Looking behind Bezels: French Windows for Wall Displays

Rodrigo A. de Almeida, Clément Pillias, Emmanuel Pietriga, Pierre Cubaud

Looking behind Bezels: French Windows for Wall Displays

Rodrigo A. de Almeida[†] Clément Pillias[†] Emmanuel Pietriga[†,‡] Pierre Cubaud[*]

[†]INRIA
91405 Orsay, France

[‡]CIRIC
7561211, Santiago, Chile

[*]CNAM
75003 Paris, France

rodrigo.almeida@lri.fr pillias@lri.fr emmanuel.pietriga@inria.fr cubaud@cnam.fr

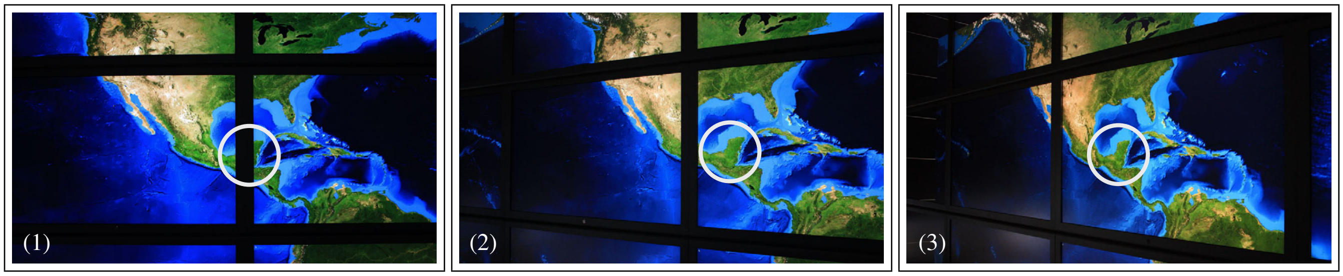


Figure 1: The grid formed by monitor bezels on wall displays is often compared to a french window. We designed two interaction techniques that transform that grid into an actual french window. On the map, (1) the Yucatán peninsula (white circle) is partially hidden by bezels. (2) With one of the techniques, GridScape, users can reveal that part of the map simply by slanting their body or moving slightly to the right. (3) Moving further right, the entire eastern cost of Mexico can be shown without any bezel occlusion.

ABSTRACT

Using tiled monitors to build wall-sized displays has multiple advantages: higher pixel density, simpler setup and easier calibration. However, the resulting display walls suffer from the visual discontinuity caused by the bezels that frame each monitor. To avoid introducing distortion, the image has to be rendered as if some pixels were drawn behind the bezels. In turn, this raises the issue that a non-negligible part of the rendered image, that might contain important information, is visually occluded. We propose to draw upon the analogy to french windows that is often used to describe this approach, and make the display really behave as if the visualization were observed through a french window. We present and evaluate two interaction techniques that let users reveal content hidden behind bezels. *ePan* enables users to offset the entire image through explicit touch gestures. *GridScape* adopts a more implicit approach: it makes the grid formed by bezels act like a true french window using head tracking to simulate motion parallax, adapting to users' physical movements in front of the display. The two techniques work for both single- and multiple-user contexts.

Keywords

Visualization, Wall-sized displays, Bezels, Motion Parallax

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces. - Graphical user interfaces.

Rodrigo A. de Almeida, Clément Pillias, Emmanuel Pietriga,
Pierre Cubaud.
Looking behind Bezels: French Windows for Wall Displays
In AVI '12: Proceedings of the working conference on Advanced visual interfaces, ACM, May 2012.
Authors Version

General Terms

Design, Human Factors.

1. INTRODUCTION

Wall-sized displays made of tiled monitors typically feature a resolution of about 100 pixels per inch, enabling the visualization of truly massive datasets [25] thanks to display capacities that range from 32 megapixels to more than 200 megapixels for the larger ones. Applications include scientific visualization, automotive or airplane design, network monitoring, geospatial intelligence, crisis management or command-and-control centers.

One drawback of these so-called *ultra-high-resolution* wall displays [24, 26] is the discontinuity caused by the bezels that frame each LCD panel. For some tasks, depending on the nature of the data being visualized, bezels can sometimes help users organize display space [14, 27, 29]. The grid formed by bezels can also help structure visual search [8]. However, bezels are a problem when displaying large images such as maps or other visualizations that span multiple monitors. They create a visual discontinuity, that can basically be treated in one of two ways [9]. The problem can be ignored entirely, displaying the picture as if monitors were juxtaposed seamlessly: this solution, called the *offset approach* (Figure 2-a and Figure 3), is simple and straightforward; it has been employed by many early platforms. However, since the panels do have seams, this method necessarily entails distortion of the rendered image, that will be proportional to the bezels' thickness. The other solution consists in taking the bezels into account [20, 27]: this solution, called the *overlay approach*, gives the overall impression that the bezels are a grid overlaid on top of a single image that spans the entire wall, as illustrated in Figure 2-b and Figure 3.

Neither of these approaches is ideal; they represent a trade-off. But to our knowledge these are the only two solutions in widespread use. One obvious way to address the problem would be to find a way to perfectly juxtapose the LCD panels. But bezels are unlikely to completely disappear soon. The ultra-thin bezels advertised in bleeding-edge products such as Samsung's 460UTN or LG's 47WV30 are still 6.9mm wide, which amounts to about 50 pixels at 100ppi, and come at the expense of resolution (1366x768 for a 47" diagonal). Projection-based systems are inherently bezel-free, but are not a viable option, as they have a low pixel density, are difficult to align, and suffer more from problems such as color drift than LCD-based tiled displays. Some researchers have tried to mitigate the negative effect of bezels by removing the plastic casing of each panel [4, 31]. However, those bezels cannot really be eliminated, but only halved due to technological constraints.

