Multiscale 3D Reference Visualization

Michael Glueck, Keenan Crane, Sean Anderson, Andres Rutnik, Azam Khan

Autodesk Research
210 King Street East, Toronto, Ontario, Canada
{firstname.lastname}@autodesk.com

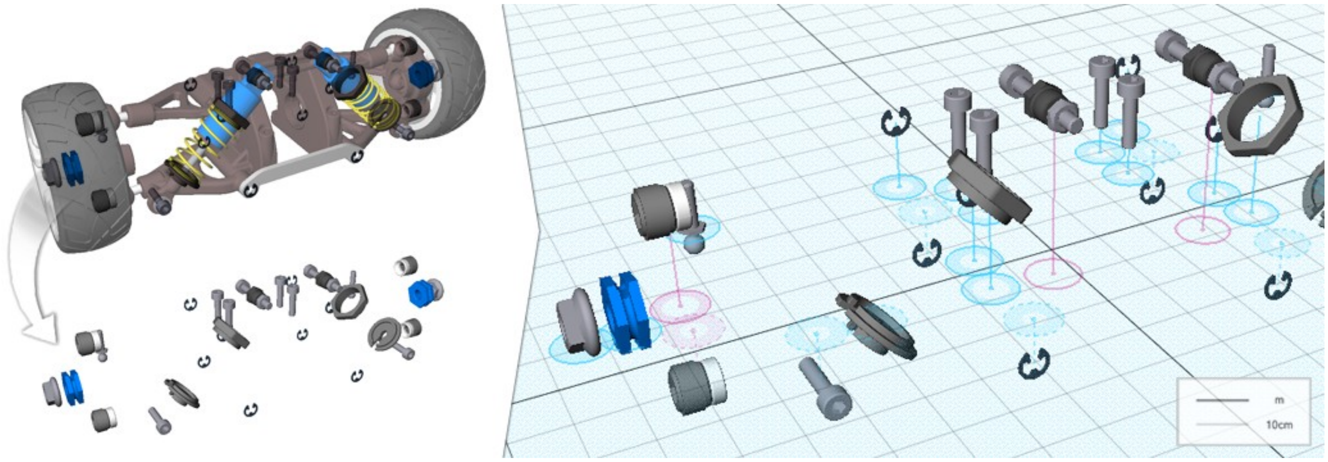

Figure 1: When sets of geometry are viewed outside a complete context, it can become difficult to determine the spatial relationships between objects. The multiscale reference grid, augmented with position pegs, provides additional depth cues to users, informing them of (i) whether an object lies above or below the grid, (ii) the distance of each object from the grid, (iii) the depth of the objects along the grid, (iv) the relative size of the objects, and (v) the approximate scale of the objects being viewed.

Abstract

Reference grids are commonly used in design software to help users judge distances and understand the orientation of the virtual workspace. Despite their ubiquity in 3D graphics applications, little research has gone into important design considerations of the 3D reference grids themselves, which directly impact their usefulness. We have developed two new techniques; the multiscale reference grid and position pegs that form a consistent foundation for presenting relative scale and position information to the user. Our design of a multiscale reference grid consistently subdivides and coalesces gridlines, based on the computation of a closeness metric, while ensuring that there are neither too many nor too few subdivisions. Position pegs extend the grid so that objects that are lying above or below the ground plane can be brought into a common environmental frame of reference without interfering with the grid or object data. We provide a stable analytic viewpoint-determined result, solving several depth cue problems, that is independent of viewing projection.

Categories and Subject Descriptors: H.5.2 [User Interfaces]: Graphical User Interfaces (GUI), 3D graphics

Additional Keywords and Phrases: 3D navigation, 3D widgets, Desktop 3D environments, virtual camera.

I3D 2009, Boston, Massachusetts, February 27–March 1, 2009.
© 2009 ACM 978-1-60558-429-4/09/0002 $5.00

1 Introduction

While advances in computing have empowered users to design and interact with objects in virtual three-dimensional space, fundamentally this experience has never left two-dimensions. Many of the perceptual cues we rely on to understand relative position and distance are not always easily conveyed in static 2D projections of 3D space.

Video games put significant effort into realistic texturing, shading, and shadowing to help with depth perception but these techniques are often not possible in authoring applications while the user is creating a new 3D model. Authoring applications must use other means to communicate depth cues that are effective on abstract and/or incomplete 3D scenes.

When confronted with an ambiguous 3D scene, the onus often falls on the user to clarify the spatial relationships presented (Figure 1). Constructing a mental model of the scene by viewing it from different viewpoints and garnering depth cues by changing the camera position is a common strategy employed by users. Unfortunately, this workflow forces the user to primarily work from memory to guide future decisions and actions. It also requires a proficiency in mentally transforming and manipulating 3D objects [Tory et al. 2006]. This can be a challenging task, particularly for users who are new to 3D [Fitzmaurice et al. 2008].

