Supporting Multiple Off-Axis Viewpoints at a Tabletop Display

Mark Hancock, Sheelagh Carpendale
University of Calgary
Calgary, AB, Canada
{msh,sheelagh}@cs.ucalgary.ca

Abstract

A growing body of research is investigating the use of tabletop displays, in particular to support collaborative work. People often interact directly with these displays, typically with a stylus or touch. The current common focus of limiting interaction to 2D prevents people from performing actions familiar to them in the 3D world, including piling, flipping and stacking. However, a problem arises when viewing 3D on large displays that are intended for proximal use; the view angle can be extremely oblique and lead to distortion in the perception of the 3D projection. We present a simplified model that compensates for off-axis viewing for a single user and extend this technique for multiple viewers interacting with the same large display. We describe several implications of our approach to collaborative activities. We also describe other display configurations for which our technique may prove useful, including proximal use of a wall or multiple-display configurations.

1. Introduction

Interest in supporting co-located collaboration via tabletop displays has been growing rapidly. Interaction with these large displays is often provided through direct-touch or stylus input. When people use this form of interaction, we necessarily expect that they will use these displays from a very close distance. This close distance implies that the viewing angle is likely to be severely off-axis. For example, with a tabletop display the traditional viewpoint is centred directly above the screen. For normal viewing from any side of the table, one's viewpoint is extremely *off-axis* or *skewed*. This skew has already been shown to be problematic for perception of lengths in 2D interfaces on tabletop displays [19]. In 3D interfaces, the problem of off-axis viewing is exacerbated. The use of standard projection techniques (perspective and orthographic) may result in significant distortion when objects are viewed from off-axis (see Figure 1).

Despite the potential difficulties, there are several reasons to think that incorporating 3D into large-display interfaces would be beneficial. We all make good use of the third dimension in the physical world: we make stacks, piles and looser groupings, and turn items over and use their other side. The need to support these types of information-handling functionalities has been well discussed [1, 9, 17].

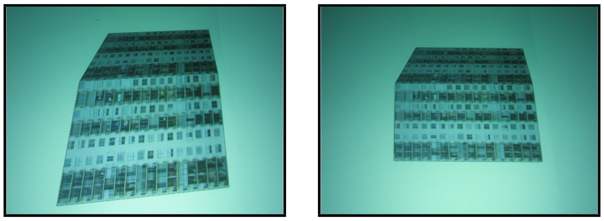


Figure 1. (left) Standard perspective projection, (right) compensated for the appropriate off-axis viewpoint. Both photographs are taken from the viewpoint of a person at one side of the table[1].

We present a simplified model to compensate for off-axis viewing of 3D objects and generalize this technique for use by many people around the display. After describing both fixed and customizable alternatives for multiple viewers, we discuss the implications of each for collaboration. Our technique can also be used to compensate for distortion for large displays in general and for multiple-display configurations.

2. Related Work

There are several areas of research that are relevant to our work. There are techniques to correct projector-displayed images onto a surface that is not perpendicular to the projector, thus *correcting off-axis projection* (of primarily 2D images) in a different way. There has been work on producing images with *alternate perspective rendering* techniques. *Fish-tank virtual reality* techniques have also been used to correct off-axis distortion using head tracking. Lastly, there are existing *3D tabletop display interfaces* that resolve the multiple-viewing-angle problem.

[1] Note that photographs and videos taken of displays containing 3D graphics from off-axis angles have the added complexity of the camera angle. We have worked with the camera to give the most realistic depiction of what one would see with the naked eye.

2.1. Correcting for Off-Axis Projection

Dorsey et al. [7] present a system that allows images to be projected onto a (possibly curved) display from an off-axis position. Lee et al. [12] extend this idea with a system that can adaptively calibrate as either the projector or display surface moves so that the image is displayed correctly, despite the angle of projection. PlayAnywhere [20] allows for off-axis projection from very close to a surface. While these systems correct distortion caused by varying the point of projection, our model compensates for varying points of view. Theoretically, these two types of corrections could work effectively in unison.