The overlay approach has often been presented using french windows as an analogy. In this paper, we propose to draw upon this analogy, and make the display actually behave as if the visualization were observed through a french window. We propose two novel interaction techniques that let users reveal content hidden behind bezels. The first technique, *ePan*, enables users to translate the entire image, displayed in overlay mode (Figure 3). Translation is controlled explicitly by interacting with a handheld device such as a smartphone. The second technique, *GridScape*, adopts a more implicit approach by pushing the metaphor further (Figure 1): the visualization is offset in depth by putting some (virtual) distance between the LCD panels and the graphical projection plane. Then, using head tracking to simulate motion parallax [12], the technique enables users to see occluded parts of the image simply by moving in front of the display or changing their body posture (see Figure 1). We first describe the design and implementation of these two techniques. Then, we report on a controlled experiment that evaluates path tracing task performance across display tiles in two conditions that require explicit input from the user to reveal hidden content, compared to a condition that relies mainly on users' physical navigation in front of the display. Finally, we discuss how the techniques can be adapted to support multiple users.

2. RELATED WORK

While the first wall-sized displays were mostly using either rear or front projection technology, e.g., [15], the trend is now to build higher-resolution displays made of tiled LCD panels. Rear-projection systems are still being developed, and have advantages such as being compatible with direct touch manipulation across tiles (an interaction typically hindered by bezels). But the higher pixel density, currently around 100dpi, the easier setup, and higher display quality of LCD panels, has made them a popular approach, enabling display capacities up to about 200 megapixels [1, 4, 9, 24, 25, 26, 28]. Such display capacity enables the visualization of large images, but also the juxtaposition of multiple datasets in coordinated views for compare and contrast tasks [29].

North and colleagues have run numerous studies on ultra-high-resolution large displays. They evaluated user performance on tasks such as path tracing, visual search and comparison, observing the behavior of users in terms of display space usage [1] and physical vs. virtual navigation [3, 4, 5] in various configurations [31]. They also evaluated the perceptual scalability of visualizations on high-pixel-density large displays [38, 39]. The main finding relevant to our context is that users do benefit from the increased display capacity and prefer physical, over virtual, navigation.

The effects of bezels have been observed and studied on desktop multi-monitor setups [14, 33]. In this context physical discontinuities help organize the workspace and do not seem to affect performance on divided attention tasks. Mouse Ether [7] and Perspective cursor [23] offer ways to facilitate pointing across display-less space such as the space covered by bezels. Potential issues caused by bezels on larger displays have been studied in depth by Bi and Balakrishnan [8]. They found them to have a negative impact on tunnel steering tasks, but not on pointing or visual search tasks, except for one particular case, when the target object is visually split across a bezel. The problem of bezels has also been mentioned in many of the above-referenced works. But no solution has been proposed to address this issue beyond the basic and unsatisfactory solution that consists in toggling between offset mode (ignoring bezels and introducing distortion) and overlay mode (rendering the visualization as if there were pixels behind them) [20, 27].

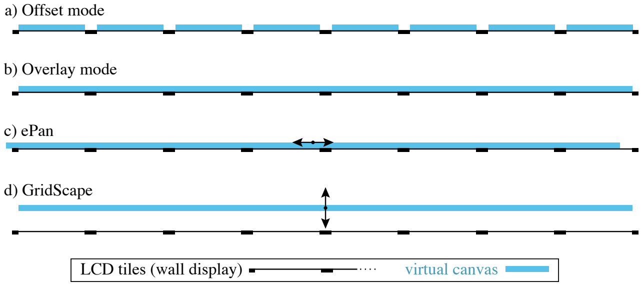


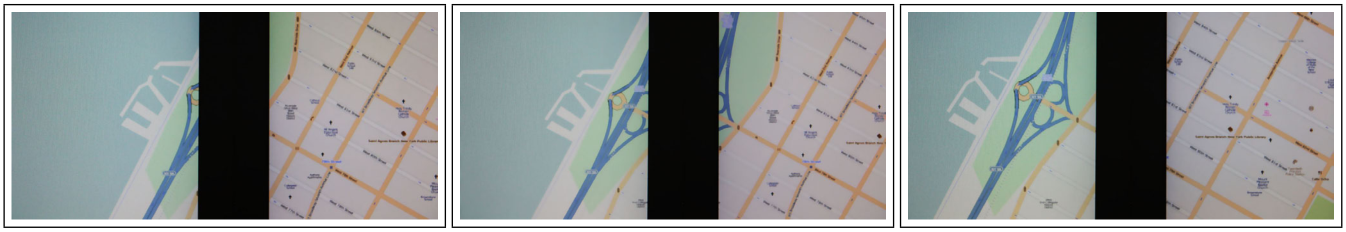
Figure 2: Conceptual representation (from above) of offset and overlay modes, ePan translation, GridScape's offset in depth.

Tiled++ [9] projects a low-resolution version of the hidden regions on the bezels themselves. However, this only provides users with some sort of coarse *preview* of hidden information. Users still require a technique to fully reveal that information, as the much-lower resolution implies that text will often be illegible and many details will be lost. The technique can nevertheless be an interesting complement to other techniques that reveal content. One significant drawback of Tiled++ is that it requires both LCD tiles with frames coated with material that can reflect projected light back *and* a set of projectors for front projection on those frames. Such a setup has a significant cost, both financial and technical (precise alignment, color calibration, maintenance of equipment, performance), that makes it impractical.

Finally, GridScape uses motion parallax, a perceptual phenomenon well studied in experimental psychology [12] that has been used in graphical interfaces as a depth cue [19, 21] to facilitate 3D tasks [2, 17, 35] or to enhance videoconferencing [11, 16]. It also enables the illusion of 3D perception in the pCubee cubic display [32], and plays a role in the E-conic [22] perspective-aware interface for multi-display environments: based on information about the location of both users and display surfaces, E-conic renders 2D graphics as if they were floating in space perpendicular to users.

3. TECHNIQUES

ePan and GridScape can be seen as extensions of the overlay approach. At a conceptual level, the metaphor is that of looking at a very large poster through a french window, or grid, of the same dimensions. The monitors become window glass panes, and the poster is a virtual 2D canvas on which the visualization is painted. In the real world, the region of the poster that one sees through one of the glass panes not only depends on the poster's position in space, but also on the observer's position. Conversely, regions hidden from the viewer by the bezels also depend on those two positions. One exception to this rule is when the poster is right behind the glass panes: in that case, the observer's position does not change anything, as *"Points which lie at the same depth as the screen are the only ones which do not "move" relative to the screen as the viewpoint changes"* [21]. This particular case corresponds to what happens in the default overlay approach described earlier.



