon in northern Arizona).

Tasks

We asked participants to perform two types of tasks: elevation comparison and line-of-sight determination. In both tasks we showed participants two points on the map, one marked in red and one marked in blue. These tasks have been used to compare the effectiveness of maps and map tools in both early [23, 25] and recent studies [11, 18, 33]. We randomly generated these tasks prior to the study.

Elevation comparison tasks required the participant to answer whether the red point or blue point was at a higher elevation. The randomly generated points were always at least 8% of the map height apart and at most 14%. In addition, we removed any trivially easy trials where one of the points fell in an open flat area such as a plateau, lake, or ocean. Participants responded to the prompt "Which Point is Higher?" by pressing buttons at the upper left-hand corner of the screen labeled "Blue" or "Red" .

Line-of-sight tasks required the participant to answer if there was a clear line of sight between one point to the other. We generated line-of-sight tasks by placing a point to the left and right of the red and blue point and performing line casts between all red points to blue points. If at least two of the line casts, and at most 6, were obstructed, we considered the trial to be sufficiently challenging. When choosing trials, we selected exactly half where there was clear line of sight and half where line of sight was obstructed. Participants responded to the prompt "Can Blue see Red?" by pressing "Yes" or "No" buttons at the upper left-hand corner of the screen.

Measures

During the trials we recorded participant accuracy in completing the tasks and the time it took to complete each task. After the trials we interviewed participants about which technique they found easiest, how difficult they found each technique, which technique they enjoyed the most, and how enjoyable each technique was.

Accuracy was measured as either correct or incorrect based on the task. We asked participants to focus on accuracy when completing trials and did not expect there to be any notable difference between technique accuracy. This is because given enough time it is possible for every technique to be equally accurate.

Time was recorded from the first touch event in a trial until the participant answered the question. Because we asked participants to focus on accuracy and made no mention of speed we expected that any noticeable differences between techniques would be reflected by differences in task time.

Perceived difficulty was measured in two ways. First, we asked participants to select which technique they found the easiest. Second, we asked participants to rate on a 1–5 Likert scale how difficult they found each technique.

Perceived enjoyability was measured similarly. First, we asked participants to select which technique they found the most enjoyable. Second, we asked participants to rate on a 1–5 Likert scale how enjoyable they found each technique.

Participants

We recruited 16 participants aged between 21 and 36 (6 male, 10 female) using a combination of university email lists, word of mouth, and snowball sampling.. Prior to the study we asked participants for simple demographic information and about their familiarity with maps and cartography. Thirteen reported experience using digital map software such as Google Maps. Five reported experience with paper maps, such as hiking trail maps. Two reported no experience with maps or mapping software.

Procedure

During the study, we compared four camera control techniques against each other: *Direct Manipulation*, *Discrete Look-From-At*, *Continuous Look-From-Forward*, and *Continuous Look-From-Towards*. In the case of each *look-from* technique, participants also had access to *direct manipulation* controls.

We asked each participant to complete 64 total trials, including 32 elevation comparison and 32 line-of-sight. We administered trials in blocks of 8, alternating between task types (one elevation comparison task, then one line-of-sight task) and changed the scene after each line-of-sight task (cycling through Canmore, Mount Wutai, Mount Waialeale, and Grand Canyon in that order). A block contained 4 trials of each type and consisted of one full sequence of locations. At the beginning of each trial we reset the participant's camera back to its initial top down position. All participants completed the same set of blocks in the same order. In each block we gave participants access to one of the four interaction techniques, and permuted the ordering of those techniques across participants using a Latin square.

Additionally, prior to starting any study blocks, we administered 3 training trials. In the first training trial we demonstrated how to use each camera control technique and asked them to practice using each technique. In the second training trial we demonstrated to participants a sample elevation comparison task and taught them how to use the study interface. In the third training trial we demonstrated to participants a sample line-of-sight task and again taught them how to use the study interface.

After completing all blocks, we debriefed the participant and asked them to complete a follow up questionnaire in which we asked about their perceived difficulty and enjoyability of each technique, and asked them to provide a short response detailing what they found easy/difficult and most/least enjoyable about each technique. On average the whole procedure took one hour to complete.

