GravitySpot: Guiding Users in Front of Public Displays Using On-Screen Visual Cues

Florian Alt\textsuperscript{1}, Andreas Bulling\textsuperscript{2}, Gino Gravanis\textsuperscript{1}, Daniel Buschek\textsuperscript{1}

\textsuperscript{1}University of Munich (LMU)
Media Informatics Group
firstname.lastname@ifl.lmu.de

\textsuperscript{2}Max Planck Institute for Informatics
Perceptual User Interfaces Group
bulling@mpi-inf.mpg.de

ABSTRACT

Users tend to position themselves in front of interactive public displays in such a way as to best perceive its content. Currently, this sweet spot is implicitly defined by display properties, content, the input modality, as well as space constraints in front of the display. We present GravitySpot – an approach that makes sweet spots flexible by actively guiding users to arbitrary target positions in front of displays using visual cues. Such guidance is beneficial, for example, if a particular input technology only works at a specific distance or if users should be guided towards a non-crowded area of a large display. In two controlled lab studies (n=29) we evaluate different visual cues based on color, shape, and motion, as well as position-to-cue mapping functions. We show that both the visual cues and mapping functions allow for fine-grained control over positioning speed and accuracy. Findings are complemented by observations from a 3-month real-world deployment.

Author Keywords

Public Displays; Interaction; Sweet Spot; Audience Behavior

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—Input devices and strategies

INTRODUCTION

Displays have become ubiquitous in public spaces, such as shopping malls or transit areas in airports and train stations. At the same time, researchers and practitioners aim to increase user uptake by providing interactive and engaging experiences [1]. This trend is further supported by sensing technologies (cameras, depth sensors, etc.) becoming available for easy and low cost integration with such displays. Sensing technology, however, has specific requirements regarding the optimal operating distance, thereby constraining the possible interaction space. For example, while touch sensors require the user to come in close proximity to the display, gestures-based interaction using Kinect allows users to position themselves freely between 0.5 m–4.0 m in front of the display. Stationary eye trackers require the user’s head to be inside the tracking box – about 30 cm×30 cm – at a distance of 70 cm in front of the screen [27]. Hence, interactive displays face the challenge of how to encourage users to position themselves in a target location within the interaction space.

Similar challenges arise in situation where public displays are deployed opportunistically. Such deployments are often constrained by the size and layout of the physical space surrounding the target location [3]. This results in displays being positioned in non-optimal spots where, for example, users cannot easily stop without blocking the way of other passers-by (cf. the butt-brush effect [29]). In such cases, it would be desirable to guide users towards less crowded areas, particularly in front of large displays. As a solution, deployments aim to either anticipate the default sweet spot, i.e. the area where users are most likely to stop as they approach the display, or...
they try to actively promote the optimal interaction area by means of explicit hints on the floor, next to the display, or on the display. Most related to our work, Beyer et al. investigated the use of dynamic visual guides to direct single users and to distribute multiple users in front of a public display [6].

We present GravitySpot, a novel approach that modifies the visual appearance of the display content based on user position. We leverage findings from human vision research that show that humans can very quickly process certain visual cues, such as color, motion, and shape. By showing the unmodified content only from a specific location in front of the display, users are made to anticipate this so-called sweet spot. Dynamic visual guides offer several benefits:

1. They allow for changing the sweet spot in an *adaptive and dynamic* manner, for example, based on the current number and position of people in front of the display.
2. They do *not require attention switches* as cues are not decoupled from the actual screen content, in contrast to, for example, hints displayed on the floor or next to the screen.
3. They are more *robust against occlusions*, since by showing the cue on the screen, users can simply re-position themselves to perceive the cue, compared to cases, where other users are standing on a cue shown statically on the floor.
4. They require *minimal hardware*. Any sensor that allows the user position to be determined can be used (e.g., Kinect).