One approach often used to help users see the spatial relationships between objects in a scene, is to display a reference grid on the ground plane centered at the origin. The grid typically shows thicker major lines with thinner minor lines between them. In a parallel viewing projection, the lines are usually drawn to fill the viewport while, in a perspective projection, the grid usually has a fixed number of lines. This allows users to quickly sense the orientation of the workspace and to see how many grid blocks there are between objects to have a sense of the distance between them. However, while this typical implementation can be found in almost any 3D graphics application, it still exhibits a number of important problems that limit its utility. For example, in a parallel projection, depth cues are explicitly removed to provide consistency in scale. This makes it very difficult to judge whether the viewpoint is above or below the grid and also how the objects in the scene relate to each other. In a perspective projection, objects in the scene often fall outside of the fixed-sized grid or completely encompass it. Small objects can fall completely between gridlines which removes any relative depth cues.

The challenge is to show reference imagery to the user that is visible and meaningful in as many viewing conditions as possible without being too distracting.

However, reference visualization is significantly more difficult than it may seem. It effectively spans a huge problem domain of all forms of 3D interactions, touching on many fundamental difficulties: being inside an object vs. being outside, how close is the viewpoint to the object, what is the user looking at and/or is interested in, egocentric vs. exocentric thinking, parallel vs. perspective viewing projections, multiscale and level-of-detail issues, what kind of data is being examined (abstract, incomplete, engineering, CAD, entertainment, medical, simulation, etc), and what is the user task (authoring, inspecting, etc.). Additional technical issues include: handling clipping planes and floating-point precision problems. Due to the multitude of conditions and problems that can arise, we focus on the particular area of exterior views of multiscale 3D scenes in both parallel and perspective projections in an authoring application setting.

While developing this work, we came to realize that we were creating an essentially infinite level-of-detail dataset and were always cognizant of the performance requirements of the system. Independent of the compute power available, it was fairly common to consume all system resources. This lead us to consider a combination of an image-based (MIP map) technique with an analytical algorithm to achieve a highly interactive (low overhead) solution.

To provide a consistent reference visualization that reinforces and strengthens the depth cues already present in the data of the scene, we enhance the 3D scene with two new techniques: *position pegs* and a *multiscale reference grid*. We present the user with these ambient elements, providing disambiguating feedback, that is always visible –even in a static display of the scene– to guide decisions and actions, while reducing the dependence on developing and working from a mental model of the scene.

2 Characteristics

Despite the prevalence of grids in commercial software, there is seemingly no literature investigating which characteristics make a grid most effective. Here we examine desirable grid properties and outline the goals of our system.

2.1 Depth Cues

Interacting with objects in our immediate environment requires us to constantly make complex egocentric and exocentric judgments about spatial relationships. Cutting and Vishton [1995] analyzed the various depth cues which have been identified in literature, ranking them based on their effectiveness in three different ranges of distance from the observer (Figure 2). Unlike the real world, the real-time 3D environments we view can range between all three distances. Not only can a user move closer or farther from an object, but it is possible that both near and distant objects are viewed in focus simultaneously. This unrealistically perfect nature of virtual 3D is what has made it such a powerful tool in industries where precision is paramount. Cutting and Vishton also noted that realistic pictures attempt to trick observers into perceiving depth within a flat surface. In response, they developed a separate ranking of depth cues specifically for pictorial sources. Since static 3D images are not so different from realistic pictures, we analyze the top three depth cues identified:

1) *Occlusion*: Also known as interposition, it occurs when one object blocks part of another object, providing a discrete measure of layering. Zhai et al. [1996] found that semi-transparent occlusion is as effective as opaque occlusion. To be useful in either case, objects are required to lie in front of one another from a given viewpoint.
2) *Relative size*: When viewing similar objects, the smaller ones are understood to be further away. This is based on the assumption of size constancy: similar objects are of the same size. In a virtual world, however, a common reference structure is needed, especially in a parallel projection.
3) *Height in visual field*: In relation to a ground plane, objects higher in the visual field appear to be farther away. The assumption in this case is that the objects are touching the ground plane, or resting on other objects that are touching the ground plane.

Relative size and relative height in the visual field both relate to assumptions we make based on our experiences in the real world. Occlusion is often used implicitly in both of these measures. In virtual worlds, however, these assumptions are not guaranteed to hold. Our strategy is to eliminate the confounding assumptions by making these three depth cues explicit, thereby maximizing the disambiguating depth cues presented to the user in a 3D scene.