7 RESULTS

Due to concern in a variety of fields about the use of null hypothesis significance testing [8, 9], we report our results using estimation techniques and report effect sizes using confidence intervals (CI) as is consistent with recent APA recommendations [2]. To do so, we compute average scores for each participant, then compute 95% bootstrap confidence intervals using the aggregate scores, applying a Bonferroni correction to control for multiple comparisons.

Accuracy

Mean accuracy values (Figure 3–top) ranged betwen 80–90%. The most accurate was *Discrete Look-From-At*, with a mean accuracy of 88% (CI=[82%,95%]) in elevation comparison trials and 87% (CI=[76%,96%]) in line-of-sight trials and the least accurate were *Continuous Look-From-Towards*, with a mean accuracy of 80% (CI=[71%,88%]) in elevation comparisons and *Direct Manipulation* with a mean accuracy of 81% (CI=[66%,91%]) in line-of-sight trials. However, variation across participants was high and overall results showed no conclusive difference in accuracy between conditions.

Time

Mean time values (Figure 3–bottom) ranged from 27s to 40s. The mean fastest techniques were *Direct Manipulation*, with a mean time of 27.6s (CI=[22.7,34.8]) for elevation comparison trials and *Discrete Look-From-At*, with a mean time of 27.8s (CI=[22.3,34.2]) for line-of-sight trials. The slowest technique was *Continuous Look-From-Towards*, with a mean time of 42.2s (CI=[31.3,55.3]) for elevation comparisons and 39.5s (CI=[30.9,49.0]) for line-of-sight trials. However, much like in accuracy there was a large degree of variation across participants and overall results showed no conclusive difference in speed between conditions.

Difficulty and Enjoyability

While no one of the three look-from techniques we tested was unambiguously superior to the others in terms of speed or error rate, our results point to differences related to perceived difficulty and enjoyability. Participants showed a strong and

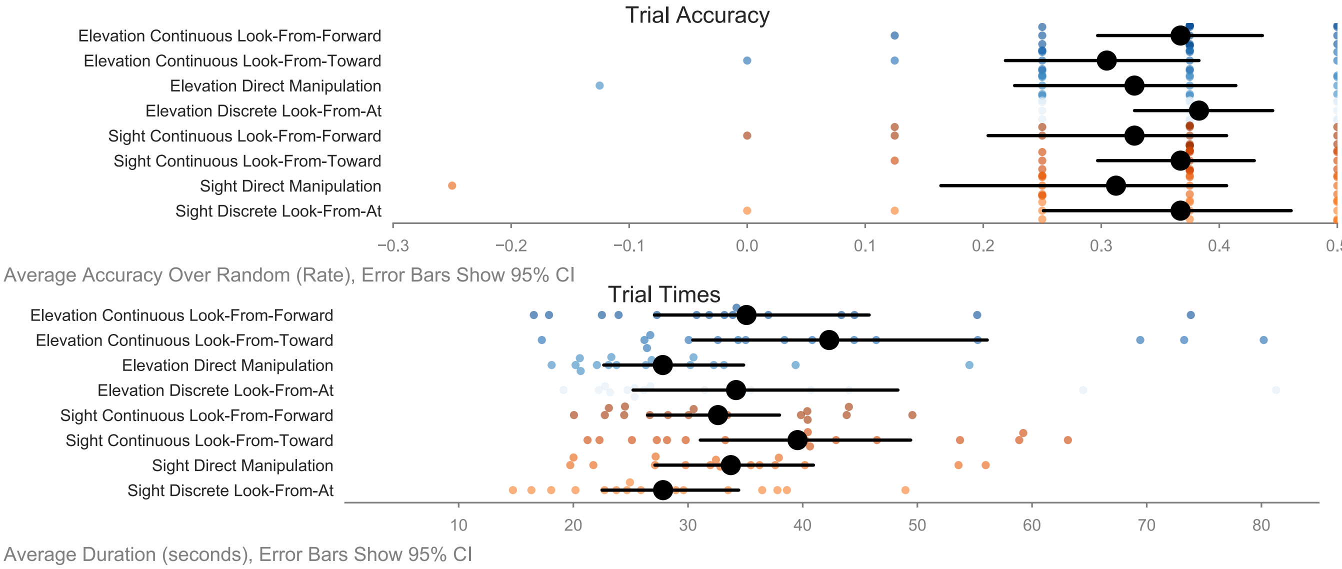


Figure 3: Average trial accuracy (difference from random) and time per participant. Error bars show 95% CIs.

conclusive preference for the simpler look-from technique, and also showed a readiness to integrate all three techniques with direct manipulation interactions.