2.2. Alternate Perspective Rendering

Agarawala et al. [3] present a way of providing multiple camera viewpoints for each object in a scene. Ryan [5] allows a static image to be created by stitching together multiple viewpoints so even single objects can be distorted and viewed at multiple angles. While these works effectively demonstrate the expressive impact of integrating multiple camera scenes, our model focuses on adjusting 3D scenes to preserve perceived shapes and sizes of 3D objects and considers both interaction and the effects on multiple people collaborating.

Zorin and Barr [21] describe many of the limitations of the use of standard perspective projection in the creation of static 3D images. Specifically, they argue that a picture of a 3D scene cannot simultaneously satisfy both properties (1) that straight lines should appear straight and (2) that objects should appear as if viewed directly. They present a formalism to manually balance these two properties. In our case, this problem is exacerbated due to (a) the desire to have a 3D image appear correct for *multiple* people viewing the image and (b) to have the "images" being viewed be interactive, and thus dynamically changing perspective.

Our work differs significantly from research dealing with the correction of viewpoint for a single person viewing a static image. In the multi-user case, we do not attempt to provide an image viewable by all users, but provide a mechanism to distort the interface in a way that better supports each user in a portion of the display. We also discuss methods to smoothly transition objects between workspaces of multiple people. Our technique relies on a much simpler technique to provide perspective correction for one user, so that extending to the multiple user case becomes easier and allows us to explore a variety of approaches to this case.

2.3. Fish Tank Virtual Reality

Previous research also explores the correction of a 3D projection based on viewing angle [6, 13] and has been dubbed "Fish Tank Virtual Reality" [18]. These systems typically use one of two approaches: they either use stereoscopic (headtracked) goggles that project an image onto two surfaces that move with the user's head motion, or they track the user's head position and correct the view for the measured eyepoint. In the former case, no off-axis correction is necessary, since the viewplane is kept perpendicular to both eyes. In the latter case, the projection must account for a potentially off-axis viewing angle, and thus these systems also provide a solution to the off-axis distortion problem for a single user. These systems have considered many variables beyond the scope of our work (including refraction and the effect of curvature of CRTs), and could be substituted for our simpler single-user solution and extended in the same way for multiple viewers.

2.4. 3D Tabletop Display Interfaces

Tabletop display interfaces have typically been designed to use only two dimensions. However, much of the interaction that takes place on traditional tables makes use of the third dimension: piling, sorting, stacking, and using orientation to communicate intention to others. There have been some interfaces and interaction techniques developed that attempt to leverage these real-world counterparts [1, 9]. These techniques focus primarily on the interaction with artifacts in a 3D world, but do not consider how those artifacts are perceived by the viewers.

The two-user responsive workbench [2] addresses the problem of different viewpoints at a table by providing correct stereoscopic 3D images to two different people. The IllusionHole can be used to integrate 2D and 3D [14] on a table by limiting the portion of the display presented at each viewing angle via a hole in the table's centre. These systems both provide stereoscopic cues via headgear and a tracking system. Our solution differs from these systems in that we focus on correcting perspective cues and we require no headgear or tracking, and thus is more suited for casual large display use. The lack of headgear also allows for more seamless collaboration, due to the possibility for eye contact.

3. Correcting Off-Axis Distortion

We present a general method to compensate for distortion caused by off-axis viewing of 3D objects projected onto a 2D surface. We first describe how to adjust this distortion for a single person's perspective and then describe several techniques for extending this method for multiple viewing angles. Note that our model for correcting distortion for a single user can be accomplished using existing techniques [3, 13]. We present a simpler model with minimal changes to the current 3D projection methods, which allows us to more easily extend the technique to multiple users and to explore a large variety of alternative projections.

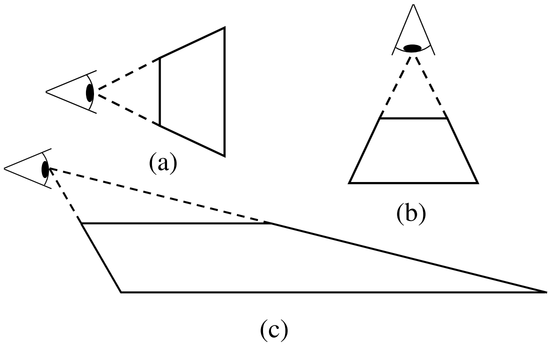


Figure 2. (a) Standard viewpoint works well for small vertical displays. (b) Standard viewpoint directly above the tabletop display. (c) Typical viewpoint for a tabletop display is close to the screen and to one side.

