GravNav: Using a Gravity Model for Multi-Scale Navigation

Waqas Javed, Sohaib Ghani, and Niklas Elmqvist
Purdue University
West Lafayette, USA
{wjaved, sghani, elm}@purdue.edu

ABSTRACT

We present gravity navigation (GravNav), a family of multi-scale navigation techniques that use a gravity-inspired model for assisting navigation in large visual 2D spaces based on the interest and salience of visual objects in the space. GravNav is an instance of topology-aware navigation, which makes use of the structure of the visual space to aid navigation. We have performed a controlled study comparing GravNav to standard zoom and pan navigation, with and without variable-rate zoom control. Our results show a significant improvement for GravNav over standard navigation, particularly when coupled with variable-rate zoom. We also report findings on user behavior in multi-scale navigation.

Keywords

Topology-aware navigation, guided navigation, zooming, panning, multi-scale spaces, space-scale diagrams.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Interaction styles*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*

1. INTRODUCTION

Human attention has many similarities to the theory of gravity: regions of high visual interest attract attention just like moths are drawn to an open flame at night, and interest wanes with distance. For example, the concept of *visual salience* as a measure of the perceptual distinctiveness of an object is key in directing our visual attention [17, 23, 25]. Furthermore, Fisher [8] found that map attention is centered on coastlines, most likely due to their irregularity and corresponding high visual interest. Despite these facts, it is curious that so few navigation techniques for computer applications take advantage of this phenomenon. Standard scrolling, panning, and zooming techniques all disregard the structure and topology of the underlying visual space, essentially making it equally easy (or equally difficult) to navigate to an empty cornfield on an interactive map as it is to navigate to an urban area.

AVI '12, May 21-25, 2012, Capri Island, Italy

In this paper, we attempt to remedy this state of affairs by presenting the concept of *gravity navigation* (GravNav) where an interest-based gravity model is used to ease navigation in large visual spaces. The basic idea is to compute an attention-gravity vector given the user's location in the visual space and the surrounding items of interest, and then use this vector to guide navigation. Similar to semantic and sticky pointing [4, 6, 20], the intuition is to speed up navigation over empty areas, and to slow it down around regions of interest. In particular, we believe the technique is especially well-suited for assisting *micro-level* navigation in multi-scale spaces [11], where small disparities at a distant magnification level may have massive impact when zooming into the space to see details and interact with visual objects.

The applicability of this family of gravity navigation techniques encompass all interactive applications that incorporate large visual spaces, such as interactive maps (e.g., Google or Bing Maps), large-scale zoomable visualizations, or detailed CAD blueprints and drawings, etc. To validate the utility and efficiency of gravity navigation, we performed a controlled user study for multi-scale navigation where we compared performance for two instantiations of the concept—gravity zooming and panning—with the standard zooming and panning operations that would be found today in the above interactive applications. We also included the OrthoZoom [2] zoom-gain interaction technique as a condition in our experiment. Our results are convincingly in favor of the new technique, showing significantly decreased completion times across all navigation scales and distances. We also report on navigation behavior for different conditions using space-scale diagrams [11], showing that gravity navigation is particularly suited for combining with OrthoZoom due to the reduced need for clutch operations.

2. RELATED WORK

Our work here draws upon existing work on general and multi-scale navigation, assisted navigation, and pointing.

2.1 General and Multi-Scale Navigation

Pan and zoom are the canonical navigation operations [11, 22], allowing the user to change the position and the magnification of the viewport, respectively. Scrolling is a special instantiation of panning restricted to one dimension, such as for navigating in large documents. Much work has been dedicated to exploring effective pan and zoom operations, including space-scale diagrams for modeling navigation behavior [11], optimal tradeoffs between zoom and pan [24], and effective methods for view navigation [10].

Most pan and zoom operations take place in *multi-scale spaces*, i.e., spaces that contain objects with scale-dependent appearances. For example, a map database provides more detail as the user zooms from a world view down to a city, neighborhood, or street.

In fact, switching between different visual representations depending on magnification level is known as *semantic zooming* [22], and could for example be used to show a document as an icon when viewed from a distance, and its actual text when viewed at high magnification. However, this multi-scale nature also makes navigation difficult since crucial navigational cues may be missing at certain levels of scale, a phenomenon known as "desert fog" [18].

Several multi-scale navigation techniques have been proposed recently. Speed-dependent automatic zooming (SDAZ) [15] is a method for automatically zooming out a document viewport while scrolling in order to maintain a constant visual flow. Guiard et al. [14] show that many of the pointing behaviors employed in fixed-scale spaces also translate to multi-scale spaces. Most recently, Appert's and Fekete's OrthoZoom [2] scroller supports variable-rate zooming across large distances and levels of scale.

2.2 Assisted Navigation

All space is not created equal: some regions in a multi-scale world will invariably be more important than other regions. For example, in a large, zoomable node-link diagram of a social network, the nodes (representing actors) and the links (representing connections between actors) will be more important than the white space separating these entities. This is the basic idea behind *topology-aware navigation* [21], where the structure of the space is used to inform and assist navigation. Moscovich et al. propose two such techniques for navigating graphs: link sliding, where users navigate by "sliding" along the visual links connecting the nodes, and bring-and-go, where the local neighborhood of a particular node can temporarily be brought into close proximity to it. Ghani et al. [13] instead use dynamic map insets for this purpose.