GravitySpot advances the state of the art as it neither requires space nor time-multiplexing between cue and content nor any overlays (silhouette, frame) since it integrates smoothly with the actual content. We compare different visual cues with regard to positioning accuracy and speed and show how to improve them by adapting the mapping between user position and cue. We conduct two controlled lab studies (n=29). Results suggest a trade-off between accuracy and speed depending on the cue. In a second study we demonstrate that by altering the mapping between user position and cue intensity, this trade-off can be overcome and accuracy (up to +51\%) and speed (up to +57\%) be enhanced. This is valuable for designers, since it allows cues to be chosen based on the display content (for example, readability of text can be preserved by choosing appropriate cues). The studies are complemented with a real-world deployment. We show that also in a real-world situation, where users are unaware of how the cues work, they can quickly and accurately position themselves.

Our contribution is threefold. First, we propose a new set of visual guidance cues that directly integrate within the display content. Second, we present two controlled lab experiments to study the efficiency of the cues and the impact of different mapping functions. Third, we present an in-the-wild deployment, demonstrating how to integrate the approach with an interactive game. We found that the approach is easily understandable to users with its efficiency being similar to the lab.

**RELATED WORK**

Our work builds on previous studies on (1) interaction models to influence audience behavior, (2) applications where interaction depends on or is influenced by a particular user location, as well as (3) positioning cues for public display applications.

**Audience Behavior**

Prior work investigated how people behave in the vicinity of interactive displays. Spatial models [25, 31] describe different zones that define the interactions offered and the information shown on the display. These model noticeably draw from Hall’s theory of proxemics [15]. Temporal models describe the interaction process. The public interaction flow model studies how groups socialize around public displays [10]. It identifies three activities – peripheral awareness, focal awareness, and direct interaction – and thresholds to be overcome by the user to proceed to the next phase. An extension is the audience funnel, which allows conversion rates between the phases to be determined [19]. The model distinguishes between a passing-by stage, followed by a stage where users are viewing and reacting. After this, subtle interactions (e.g. to find out how interaction works), direct interaction, and eventually follow-up actions may occur.

Most previous work on modeling implicitly assumes that users are not only able to identify how interaction works but also to understand where they need to position themselves. Our work is based on the observation that attention is a crucial prerequisite for user positioning. Only after users notice the display it makes sense to focus on guidance. Our approach allows the stages defined in prior models to be refined by a positioning phase. This phase is not necessarily a part of the ultimate interaction step only but may span across multiple phases or zones. For example, positioning cues can already guide users before they notice that the display is interactive.

Koppel et al. [26] showed that, by changing the display configuration, audience behavior could be altered, for example, how people approach, position, and interact with the displays. Our method creates a similar effect but without the need to reconfigure the display, which is not feasible on-the-fly.

**Location-Aware Display Deployments**

A lot of display applications exist in which the interaction depends on the user location. Beyer et al. detect the user’s position in front of a cylindrical display to let users draw a flower pattern on the screen [5]. GazeHorizon enables users to interact with the screen content based on gaze [36]. SpiderEyes is a toolkit for designing proximity-aware collaborative display applications [14], using the Kinect to determine the user’s distance to the screen and adapt the content visualization accordingly. The Clavier is a walkable piano projected on a path [16]. A projection of the keyboard communicates the interaction area in which light sensors would detect user movements and then trigger auditory output. The Proxemic Peddler is an advertising display that makes the content adapt or move as users change their position [34]. The aim is to raise attention and foster (touch) interact with the content. Brudy et al. presented a system using the position of multiple users in front of a public display to increase privacy. One sample application they present is a spotlight that makes only areas of the screen visible that users obstruct with their body [11].

Users can also benefit from positioning in the context of visualizations on large wall displays. Ball et al. found increased physical navigation to improve user performance [4]. Further work found that in perception estimation tasks, users should
We designed GravitySpot with these various application areas in mind, considering that visual cues for guiding the user should (1) not obscure the actual user interface, (2) be shown on the display itself in order not to be overlooked by users, (3) not require any attention switches, and (4) not be textual so as to minimize the cognitive load and be language-independent.

Positioning in Front of Public Displays

Few previous works studied means to influence user position in front of displays. Beyer et al. showed that people tend to position themselves in a way so that they can optimally perceive the content of a planar display and there is evidence that this also holds for non-planar displays [5]. As a result of this, sensors are usually placed so that they can best sense the user and their interaction – for example, a Kinect placed below or above the screen [2, 20, 26, 30, 32] or cameras above the display [5, 7]. Some works employ very wide displays where one sensor cannot easily cover the entire interaction area. In such cases, several sensors are usually combined [30].