Directly reflecting the Likert responses (Figure 4) 10 of 16 participants responded that *Discrete Look-From-At* was the easiest technique to use, 3 responded *Direct Manipulation*, 2 *Continuous Look-From-Forward*, and just 1 *Continuous Look-From-Towards*. Similarly, 9 of 16 participants ranked *Discrete Look-From-At* as the most enjoyable control scheme, while 3 each preferred *Continuous Look-From-Forward* and *Continuous Look-From-Towards*, and only 1 ranked *Direct Manipulation* as the most enjoyable technique.

Direct Manipulation. The most common positive comment of *Direct Manipulation* was that participants found the controls familiar (16 participants). While other comments noted specific functions (pan, zoom, or rotate) felt great to use (3 participants). Additionally, P4 noted that it gave them very "fine granularity to control things". Negative comments noted that *Direct Manipulation* was "reliable but slow", required multiple steps (4 participants), was "tedious" (2 participants), or wasn't precise enough (2 participants).

Discrete Look-From-At. Positive comments noted that *Discrete Look-From-At* was effective at very quickly zooming in to their target destination (P6) while rotating them to where they wanted to look, all in one interaction (3 participants). Half of the participants noted that the interaction felt easy and fast to use (8 participants). Participants also remarked that it felt well-suited for line-of-sight tasks (2 participants) and that while on top of one point they could select the other point to quickly flip their view (2 participants). They also noted that the results of their actions were always as expected (5 participants). Negative comments noted that it could take a few interactions to get used to (P7) and that they initially found it disorienting (P2).

Continuous Look-From-Forward. Positive comments about mentioned that *Continuous Look-From-Forward* was simple to use (7 participants) and that it worked as a straightforward alternative to pitch (3 participants) and zoom (3 participants). However, negative comments noted that, like *Direct Manipulation* it required a large number of interactions (8 participants).

Continuous Look-From-Towards. Positive comments noted that *Continuous Look-From-Towards* had great control of the different axes of motion (P11) made easier by a fixed point of rotation (P16). Some participants remarked that it felt very powerful (6 participants) and responsive (3 participants). However, negative comments overwhelmingly noted that the rotational controls could be confusing or disorienting (14 participants).

8 OBSERVATIONS

When completing trials, the most important factor in determining success was how well the participants were able to keep both of the target points visible during a transition. Keeping both points in view allowed participants to maintain their frame of reference, even as they pivoted the camera to more easily assess line-of-sight or elevation. While keeping points in frame was generally straightforward, it was often more challenging near concave and vertical terrain features.

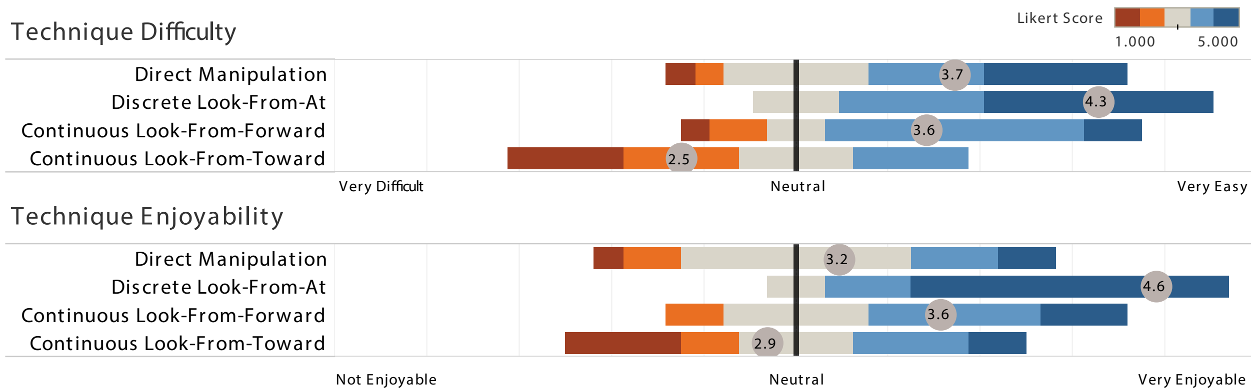


Figure 4: Likert survey results. Circled values indicate average Likert scores.