Ishak and Feiner propose *content-aware scrolling* [16], where scrolling in a document is modulated by the document content itself. While scrolling in their approach is seen as movement along a one-dimensional path, the path itself as well as the speed and zoom level is computed to cover the important regions in the document.

This kind of assisted navigation has also been employed for 3D navigation in virtual environments. Galyean [12] uses a "river analogy" where the user is pulled along a pre-computed path, while still being able to deviate slightly from it. Elmqvist et al. [7] study the impact of assisted navigation on recall of a 3D world, and found that fully or semi-assisted navigation allowed the user to focus more cognitive resources on the world instead of on navigation.

2.3 Pointing

Pointing is the acquisition of targets on a screen using a mouse or other pointing device. Most pointing techniques are evaluated using Fitts' law [9, 19]. While pointing is not strictly relevant to navigation, Guiard et al. [14] recently showed that Fitts' law also applies to *multi-scale pointing*, which is essentially a navigation task. Consequently, several recent navigation studies have been based on this fact [2, 14]. We adopt this practice in our user study.

The pointing equivalent of gravity navigation is *semantic pointing* [4], where knowledge about the underlying user interface elements on the screen can be used to improve pointing performance by varying the control-display ratio between mouse and pointer movement. Other related work includes sticky targets [20, 27] that ease acquisition, force-enhanced targets [1] that attract the pointer, and extensions to 3D picking in games and virtual environments [6]. The difference between sticky/semantic/force-enhanced pointing and gravity navigation is that the former operates on pointing in screen space whereas our technique operates on multi-scale navigation operations in large visual spaces. However, since the techniques are orthogonal, they can certainly be combined.

3. GRAVITY NAVIGATION

The intuition behind gravity navigation (GravNav) is to use a degree-of-interest (DOI) function for the local neighborhood of visual items around the current position of the viewport to calculate a "gravity vector" that models the attention of the user. This vector is then used to modulate user interaction in motor space to influence navigation. The result is that objects in the visual space create "gravity wells" that attract the user's viewport during navigation (Figure 1). Below we describe each of these phases in turn.

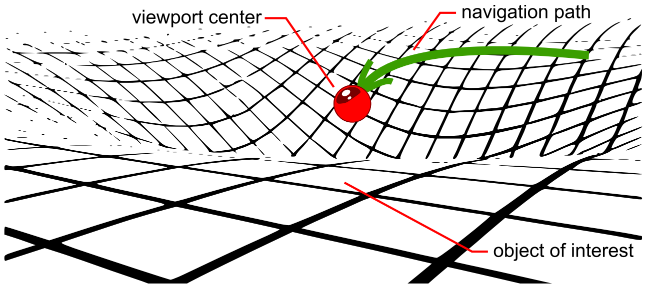


Figure 1: Illustration of an attention gravity well for an object.

3.1 Degree of Interest

GravNav uses a degree-of-interest (DOI) function to model human attention in a visual space. This takes the shape of a function $DOI(p) \in \mathbb{R}$ that returns the interest for any position $p \in \mathbb{R}^2$ on the visual space. Most commonly, the DOI function is defined on a set S of discrete visual shapes $s \subseteq \mathbb{R}^2$ in the space as $DOI(s) = 1$.

For a structured visual space such a map, we can utilize our higher-level knowledge of the space and its salient features to define the DOI function. Consider the following example function for a map consisting of sets of cities C and roads R:

$$DOI_{map}(s) = \begin{cases} \text{population}(s) & s \in C \\ \text{roadSize}(s) & s \in R \\ 0 & \text{otherwise} \end{cases}$$

In this example, we assign interest to points inside a city based on its population, and interest to points along a road based on the road size (normalized appropriately depending on the application).

While our emphasis in this work is on assisting navigation in structured visual spaces, we may also want to apply gravity navigation to entirely unstructured visual spaces. In such situations, we calculate the *visual saliency* [23, 25] of the graphical features in the space to find the regions that are perceptually important to the human visual system. In other words, we use a function of the format $DOI_{saliency}(p) = \text{saliency}(p)$. There are several approaches to calculating saliency, including gradient magnitude, entropy, and image complexity (examples include [5, 17, 26]). In the example in Figure 2, we simply use gradient magnitude (an edge detection algorithm) to calculate saliency for a photograph, and then segment the image into regions. This approach can be applied to any image.