More recently, and closely related to our own work, Beyer et al. showed that dynamic visual guides (frames and ellipses), both on planar [6] as well as on cylindrical displays [7], lead to users positioning themselves centrally in front of these guides and can be used to manipulate the audience. However, the proposed approach only allows for positioning parallel to the display (x-position). In contrast, our work also allows the distance to the display to be controlled (z-position). In addition, frames and ellipses constantly overlay the user interface whereas our cues get weaker and finally disappear as the user reaches the sweet spot. Finally, our approach also allows the trade-off between speed and accuracy to be controlled.

GazeHorizon is an application that uses a webcam for gaze-based interaction with public displays [36]. To position users the authors used different cues, including floor labels, explicit on-screen distance information, and a mirror video feed overlaid with a face outline. While floor labels were usually overlooked by users, distance information worked better but required considerable time for correct positioning. In contrast, our work provides a thorough investigation and comparison of different positioning cues. Our cues can be employed to a user interface without obstructing any information and that can be constantly applied, leading to users staying in focus.

GUIDING USERS USING VISUAL CUES

Findings in cognitive psychology suggest that the human visual system can rapidly process a visual scene to extract low level features, such as color, shape, orientation, or movement, without the need to cognitively focus on them [28]. We aim to leverage this ability by mapping a user’s current position to visual cues shown on the display.

Psychological Foundations

Our work exploits effects of attentive and pre-attentive visual perception, as introduced by Neisser [21] and confirmed by Treisman [28]. Neisser describes the process of visual perception as a two-step process. First, simple features of a scene are perceived, such as separating textures or the distinction between an object and its background (figure-ground perception). This stage is pre-attentive and characterized through parallel processing. It results in a set of features not yet associated with specific objects [18]. Second, users associate features to scenes, directing attention serially towards the different scene objects.

There is no consent in research literature as to which features are perceived pre-attentively [12]. There is strong evidence that the list of tasks working pre-attentively presented by Neisser is not conclusive. Hence, also the distinction between pre-attentive and non pre-attentive features is rather blurry. Research that aims to make this distinction includes the work from Wolfe [35]. He presents a list of 28 features, separated into likely, possible, and unlikely candidates for pre-attentive perception. As Wolfe noted himself, for many cases there is only little evidence since results stem from single publications – so the list may have to be extended in the future.

We base our research on the work of Nothdurft on the role of visual features during pre-attentive visual perception [22]. Nothdurft classified pre-attentive features into three categories: color, shape, and motion.

Selection of Visual Cues

We selected five visual cues according to Northdurf’s categories (see Figure 2). According to Wolfe, all of these cues are likely to be perceived pre-attentively.

Color

Public displays often contain monochrome content, such as text. Hence, we opted for brightness and contrast as color cues, since these have a smaller impact on readability. To also consider features that affect the color information of multicolor content, we included saturation (see Figure 2b-d).

Shape

We selected shape features that alter the form of content and can be applied to content post-hoc. In particular we chose pixelation and distortion. While pixelation simply decreases the

Figure 2. Visual cues investigated in this work (shown on a test pattern for intelligibility only): Original test pattern (a); color cues: brightness (b), contrast (c), and saturation (d); shape cues: pixelation (e) and distortion (f).
resolution of the content, distortion applies a non-affine mapping function (see Figure 2e-f). Both cues have a strong impact on readability. Based on the font size, content becomes only readable near the sweet spot (10–20 cm).

**Motion**
Finally, as a motion cue, we opted for jitter that moves content with a frequency of 5 Hz along the screen axes. Based on the distance of the user from the sweet spot, the effect intensity is increased by adapting the motion amplitude.

**Baseline**
We compare these cues with two baselines from prior work. We opted for on-screen cues, since they were shown to work best in public settings [36]. The first cue is a compass-like arrow on the display that points to the direction in which users should move to reach the sweet spot. The arrow is slightly tilted in z-direction, indicating that “up” means moving forward. The second cue is a simple text telling users whether they should move ‘forward’, ‘backward’, ‘left’, or ‘right’.