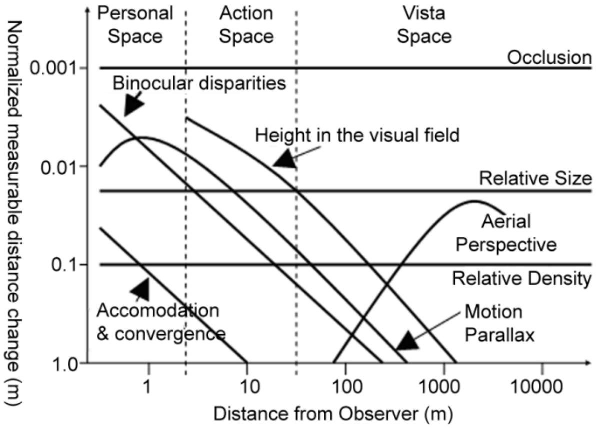

Figure 2: Relative strength of depth cues (adapted from Cutting et al. [1995])

2.2 Goals

Within the scope of our investigation, we propose a reference visualization solution that provides users with salient information regarding the spatial organization and relationships between objects in a 3D scene. To summarize our goals, we sought to meet the following criteria:

1) Provide reference feedback in a global context and represent real-world units;

2) Support both perspective and parallel projections;
3) Display grid on a fixed plane with consistent and reasonable subdivisions in screen space;
4) Determine grid spacing solely on camera position (navigation operation independent)
5) Update grid division spacing interactively and smoothly;
6) Provide salient feedback regarding the height of objects from the grid and relative position of objects on the grid;
7) Present feedback always and everywhere.

3 Reference Visualization

3.1 Multiscale Reference Grid

Hagen [1991] stressed the importance of *ecological validity* in order to achieve realism of composition in 3D images and guidelines were proposed. While realism is not our goal, we attempted to adhere to the guidelines applicable to dynamic 3D environments. In particular, that there should be only one horizon and that a horizontal ground plane is always present. Wanger et al. [1992] corroborate the importance of providing users with an environmental frame of reference, such as a grid. We first describe the design, then we outline the multiscale aspects.

3.1.1 Appearance

The grid lies on the conceptual ground plane of the scene, through the origin. A legend is displayed in the lower right hand corner indicating the scale represented (Figure 3). The scale of major and minor lines is determined by a robust algorithm (Section 3.1.2).

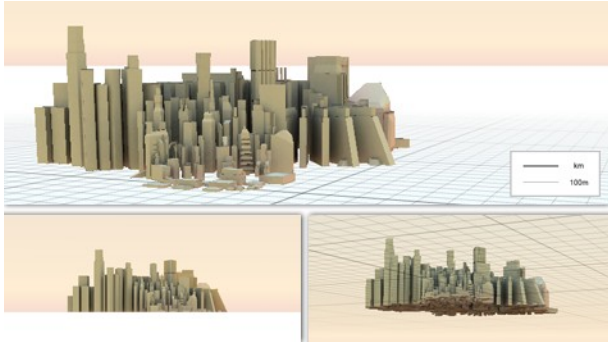

Figure 3: (top) Reference grid from above, with horizon and scale legend, (bottom-left) side view, and (bottom-right) from below.

Since the grid appears to vanish into the white “ground” before it has reached the horizon, an explicit horizon line is defined by rendering the upper region as a “sky”. This helps prevent the perception that a distant object, beyond the visible extents of the grid, is floating. As the camera passes under the grid, sky is visible behind the grid (Figure 3). By rendering both the “ground” and the “sky”, it becomes clear whether the camera is completely above or below the grid.

The grid has a semi-transparent quality [Zhai et al. 1996] to it, using a stippled pattern to leverage the partial-occlusion depth cue in two ways. First, it helps to determine where small objects, which do not pass through grid lines, intersect the grid. Second, this feedback discretely supports judging whether an object lies above or below the grid, regardless of their color.

The grid lines fade out the further they are from the camera. This fading effect makes use of an additional depth cue: *aerial perspective*. It also reduces anti-aliasing artifacts when grid lines become too dense in the distance in perspective projection. Additionally, this effect provides valuable depth information in parallel projection, allowing users to discern in which direction the grid recedes from the camera (Figure 4).

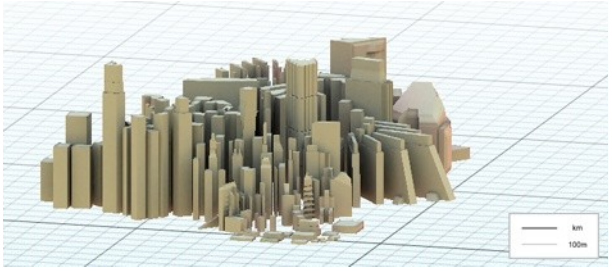

Figure 4: The grid fading in the distance provides additional depth cues in parallel projection.

As a multiscale object, an individual gridline will transition from being a minor line to a major line, or vice versa, as the viewpoint moves closer to or further from the grid (during zoom-in or out operations, for example), or closer to the grid. To minimize the visual disruptions that changes in the grid scale may cause, we smoothly interpolate the opacity of the grid lines using an inverse sigmoid function. The function ensures differentiability between major and minor lines, while providing a quick fade in, a plateau at semi-transparency, and then a quick transition to the darkest, major lines. The function is defined by the equation:

$$\alpha = -k\left(\ln\left(\frac{1}{(y+m)} - n\right) + c\right) \qquad (1)$$

where α is the opacity value ranging in [0,1]. The variable y , also in the range $[0,1]$, represents how far a minor grid line is along the transition from invisible, at 0.0, to becoming a major grid line, at 1.0 (based on the algorithm described in Section 3.1.2). In our implementation, we found that a visually pleasing effect was achieved with variable values of $k = -0.125,\ c = 1.5,\ m = 0.01,\ n = 0.995,$.