3.1. Correcting for One User

The standard method for projecting 3D graphics onto a 2D surface assumes that the viewpoint is at the centre and directly in front of the display (see Figure 2). A common method to achieve this projection is by transforming points in a 3D model to a canonical viewing volume [10]. For a perspective projection, the near plane is assumed to be perpendicular to the eye and so has the equation $z = -N$, where N is the distance from the eye to the near plane. Thus, all points in the model are projected by intersecting the line from the eye to the point with this near plane. Such a line would have parametric equations:

$$x = P_x t, y = P_y t, z = P_z t$$

Thus, this intersection would result in a projected point (x', y') with pseudodepth z' as follows:

$$x' = N\frac{P_x}{-P_z}, \quad y' = N\frac{P_y}{-P_z}, \quad z' = \frac{aP_z + b}{-P_z}$$

The corresponding transformation matrix is:

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

When viewing a display from off-axis, the assumption that the viewer is directly in front of the display is invalid. The degree to which the viewer is off the centre axis is particularly high when viewing a large display from one side. We introduce a method for rendering 3D objects that, instead of using a perpendicular near plane, uses an arbitrary near plane with the equation:

$$Ax + By + Cz = D$$

In order to preserve the property that the z-axis intersects the plane at a distance of N from the eye, we set $C = -1$ and $D = N$. Thus, A and B represent the slope of the plane in the x and y directions, respectively. Points in the model are again projected by intersecting the line from the eye to the point with this arbitrary near plane. This intersection results in the projected point (x', y', z'):

$$x' = N\frac{P_x}{AP_x + BP_y - P_z}, \quad y' = N\frac{P_y}{AP_x + BP_y - P_z}$$

$$z' = \frac{aP_z + b}{AP_x + BP_y - P_z}$$

The transformation matrix is thus only slightly modified:

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & a & b \\ A & B & -1 & 0 \end{bmatrix}$$

With this simple modification to the projection matrix, 3D objects can be rendered to compensate for off-axis viewing. This method introduces no added complexity and does not interfere with the response time of interaction.

3.2. Correcting for Multiple Users

The above method allows 3D objects to be rendered correctly for a single off-axis viewpoint, but tabletop displays lend themselves to many people gathering around them, each with their own viewing angle. Thus, a single viewpoint rendering may not be sufficient. Objects can each be rendered with a different perspective transformation, depending on the position of the object. By altering the perspective matrix in proportion to the object's position, this method essentially provides an arbitrarily-shaped near surface. We present alternative methods of projecting 3D objects for multiple users at a table.

3.2.1. Partitioning Viewpoints. One method of altering the perspective for many users is to provide several dedicated areas that each optimize the viewing angle for a particular portion of the display. For example, an obvious partition for a rectangular table is to divide the table in four parts and optimize the view for the closest side to each part. Essentially, each partition provides a different "window" through which to look at the underlying 3D model. Thus, the eye position of each partition can be chosen in three different ways.

The partitions can be aligned so that the near planes of each provide the boundaries (see Figure 3 and Figure 6c). This method results in view volumes that intersect one another. When an object is within the bounds of an intersection, the objects can either be displayed at all viewpoints, or some decision must be made as to which viewpoint to use.

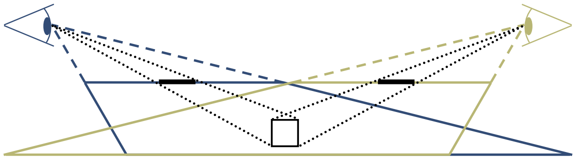


Figure 3. Partitions aligned with touching near planes result in intersecting view volumes.