Navigating Concave Features

Concave terrain such as bowls or valleys can pose navigation problems when the camera comes close to the surface of the terrain. Once the camera dips into a concave region, the sides of the feature tend to obscure the surrounding terrain, hiding landmarks and making it difficult for viewers to maintain a clear frame of reference.

In our study, participants most often became trapped by concave features in the *Direct Manipulation* condition, either by zooming in too quickly or by dramatically pitching the camera down into a valley. With *Discrete Look-From-At* controls, participants sometimes encountered similar issues when choosing a *look-from* location deep in a valley. Once they had entered a concave region, the terrain often obscured one or both of the target points and participants generally found it easier to reset the camera back to a top-down view than to attempt to backtrack and locate them. This occurred most frequently in the Rocky Mountains and Mount Waialeale environments where the terrain had tight concave features that were not readily apparent from above.

Some participants learned to avoid entering concave features in the *Direct Manipulation* condition by pitching the camera slightly before zooming in, mimicking the behavior of *Continuous Look-From-Towards/Forward*. This initial camera tilt made it easier for those participants to identify regions that could potentially trap the camera before dropping into them. In the *Continuous Look-From-Towards/Forward* conditions, meanwhile, participants were typically able to escape concave regions by interactively reversing the transition once they noticed an occlusion.

Navigating Vertical Features

Participants also experienced problems when the target points were placed on the sides of vertical terrain features like cliff faces or canyon walls. In these cases, the vertical features often obscured the target points from most directions, greatly reducing the number camera positions from which a viewer could see both points. This occurred most frequently in the Grand Canyon, where points along the canyon walls were often only visible from above or from a small number of locations within the canyon itself. It also occurred in a few places on Mount Wutai where points along the edge of a plateau could only be seen if the camera was positioned on that side of the feature.

Navigating along these vertical features (especially those in the canyon) was difficult in all conditions. However, it was particularly challenging with *Direct Manipulation* controls, where positioning and angling the camera to see both points simultaneously required considerable manual dexterity and planning. The three *look-from* techniques somewhat mitigated this challenge, since participants could typically ensure that at least one of the points remained continuously in view by choosing a *look-from* location immediately adjacent to it. The *Continuous Look-From-Towards/Forward* techniques also allowed participants to refine and reverse the camera movement and make corrections whenever the vertical feature obscured the second point.

9 DESIGN CONSIDERATIONS

Based on participants' feedback, as well as our own experience iteratively designing and testing numerous different *look-from* techniques (including the three presented in detail here), we provide several considerations for the design of future *look-from* techniques for touch maps on mobile devices.

D1. Keep *look-from* points in frame during transitions. We find that maintaining the visibility of the original selection throughout transitions preserves a common reference point and helps reduce viewers' sense of disorientation. In our implementations, we achieve this by ensuring that the final camera position, falls slightly behind the *look-from* point, allowing that point to stay in view even at the end of the transition.

When a transition involves a specific *look-at* point, keeping it visible is also beneficial. In cases where terrain blocks the line of sight between the *look-from* and *look-at* points, we find that shifting the final camera position both upward and backward can help maintain the visibility of both. Not including this sort of adjustment often results in views where the *look-at* position is entirely occluded.

D2. Sequence camera rotation/pitch to minimize roll. Simultaneously rotating and pitching the camera can result in visible roll where the camera tilts such that the horizon appears diagonal and few points from the initial view remain in frame. This was very apparent in our initial explorations of *look-from-towards*, where pilot testers complained that they had a hard time maintaining a clear frame of reference, especially after performing rotations greater than ±90°. Staging rotations at the beginning of a transition minimizes this issue by aligning the horizon and *look-at* direction so that the camera never appears tilted. This makes it possible for pitch to be smoothly interpolated throughout the animation or adjusted to ensure that the *look-from* location always remains in frame.