Given an image segmented into a set of regions S with a DOI function defined for every point $p \in \mathbb{R}^2$, we calculate the interest of each region s by simply integrating over space:

$$DOI(s) = \int_{p \in s} DOI(p)\, dp$$

3.2 Gravity Model

Given a DOI function, we now want to predict the user's attention in the current view. We make the reasonable assumption that

Figure 2: Visual saliency map (right) for a photograph (left) calculated using edge-detection based on gradient magnitude.

during navigation, the user's attention is focused on the local neighborhood around the cursor (mouse pointer or touch point, respectively), represented by a point p_c in screen coordinates. We now calculate a resulting *attention gravity vector* $\overline{g}(p_c)$ as

$$\overline{g}(p_c) = \sum_{s_i \in S} \frac{DOI(s_i)}{|\overline{d_i}|^2} \cdot \frac{\overline{d_i}}{|\overline{d_i}|}$$

where $\overline{d_i}$ is a 2D vector defined as $\overline{d_i} = \overline{p_c} - \overline{p_i}$, and $\overline{p_i}$ is the point in s_i (in screen coordinates) that is closest to the cursor position p_c. This is an adaptation of the law of gravity, which has the below form for the gravitational force F between two bodies with mass m_1 and m_2 at a distance r from each other:

$$F = G\frac{m_1 m_2}{r^2}.$$

Figure 3 shows a visualization of the gravity field sampled at regular intervals in a simple visual space consisting of five square shapes with connecting paths. Unlike the gravity wells caused by heavenly bodies, our attention gravity also has gravity troughs arising from paths in the visual space.

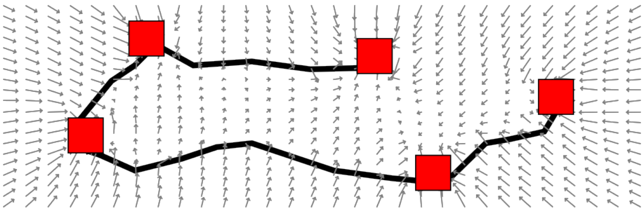


Figure 3: Gravity vector field for a sample scene. Paths cause gravity troughs in the same way shapes cause gravity wells.

In a real implementation for a very large visual space, it is often practical to use a spatial index (such as a quadtree) to eliminate distant shapes from the computation; these will have a negligible impact on the resulting vector anyway due to the inverse proportion to the square of the distance.

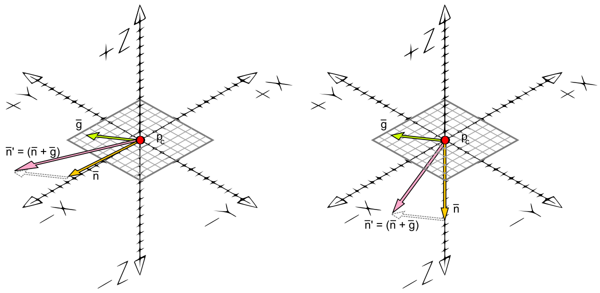
3.3 Motor Space Model

The last step of the process is to use the attention gravity vector $\overline{g}$ computed above from the DOI function to guide multi-scale navigation. Similar to semantic [4], sticky [20, 27], and force-enhanced [1] pointing, our approach is to manipulate the control-display (CD) mapping to achieve this. We identify two main approaches to achieve this:

- **CD gain:** Modulate the motor space input vector by the projection of the gravity attention vector. This mechanism is used in semantic pointing, and will make navigation towards salient targets easier, and navigation away harder.

- **CD direction:** Add the gravity attention vector to the motor space input vector. Similar to force-enhanced pointing, this will actively change not just the magnitude of the motor interaction, but also its direction, thereby actively guiding the user's navigation towards salient features.

For both approaches, the scaling factor applied to the gravity attention vector will be central for achieving the desired effect. This scaling factor should be determined empirically.



(a) Gravity panning. (b) Gravity zooming.

Figure 4: Gravity zoom and pan implementation. The attention gravity vector $\overline{g}$ (green) is operating on the user's navigation vector $\overline{n}$ (yellow) from position p_c, resulting in $\overline{n}'$ (pink).

3.4 Implementation: Gravity Zoom and Pan

We have implemented GravNav as two navigation techniques: *gravity pan* and *gravity zoom*. Figure 4 shows our approach. For panning, we take the user's input vector in motor space $\overline{n} = (\Delta x, \Delta y)$ (where Δx and Δy are the cursor position deltas in motor space) and simply add the attention gravity vector $\overline{g}$, suitably scaled (we take the logarithm of the magnitude), to create an actual navigation vector $\overline{n}'$ that is used to transform the viewport's position.

For zooming, we view the navigation vector $\overline{n}$ from motor space as a 3D vector pointing into (for zoom-in) or out of (for zoom-out) the screen along the z (scale) axis (i.e., $\overline{n} = (0, 0, \Delta x)$ if using horizontal cursor movement to zoom). Again, we perform a vector addition with the attention gravity vector, resulting in an actual navigation vector $\overline{n}'$. It is often advantageous, however, to not enable gravity effects for zoom-out operations.

4. CONTROLLED STUDY

We conducted a user study to poll the effectiveness of gravity navigation in comparison to traditional multi-scale navigation techniques. To ensure that we followed an accepted and well-known design, we based our experiment on the OrthoZoom [2] user study with three important differences:

- **Constant target size:** In the OrthoZoom experiment, Appert and Fekete varied the target width to control pointing difficulty based on Fitts' law [9, 19]. However, they found no such interaction. We decided against variable target width due to our emphasis on navigation and not pointing.

- **Two-dimensional:** OrthoZoom is a one-dimensional zooming technique, and for this reason, the OrthoZoom experiment is restricted to the user panning vertically and then zooming in (using OrthoZoom or SDAZ [15]) to be able to select the target. However, because gravity zooming is a two-dimensional navigation technique, we instead allow for panning the view in both vertical and horizontal directions. Nevertheless, we included a 2D-adaptation of OrthoZoom as an independent factor in itself in the experiment.