Alternatively, the partitions can be aligned so that the separate view volumes do not intersect (see Figure 4 and Figure 6d). This model has the advantage that objects cannot be within two views at the same time. However, when crossing the boundary of two views, the change in projected position can be both large and discontinuous, which may make interaction confusing. Also, this method creates a volume in the model between the partitions where an object can exist without being visible.

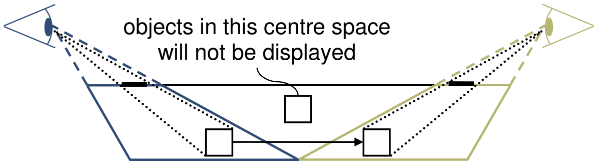


Figure 4. Partitions aligned with touching far planes result in disjoint view volumes. Objects moving between between view volumes may appear to jump.

A third method of providing correct perspective for multiple partitions is to keep the eye position above the centre of the table, but to slope the near plane differently in each partition (see Figure 5 and Figure 6e). Essentially this provides a view in each partition that is set for the correct viewing angle, while avoiding intersection of view volumes and discontinuities when objects move across boundaries. This method still compensates for off-axis distortion, but provides smoother interaction. However, because the near plane is distorted, applications requiring realism may prefer a partition with a "correct" eye position for each side.

A side effect of not moving the eye position as the slope changes is that objects become further from the near plane as they are moved closer to the table's edge. This effect can be corrected by adjusting the near plane distance so that the centre of the object is always projected to the original near plane. This new near plane distance, N', can be calculated by substituting the centre's projected point in standard perspective $(x', y', -N)$ into the equation for the plane:

$$N' = Ax' + By' + N$$

All three partitioning methods can be achieved by setting either A or B to the desired slope of the near plane.