Overlay mode, default position — Offset mode — Overlay mode, panned leftward with ePan

Figure 3: Close-up on junction between Henry Hudson Parkway and W79th street in Manhattan. Left (overlay mode, default position): the junction is hidden behind a bezel. Middle (offset mode): the junction can also be revealed by switching to offset mode. This removes any occlusion, but roads are no longer continuous, making them difficult both to understand and follow. Right (panned leftward): the junction can be revealed by invoking ePan through a simple drag gesture.

3.1 ePan

The ePan technique lets users pan the entire virtual canvas with finger drag gestures. The canvas is positioned right behind the glass panes, and gets translated within that plane (Figures 2-c and 3). Users can perform the gestures either directly on the wall, provided its surface is touch-enabled, or using a handheld device. The first option does not require users to carry a handheld device, but forces them to be within arm's reach of the wall when invoking ePan. The second option enables them to invoke the technique from anywhere. Control-display gain between the handheld's surface and the wall display is set so that a typical finger drag on the handheld will translate the entire canvas by slightly more than twice a bezel's width. This ensures that users can comfortably reveal all hidden information, plus some context, with a single drag gesture.

ePan can behave in either of two ways when releasing the finger from the surface (handheld device or wall) after a drag. The virtual canvas can stay in its current position, or it can revert back to its original position automatically. The second behavior is interesting when users only want to have a brief look at what is usually hidden behind bezels, as the cost to revert to the original state is much lower than with the first behavior.

3.2 GridScape

The second technique, GridScape, virtually moves the canvas "behind" the display, typically 20 to 30 cm (Figure 2-d). This depth is enough to produce a motion parallax effect [12] when the user walks or leans in front of the wall: bezels appear to be on a front plane, and as the user's point of view changes, regions of the canvas that were previously hidden behind bezels get progressively revealed while other areas get occluded, as illustrated in Figures 1 and 4. GridScape relies on physical navigation to achieve this effect. Physical navigation happens naturally in front of large displays [6], and we believe that it can help users gain a global comprehension of the displayed scene thanks to the human brain's ability to build a coherent model of the world even if some objects get hidden as the observer moves [34]. This ability should help minimize the cognitive burden associated with having to deal with the discontinuities caused by bezels.

Users can choose to have GridScape "always on", or to activate it on demand. They can also adjust the depth of the virtual canvas according to their preference by, e.g., sliding their finger upward on a handheld device. The canvas' depth is controlled through a non-linear function of dragging distance. The function allows a maximum depth of one meter and has a higher slope at lower values, to allow quick access to mid-range depths. When the finger is released, the image comes back to its original position, automatically deactivating the technique, thus minimizing the cost to revert to the original view state, as for ePan.

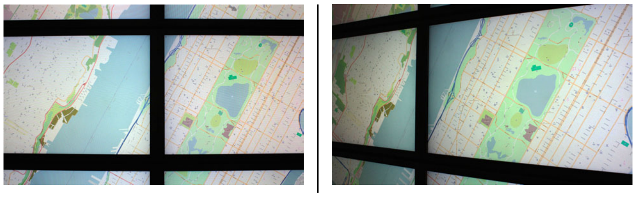


Figure 4: Revealing the same junction as in Figure 3 simply by performing physical navigation using GridScape.

3.3 Prototype

Both techniques can be implemented using various hardware and software solutions. We developed a fully-functional prototype for the display wall in our laboratory, that consists of 32 high-resolution 30" LCDs laid out in an 8×4 matrix, 5.5 meters wide and 1.8 meters high (see Figure 5). This wall can display 20480×6400 pixels, and is driven by a cluster of 16 computers, each equipped with two high-end nVidia 8800GT graphics cards.

While the two techniques are conceptually easy to understand, implementing them on such ultra-high-resolution wall displays requires distributing very large mipmaps on the multiple GPUs and computers that drive the wall, to eventually form a single coherent image. This image has to be rendered at a high frame-rate, typically 60fps, as otherwise the visual metaphor would die. GridScape and ePan are developed in C++ using Equalizer [10], a framework for OpenGL rendering on clusters of computers. The virtual canvas is made of a large rectangular mesh on which textures are applied. The two techniques are implemented by translating that mesh orthogonally to (GridScape), or within (ePan), the wall display's plane. In GridScape, the camera's frustum adapts to the user's 3D head position and orientation. This information is obtained from a real-time motion-tracking system, as detailed later.

Most graphics cards can't handle textures large enough to cover wall-sized displays, because of hardware video memory limitations and restrictions on maximum texture size in OpenGL. Even if it were possible (using for instance lossy texture compression), the time to transfer new tiles when the visualization changes would be prohibitive. As panning is locally bounded to the width of about two bezels on all sides of each viewport in the case of ePan, each tile only has to load and display a relatively small region of the virtual canvas. Thus, cropping the canvas to that region viewport-wise provides a simple solution to the problem of limited memory resources. Things are more complex for GridScape, since the perspective projection requires, in extreme situations, loading and rendering the entire canvas on a single screen, putting a heavy load on the associated cluster node. However, taking into account the change in scale induced by the perspective adaptation, it is never the case that a screen has to render an image as big as the original canvas in terms of resolution. Several tiling techniques were tested and discarded. We eventually developed an algorithm based on a quadtree representation of the scene that allows loading images of arbitrary size. Tiles at different mipmap levels [37] are stored in a single big texture, which works as a paged virtual memory. A *GLSL shader* uses this representation as if it were a real texture. Compared to a tiling approach, it is faster (all tile management is done on the GPU, sparing CPU cycles), more memory efficient (only the mipmap levels actually used are loaded) and easier to use since there is no need to split the mesh into smaller ones that use a single texture, as would be the case with tiling solutions.