Finally, when the camera lies close to the grid and approaches a view direction parallel to the grid, the opacity of the grid is reduced. When the camera becomes coplanar with the grid, a single line is drawn to represent the entire grid. This provides the user with a smooth visual transition as the camera passes through the grid (Figure 3).

3.1.2 Grid Scale

To ensure we provide constant feedback, as outlined in our goals, the spacing between grid lines must be dynamically updated. If the spacing was constant, we could zoom between two gridlines and we would no longer see the grid.

Ideally, grid spacing should remain roughly constant as the user navigates the space, while still giving an accurate sense of scale and location. That is, overall, while zooming out (for example), the number of grid lines should not continuously increase. There will be a number of gridlines drawn (or “gridline density”) that could be considered to be legible at any given time.

Importantly, we seek a scheme that is invariant with respect to the camera trajectory. In other words, a given camera configuration, including the position, orientation, and distance from the grid, should always yield the same grid spacing, regardless of the path taken to reach that configuration. This requirement would forbid many schemes which, for example, update grid spacing during a translation but not a rotation.

This may seem like a simple goal to achieve. However, as stated in the Introduction, this problem touches on many of the fundamental difficulties in 3D interaction. For example, it may be clear how far the viewpoint is from the grid when the camera is looking directly at the grid, but how close is the viewpoint to the grid when looking parallel to the ground plane? If the true distance is used, then there will be too many gridlines at some point, or not enough at others. We attempted several solutions, but it was difficult to find one which performed well under all scenarios. For example, some worked well during panning, but failed during orbiting, and others worked well at high angles of incidence, but poorly at low angles. To develop a consistent solution, we chose a screen-space based approach. We introduce the use of a grid spacing scheme determined by the minimum screen-space derivatives of the plane coordinates (discussed below). This approach is inspired by *MIP map* texture filtering [Williams 1983], which selects a texture's level of detail according to the screen space derivative of texture coordinates. This approach is clearly path-invariant since it depends exclusively on the current camera configuration. We have also found that this scheme avoids many of the corner cases and singularities of more ad-hoc methods for determining grid spacing.

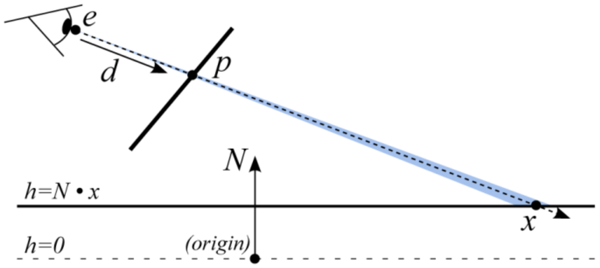

Figure 5: Diagram of grid scaling variables used in the multiscale algorithm.

In particular, we compute a scalar quantity, *α,* that characterizes the rate of change of the grid plane's local coordinates relative to an infinitesimal change in window coordinates at a given screen-space location w_x, w_y. This quantity is linear in the sense that if α becomes twice as large we will need twice as many grid lines in each direction to retain constant spacing in screen space. In practice, we pick our scale by evaluating α at the closest visible point on the plane (i.e., where the plane's local coordinates are changing fastest). This value is then used to select our grid spacing from a pre-defined range of scales that might, for instance, correspond to metric units of length (millimeters, centimeters, etc.).

Using normalized window coordinates in $[-1,1]^2$, the map $\alpha : [-1,1]^2 \rightarrow \mathbb{R}^+$ is constructed as follows. Let the grid plane be defined by $N \cdot x = h$ where $N \in \mathbb{R}^3$ is the normal and $h \in \mathrm{R}$ is the normal offset. Consider any point p on the viewing plane through which the plane of the grid is visible, and suppose that the corresponding window coordinates are (w_x, w_y). We can intersect a ray from the observer through p to get the local coordinates x on the grid plane (Figure 5). Specifically, if e is the location of the observer and $d = p - e$ is the direction of the ray, then we have

$$x = e + \frac{h - N \cdot e}{N \cdot d} d \qquad (2)$$

To find points along the ray $e + td$ we can apply the inverse modelview-projection transformation to homogeneous clip coordinates associated with our window coordinates w_x and w_y (cf. Appendix F of [Nieder et al. 1993] and the `man` page for `gluUnProject`). Let $A = M^{-1}P^{-1}$ where M is the current homogeneous modelview transformation and P is the current perspective projection transformation. We can then use

$$\hat{e} = A[\, w_x, w_y, -1,1 \,]^T \qquad (3)$$

and

$$\hat{p} = A[\, w_x, w_y, 1,1 \,]^T \qquad (4)$$

to determine our ray origin and direction, where $\hat{}$ indicates coordinates prior to a homogeneous divide.