- **Distractors:** In contrast to the OrthoZoom experiment, in our experiment we included distractor objects that were black squares of the same size and having the same gravity as that of the target. This decision is motivated by the fact that the resultant gravity vector, and hence the technique, is effected by the presence of these distractor objects. Further, we wanted to measure the performance of gravity zooming in a realistic scenario, where a target is normally surrounded by many distractors and the number of these distractors generally increases with the distance from the target.

In this section, we describe our study participants, equipment, and methods, followed by results in the next section.

4.1 Participants

We recruited 12 participants (7 male, 5 female) from the student population at our university (ages 21 to 28, average 25 years). Participants were all self-selected volunteers, had normal or corrected-to-normal vision, were not color blind (self-reported), and were paid $10 upon completing a full session. No particular skills were required other than basic knowledge of operating a computer.

4.2 Apparatus

The experiment was conducted on a desktop computer with 3 GHz processor, 4 GB RAM, and Microsoft Windows XP. The experimental application was 800×800 pixels in size, centered on a 19" LCD monitor set to 1280×1024 resolution. The participants were provided a standard mouse to interact with the application.

4.3 Task

The goal with our experiment was to study the performance of gravity navigation in what we call *micro-level navigation*: situations where the user has decided on a destination, and needs assistance navigating there. This is to contrast with *macro-level navigation*, where the task is wayfinding in the visual space to determine the next destination. While micro-level navigation does not capture the full spectrum of applications that gravity navigation supports, we think it is an example of where the technique is most useful.

Based on this reasoning, we designed a task where the objective was to navigate the viewport from an initial zoomed-out overview to a zoomed-in detail view of a particular target and then click on it. The target was surrounded by distractor objects, the number of which was linearly dependent on the distance from the target. In particular, we placed $2*N$ objects at a random position on the periphery of every N-th imaginary circle, centered at the target with a radius of $2^N W$ (where W is the constant target width, $N = \{1, 2, ..., ID\}$, and ID is a factor used in the experiment). It is important to note that the number of distractor objects inside the viewport depends on the zoom factor and the position of the viewport. However, we used all the distractor objects, whether inside or outside the viewport, while calculating the resultant gravity vector.

At the start of each trial, the target was shown somewhere along the perimeter of an imaginary circle centered on the screen and with a radius corresponding to 45% of the viewport dimension. The center point of the view was placed a specific target distance away from the view (governed by the Index of Difficulty ID) and zoomed out sufficiently to bring the target in view on this imaginary screen-space circle. The intuition behind the task was to force the user to navigate the center of the viewpoint from different distances ID through pan and zoom operations until the target was reached.

The target itself was a square the size of 10% of the viewport size at 1:1 zoom factor; the square was red in color whenever viewed at less than full scale factor, and blue when viewed at 1:1 zoom factor or closer. The target could only be clicked while in the blue stage, i.e., when the user's view was zoomed in to its natural 1:1 scale. This was done to force the user to zoom and pan the view the full distance from overview to detail view. Because the target was too small to see in the zoomed-out views, we added a collection of successively larger concentric rings centered around the target; these were placed and sized so that at least one was always visible at any zoom level, and gave participants visual feedback about their navigation when zooming and panning around the target. Furthermore, these rings also give an indication of the target's position whenever it fell outside the viewport, similar to the Halo technique [3].

4.4 Factors

We included the following three factors in our experiment.

4.4.1 Navigation Technique (T)

The obvious factor to control was the navigation technique used:

- **Standard navigation (SN):** We used a standard pan and zoom implementation where a left mouse drag panned the view in world space proportional to the amount of pixels dragged in screen space (i.e., so that the space moved at the same speed as the cursor movement), and a right mouse drag zoomed the view in (right horizontal movement) and out (left horizontal movement). As is consistent with most standard pan and zoom implementations, the zoom operation caused the view to zoom towards the initial zoom point of the mouse cursor (not the center of the viewport). The mouse wheel was disabled for zooming to allow for variable zoom control.

- **Gravity navigation (GN):** Here we used the gravity zooming and panning operations. The pan and zoom interactions were similar to the standard navigation case above (left drag for pan and right drag for zoom), but our gravity model modulated both the speed and direction of all operations. We set constants for the gravity model based on pilot testing.

4.4.2 Zoom Control Technique (Z)

We suspected that zoom control may be a significant factor for navigation performance because of the tradeoff between high speed and error correction. In other words, while high zoom speed will obviously allow for shorter completion times and thus more efficient navigation, it may also cause significant time costs when correcting for navigation errors (i.e., zooming into a position that initially appears close to the target may eventually result in the target leaving the edge of the screen when zoomed sufficiently). For this reason, we chose to include two levels for this factor:

- **Standard zoom control (SZ):** Here the zoom interactions (dragging right for zoom in, dragging left for zoom out) caused a constant amount of zooming (zoom-constant = 0.01) for the same movement along the horizontal axis.
- **OrthoZoom (OZ):** More recent work in multi-scale navigation (see the Related Work section) allow for variable-rate zoom control. We adopted the OrthoZoom technique [2] since it has recently been shown to be the most efficient such technique compared to, e.g., SDAZ [15]. In our implementation, as in the original paper, drag movement along the horizontal axis is scaled by the pointer's vertical distance from the original pointer position and transformed into zoom operations. In other words, the view can be zoomed in faster by

simply increasing the vertical distance from the initial button press position while dragging horizontally, and slower by decreasing the distance. We used the default constants for OrthoZoom as presented by Appert and Fekete.