D3. Make *look-from* interactions reversible where possible. The inability to reverse after completing a camera move was a major shortcoming of both our *look-from-at* and *look-from-towards* implementations, both of which require viewers to use direct manipulation interactions or a hard reset to return to a top-down view. Reversible *look-from* interactions, meanwhile, can make it easier to recover from mistakes and support more seamless transitions between top-down, oblique, and ground-level views in both directions.

D4. Preserve compatibility with existing direct manipulation techniques where possible. While *look-from* techniques can streamline common transitions between top-down, oblique, and ground-level views, they are less suited to search and navigation tasks that require repeated zooming and panning. Moreover, direct manipulation interactions can be very useful for making small adjustments to views produced using *look-from* interactions, allowing viewers to independently pan, zoom, or pitch the camera after transitioning down to an oblique view or rotating around a ground-level point of interest.

Based on our experience, adding *look-from* navigation as quasi-modes initiated via a distinct interaction such as a long-press or double-tap facilitates these transitions well, allowing viewers to quickly enter and exit a *look-from* navigation mode while retaining access to the standard vocabulary of map interactions. However, the vocabulary of simple gestures is relatively limited, and many mapping tools and operating systems already assign functions to gestures like double-taps (often mapped to zoom) or long presses (often used to trigger context menus). This limitation may make it difficult to add quasi-modal *look-from* gestures to some existing systems without relying on additional hardware inputs or interface elements. Moreover, it may make it difficult to support multiple different kinds of *look-from* gestures simultaneously in an interface.

10 ALTERNATIVE CONTROL SCHEMES

While we considered just three examples of *look-from* techniques, variants of these approaches may also merit further examination—especially approaches that relax the requirement of using a single touch gesture. For example, using multiple touches (either sequentially or bimanually) could permit continuous versions of *look-from-at*, where one touch sets the *look-from* point, and a second touch gesture sets the *look-at* location and interactively advances the camera.

There also exists a rich space of alternative input schemes for more complex *look-from* techniques, including alternatives to our version of *Continuous Look-From-Towards*. In our implementation, using the angle between the cursor and the initial touch-down to control rotation often resulted in confusion, since rotations greater than 90° inverted viewers' frame of reference. Once at ground level, viewers often expected to be able to rotate right or left by dragging in the corresponding direction, when in fact they needed to drag clockwise or counter-clockwise around the initial point. A Cartesian input scheme that used forward/backward movement to move the camera along the path and left/right motion to control rotation might align more closely with common mental models of camera movement.

Finally, although we considered input schemes that rely only on finger position, commodity touch-enabled devices generally also capture touch pressure and information that can be used to reliably estimate finger pitch and pose [35]. These additional input factors could support even more nuanced transitions, such as using finger pitch or pressure to dynamically vary the vertical height of the camera path.

11 CONCLUSION

Look-from camera control techniques can serve as lightweight additions to *Direct Manipulation* controls, allowing viewers to simplify common and challenging map navigation tasks using simple gestures. By integrating camera zoom, pan, and pitch and building camera paths based on one or two viewer-specified points on the map, these techniques can support smooth transitions between top down, oblique, and ground-level views. In this paper, we developed three variants of *look-from* controls: *Discrete Look-From-At*, *Continuous Look-From-Forward*, and *Continuous Look-From-Towards*. Our initial studies highlight the utility of *Discrete Look-From-At* and suggest four design considerations for successful *look-from* interactions, as well as opportunities for future techniques.

ACKNOWLEDGMENTS

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [RGPIN-2016-04564] as well as our colleagues who helped test and give feedback on the design of the interaction techniques.

REFERENCES

[1] Pablo Abend, Tristan Thielmann, Ralph Ewerth, Dominik Seiler, Markus Mühling, Jörg Döring, Manfred Grauer, and Bernd Freisleben. 2012. Geobrowsing behaviour in Google Earth-A semantic video content analysis of on-screen navigation. *GI_Forum* (2012), 2–13.

[2] American Psychological Association. 2017. *Publication Manual of the American Psychological Association* (sixth ed.). American Psychological Association, Washington.