4. EMPIRICAL EVALUATION

As mentioned earlier, bezels cause two types of problems. In overlay mode, they hide potentially useful or even important information. This problem can be partially addressed by switching to offset mode, as bezels do not hide anything any more: users can perform visual searches with the guarantee that all information items are visible, and then revert back to overlay mode. There are some issues when target objects are split on the two sides of a bezel [8], making it difficult to recognize the object even when performing attentional visual search tasks, but such configurations are relatively infrequent. Providing a toggle between overlay and offset modes is not sufficient, however, to address the second problem, i.e., that bezels cause discontinuities that make it difficult to perform some very common tasks on wall displays such as interpreting information items that span multiple tiles, or relating information items located on different tiles.

Following a route on a map, or finding a path between two nodes in a tree or graph, are common examples of such tasks, that neither overlay nor offset modes support in a satisfactory manner. As illustrated in Figure 3, bezels significantly hinder this type of tasks, because of the Poggendorff illusion [13], because of the Gestalt law of continuity [34], and because there is uncertainty about connectivity when two or more paths cross behind a bezel.

We conducted a path tracing task experiment to compare the performance and limits of ePan and GridScape, against existing techniques. This task is similar to other tasks studied in previous laboratory experiments that have evaluated user performance in such contexts [31, 9]. Our task operationalization is tailored to the specific research questions related to the study of bezels on wall displays, how to reveal and relate information hidden behind them. The goal of this experiment is to evaluate how easy the various techniques make it for users to follow paths of varying complexity that span multiple display tiles. We compare conditions that require explicit input from the user to reveal hidden content, to conditions that rely mainly on implicit information gathered from users' physical navigation. The task is performed on a 2D dataset, as we are mostly interested in solving the problem of data being occluded by bezels on wall displays. Motion parallax is one possible answer to this problem, and the fact that it provides a strong depth cue that can be used to enhance 3D visualization is another research question, addressed elsewhere [2, 17, 35].

To our knowledge, overlay and offset modes are the only existing solutions to address bezel-related issues on wall displays. Including both techniques as separate conditions in the experiment would not make much sense, as neither of them, taken alone, makes it possible to complete path tracing tasks. Overlay makes it challenging, if not impossible, because paths often cross behind bezels. Offset is not an option either, as it introduces significant distortion in the perceived image. This does not prevent, but often makes more difficult, relating path sections on opposite sides of bezels as those are no longer aligned. We thus decided to combine the two modes in a single technique, that we refer to as OFFSET. This basic technique enables users to dynamically toggle between overlay (default) and offset modes, i.e., between alignment-preserving and visibility-preserving modes. We thus compared three techniques: GRIDSCAPE, EPAN, and OFFSET. As we wanted to focus the comparison on *explicit virtual control* versus *implicit physical navigation*, we selected activation methods and low-level controls as similar as possible between the three techniques, to avoid introducing unwanted bias. Thus, all techniques are activated using a handheld device. OFFSET switches from overlay to offset mode when participants touch the handheld's surface, and automatically reverts back to overlay mode when they lift their finger from that surface. Similarly, EPAN and GRIDSCAPE revert the view back to its original configuration on finger release (see previous section for details).

4.1 Apparatus

Hardware. The display wall used in the experiment is the one described earlier in Section 3.3. Our goal is to identify the performance characteristics of each technique from the user's perspective. It is thus essential that each technique operates equally well from a purely technological perspective. In all conditions, we use a VICON motion-capture system to track passive IR retroreflective markers and provide the 3D coordinates of the participant's head with sub-millimeter accuracy at 200Hz. Indeed, although gesture recognition technologies are constantly improving, such a system is still necessary to get reliable and precise 3D position/orientation information. The system is used both to log participants' physical navigation [4] and as a head tracker for the GRIDSCAPE condition. The handheld device is an iPod Touch. As activation does not require high-precision input, participants can operate the device using a single hand (we systematically choose their dominant one for the experiment) with their arm in a relaxed position (they do not need to look at the device, thus avoiding problems of divided attention).

Software. The experiment uses the implementation described earlier. Input commands from the iPod Touch and from the experimenter's console (to start and stop trials, and input answers to the log system) are sent to this program using the OSC communication protocol. Neither the VICON motion tracking system nor the graphics rendering pipeline showed any noticeable lag, and both yielded high-enough refresh rates ($\geq$ 60Hz) that there was no problem of spatial and temporal accuracy in head tracking [2] and adjustment of the rendering in the GRIDSCAPE condition.

4.2 Participants

Twelve volunteers (six female), from 24-38 years (avg. 31.08, med. 31.5), all daily computer users with normal or corrected-to-normal vision, no color blindness, served in the experiment.

4.3 Task & Procedure

In a scene containing eight lines that cross each other multiple times, participants are instructed to follow one of those lines from one extremity of the wall to the other (see Figure 5). At the beginning of each trial, the display is blank. Participants are instructed to position themselves either near the left or right extremity of the display depending on *Direction*: trials alternate between left-to-right and right-to-left to minimize unnecessary travel in the room, and because we want to evaluate the influence of direction as a secondary factor. Once participants are ready, the scene is revealed to them and the trial starts. The line to follow is indicated by a prominent red arrow (Figure 5, top-left corner), and each possible exit point on the other side is labeled by a letter (A-H). Participants are encouraged to minimize the number of errors, while going as

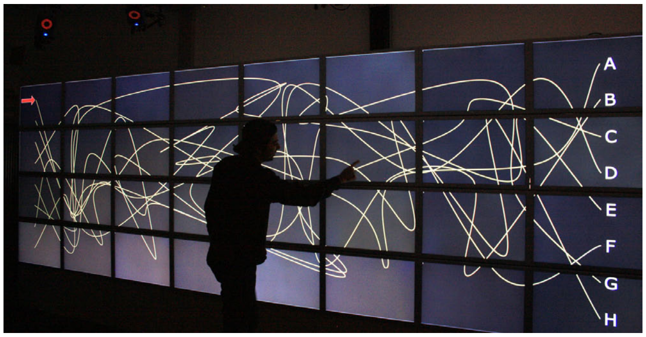


Figure 5: Example path tracing trial, *Direction*= LR.

fast as possible. To avoid them rushing through the experiment, a 10-second delay penalty is enforced before the next trial can start (practice trials excluded) in case of an erroneous answer.