We should note that OrthoZoom was originally labeled as a one-dimensional scroll technique, whereas we use it for zooming in a 2D visual space. The adaptation is simply that the user selects a position to zoom into through the initial mouse position, and then uses the zoom technique to control the one-dimensional zoom factor.

4.4.3 Index of Difficulty (ID)

The Index of Difficulty (ID) of a navigation trial captures the distance between the initial position of the viewport (based on the center point of the viewport) and the target to navigate to. Using Fitts' law [9, 19], we can simply find the distance D to travel as $D = 2^{ID}W$, where W is the (constant) target width. A recent trend has been to study very large indices of difficulty that correspond to navigation in multi-scale spaces. Therefore, we adopted the same indices as in the OrthoZoom study, i.e., 10, 15, 20, 25, and 30 bits.

4.5 Experimental Design

We used a full-factorial within-participants design:

	12	participants
×	2	**Navigation Technique** T (SN, GN)
×	2	**Zoom Control Technique** Z (SZ, OZ)
×	5	**Index of Difficulty** ID (10, 15, 20, 25, 30)
×	5	repetitions
	1200	Total trials (100 per participant)

We organized the trials in blocks based on zoom control technique so that participants would only have to utilize one type of zoom technique at a time. Block order and the order of navigation technique and ID levels within each block were randomized across participants to counteract learning effects. During the experiment we collected the time it took participants to complete a task. We also collected full cinematic interaction data during each trial.

4.6 Hypotheses

H1 *GN will be faster than SN.* We believe that gravity navigation will outperform standard navigation because gravity navigation corrects for the small pointing errors that grow to large navigation errors over long distances.

H2 *OZ will be faster than SZ.* Based on previous results [2] where OrthoZoom was superior to SDAZ, we are confident that OrthoZoom will also outperform constant-rate control.

H3 *GN will benefit more from OZ than SN.* We speculate that the combination of gravity navigation and OrthoZoom will be particularly effective since it will allow the user to make large-scale zoom movements without worrying about overshooting or losing the target.

4.7 Procedure

Participants first received general instructions about the experiment and the trials. They were then given a demonstration on how to solve a trial using the different zoom techniques. As is customary in similar experiments, we did not inform participants that the experiment incorporated a navigation assistance technique.

After the demonstration, participants received five training trials in each combination of zoom control Z and navigation technique T. They were asked whether they felt ready to move on to the timed trials; no participant stated that they were not. Each trial started with an on-screen instruction for the zoom control technique to use (standard or OrthoZoom), and users were asked to click a target centered on a screen to proceed. This ensured that the user's cursor was located in a neutral position in the center of the screen at the start of a trial. They could also rest indefinitely between trials while on this screen. After clicking the target, the timer was started and the task was shown to the participant. Clicking the blue target ended the trial, stopping the timer and recording the data.

A typical experimental session lasted 40 minutes, including training. After finishing all trials, participants were issued a questionnaire asking them to give feedback and comments on the experiment and the techniques.

4.8 Design Decisions

A large number of decisions have to be made when designing any user study. Here we discuss the major such decisions and what factors motivated us in our choice for each of them:

- **Pointing task:** Multi-scale navigation experiments in the literature have many similarities to pointing experiments, i.e, they are often based on Fitts' law [9, 19] and use metrics such as distance, target width, and completion time. While we chose a similar design for this evaluation, it is important to remember that our work is on navigation and not pointing techniques. We are asking the user to navigate a viewport through a multi-scale space, not primarily to click on a particular target on the screen. As such, our emphasis is on understanding the dynamic behavior of users when zooming and panning through such spaces rather than their pointer's movement across the screen.

- **No pointing assistance:** Pointing techniques such as semantic, sticky, and force-enhanced pointing help in acquiring the target to navigate to, and one can argue that such techniques therefore should be included in the evaluation. There are two reasons for excluding such techniques: we felt that (1) pointing assistance is orthogonal to navigation assistance, and therefore beyond the scope of this experiment; and that (2) pointing assistance techniques are not defined for multi-scale spaces and, in particular, for situations where targets are outside the viewport. We still expect pointing assistance techniques to result in even better task performance.

5. RESULTS

Our results come in several forms: quantitative completion times, dynamic navigation behavior, and subjective feedback. Below we report on each of these categories in turn.

Factors	**df, den**	**F**	**p**
Navigation technique (T)	1, 11	54.648	***
Zoom control technique (Z)	1, 11	6.0184	*
Index of Difficulty (ID)	4, 44	170.23	***
T * Z	1, 11	29.611	***
T * ID	4, 44	26.873	***
Z * ID	4, 44	7.0477	***
T * Z * ID	4, 44	9.7175	***

* = $p \leq 0.05$, *** = $p \leq 0.001$.

Table 1: Effects of factors on completion time (ANOVA).