[3] Henrik Buchholz, Johannes Bohnet, and Jürgen Döllner. 2005. Smart and physically-based navigation in 3D geovirtual environments. In *Ninth International Conference on Information Visualisation (IV'05)*. IEEE, 629–635. https://doi.org/10.1109/IV.2005.117

[4] Jonas Buddeberg, Bernhard Jenny, and Wesley Willett. 2017. Interactive shearing for terrain visualization: an expert study. *GeoInformatica* 21, 3 (jul 2017), 643–665. https://doi.org/10.1007/s10707-016-0283-9

[5] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. 1991. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems* 9, 2 (apr 1991), 99–122. https://doi.org/10.1145/123078.128726

[6] David B. Christianson, Sean E. Anderson, Li-wei He, David H. Salesin, Daniel S. Weld, and Michael F. Cohen. 1996. Declarative Camera Control for Automatic Cinematography. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1 (AAAI'96)*. AAAI Press, 148–155. http://dl.acm.org/citation.cfm?id=1892875.1892897

[7] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. 2008. A review of overview+detail, zooming, and focus+context interfaces. *Comput. Surveys* 41, 1 (dec 2008), 1–31. https://doi.org/10.1145/1456650.1456652

[8] Geoff Cumming. 2014. The new statistics: Why and how. *Psychological science* 25, 1 (2014), 7–29.

[9] Pierre Dragicevic, Fanny Chevalier, and Stephane Huot. 2014. Running an HCI Experiment in Multiple Parallel Universes. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 607–618. https://doi.org/10.1145/2559206.2578881

[10] Jörg Edelmann, Andreas Schilling, and Sven Fleck. 2009. The DabR - A multitouch system for intuitive 3D scene navigation. In *2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*. 1–4. https://doi.org/10.1109/3DTV.2009.5069671

[11] James D. Eynard and Bernhard Jenny. 2016. Illuminated and shadowed contour lines: improving algorithms and evaluating effectiveness. *International Journal of Geographical Information Science* (apr 2016), 1–21. https://doi.org/10.1080/13658816.2016.1144885

[12] Michael Gleicher and Andrew Witkin. 1992. Through-the-lens camera control. In *ACM SIGGRAPH Computer Graphics*, Vol. 26. 331–340. https://doi.org/10.1145/142920.134088

[13] Benjamin Hagedorn and Jürgen Döllner. 2008. Sketch-based navigation in 3D virtual environments. In *Smart Graphics*. Springer Berlin Heidelberg, Berlin, 239–246. https://doi.org/10.1007/978-3-540-85412-8_23

[14] Andrew J. Hanson and Eric A. Wernert. 1997. Constrained 3D navigation with 2D controllers. In *Proceedings of the 8th Conference on Visualization '97 (VIS '97)*. IEEE Computer Society Press, Los Alamitos, CA, USA, 175–ff. http://dl.acm.org/citation.cfm?id=266989.267052

[15] Dieter Hildebrandt and Robert Timm. 2014. An assisting, constrained 3D navigation technique for multiscale virtual 3D city models. *GeoInformatica* 18, 3 (jul 2014), 537–567. https://doi.org/10.1007/s10707-013-0189-8

[16] Robert J. K. Jacob, Linda E. Sibert, Daniel C. McFarlane, and M. Preston Mullen. 1994. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction* 1, 1 (mar 1994), 3–26. https://doi.org/10.1145/174630.174631

[17] Jacek Jankowski and Martin Hachet. 2015. Advances in interaction with 3D environments. In *Computer Graphics Forum*, Vol. 34. 152–190. https://doi.org/10.1111/cgf.12466

[18] Nico Li, Wesley Willett, Ehud Sharlin, and Mario Costa Sousa. 2017. Visibility perception and dynamic viewsheds for topographic maps and models. In *Proceedings of the 5th Symposium on Spatial User Interaction - SUI '17*. ACM Press, New York, 39–47. https://doi.org/10.1145/3131277.3132178

[19] Jock Mackinlay, Stuart K. Card, and George G. Robertson. 1990. A semantic analysis of the design space of input devices. *Human-Computer Interaction* 5, 2 (1990), 145–190. https://doi.org/10.1207/S15327051HCI0502&3_2

[20] Jock D. Mackinlay, Stuart K. Card, and George G. Robertson. 1990. Rapid Controlled Movement Through a Virtual 3D Workspace. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH Computer Graphics)*. ACM, New York, NY, USA, 171–176. https://doi.org/10.1145/97879.97898