**Apparatus**
To evaluate how well users could be guided using visual cues we implemented the GravitySpot prototype. The C# prototype consists of (1) a tracking module that measures users’ 2D position in real-time using Kinect and (2) a rendering module that allows any of the aforementioned visual cues to be applied to the display content. The intensity of the cue depends on the current distance of the user to the target position. We implemented different mappings (Figure 3), where the minimum is defined by the target spot and the maximum by the largest distance at which the user can still be sensed.

**Sensor Calibration**
We use the Kinect skeleton data to calculate the user position (x- and z-coordinate) in the 2D space in front of the display. To cover as much space as possible, we support the use of multiple Kinects — for example, with two Kinects a visual angle of up to 90° can be covered. We implemented a calibration tool that allows position information obtained from multiple Kinect sensors to be transformed into an x/z user position. For calibration we use triangulation based on 2 reference points.

To be able to change the sweet spot during runtime, our prototype allows a rectangular area to be defined within the field of view of the Kinect sensors. Arbitrary locations within this area can then be selected as sweet spots.

**Mapping Between Position and Cue**
During first tests, we noticed that the visual cues were subject to a trade-off between speed and accuracy of guiding a user to the sweet spot. To investigate this phenomenon in more detail we decided to implement different position-to-cue mapping functions. The functions were designed in such a way as to improve the visual cues so that the users find the sweet spot faster and/or more precisely. We chose four mapping functions (Figure 3): **Linear, SlowStart, QuickStart** and **Scurve**.

**Linear mapping function.** The linear mapping function was chosen as a baseline. The Euclidean distance $x$ of the user is linearly mapped to the intensity of the visual cue.

$$linear(x) = x$$

**S-shaped mapping function.** The s-shaped mapping function is a combination of the quick start and slow start mapping functions. We expect it to provide clearly visible changes at great distances and accurate feedback when the user draws near to the sweet spot. In the center span this function keeps a steady increase and does not fall flat. As a result, we avoid areas where the user receives no feedback on position changes. We expect this function to provide a good combination of speed and accuracy, while outperforming the linear function.

$$scurve(x) = \frac{(2(x - 0.5))^2 + 2(x - 0.5) + 2}{4}$$

**Approach**
To evaluate GravitySpot we designed three studies. In the first controlled laboratory study we compare the different cues with regard to positioning time and accuracy. Furthermore, we collect user feedback with respect to how easy it is for them to understand the different cues. The anticipated measures required a controlled setting where lighting was kept constant and where participants were able to approach the display from constant distances and angles. The study followed a repeated measures design.
In the second study we investigate the influence of the mapping functions that determine how the user position is mapped to the cue shown on the display. We are particularly interested whether positioning speed and accuracy could be further increased by applying the mapping functions. We also investigate whether the accuracy of fast cues can be increased and vice versa. We believe this to be valuable for designers who want to work with particular cues and show specific content. We selected two cues – brightness and pixelate – that users considered to work best in the first study.

As prior work showed user behavior to often differ in the real world as opposed to the lab [17, 20], we validated the ecologic validity of our findings through an in-the-wild deployment.

Applications
For the lab studies we needed an application that required users to position themselves precisely while timing measurements for a given task could be taken. For the in-the-wild deployment we needed an application that was engaging and easy-to-understand while at the same time requiring minimal interaction techniques, since these are in general very difficult to communicate in a public setting [32].

Spot-the-Difference
For the lab study we implemented a Spot-the-Difference game. In this game the screen shows computer-generated images of two shelves that contain a number of items in different colors. The position and color of the items on each shelf can be modified. The task of the user is to spot all items which differ between the right and the left shelf. We do not allow any input, such as touching or pointing at the respective items. This is because measurements may be affected by the recognition accuracy of the system or users would be required to leave the sweet spot. Instead, users are asked to notify the experimenter verbally once they find the solution. The game was shown on a 78” projection screen with a resolution of 1600×1200 px.