As we want participants to freely move in front of the display during trials, they are instructed to say the answer out loud as soon as they find it, so as to avoid problems of divided attention entailed by the use of a handheld device to input answers, which would also be tiresome. Using a rolling stand would have been even more cumbersome than a handheld device, and would have introduced noise in the collected data had it been setup in a fixed location. Moreover, it was observed to have an effect on user behavior in previous experiments [31]. The experimenter stops the trial as soon as he hears the participant's answer. Though this method introduces some delay before the trial timer is actually stopped, this is typically of the order of 250 milliseconds, which is negligible compared to the average task completion time.

This experiment is a 3 × 2 × 2 within-subject design with the following factors: *Tech*nique (EPAN, GRIDSCAPE, OFFSET), *Difficulty* (EASY, HARD), and *Direction* (LR, RL) for left-to-right and right-to-left, respectively.

Task difficulty is controlled as described in Section 4.4. We group trials into three blocks, one per technique (*Tech*), so as not to disturb participants with too many changes between interaction techniques. The presentation order of *Tech* is fully counterbalanced with 6 combinations across the 12 participants. Within a *Tech* block, each participant sees two sub-blocks: a practice block containing 4 trials of increasing *Difficulty*, and a second block of 12 actual measurements. In the latter block, the presentation order of *Difficulty* is counter-balanced across techniques using a pseudo-random sequence of *Difficulty* conditions. Before each *Tech* condition, the experimenter took 2-3 minutes to explain the technique to be used next. The average per-participant duration of the experiment was 50 minutes.

To summarize, we collected 3 *Tech* × 2 *Difficulty* × 2 *Direction* × 3 replications × 12 participants = 432 trials for analysis. We measured task completion *Time* and *Error*s. Participants had to fill a post-hoc questionnaire. They were asked to rank the techniques by order of preference, and were encouraged to make additional observations and comments.

4.4 Path Generation and Rendering

The start and endpoints of the eight lines making a trial are evenly spaced along the vertical axis (see Figure 5). The lines are Catmull-Rom splines defined by nine control points each. Their vertical position coordinates are chosen randomly within the display's bounds (uniform distribution). The horizontal coordinates are chosen randomly from Gaussian distributions with evenly-spaced centers, and widths determined by a trapezoidal function so that horizontal oscillations are almost null near the lines' extremities and greater in the middle of the wall. Finally, the Fisher-Yates shuffling algorithm is used to make a random permutation of the vertical positions of all lines' endpoints, so that the start index of a given line is different from its end index [18].

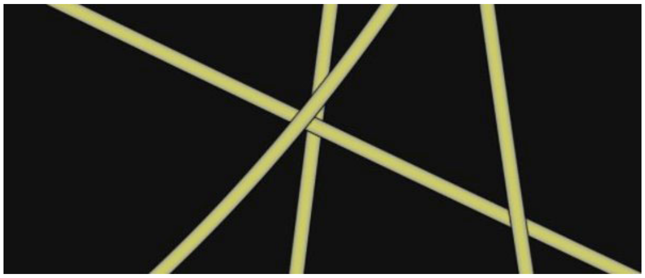


Figure 6: Stroked paths (thin black contour) and shadows help disambiguate intersections.

The number of intersections between the selected line and the other seven lines is used as a metric for assessing difficulty [36]. Intersections are evaluated through the number of pixels that participate in the crossings, as opposed to the intersection of two *mathematical* curves, as we find the first measure to be a better representative figure of the visual experience of seeing an intersection displayed on a computer screen. We first generated a large set of trials (40,000) to gather statistics that would then help us characterize visual complexity for this task: length of a line (in pixels), number of pixels of a line involved in an intersection, percentage of line length hidden behind bezels, percentage of pixels involved in an intersection hidden behind bezels.

For each trial, we select the line featuring the most intersections behind bezels as the line that the participant will be asked to follow. The mean value of hidden intersections for our 40,000-trial set was $h = 19.7$ pixels (standard deviation $SD = 6.5$). We then generated a new set of trials, in which we tried to identify both EASY and HARD conditions.

HARD trials were those in which the line to follow featured a high number of intersections: $h > mean + 2.5 \times SD$, i.e., more than 38. EASY ones were those in which that line presented a low number of intersections: $mean - 2.0 \times SD < h < mean - 1.5 \times SD$, i.e., between 7 and 10. A second filter kept only the trials that featured a line to follow with approximately 16% of its pixels hidden behind bezels[1]. Finally, we checked that all trials for a given condition (EASY or HARD) were indeed of roughly the same difficulty by running a pilot study with two participants (not involved in the main experiment) and only one technique (EPAN) to compare trials according to the sole criterion of difficulty. Some trials were eliminated for being too easy, not requiring any interaction (i.e., one could achieve the task just by looking at the display). Others were eliminated for being too difficult (e.g., two lines overlapping over a long distance). Difficulty was assessed through task completion time and error rate. Approximately 20% of trials had to be replaced by other trials taken from the set (new trials were checked in the same conditions).

All lines are painted yellow and are 14 pixels/3.5mm in width. They are decorated with a solid black border and a shadow effect simulated using alpha-compositing (Figure 6). We chose this graphical design to avoid ambiguities inherent to complex path crossings, and to make them look like roads on a map, with clear indications about what line is above the other.

[1] This ratio corresponds to the ratio of pixels behind bezels over the total number of pixels of the wall display used in the experiment.

4.5 Hypotheses

We made the following hypotheses, based on the theory of visual perception [34], prior empirical evidence and our own experience using the three techniques:

- H_1: EPAN performs better than OFFSET as the mechanism to disambiguate paths crossing behind bezels should be less error-prone with EPAN: better support for the Gestalt law of continuity, no side-effect misalignments. See Figure 3 for a detailed example comparing the two approaches.
- H_2: GRIDSCAPE also performs better than OFFSET, enabling the user to focus on her task, as revealing what is behind bezels with the former is strongly integrated in the physical navigation process (Figure 4) that naturally occurs in front of wall displays [4].
- H_3: There should not be any significant effect of direction (left-to-right or right-to-left) on performance.