5.1 Completion Times

We averaged the completion times for each condition and participant across repetitions and performed a repeated-measures analysis of variance (RM-ANOVA, all assumptions of the test were fulfilled). Figure 5 shows boxplots of completion times for different conditions and Table 1 gives the main and interaction effects of the factors on completion time; as can be seen in the table, all effects were significant. In general, gravity navigation was an average of 25.6% faster than standard navigation across all indices of difficulty. We analyzed the pairwise differences for the *ID* factor and found, not surprisingly, that all levels were significantly different from each other (Tukey HSD, $p < 0.05$).

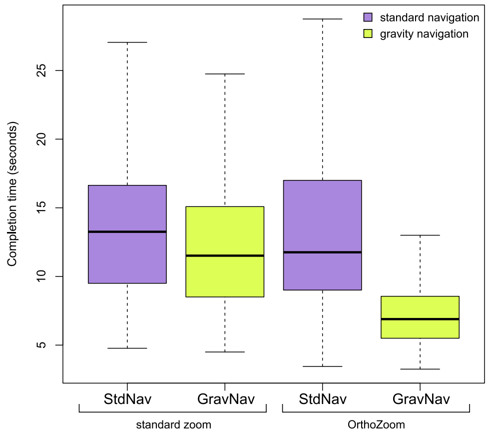


Figure 5: Completion times vs. navigation and zoom technique.

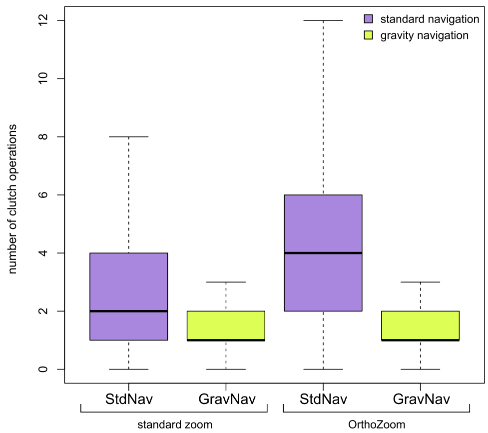


Figure 6: Clutching vs. navigation and zoom technique.

The significant interaction between navigation technique T and zoom control technique Z indicated that there were combinations of T and Z that exhibited diverging behavior. We compared these combinations using a posthoc test (Tukey HSD) and found that the condition GN + OZ was significantly ($p < .05$) faster than all other combinations, followed by GN + SZ ($p < .05$), and then SN + OZ and SN + SZ having no significant time differences ($p = .904$).

5.2 Navigation Behavior

The dynamic navigation behavior for gravity navigation was different from standard navigation. Figure 6 shows a boxplot of the number of clutch—unique sequences of mouse press-drag-release, not counting the initial and final clicks—operations for the different conditions. The T factor exhibited a significant difference (Friedman test, $p < .05$), whereas there was no significant difference for Z (Friedman test, $p = .145$). Surprisingly, standard navigation exhibited *more* clutch operations for OrthoZoom than for constant zoom rate (probably due to overshooting, see more below).

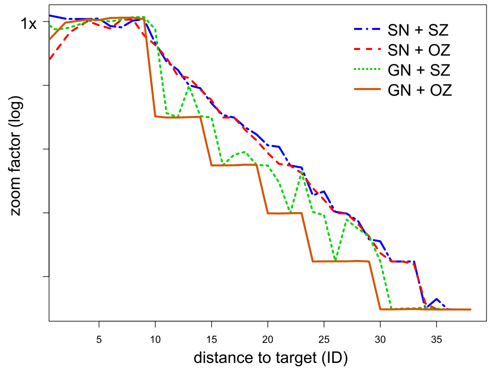


Figure 7: Aggregated space-scale performance (zoom factor vs. world distance from viewport center to target) for all factors and participants.

Figure 7 shows aggregated performance for the different techniques in a space-scale diagram [11]. While this figure does not incorporate timing, it is still clear that gravity navigation seems to converge faster towards the target.

5.3 Subjective Feedback

As discussed earlier, we did not inform participants about the gravity navigation technique and we randomly chose between SN and GN navigations in all blocks. Participants were generally favorable in regards to OrthoZoom (which we explicitly had to train participants in using and also used for blocking), saying that it gave them much more fine-grained control over zooming than for constant-rate zoom. On the other hand, participants felt that they often overshot the target with OrthoZoom, one participant stating that the "the [OrthoZoom] task sometimes got out of control."

Furthermore, several participants perceived that some trials were easier than others, one stating that "sometimes it felt as if my cursor snapped onto the target, whereas other times not so much." The participant went on to state that "maybe it was just my imagination." Another thought that targets seemed to "pull in the cursor." However, other participants did not seem to pick up on this discrepancy. One participant remarked that his strategy of quickly selecting an initial zoom point sometimes paid off, and sometimes did not; we speculate that successful trials were mostly those where gravity navigation corrected for a rough initial zoom point.