Trivia Game
For the in-the-wild deployment we implemented a Trivia game (Figure 4) in which questions are shown on the display (55”, LCD, 1600×1200 px) as soon as users enter the interactive area. Answers are shown in the form of still images to which the system applies the corresponding visual cue. A sample question could ask for the tallest building in the US and then show an image of the new World Trade Center in New York. Answers are shown as soon as users do not alter their position any more, assuming that at this point they reached the (subjective) current best position. We use a time-based threshold to decide when to display the answer, i.e. users have to stop for at least 1.5 s. Answers are shown for five seconds before the next question is displayed. For each question we make sure that the new sweet spot has a minimum distance to the old sweet spot, so that users need to alter their position.

LAB STUDY I: ACCURACY AND POSITIONING TIME
Participants
In total, 15 people (six female) participated in the study. Participants were students and employees with an average age of 23 years (std.dev.=2.8). Two participants owner a Kinect and seven wore glasses or contact lenses.
We first compared our cues to the baselines: Bonferroni-corrected post-hoc tests revealed that the differences between distort and the baselines were not significant (arrow: \(p=0.194\), text: \(p=0.226\)). In contrast, all other cues showed significant differences to both baselines (all \(p<0.01\)). Distort was not significantly different from pixelate, and all other visual cues also showed no significant differences between them. These results show the existence of two “groups” of visual cues with respect to task completion time: 1) slower ones (distort, pixelate), and 2) faster ones (brightness, contrast, jitter, saturate).

**Positioning Accuracy**

We analyzed mean euclidean distances to the sweet spot per cue for each participant (see Figure 6–right). Greenhouse-Geisser corrected ANOVA found a significant main effect of visual cue (\(F_{3,356.43,0.623}=36.333, p<0.001\)). Bonferroni-corrected post-hoc tests revealed that the differences between distort and the baselines were not significant (arrow: \(p=0.194\), text: \(p=0.226\)). In contrast, all other cues showed significant differences to both baselines (all \(p<0.01\)).

Furthermore, distort and pixelate were not significantly different, but they were both significantly more accurate than all other cues (brightness: \(p<0.05\), all others: \(p<0.005\)). Brightness was significantly more accurate compared to jitter (\(p<0.05\)), but not compared to saturate (\(p=0.241\)) and contrast (\(p=0.076\)). There were no significant differences between saturate, contrast and jitter.

In conclusion, similar to task completion times, this analysis revealed two main groups of visual cues, as can be also derived from users’ trajectories (Figure 7): 1) More accurate ones (distort, pixelate), and 2) less accurate ones (saturate, contrast, jitter), with brightness as a compromise.

**Questionnaire**

In the following we analyze the questions users had to answer with regard to each cue during the study.

**Correlation between position and visualization:** There was a significant difference (\(\chi^2(7)=24.289, p=0.001\)) depending on the cue. Post-hoc tests revealed that jitter was ranked significantly worse than arrow, text, and brightness (all \(p<0.05\)). Arrow received the best median rank (2), followed by brightness and text (both 3). Jitter was ranked worst (7).

**Accuracy of visualization:** We found no significant difference depending on the cue (\(\chi^2(7)=10.378, p=0.168\)). Arrow, brightness, pixelate, and text were ranked with median 4. Distort/saturate received 5, followed by contrast (6) and jitter (7).

**Changes in visualization:** We discovered a significant effect for the cue (\(\chi^2(7)=30.000, p<0.001\)). Post-hoc tests revealed that jitter was ranked significantly worse than arrow, text, and pixelate (all \(p<0.05\)). Arrow and text received the best median rank (2), followed by pixelate (3), brightness (4), distort (5), contrast/saturate (6), and jitter (7).

Overall, jitter stands out: despite good performance with regard to task completion time it was perceived as less clear as the rest in all questions. From this we conclude that designers need to be particularly careful when applying this cue. Future work could further investigate this cue by (a) modifying the frequency of the movement and (b) applying this cue to particular objects rather than the entire screen.

---

**Results**

**Task Completion Times**
We first analyzed mean task completion times per cue for each participant (see Figure 6–left). Greenhouse-Geisser corrected ANOVA found a significant main effect of visual cue (\(F_{3,101.43,0.413}=23.631, p<0.001\)).