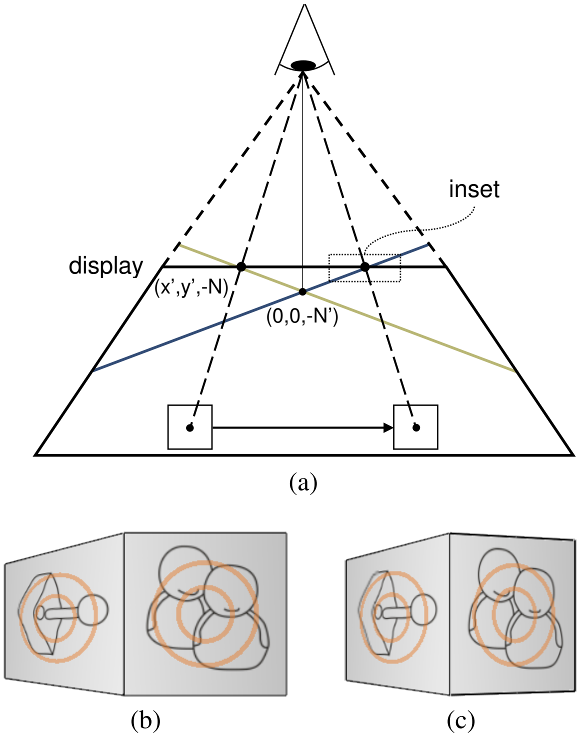
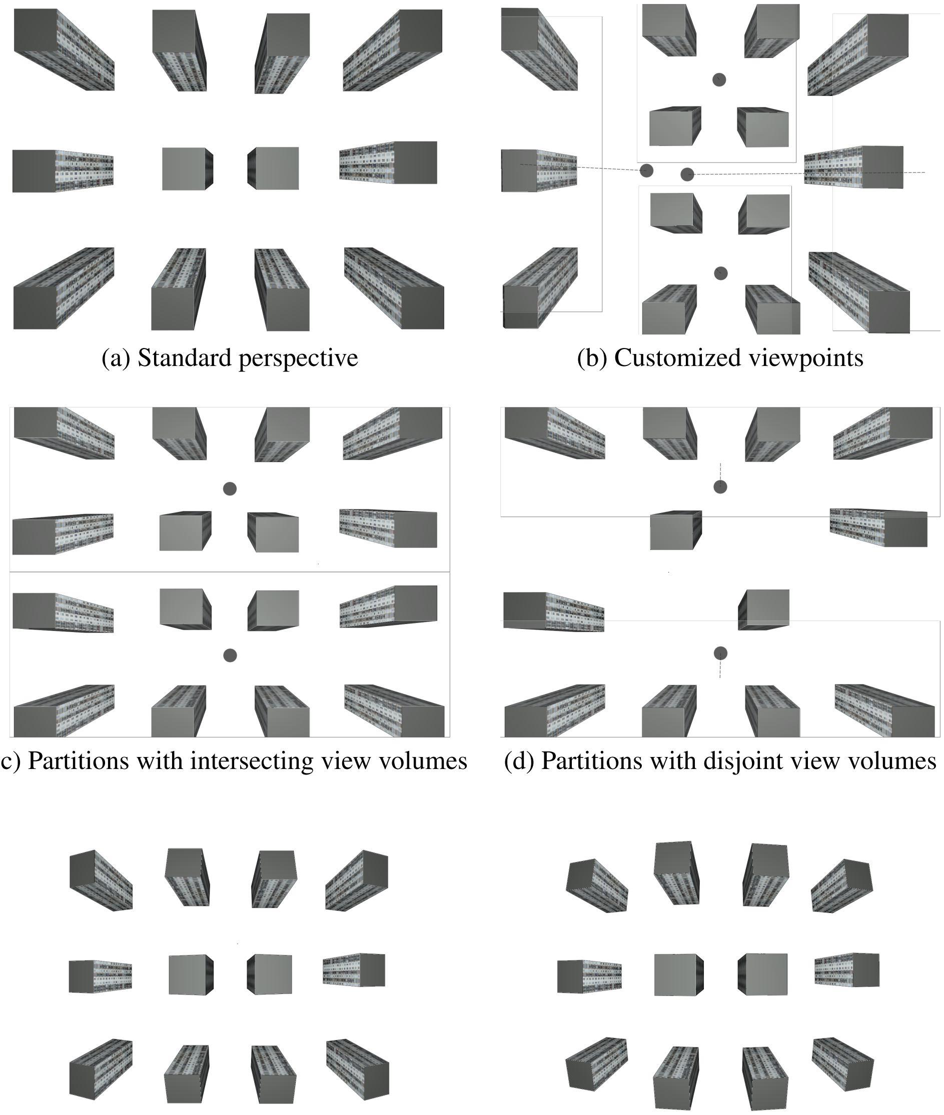


Figure 5. (a) The slope of the near plane can be adjusted to be appropriate for the adjacent table edge. Objects moving across the partition boundary do not jump and are not replicated. Insets show how the object would appear using (b) standard perspective and (c) when projected to the sloped plane.

3.2.2. Continuous Viewpoints. Instead of dividing the table into discrete parts, objects can be projected so that the viewing direction changes continuously. That is, objects can be projected so that the viewpoint is determined by the rotation (θ) about the z-axis. This method prevents objects from having to cross partition boundaries and suddenly switch viewpoints; instead, the transition is smooth across the entire display. The transition can also be made smooth at the centre of the display by adjusting the plane's slope (m) according to the distance (r) of the object from the centre of the display. This method essentially provides a hemispherical near plane (see Figure 7 and Figure 6f) and can be achieved by setting $A = mr \cdot \cos\theta$ and $B = mr \cdot \sin\theta$.

3.2.3. Customizable Viewpoints. We provide customizable views to allow users to manipulate the slope of the near plane and resize the area of influence of this near plane (see Figure 8 and Figure 6b). This slope can be controlled with a virtual handle that can be adjusted to provide the appropriate view. Again, it is possible to implement customizable views so that the eye position moves as the slope changes. However, it can be advantageous to not move the eye loca-