6. DISCUSSION

Summarizing our results, we see the following:

- Gravity navigation exhibited significantly lower completion times than standard navigation (confirming **H1**);
- Variable-rate zoom control using OrthoZoom [2] resulted in significantly faster completion times than standard constant-rate zoom control (confirming **H2**); and
- Gravity navigation with OrtoZoom was significantly faster than gravity navigation with constant-rate zoom control whereas no significant such difference was found for standard navigation; in addition, gravity navigation with constant-rate zoom control was faster than standard navigation with OrthoZoom (confirming **H3**).

6.1 Explaining the Results

Our findings confirmed our hypotheses, and thus our reasoning behind them were correct. We were also able to confirm our intuition through observations and the cinematic replay data for each trial. For **H1**, when zooming into the space in any condition, the user would first place the mouse pointer on the target and start the zoom operation (by pressing the right mouse button and dragging to the right, possibly changing the zoom-ratio using vertical mouse movement). Regardless of condition, this initial target acquisition to start zooming would invariably only be a rough estimate. Even if the distance between the mouse pointer and the target was very small in screen distance, this still translated to a very large distance in actual world coordinates.

For gravity navigation this is no issue, since the gravity model is multi-scale and operates in screen coordinates, thus allowing it to automatically adapt to whatever scale is used to view the space and quickly correct such pointing errors. However, for standard navigation even small pointing errors in screen space remain uncorrected and rapidly amount to a large screen distance as the user continues to zoom. This resulted in either having to stop the navigation to recenter the view or change the zoom position, or, worse, having to backtrack by zooming out if the target left the edge of the screen. Regardless, this explains why gravity navigation performed an average of 26% faster than standard navigation.

We were also able to confirm that the OrthoZoom variable-rate zoom control technique was significantly more efficient than constant-rate zoom control (**H2**). This was already known from Appert and Fekete's initial user study [2], but it serves as a useful assertion that GravNav and OrthoZoom do not conflict; in fact, as we discuss below, the two techniques complement each other well. The one surprising finding was that we saw no significant improvement for standard navigation with OrthoZoom compared to constant-rate zoom control. However, we attribute this to the two-dimensional nature of our task making it more difficult, where Appert and Fekete studied a strictly one-dimensional scrolling task.

Finally, we also found that OrthoZoom indeed benefited gravity navigation more than it benefited standard navigation (**H1**). We believe this ties in with the multi-scale nature of the navigation task discussed above: for standard navigation, the fact that small pointing errors quickly gave rise to large navigation errors meant that users were unable to take full advantage of the higher zoom speed made possible by OrthoZoom. In fact, Figure 6 shows that they used *more* clutch operations with OrthoZoom, and there were no significant differences in completion time between these two conditions. This suggests that OrthoZoom may be causing users to overshoot targets more often, causing targets to leave the viewport and the user to have to backtrack by zooming out or panning blindly in the direction of the lost target.

In contrast, gravity navigation marries well to OrthoZoom since the multi-scale error correction in the technique makes large-scale, ballistic zoom operations possible without excessive backtracking and target loss. In fact, we think that one outcome of our work is the suggestion that designers consider pairing GravNav with OrthoZoom for practical implementations of the navigation technique.

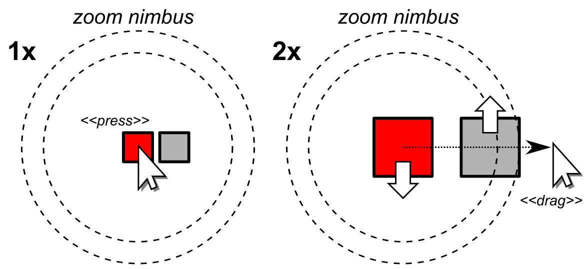
6.2 Generalizing the Results

While our findings are encouraging and confirm our intuition, we still have to explore whether and how these results hold in general multi-scale navigation. First of all, our study focused on micro-level navigation, where the user has already decided on which target to focus on, and merely needs assistance in navigating to the appropriate scale and position. In our task, this was mirrored by the fact that the target was visible on the screen at the start of the trial. In general multi-scale navigation, this is certainly not always the case: the user may have to navigate to off-screen targets causing the navigation task to become more complex. Future evaluations are needed to study such tasks.

Having said that, we still firmly believe in the general utility of gravity navigation: our results clearly show that once the user has reduced ambiguity by settling on a particular target, the technique is significantly more efficient than standard pan and zoom in rapidly zooming and panning to it. In orer to better support target ambiguity while zooming, we propose an extension to gravity navigation that we call *zoom branching* (discussed next) that provides simple gestures for choosing between multiple targets without breaking the flow of the navigation. However, since these gestures are perpendicular to the zoom axis, zoom branching cannot be combined with OrthoZoom, which uses that same approach for rate control.

6.3 Extension: Zoom Branching

The main weakness of gravity navigation is the same as pointing assistance techniques: in the presence of many potential targets, attention gravity may cause navigation to become slower, not faster. For multi-scale spaces this is particularly problematic: even if a target looks like it is isolated from other targets, this may not turn out to be the case when the user starts to zoom into the space towards that target. Another potential target that was previously too close to the intended target to be discernable from it will eventually start to separate into an object of its own as the user zooms closer.



(a) Starting to zoom. (b) Zoom branching.

Figure 8: Zoom branching for a situation with two potential targets. The white arrows on top of the targets in (b) indicate the gesture to perform for centering the view on that target. Once the grey target leaves the outer ring of the nimbus, the branch gestures disappear.