We first compared our cues to the baselines: Bonferroni-corrected post-hoc tests revealed significantly shorter task completion times for contrast and saturate than for arrow (both \(p<0.05\)). Moreover, task completion times for jitter (\(p<0.005\)), brightness, contrast, and saturate (all \(p<0.001\)) were significantly shorter than for text. These results suggest that visual cues can significantly speed up guiding users to a defined sweet spot, compared to textual or symbolic cues.

Further differences were found between cues: distort and pixelate were significantly slower than the rest (distort vs jitter: \(p<0.005\), distort vs rest: \(p<0.001\); pixelate vs others:
We were interested whether accuracy and/or task completion time could be further increased by using different mapping functions. This is potentially valuable information for a designer because it allows (1) a visual cue to be selected based on the specific positioning accuracy and task completion time required by the application and (2) a cue that preserves readability to be chosen. We selected two cues based on user ratings from the first study. To this end, we compared the four mapping functions for the pixelate and the brightness cue.

14 participants (six female) with an average age of 23.1 years (std.dev.=2.87) were recruited via mailing lists and Facebook.

**Task, Setup and Procedure**

As for the first study, we again used the Spot-the-Difference game to compare different mapping functions and we used the same setup depicted in Figure 5.

**Results**

**Task Completion Time**

We analysed mean task completion times per cue and mapping per participant. Greenhouse-Geisser corrected ANOVA revealed significant main effects for cue ($F_{1,13}=148.760$, $p<0.001$), mapping ($F_{2,247.29.208}=23.896$, $p<0.001$), and cue × mapping ($F_{2.362.30.710}=9.389$, $p<0.001$). Bonferroni-corrected post-hoc tests showed that brightness led to significantly faster task completion times than pixelate ($p<0.001$), matching the findings from the first lab study. Regarding mappings, SCurve was not significantly different from linear, but all pairwise comparisons were significant (quick vs linear, slow vs linear, slow vs SCurve: $p<0.05$, all others: $p<0.001$).

The directions of the mappings’ influences matched our expectations. QuickStart resulted in significantly shorter task completion times than linear, while SlowStart resulted in significantly longer ones. Hence, guiding speed can be significantly influenced by choosing different mappings. The results also suggest that speeding up adaptation has a larger influence on slower cues (QuickStart: +51% with pixelate, +7% with brightness), while slowing down is stronger for faster ones (SlowStart: -62% with brightness, -31% with pixelate).

**Positioning Accuracy**

As in the first lab study, we analyzed mean euclidean distances to the sweet spot per cue for each participant. Greenhouse-Geisser corrected ANOVA revealed significant main effects for cue ($F_{1,13}=369.280$, $p<0.001$), mapping ($F_{2,672.21.370}=108.359$, $p<0.001$), and cue × mapping ($F_{2,311.30.048}=12.827$, $p<0.001$). Bonferroni-corrected post-hoc tests revealed that brightness resulted in significantly lower accuracy than pixelate ($p<0.001$). This matches findings from the first lab study. Comparing the mappings, we found no significant difference between linear and SCurve, but significant differences for all other comparisons ($p<0.001$).
We used the most accurate cues (pixelate, distort, jitter) and selected SlowStart, QuickStart, and the SCurve (which yielded similar results as linear) as mapping functions. As a baseline, we selected arrows due to language independency and results similar to text. This resulted in ten experimental conditions (3 cues × 3 mappings + baseline). Conditions were randomly selected for each user. If users played subsequent games, the same cue was used.

Results

During the deployment 775 games were played in a total of 234 sessions. Overall, the most games were completed with pixelate (343), followed by distort (243). The least games were completed with baseline (121) and jitter (68).

Observations

To understand how to best integrate the cues, we tried different initial screen layouts during the deployment. In particular we compared showing the cue immediately as users entered the visual field of the camera to a screen that first explained the game to them. Showing the cue immediately led to more people interacting, since the motion caused by the movement seemed to attract the attention of passersby. We furthermore found that the different cues attract a user's attention to different extents. Cues that have stronger effects on the image (e.g., pixelation) seem to work better than more subtle cues (e.g., distort). This can be exploited by designers to attract more or less people to the display, for example based on the overall number of people in the vicinity or the application. Finally, cues seem to differ in attractiveness, reflected by how often people played in the different conditions. Pixelate had a quite immersive effect, leading to people playing on average more questions before leaving than for jitter, distortion, or arrow.