We did not formulate an hypothesis on the performance ordering of GRIDSCAPE and EPAN as the two techniques have their own advantages and drawbacks. GRIDSCAPE might feel more intuitive to use than EPAN, being naturally integrated with physical navigation, and possibly providing a more holistic view on the scene (supported by the french window metaphor). But GRIDSCAPE also constrains users' movements, as the visibility of a region is strongly coupled with their body position and head orientation. This might actually hinder task performance.

4.6 Results

Prior to our analysis, we checked for unwanted effects from secondary factors such as fatigue or asymmetrical transfer, and did not find any. In all analyses, we handle participant as a random variable, using the standard repeated measures REML technique, and perform Tukey HSD post-hoc tests for pairwise comparisons.

We perform a multi-way ANOVA for the following model: *Time* ∼ *Tech* × *Difficulty* × *Direction* × Rand(Participant). *Tech* has a significant effect on *Time* ($F_{2,22} = 5.87$, $p < 0.01$). EPAN (avg. $35.9s$) is significantly faster than both GRIDSCAPE (avg. $41.4s$) and OFFSET (avg. $42.3s$). As expected, *Difficulty* has a significant effect on *Time* ($F_{1,11} = 153.1$, $p < 0.0001$) with EASY trials being significantly shorter (avg. $26.8s$) than HARD trials (avg. $53s$). Confirming H_3, we do not observe a significant effect of *Direction* on *Time* and this factor is omitted in the remainder of this analysis.

We find a significant *Tech* × *Difficulty* interaction effect ($F_{2,22} = 3.79$, $p < 0.04$) due to the degradation of GRIDSCAPE's performance between the EASY and HARD conditions, illustrated in Figure 7-a: while GRIDSCAPE has an average task *Time* close to that of EPAN for the EASY condition (avg. $25.5s$ vs. $24.6s$, respectively), with OFFSET performing significantly worse (avg. $30.4s$), the performance of GRIDSCAPE actually gets worse than that of OFFSET (avg. $57.7s$ vs. $54.4s$, respectively), with EPAN still performing significantly better than both of them (avg. $47.2s$).

In terms of errors, a nominal logistic regression reveals a significant effect for *Difficulty* only ($\chi^2 = 18.23, p < 0.0001$). As shown in Figure 7-b, participants make a very low and similar number of errors in EASY trials across techniques. Errors are naturally more numerous in HARD trials. There are more errors with OFFSET than the two other techniques, but this difference is not significant.

4.7 Subjective Feedback

8 participants out of 12 ranked EPAN as their preferred technique for this task, with 2 ranking it third. GRIDSCAPE also got positive comments and was ranked first by 5 participants (they were allowed to rank more than one technique at the same position). OFFSET got much less enthusiastic ratings: it was ranked last by 8 participants and never ranked first.

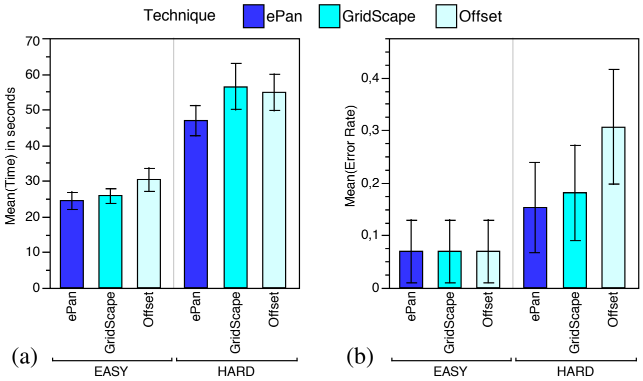


Figure 7: (a) *Time* (seconds) per *Tech* × *Difficulty*, (b) *Error* rate per *Tech* × *Difficulty*. Error bars show the 95% confidence limit of the mean.

Several participants found GRIDSCAPE to be *"more intuitive"*, *"good for accuracy"* but also *"more fluid than the others"*, though one participant mentioned that he would have liked to practice it more and another said that he did not understand why *"it was moving with [him] when [he] went to the right"*. Even though participants did not perform better with this technique, they did enjoy using it, probably because of the novelty effect and the positive impression it makes on first-time users.

EPAN was described by one participant as *"fast and efficient to resolve ambiguities"*. Participants also made comments about the way in which they interacted with the technique. One participant wanted to replace elastic control in EPAN by free pan (zero order of control with clutch) because the participant *"did not want to release [his] finger from the iPod, otherwise [he] is lost"*. As mentioned in Section 3.1, this behavior is implemented, but was not made available in the experiment to avoid potential sources of noise and bias between techniques. Several participants also commented about how entertaining the task was, looking more like a game than an experiment, which correlates well with the observed level of engagement for most participants.

5. DISCUSSION

Results confirm H_1: EPAN performs significantly better than OFFSET in terms of task completion time. Figure 7-b suggests that OFFSET is more error-prone than the other techniques, but we cannot claim this as we did not observe a statistically significant difference between techniques in terms of errors, probably because participants did follow our instructions to make as few errors as possible very seriously. H_2 was not confirmed, as GRIDSCAPE did not perform significantly better than OFFSET. Actually, as shown in Figure 7, GRIDSCAPE performed relatively well in EASY trials, but its performance degraded considerably in HARD trials.

Note that these results are not in contradiction with those from earlier studies [2, 17, 35] about the usefulness of motion parallax for path tracing tasks. Those experiments were assessing the effectiveness of motion parallax as a depth cue for tracing paths in 3D visualizations. Our task was a 2D path tracing task, where motion parallax is used not so much as a depth cue, but as a behavior enabling users to reveal regions of images and other types of 2D visualizations. In our case, the illusion of depth obtained with GRIDSCAPE is a means rather than an end.

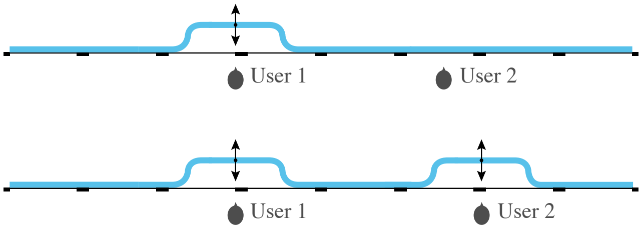


Figure 8: (top) locally-bounded transformations allow User 1 to look behind bezels without disturbing User 2. (bottom) User 1 and User 2 can invoke the technique independently.