Finally, we define α as

$$\alpha = \max\left(\left\| \frac{\partial x}{\partial w_x} \right\|_2, \left\| \frac{\partial y}{\partial w_y} \right\|_2 \right) \qquad (5)$$

which we evaluate analytically by solving for $\hat{e}$ and $\hat{p}$, substituting the resulting expressions into the expression for x, and finally writing out the partial derivatives of x needed to define α explicitly in terms of window coordinates and the current transformation matrices. (For simplicity, we have also tried approximating these derivatives numerically, but found such an approximation to be too noisy at shallow glancing angles.) This expression approximates the maximum rate of change of the plane's local coordinates and is similar to functions used for MIP map level selection.

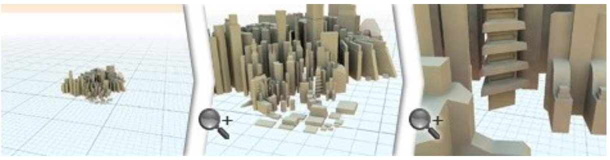

Figure 6: Reference grid while zooming multiple scales.

Since this algorithm depends exclusively on the window coordinates, the current transformation, and the plane geometry, it provides a consistent definition of grid spacing that is independent of the camera path or other external factors. This allows the reference grid to transition across multiple scales of geometric data seamlessly (Figure 6).

3.2 Position Pegs

Position pegs are an ambient feedback mechanism designed to complement the planar nature of the reference grid. When geometry in the scene does not intersect with the reference grid, it becomes difficult to understand spatial relationships between it and other objects in the scene. In particular, powerful depth cues, such as relative size and height in the visual field are rendered ineffective. Position pegs augment the reference grid by indicating the projected position of the object on the grid, as well as the object's height from the grid. By providing the user with these two cues, the height and depth of the object becomes unambiguous and judgments can quickly be made concerning the relative size and position of objects in the scene.

In keeping with principles of ecological validity, every object in the scene should rest on the reference grid directly or on another object that lies on the reference grid [Hager 1991]. Since it is not possible for every object in the scene to fulfill this property, position pegs act as abstract proxies connecting every object in the scene to the grid. Tory et al. [2006] support this finding. Based on the results of their user study, they hold that shadows, augmented with concrete height cues, would facilitate more accurate relative position estimates than shadows alone.

While shadows allow us to apply learned real-world relationships to virtual environments, these same experiences also dictate how we expect shadows to appear and behave. Tory et al. [2006] used projective shadows cast downwards onto other objects to provide height cues. However, when considering that objects can lie above or below the reference grid, real world assumptions typically do not apply. For example, looking down at an up cast shadow is uncommon in the real-world. Extending this example, shadows cast onto a grid simultaneously from above and below is improbable. An even stranger case is viewing an object through the grid, partially occluded by its own shadow.

Since position pegs operate properly without implicit light sources, they are effective when objects are either above or below the grid. Position pegs can also better communicate relative height in more situations than a conceptual light-source and shadow model. Finally, in contrast to shadows, position pegs will continue to provide clear and consistent feedback even if in-scene shadows, based on user-defined light sources, are enabled.

3.2.1 Appearance

Position pegs are designed to abstractly represent the shadows of scene geometry. By using an abstract representation, we are able to leverage the depth cues shadows provide and provide more concrete height cues, while avoiding many of the real world constraints inherent in shadows.

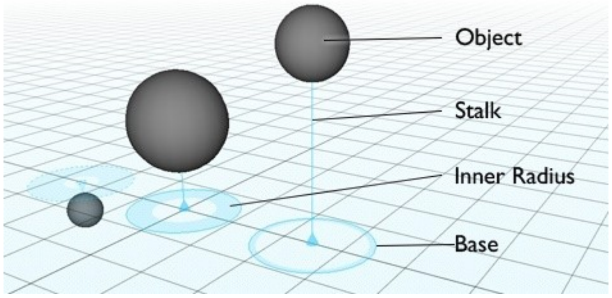

Figure 7: Composition of a position peg. The inner radius indicates the distance of the object from the grid.

Position pegs consist of two main parts: a base and a stalk. The base appears as a disk, co-planar with the grid surface. Each base represents the projection of an object orthogonally onto the grid. Comparisons among position peg bases provide information about the relative depth of objects along the grid. Secondly, the shaded inner region of the base coarsely indicates the relative height of objects to one another. The more that is shaded, the closer the object is to the reference grid when compared to the most distant object from the grid. Thus, as the salience of the stalks degrades due to foreshortening, the base fills in the missing information, and vice versa (Figure 11). The fixed size of all bases is constant in screen space, but can be resized by the user. The stalk provides finer grain height information about the object it represents. The length of the stalk instantly communicates distance from the reference grid. In addition, the stalk visually ties the object to its base (Figure 7).