(a) Standard perspective

(b) Customized viewpoints

(c) Partitions with intersecting view volumes

(d) Partitions with disjoint view volumes

(e) Partitions with sloped planes only

(f) Continuous (spherical) viewpoint

Figure 6. (a) shows a series of 9 buildings using a standard perspective projection, which works well for someone standing directly above the table, but no one else. (b) shows four areas, each with its own customized viewpoint. The top and bottom areas are best viewed from the top and bottom respectively. The left area is best viewed from near the centre of the display and the right area is best viewed from the left side. (c) shows a partitioning of these same building into two parts. The top partition is best viewed from the top side, and the bottom is best viewed from the bottom. Note that in this correction, the view volumes intersect and so the middle buildings are replicated in both views. (d) shows two partitions with the same optimal viewing angles, but with non-intersecting view volumes. (e) shows a partitioning into four parts (left is best viewed from left, top from top, etc.), but only the planes are sloped; the eye position is kept at the centre and above the display. (f) shows a continuously changing viewpoint as objects move across the display. Objects are projected so they are best viewed along the axis from the centre of the display to the object.

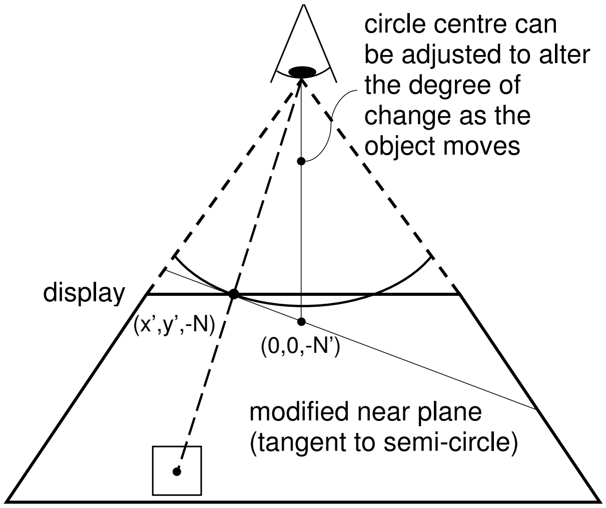

circle centre can be adjusted to alter the degree of change as the object moves
display
(x',y',-N)
(0,0,-N')
modified near plane (tangent to semi-circle)

Figure 7. In continuous spherical perspective projection, as objects are moved toward the edges of the display, the near plane they are projected to increases in slope. As they move toward the centre, the near plane becomes a standard perspective. Objects will appear more correct the closer they are to the viewer's edge.

tion so that the objects do not change their projected positions as the handle is dragged.

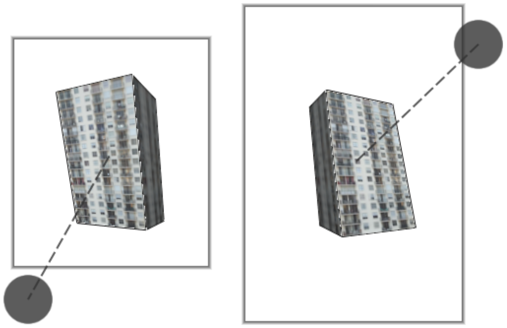


Figure 8. The left area is a custom view corrected for a viewpoint at the bottom-left and the right custom area is for a viewpoint at the top-right.

3.3. Effect on Collaboration

Choices made between off-axis projections influence the environment and how people coordinate and communicate within that environment. We discuss some of the implications and limitations of the corrections we suggest for multiple viewers.

3.3.1. Viewpoint Relativity. In all of the suggested multiple-viewer corrections, at any point in time, each object appears "correct" for only one viewpoint around the table. In order to make it appear correct for another, a person must either move the object along the surface or move a handle to adjust that viewpoint. Thus, our solution solves the multiple-viewer problem in a very different way than providing different views for each person (as in [2, 11]). The effect of our approach is that an object in "someone else's" space would still appear distorted (in fact, more so than in a standard perspective projection). This increased distortion may be a disadvantage, but on the other hand, may actually provide an added hint to a collaborator not to expect the object to appear correct. For example, a person watching the actions of another across the table would see a highly distorted image, and therefore not expect to see (say) the opposite side of a virtual cube, as they would with a real object.