We augment gravity zooming with a specific interaction technique called *zoom branching* to manage this situation. Figure 8 shows an illustration of the idea. The technique uses a *zoom nimbus*—a screen-space circle—centered on the initial point selected for the zoom operation. Whenever the situation arises during zooming that a potential target s (defined by $DOI(s) > 0$) is about to leave the nimbus, the technique provides the user with the option to perform an input gesture perpendicular to the zoom axis to select which of the potential branches to choose for continued zooming. These gestures are visually represented as arrows on top of the targets and are visible for a certain zoom distance. If no action is taken other than continued zooming, the main (center) target will be selected. If the user performs the gesture, the view will instead be centered on the target leaving the nimbus. Significantly, the fact that the gesture is orthogonal to the zoom axis means that the gesture can be performed without interrupting the zoom operation.

Even though we present zoom branching in the context of gravity navigation, it should be noted that the technique is not specifically tied to gravity navigation. The technique could just as well be paired with standard pan and zoom.

7. CONCLUSIONS AND FUTURE WORK

We have presented a novel family of multi-scale navigation techniques that we call GravNav techniques, short for *gravity navigation*. Gravity navigation utilizes the topology of the underlying visual space to assist navigation by calculating an attention vector depending on the viewport position in the space and the surrounding objects of interest. This vector is then used to modulate the speed or even the direction of the user's interactions to facilitate navigation to objects of interest. We evaluated the performance of gravity navigation implementations and found that they were significantly more efficient compared to standard pan and zoom, particularly when paired with variable-rate zoom gain.

In general, topology-aware navigation is a promising idea and much work remains in fully mapping out its design space. Our future work will continue to utilize knowledge about the structure of the visual space to improve navigation. In particular, we intend to study mechanisms for disambiguating between multiple potential targets without breaking the overall flow of interaction.

8. ACKNOWLEDGMENTS

Thanks to Keegan for gracefully agreeing to appear in Figure 2. He is a very good boy, usual payment in milkbones to follow.

9. REFERENCES

[1] D. Ahlström, M. Hitz, and G. Leitner. An evaluation of sticky and force enhanced targets in multi target situations. In *Proceedings of the ACM Nordic Conference on Human-Computer Interaction*, pages 58–67, 2006.

[2] C. Appert and J.-D. Fekete. OrthoZoom scroller: 1D multi-scale navigation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 21–30, 2006.

[3] P. Baudisch and R. Rosenholtz. Halo: a technique for visualizing off-screen objects. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 481–488, 2003.

[4] R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, pages 519–526, 2004.

[5] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18:193–222, 1995.

[6] N. Elmqvist and J.-D. Fekete. Semantic pointing for object picking in complex 3D environments. In *Proceedings of the Graphics Interface Conference*, pages 243–250, 2008.

[7] N. Elmqvist, M. E. Tudoreanu, and P. Tsigas. Evaluating motion constraints for 3D wayfinding in immersive and desktop virtual environments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1769–1778, 2008.

[8] D. Fisher. Hotmap: Looking at geographic attention. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1184–1191, 2007.

[9] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

[10] G. W. Furnas. Effective view navigation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 367–374, 1997.

[11] G. W. Furnas and B. B. Bederson. Space-scale diagrams: Understanding multiscale interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 234–241, 1995.

[12] T. A. Galyean. Guided navigation of virtual environments. In *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pages 103–104, 1995.

[13] S. Ghani, N. H. Riche, and N. Elmqvist. Dynamic insets for context-aware graph navigation. *Computer Graphics Forum*, 30(3):861–870, 2011.

[14] Y. Guiard, M. Beaudouin-Lafon, J. Bastin, D. Pasveer, and S. Zhai. View size and pointing difficulty in multi-scale navigation. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pages 117–124, 2004.

[15] T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 139–148, 2000.

[16] E. W. Ishak and S. Feiner. Content-aware scrolling. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 155–158, 2006.

[17] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1254–1259, 1998.

[18] S. Jul and G. W. Furnas. Critical zones in desert fog: Aids to multiscale navigation. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 97–106, 1998.

[19] S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139, 1992.

[20] R. L. Mandryk and C. Gutwin. Perceptibility and utility of sticky targets. In *Proceedings of the Graphics Interface Conference*, pages 65–72, 2008.

[21] T. Moscovich, F. Chevalier, N. Henry, E. Pietriga, and J.-D. Fekete. Topology-aware navigation in large networks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2319–2328, 2009.

[22] K. Perlin and D. Fox. Pad: An alternative approach to the computer interface. In *Computer Graphics (ACM SIGGRAPH '93 Proceedings)*, pages 57–64, 1993.

[23] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[24] J. J. van Wijk and W. A. A. Nuij. Smooth and efficient zooming and panning. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 15–22, 2003.

[25] M. Verm and P. W. McOwan. A semi-automated approach to balancing of bottom-up salience for predicting change detection performance. *Journal of Vision*, 10(6):3, 2010.

[26] D. Walther, U. Rutishauser, C. Koch, and P. Perona. Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding*, 100(1-2):41–63, 2005.

[27] A. Worden, N. Walker, K. Bharat, and S. Hudson. Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 266–271, 1997.