Position pegs below the reference grid are rendered with higher transparency than those above the grid. In addition, position pegs below the grid are also rendered with dashed lines. A cone is rendered in the center of the base to indicate the direction in which the object lies (above or below the grid). These techniques try and best leverage the partial-occlusion depth cue [Zhai et al. 1996] to indicate which objects are above the grid, which are below, and where these objects intersect the grid, without fully occluding the grid itself.

3.2.2 Aggregation

Position pegs also operate seamlessly in a multiscale environment through the ability to aggregate with one another when objects in the scene begin to cluster too closely to one another, when a user zooms out for example. Position pegs aggregate when their bases begin to overlap excessively. Since position pegs are projected onto the grid, this is reduced to a two-dimensional clustering problem. Changing the size of position peg bases allows users to control when position pegs aggregate.

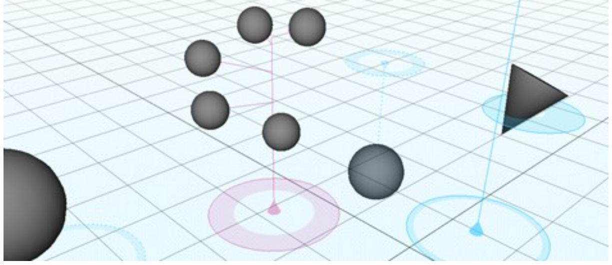

Figure 8: An aggregated position peg. Branches from the stalk indicate the individual height of objects.

Aggregated position pegs are displayed in a different color from normal position pegs. The base of an aggregated position peg represents the center of mass of all combined objects. The stalk of an aggregated position peg is the maximum height of all combined objects, with a branch coming off to meet the center of mass of each individual object. This allows the relative height of the individual objects to be inferred (Figure 8).

3.2.3 Disambiguation & Additional Cues

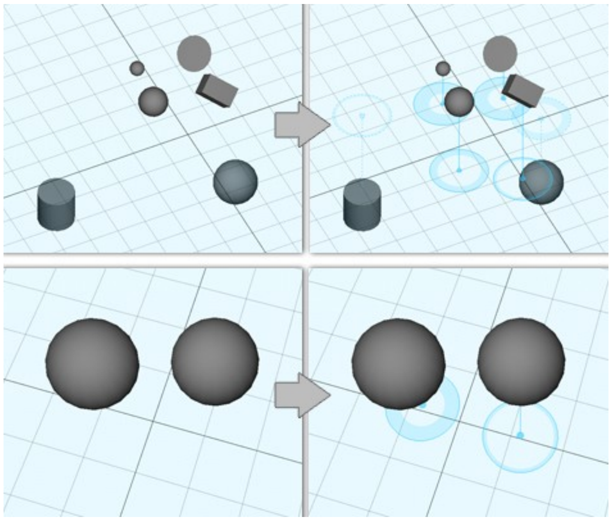

Figure 9: Two ambiguous scenarios: (top) parallel projection; (bottom) perspective projection.

Position pegs allow users to determine the position of objects in the scene without changing their viewpoint. Parallel projection, which inherently offers users fewer depth cues than perspective projection, stands to benefit as well (Figure 9).

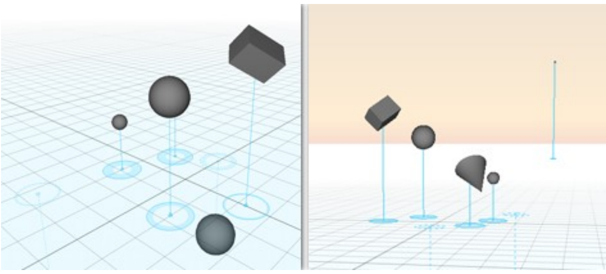

Figure 10: Position pegs reveal height information about occluded objects and those outside the viewport (left), and also provide context for very distant objects (right).

Position pegs present users with height information about nearby, but out-of-viewport objects, and those occluded by other objects. Distant objects also benefit from position pegs (Figure 10).

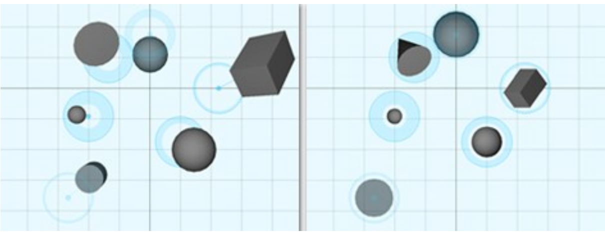

Figure 11: Position peg bases convey height information when stalks are occluded: (left) perspective projection, (right) parallel projection.

Position pegs continue to provide information about the relative height of objects in the scene even when the stalks are partially or fully occluded. Again, this is especially helpful when viewing the scene in parallel projection (Figure 11).