In general, observed numbers indicate that users preferred playing the game with the pixelate and distort cues. Most games (41) were completed for pixelate and SCurve mapping, and for distort and QuickStart (31). Over all mappings, SlowStart led to fewer completed games (63) than QuickStart and SCurve (both 71). This suggests that slow initial adaptations may result in a less engaging / motivating gaming experience. Designers should thus base their decision for a particular mapping both on the required accuracy and application purpose.

Quantitative Findings

We focused on the accuracy of the different cues and mapping functions. We did not compare task completion times since we could not control for the time it took users to read and think about the questions. To account for the different sample sizes for the cues / mappings (see above) we report the following analyses / ANOVAs based on estimated marginal means (weighted means) instead of unweighted means, using SPSS.
Regarding accuracy we found no significant effect of cue \( (F_{3,224}=1.350, p>0.05) \). Averaged over all mappings, pixelate was most accurate (mean dist. 0.21 m), followed by distort, baseline (both 0.22 m), and jitter (0.25 m). We found a significant effect of mapping on distance \( (F_{3,224}=6.011, p<0.01) \). Averaged over all cues, SCurve (0.19 m) was more accurate than SlowStart (0.23 m) and QuickStart (0.25 m). The difference between QuickStart and the others was significant (Bonferroni-corrected post-hoc tests, \( p<0.05 \)).

In summary, findings from our deployment confirm most results from the lab. In particular, the mapping functions can indeed enhance accuracy in the intended way. For example, accuracy in the deployment was on average 50% higher than in the first lab study with the standard mapping (e.g., pixelate – lab: 0.33 m, deployment: 0.21 m; distort – lab: 0.31 m, deployment: 0.22 m). Furthermore, differences in accuracy between cues are comparable for the lab and in-the-wild. From this we conclude that our approach is in fact capable of enabling interaction that requires accurate positioning of the user due to narrow interaction spaces, for example, eye tracking where users need to position themselves in a 30 cm \( \times \) 30 cm area.

**LIMITATIONS AND FUTURE WORK**

First, we focused on single-user interaction. We configured our system to recognize and react to the first person to arrive. In future work, multiple users could be supported, which is beneficial for very large screens or screens employing multiple sensors and sweet spots. We believe the major challenge to be the relationship between cue and user. To make the relationship understandable for users, future work could investigate proximity or kinaesthetic matching. The latter approach is particularly promising since recent work showed that user representations on interactive public displays attract significantly more visual attention than other screen content [33].

Second, we only investigated playful applications. Yet the approach is in theory easily applicable to other applications, such as information displays. Text content may require further investigation, since some cues impact on readability. Optimal perception may require high accuracy, hence reducing speed.

Third, we employed our approach to the entire user interface, thus making it very prominent. Hence, we cannot draw any conclusion how well the approach works in situation where it is only applied to parts of the UI and where users may more easily oversee it. For example, an information display in an art gallery could provide textual information on an exhibit alongside with an image of the artist. Future work could investigate how well applying the cue only to the image works.

**IMPLICATIONS FOR DESIGN**

Our results show that designers can guide users with different cues, and that they should consider mapping functions to tune these cues with respect to speed and accuracy. Figure 9 summarises the resulting trade-offs, allowing designers to choose the setup that suits their needs best. Apart from accuracy and speed, cues should be considered regarding *readability*. For textual content, color cues seem more appropriate than shape-changing ones. In contrast, the latter seem to not only attract more attention (usually desirable for any public display app) but also to be more *entertaining and engaging* for users, making these cues particularly suitable for playful applications.

**CONCLUSION**

We presented GravitySpot – an approach to guide users in front of public displays using visual cues. The approach was evaluated in both lab and field experiments. The results suggest that the approach can ease the deployment of arbitrary kinds of sensors that have particular requirements regarding interaction distance but also allows the content and type of application to be considered.

**ACKNOWLEDGEMENTS**

We thank Mohamed Khamis, Christian Becker, Joseph Longden, Max von Bülow, and Benjamin Bisinger for help with conducting the studies and creating the video figure.

**REFERENCES**