[21] Anthony Martinet, Gery Casiez, and Laurent Grisoni. 2012. Integrality and separability of multitouch interaction techniques in 3D manipulation tasks. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (mar 2012), 369–380. https://doi.org/10.1109/TVCG.2011.129

[22] James McCrae, Igor Mordatch, Michael Glueck, and Azam Khan. 2009. Multiscale 3D Navigation. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games (I3D '09)*. ACM, New York, NY, USA, 7–14. https://doi.org/10.1145/1507149.1507151

[23] Richard J. Phillips, Alande Lucia, and Nicholas Skelton. 1975. Some objective tests of the legibility of relief maps. *The Cartographic Journal* 12, 1 (jun 1975), 39–46. https://doi.org/10.1179/caj.1975.12.1.39

[24] Jeffrey S. Pierce and Randy Pausch. 2004. Navigation with place representations and visible landmarks. In *IEEE Virtual Reality 2004*. 173–288. https://doi.org/10.1109/VR.2004.1310071

[25] L. M. Potash, J. P. Farrell, and T. S. Jeffrey. 1978. A technique for assessing map relief legibility. *The Cartographic Journal* 15, 1 (jun 1978), 28–35. https://doi.org/10.1179/caj.1978.15.1.28

[26] Jef Raskin. 2000. *The humane interface : new directions for designing interactive systems.* Addison-Wesley. 233 pages.

[27] Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. 2009. A screen-space formulation for 2D and 3D direct manipulation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*. ACM Press, New York, 69. https://doi.org/10.1145/1622176.1622190

[28] Bruno Marques Ferreira da Silva, Selan Rodrigues dos Santos, and Jauvane Cavalcante de Oliveira. 2009. Using a physically-based camera to control travel in virtual environments. *Symposium on Virtual and Augmented Reality (SVR)*.

[29] Russell Turner, Francis Balaguer, Enrico Gobbetti, and Daniel Thalmann. 1991. Physically-based interactive camera motion control using 3D input devices. In *Scientific Visualization of Physical Phenomena*. Springer Japan, Tokyo, 135–145. https://doi.org/10.1007/978-4-431-68159-5_8

[30] Benjamin Walther-Franks, Marc Herrlich, and Rainer Malaka. 2011. A multi-touch system for 3D modelling and animation. In *Smart Graphics. SG 2011, Lecture Notes in Computer Science, vol 6815*, Dickmann L. et al. (Ed.). Springer, Berlin, Heidelberg, Berlin, Heidelberg, 48–59. https://doi.org/10.1007/978-3-642-22571-0_5

[31] Ming Wan, Frank Dachille, and Arie Kaufman. 2001. Distance-field based skeletons for virtual navigation. In *Proceedings of the Conference on Visualization '01 (VIS '01)*. IEEE Computer Society, Washington, DC, USA, 239–246. http://dl.acm.org/citation.cfm?id=601671.601708

[32] Colin Ware and Steven Osborne. 1990. Exploration and virtual camera control in virtual three dimensional environments. *ACM SIGGRAPH Computer Graphics* 24, 2 (feb 1990), 175–183. https://doi.org/10.1145/91394.91442

[33] Wesley Willett, Bernhard Jenny, Tobias Isenberg, and Pierre Dragicevic. 2015. Lightweight relief shearing for enhanced terrain perception on interactive maps. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, New York, 3563–3572. https://doi.org/10.1145/2702123.2702172

[34] Dongbo Xiao and Roger Hubbold. 1998. Navigation guided by artificial force fields. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, New York, 179–186. https://doi.org/10.1145/274644.274671

[35] Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 47–50. https://doi.org/10.1145/2817721.2817737

[36] Georgios N. Yannakakis, Héctor P. Martínez, and Arnav Jhala. 2010. Towards affective camera control in games. *User Modeling and User-Adapted Interaction* 20, 4 (oct 2010), 313–340. https://doi.org/10.1007/s11257-010-9078-0

[37] Robert Zeleznik and Andrew Forsberg. 1999. UniCam—2D gestural camera controls for 3D environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics - SI3D '99*. ACM Press, New York, 169–173. https://doi.org/10.1145/300523.300546