4 Related Work

4.1.1 Multiscale Reference Grid

As mentioned earlier, previous grid investigations are primarily from commercial applications. For example, Autodesk Maya provides a simple grid of fixed-size for global orientation. 3DVia Shape is a single-perspective-view based modeling program featuring a ground-plane grid that extends “infinitely”, fading off towards the horizon. Major and minor lines dynamically subdivide based on the camera position during pan and zoom operations, but non-interactively during orbit operations. Zooming is limited at both near and far extremes and orbiting is only possible around pivot points lying co-planar with the grid. Another example is found in Autodesk Inventor. Primarily operating in a parallel viewing projection, it contains a very large, finite grid while in modeling mode. As the camera approaches the grid, it subdivides dynamically. Inherent with parallel projection is the lack of a foreshortening effect, which makes it difficult to determine whether one is viewing the scene from above or below the grid, and the relationship between objects. Luxology Modo also implements a multiscale grid and the major and minor lines dynamically subdivide during panning and zooming operations, but not while orbiting. Due to this scheme, the spacing of subdivisions is not always ideal and is not invariant to the camera path. While the viewport updates to show the correct units represented by the grid lines, the inconsistency of spacing might confuse users since the reported scale is no longer accurate.

In contrast, we present an ambient reference grid that overcomes the shortcomings enumerated above. Real-world units are represented in a dynamically subdividing grid that provides depth cues in both parallel and perspective projection, while supporting unrestricted camera operations and unrestricted pivot points.

Infinitely zooming into a checkerboard texture was presented by Cunzi et al. [2003] as part of a technique, called Dynamic Canvas, creating an immersive 3D motion effect in non-photorealistic walkthroughs of 2D scenes. Like Dynamic Canvas, our implementation also draws inspiration from MIP mapping, but can be seen as a 3D geometric analogue.

4.1.2 Position Pegs

Researchers have investigated the use of shadows and reflections to provide positional information in 3D scenes. User studies have shown that shadows, when projected on a nearby background, are powerful cues for accurately determining the position of objects [Wanger et al. 1992; Hubona et al. 2000; Wickens et al. 1989]. This beneficial effect diminishes dramatically when shadows are not projected onto flat surfaces or multiple light sources are used [Murray 1994; Hubona et al. 2000]. Additional studies by Tory et al. [2006] indicated that shadows resulted in significantly faster user responses than the other techniques studied, while providing coarse accuracy.

Herndon, et al. [1992] investigated the use of shadows not just to inform users of positional information, but also to provide a means of object manipulation. In their system, shadows were cast on all walls and floors of a virtual “room”. Moving the shadow would perform a constrained translation of the object along that plane. In addition, they explored the value of providing reflections instead of simple silhouettes, to allow users to view hidden parts of the object. Similarly, Ritter et al. [2003] used the reflections of objects to inform the user of occluded views of an object to aid in quicker inspection.

The use of real-time shadows as a tool for providing additional depth information has also been adopted in the field of augmented reality [e.g., Haller et al. 2003; Naemura et al. 2002]. In their augmented reality system, Ayatsuka et al. [1996] varied the characteristics of a shadow to inform users of an object’s height from a surface. These artificial shadows were always projected directly below the object, and the ratio of the umbra to penumbra of the shadow informed the user of height. This technique was found to be more effective and rated more natural than projecting shadows onto the walls of a virtual room. Ayatsuka et al. [1998] presented a generalized and expanded version of this technique to apply to any 3D environment.

Wanger [1992] studied whether shadow precision had an impact on informing users about objects, finding evidence to suggest that simply the existence of a shadow, regardless of shape or sharpness was sufficient to provide height information about an object.

Our work differs from the previous work as we enhance the ambient grid with an abstract representation of shadows, allowing coarse grain height estimates to be inferred for any object in a 3D scene independent of the viewing projection.

Together these techniques eliminate many of the ambiguities present in dynamic and static virtual 3D scenes.

5 Initial Impressions

We carried out an informal evaluation of the multiscale reference grid and position peg technique. Six volunteers participated, ranging in their experience with 3D authoring applications. All participants mentioned that the multiscale reference grid reacted smoothly to their navigation and faded off in the distance. One participant really liked that the sky was visible from underneath the grid. Two participants commented that the grid appeared to end abruptly, and that it seems disjoint from the horizon in certain instances.

Position pegs were seen as useful by all participants. One likened the system to an overlaid orthographic top view in perspective projection. However, there was also agreement that certain features were initially confusing. In particular, the color distinction between single and aggregated pegs, and the mapping between the inner radius and height. One participant felt that a thicker inner radius should indicate a higher object. The expert 3D users desired to have more control over the behavior of the position pegs, in particular the tolerance at which aggregation occurred and the ability to toggle them on and off. In addition, they desired more precise feedback from the position pegs. One participant said he wished numbers indicating the exact height from the reference grid would be displayed next to each peg.

Overall, initial reaction to the ambient reference visualization was positive. Both novice 3D users commented that the combination provided a good reference point, and that they did not feel as easily lost. The more advanced users also mentioned that new users would likely find the system helpful and intuitive. While some participants did not feel the position pegs were completely necessary in perspective projection, all agreed that position pegs greatly helped in understanding parallel projection layouts.