Overall, these results indicate that for path tracing tasks that are not too complex, both EPAN and GRIDSCAPE improve performance significantly compared to basic overlay and offset modes. In such situations, the fact that GRIDSCAPE can be controlled through simple head and body movements alone has two main advantages. First, the reliance on an easy-to-understand metaphor that smoothly integrates with physical navigation makes the technique interesting for public walk-up-and-use systems where input devices might not be available. Second, it can leave the hands free to perform other tasks, such as pointing, virtual navigation and other interactions that require input devices and possibly involve both hands already [25]. For more complex path tracing tasks, EPAN still provides the best performance figures. GRIDSCAPE degrades significantly, actually yielding higher completion times than OFFSET. Though GRIDSCAPE did seem to provide users with a more holistic view of the scene – participants could easily follow paths from tile to tile without resorting to elaborate strategies as was the case with OFFSET – it seems that, despite its integration with the physical navigation patterns that users naturally adopt in front of wall displays [5, 6], the additional physical effort in terms of body and head movements resulted in poor performance for difficult trials.

6. SUPPORTING MULTIPLE USERS

One limitation of all techniques studied so far (OFFSET included) is that they work for a single user only. Wall displays being natural platforms for cooperative work, it is important that the techniques can function in multi-user scenarios: a user should be able to activate a technique to look behind bezels near her without interfering with other users' activities, i.e., without changing the visualization in the regions they are focusing on.

A straightforward solution for OFFSET consists in toggling between overlay and offset modes only for the tiles located within the field of view of the user who is invoking the technique. Changes are applied locally, based on the user's position and orientation relative to the wall, and do not affect the other users' focus area – except for a user who would be standing back, looking at an overview of the entire collaborative workspace, but then such a user would arguably not be strongly impacted by those small-scale visual changes.

This idea of per-user, locally-bounded visual transformations can also be adapted to the other two techniques. One solution for EPAN is to apply a locally-bounded translation to the rectangular subregion corresponding to the user's visual focus area: the user can choose to trigger EPAN in cooperative mode, which only translates that subregion (again computed from her position and head orientation) in the virtual canvas' plane, instead of translating the whole representation as the original EPAN technique does. Using a continuous version of Stretchable Rubber Sheets [30], the immediate surroundings of this subregion get distorted (compressed or stretched depending on direction) to achieve a smooth transition with the other regions that form this user's context and the other users' focus area, that the transformation has left unaffected. Each user is free to invoke EPAN independently, for her own purposes.

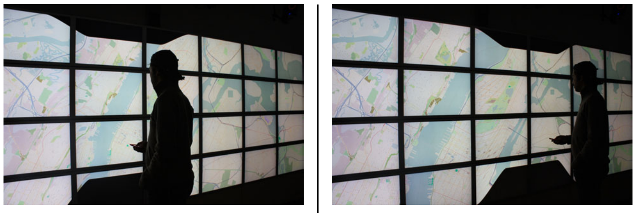


Figure 9: Locally-bounded version of GridScape, centered on the user's position. Other regions remain unaffected.

A similar approach can be employed for GRIDSCAPE, offsetting (in depth) only the subregion that is in the user's focus area, as illustrated in Figure 8 (top). This case requires a more elaborate mapping, using fisheye-like 1D distortion to smoothly integrate the locally transformed view into the surrounding context (though here the focus area gets zoomed-out to enable the motion parallax effect, while it is usually magnified in fisheye-based focus+context visualizations). Once offset in depth, the subregion's outer boundaries remain fixed, with only the offset plane and surrounding transition areas adapting to the user's head and body movements to simulate motion parallax. In our implementation, we achieve this effect by distorting the rectangular mesh on which the image is mapped, again using a locally-bounded transformation. Applying this distortion requires subdividing the originally very simple mesh so as to provide a smooth, continuous transition between offset and unaffected regions. Figure 8 (top) illustrates the deformation applied to the mesh; Figure 9 depicts the effect for one user in two different locations. As shown in Figure 8 (bottom), each user is free to invoke this technique independently, as for EPAN.

7. SUMMARY AND FUTURE WORK

The bezels that frame each monitor on ultra-high-resolution wall displays cause problems of occlusion or distortion. Those bezels regularly decrease in size, but are unlikely to completely disappear soon. We designed two techniques that enable users to look at information hidden behind them. We showed that for path tracing tasks of moderate complexity, both the technique that relies on explicit user actions (EPAN) and the one that relies on more implicit information from users' physical navigation (GRIDSCAPE) do improve performance compared to the basic overlay/offset techniques in use today. For more complex tasks, EPAN has a clear advantage over all other techniques. The techniques can be adapted to multi-user scenarios, and future work will focus on further exploration of solutions for such contexts of use, e.g., what are the most efficient distortion drop-off functions, and how to handle overlapping regions when two users are near one another.

Acknowledgements

We wish to thank Olivier Chapuis for helpful comments on early drafts of this paper. We also wish to thank all the volunteers who participated in our experiments. This work was supported by a Région Île-de-France / Digiteo grant.

8. REFERENCES

[1] Andrews, C., Endert, A., and North, C. Space to think: large high-resolution displays for sensemaking. In *Proc. CHI '10*, ACM (2010), 55–64.

[2] Arthur, K. W., Booth, K. S., and Ware, C. Evaluating 3d task performance for fish tank virtual worlds. *ACM Trans. Inf. Syst. 11* (July 1993), 239–265.

[3] Ball, R., and North, C. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *Proc. CHI '05 Extended Abstracts*, ACM (2005), 1196–1199.

[4] Ball, R., and North, C. Visual analytics: Realizing embodied interaction for visual analytics through large displays. *Comput. Graph. 31* (June 2007), 380–400.

[5] Ball, R., and North, C. The effects of peripheral vision and physical navigation on large scale visualization. In *Proc. GI '08*, Canadian Information Processing Society (2008), 9–16.