3.3.2. Sensitivity to Frames. If a 3D scene is projected inside a framed box, people tend to compensate for off-axis inconsistencies. This effect may be due to the tendency to achieve shape constancy under object rotations [16]. Since the tabletop display is itself a frame, this could lead to an expectation different from the correction that we provide. Because the frame is a fixed physical entity, viewing the frame from different viewpoints (i.e. by different people) cannot be corrected for. However, because of the size of the tabletop display and the proximity of the people to the display, the frame may be too far separated from the objects being rendered to have any effect.

Framing can also be an issue in our customizable viewpoint correction, since we use a rectangular box (parallel to the edges of the display) to indicate the affected areas. It may be preferrable to use a frame that matches the correction to provide a better visual cue to the people at the display. Matching the frame to the correction may also improve a person's ability to parse objects in "someone else's" area, despite the increased distortion.

3.3.3. Fixed vs. Customizable Corrections. Both partitioned and continuous views provide a fixed correction for the entire display. These fixed corrections can be chosen based on an expected scenario of use. With these solutions, users will not have to learn additional controls and may be mostly unaware that the distortion has been corrected. Thus, this correction may become invisible to the user.

It can also be beneficial to allow people to control what areas of the screen are best viewed from what side. By providing this freedom, natural communication gestures may become available. For example, with this solution, it is possible for people to set up personal areas within which artifacts look correct as they work independently, and then can share their work by adjusting the viewpoint to be correct for another person. In general, providing this ability supports many of the mechanics of collaboration [15], including gestural messages ("this is what mine looks like"), visual evidence ("this is how they were looking at it"), and obtaining and reserving resources ("I'll look at this area from my viewpoint"). However, providing this added freedom can also add cognitive load, in that portions of the display can be set for different viewpoints, and users are assigned the added task of adjusting to the correct viewpoint.

3.4. Other Uses of Perspective Correction

Many wall-sized displays allow for interaction at close distances [4, 8]. Our technique could be used to adjust a 3D scene on such a wall to compensate for this proximity for either a single viewpoint or for multiple viewpoints as with the tabletop display. This setup would differ from the table in that the correction would likely only be necessary along one axis (horizontal adjustments, but not vertical). The view could also be coupled with such technologies as motion detection or gaze awareness to make it possible for a person's perspective view to stay correct as they explored 3D models.

It is also possible to use our technique to display a 3D scene on multiple displays configured at different angles. For example, several screens could be placed facing one another to create a long column with an adjacent column at the base (Figure 9). Two people could stand at each end of the display, and one of our multiple-user corrections could be used. Other display combinations are also possible, such as a tabletop display with an adjacent wall display or a cube with projections on each side.

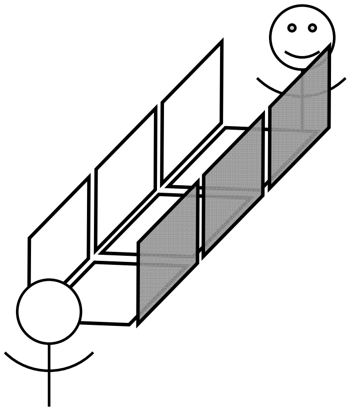


Figure 9. Our off-axis distortion correction can be applied to 3D scenes viewed across many displays, each at a different angle with respect to the viewers.

4. Conclusion

Our simplified model for correcting perspective projections for varying viewpoint positions provides support for proximal interaction with 3D interfaces on tabletop displays, and allows us to extend these ideas into a collaborative setting. Providing corrected views is a step toward making interactive 3D more viable and generalizes to many display configurations, including walls, tables and multiple configurable displays. We provide both continuous and discrete options for fixed corrections that allow interactions with perspective views automatically corrected to the closest side of the display, and customizable options that provide collaborators with many possibilities for adjusting viewpoints and the affected display regions, supporting both shared and personal views.

We would like to encourage further exploration into physical advantages of 3D interaction and enable empirical and observational studies that will further our understanding of proximal use of 3D on large displays.

5. Acknowledgments

We would like to thank Natural Science and Engineering Research Council of Canada, Alberta's Informatics Circle of Research Excellence, Alberta Ingenuity, the Canadian Foundation of Innovation, and SMART Technologies for research support. We also thank members of the iLab for their valuable input.

References

[1] A. Agarawala and R. Balakrishnan. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proc. CHI*. ACM Press, 2006.

[2] M. Agrawala, A. C. Beers, I. McDowall, B. Fröhlich, M. Bolas, and P. Hanrahan. The two-user responsive workbench: Support for collaboration through individual views of a shared space. In *Proc. SIGGRAPH*, pages 327–332, New York, NY, USA, 1997. ACM Press.

[3] M. Agrawala, D. Zorin, and T. Munzner. Artistic multiprojection rendering. In *Proc. Eurographics Workshop on Rendering Techniques*, pages 125–136, London, UK, 2000. Springer-Verlag.

[4] A. Bezerianos and R. Balakrishnan. View and space management on large displays. *IEEE CG & A*, 25(4):34–43, 2005.

[5] P. Coleman and K. Singh. Ryan: Rendering your animation nonlinearly projected. In *Proc. NPAR*, pages 129–156, New York, NY, USA, 2004. ACM Press.

[6] M. Deering. High resolution virtual reality. In *Proc. SIGGRAPH*, pages 195–202, New York, NY, USA, 1992. ACM Press.

[7] J. O. Dorsey, F. X. Sillion, and D. P. Greenberg. Design and simulation of opera lighting and projection effects. In *Proc. SIGGRAPH*, pages 41–50, New York, NY, USA, 1991. ACM Press.

[8] F. Guimbretière, M. Stone, and T. Winograd. Fluid interaction with high-resolution wall-size displays. In *Proc. UIST*, pages 21–30, New York, NY, USA, 2001. ACM Press.

[9] M. Hancock, S. Carpendale, and A. Cockburn. Shallow-depth 3D interaction: Design and evaluation of one-, two- and three-touch techniques. In *Proc. CHI*, pages 1147–1156, New York, NY, USA, 2007. ACM Press.

[10] F. S. Hill, jr. *Computer Graphics Using Open GL*. Prentice Hall, 2nd edition, 2001.

[11] Y. Kitamura, T. Konishi, S. Yamamoto, and F. Kishino. Interactive stereoscopic display for three or more users. In *Proc. SIGGRAPH*, pages 231–240, New York, NY, USA, 2001. ACM Press.

[12] J. C. Lee, S. E. Hudson, J. W. Summet, and P. H. Dietz. Moveable interactive projected displays using projector based tracking. In *Proc. UIST*, pages 63–72. ACM Press, 2005.

[13] M. McKenna. Interactive viewpoint control and three-dimensional operations. In *Proc. SI3D*, pages 53–56, New York, NY, USA, 1992. ACM Press.

[14] K. Nakashima, T. Machida, K. Kiyokawa, and H. Takemura. A 2D-3D integrated environment for cooperative work. In *Proc. VRST*, pages 16–22, New York, NY, USA, 2005. ACM Press.

[15] D. Pinelle, C. Gutwin, and S. Greenberg. Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. *ToCHI*, 10(4):281–311, 2003.

[16] Z. Pizlo. A theory of shape constancy based on perspective invariants. *Vision Research*, 34(12):1637–1658, June 1994.

[17] L. Terrenghi, D. Kirk, A. Sellen, and S. Izadi. Affordances for manipulation of physical versus digital media on interactive surfaces. In *Proc. CHI*, pages 1157–1166, New York, NY, USA, 2007. ACM Press.

[18] C. Ware, K. Arthur, and K. S. Booth. Fish tank virtual reality. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 37–42, New York, NY, USA, 1993. ACM Press.

[19] D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Perception of elementary graphical elements in tabletop and multi-surface environments. In *Proc. CHI*. ACM Press, 2007. To Appear.

[20] A. D. Wilson. PlayAnywhere: a compact interactive tabletop projection-vision system. In *Proc. UIST*, pages 83–92. ACM Press, 2005.

[21] D. Zorin and A. H. Barr. Correction of geometric perceptual distortions in pictures. In *Proc. SIGGRAPH*, pages 257–264, New York, NY, USA, 1995. ACM Press.