[6] Ball, R., North, C., and Bowman, D. Move to improve: promoting physical navigation to increase user performance with large displays. In *Proc. CHI '07*, ACM (2007), 191–200.

[7] Baudisch, P., Cutrell, E., Hinckley, K., and Gruen, R. Mouse ether: accelerating the acquisition of targets across multi-monitor displays. In *CHI '04 Extended Abstracts*, ACM (2004), 1379–1382.

[8] Bi, X., Bae, S.-H., and Balakrishnan, R. Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection. In *Proc. CHI '10*, ACM (2010), 65–74.

[9] Ebert, A., Thelen, S., Olech, P.-S., Meyer, J., and Hagen, H. Tiled++: An enhanced tiled hi-res display wall. *IEEE TVCG 16*, 1 (2010), 120–132.

[10] Eilemann, S., Makhinya, M., and Pajarola, R. Equalizer: A scalable parallel rendering framework. *IEEE TVCG 15*, 3 (2009), 436–452.

[11] Gaver, W. W., Smets, G., and Overbeeke, K. A virtual window on media space. In *Proc. CHI '95*, ACM Press/Addison-Wesley Publishing Co. (1995), 257–264.

[12] Gibson, E., Gibson, J., Smith, O., and Flock, H. Motion parallax as a determinant of perceived depth. *Journal of Experimental Psychology 58*, 1 (1959), 40–51.

[13] Green, R., and Hoyle, E. The Poggendorff illusion as a constancy phenomenon. *Nature 200* (1963), 611–612.

[14] Grudin, J. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proc. CHI '01*, ACM (2001), 458–465.

[15] Guimbretière, F., Stone, M., and Winograd, T. Fluid interaction with high-resolution wall-size displays. In *Proc. UIST '01*, ACM (2001), 21–30.

[16] Harrison, C., and Hudson, S. E. Pseudo-3d video conferencing with a generic webcam. In *Proc. IEEE Multimedia (ISM)*, IEEE Comp. Soc. (2008), 236–241.

[17] Hassaine, D., Holliman, N. S., and Liversedge, S. P. Investigating the performance of path-searching tasks in depth on multiview displays. *ACM Trans. Appl. Percept. 8* (November 2010), 1–18.

[18] Knuth, D. E. *The Art of Computer Programming volume 2: Seminumerical algorithms.* Addison–Wesley, 1969, 124–125.

[19] Lee, J. C. Hacking the nintendo wii remote. *IEEE Pervasive Computing 7* (July 2008), 39–45.

[20] Mackinlay, J. D., and Heer, J. Wideband displays: mitigating multiple monitor seams. In *CHI '04 Extended Abstracts*, ACM (2004), 1521–1524.

[21] McKenna, M. Interactive viewpoint control and three-dimensional operations. In *Proc. I3D '92*, ACM (1992), 53–56.

[22] Nacenta, M. A., Sakurai, S., Yamaguchi, T., Miki, Y., Itoh, Y., Kitamura, Y., Subramanian, S., and Gutwin, C. E-conic: a perspective-aware interface for multi-display environments. In *Proc. UIST '07*, ACM (2007), 279–288.

[23] Nacenta, M. A., Sallam, S., Champoux, B., Subramanian, S., and Gutwin, C. Perspective cursor: perspective-based interaction for multi-display environments. In *Proc. CHI '06*, ACM (2006), 289–298.

[24] Nam, S., Deshpande, S., Vishwanath, V., Jeong, B., Renambot, L., and Leigh, J. Multi-application inter-tile synchronization on ultra-high-resolution display walls. In *Proc. MMSys '10*, ACM (2010), 145–156.

[25] Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., and Mackay, W. Mid-air Pan-and-Zoom on Wall-sized Displays. In *Proc. CHI '11*, ACM (2011), 177–186.

[26] Pietriga, E., Huot, S., Nancel, M., and Primet, R. Rapid Development of User Interfaces on Cluster-driven Wall Displays with jBricks. In *Proc. EICS '11*, ACM (2011), 185–190.

[27] Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., and Tan, D. The large-display user experience. *IEEE CG&A 25* (July 2005), 44–51.

[28] Rønne Jakobsen, M., and Hornbæk, K. Sizing up visualizations: effects of display size in focus+context, overview+detail, and zooming interfaces. In *Proc. CHI '11*, ACM (2011), 1451–1460.

[29] Sandstrom, T. A., Henze, C., and Levit, C. The hyperwall. In *Proc. Conf. on Coordinated and Multiple Views In Exploratory Vis. '03*, IEEE (2003), 124–133.

[30] Sarkar, M., Snibbe, S. S., Tversky, O. J., and Reiss, S. P. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *Proc. UIST '93*, ACM (1993), 81–91.

[31] Shupp, L., Andrews, C., Dickey-Kurdziolek, M., Yost, B., and North, C. Shaping the display of the future: The effects of display size and curvature on user performance and insights. *Human Computer Interaction 24* (April 2009), 230–272.

[32] Stavness, I., Lam, B., and Fels, S. pCubee: a perspective-corrected handheld cubic display. In *Proc. CHI '10*, ACM (2010), 1381–1390.

[33] Tan, D., and Czerwinski, M. Effects of visual separation and physical discontinuities when distributing information across multiple displays. In *Proc. INTERACT'03* (2003), 252–255.

[34] Ware, C. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2004. 2nd Edition.

[35] Ware, C., and Franck, G. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans. Graph. 15* (April 1996), 121–140.

[36] Ware, C., Purchase, H., Colpoys, L., and McGill, M. Cognitive measurements of graph aesthetics. *Information Visualization 1* (June 2002), 103–110.

[37] Williams, L. Pyramidal parametrics. In *Proc. SIGGRAPH '83*, ACM (1983), 1–11.

[38] Yost, B., Haciahmetoglu, Y., and North, C. Beyond visual acuity: the perceptual scalability of information visualizations for large displays. In *Proc. CHI '07*, ACM (2007), 101–110.

[39] Yost, B., and North, C. The perceptual scalability of visualization. *IEEE TVCG 12* (September 2006), 837–844.