6 Discussion and Future Work

While we have shown the promise of multiscale reference grid and position pegs to be highly effective at providing users with height information for objects in a scene, there are two cases where their behavior could be improved.

The first is when the camera is positioned co-planar to the reference grid and the view direction is parallel to the grid. In this case, no addition height information can be obtained from position pegs over and above those already provided by the geometry in the scene. Moreover, as it is no longer possible to distinguish the intersections of the position pegs and the multiscale reference grid, depth information is no longer available. An additional supplemental multiscale reference grid could be implemented, appearing on an orthogonal plane when the viewing angle approaches parallel to the primary multiscale reference grid. The drawback of this scheme is that it requires users to continually re-evaluate relationships with respect to alternating frames of reference. More suitable solutions to this problem could be investigated.

Second, the position peg and multiscale reference grid combination does not perform well when many objects in the scene are organized orthogonally in relation the grid. In this case, position pegs are much more likely to aggregate and it becomes difficult to easily distinguish which position peg belongs to which object. While allowing users to choose the primary plane in which the multiscale reference grid appears, it may not be robust enough to handle the multitude of datasets which can be represented in virtual 3D space. Future research might address optimal ambient reference configurations for different spatial layouts of data. Perhaps there are commonly occurring patterns for which specialized feedback schemes could be developed.

Finally, a more formal user evaluation would help refine which properties could be augmented further.

We believe the lack of a robust commercial implementation of a multiscale reference grid demonstrates the difficulty involved in developing a system that nicely handles a large set of criteria. While it may be impossible to provide some desired properties, we have met our initial goals with the design presented here. As we expand on the types and complexity of data sets supported by this system, we expect to be confronted by additional fundamental 3D interaction problems. Specifically, the camera model and the reference visualization will most likely become more closely tied together.

In working with some multiscale data sets, we also found the need for an additional type of feedback, beyond that provided by the position pegs, analogous to the Halo technique [Baudisch and Rosenholtz 2003] to indicate where in 3D-space additional data may exist.

While we had explored some grid labeling techniques, these too were challenging in a multiscale environment. We expect that the investigation of a more general “legibility” property for visually complex scenes will be a fruitful area of future research.

7 Conclusions

By understanding the depth cues needed, we were able to determine a set of effective reference visualization characteristics needed to understand relative size and position information from an abstract 3D scene in either parallel or perspective viewing projections.

We then designed the multiscale reference grid and position peg components to provide the needed cues and developed the algorithms to support the design. The multiscale reference grid is a consistent and ever-present frame of reference for users. Position pegs overcome the planar limitations of the grid, providing height and depth information for all objects in the scene. This scheme is particularly effective for parallel projections and abstract/incomplete 3D scenes, where objects are disjoint or lack a larger context.

To support the behavior of the multiscale reference grid, we developed an algorithm for determining scale in a multiscale 3D scene based only on the position of the camera. For scalability of the position pegs, we implemented an aggregation technique that reduces clutter while increasing a sense of the distribution of objects in the scene.

The multiscale reference grid, augmented with position pegs, provides additional depth cues to users, informing them of (i) whether an object lies above or below the grid, (ii) the distance of each object from the grid, (iii) the depth of the objects along the grid, (iv) the relative size of the objects, and (v) the approximate scale of the objects being viewed.

These contributions impart insights into important design considerations of 3D reference grids creating a more useful and more consistent foundation for presenting relative scale and position information to the user.

Acknowledgments

We would like to thank Joannes Chan, Thomas White, Ryan Schmidt, James McCrae, Justin Matejka, Tovi Grossman, George Fitzmaurice, and Gord Kurtenbach for their feedback and suggestions.

References

AYATSUKA, Y., MATSUOKA, S., AND REKIMOTO, J. 1996. Penumbrae for 3D interactions. In *Proceedings of the 9th Annual ACM Symposium on User interface Software and Technology* (Seattle, Washington, United States, Nov. 06 - 08, 1996). UIST '96. ACM, New York, NY, 165-166.

AYATSUKA, Y., MATSUOKA, S., AND REKIMOTO, J. 1998. Layered Penumbrae: An Effective 3D Feedback Technique. In *Proceedings of the Third Asian Pacific Computer and Human interaction* (Jul. 15 - 17, 1998). APCHI. IEEE Computer Society, Washington, DC, 202.

BAUDISCH, P. AND ROSENHOLTZ, R. 2003. Halo: a technique for visualizing off-screen objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, Apr. 05 - 10, 2003). CHI '03. ACM, New York, NY, 481-488.

CUNZI, M., THOLLOT, J., PARIS, S., DEBUNNE, G., GASCUEL, J.-D., AND DURAND, F. 2003. Dynamic canvas for immersive non-photorealistic walkthroughs. In *Proceedings from Graphics Interface.* GI '03, A K Peters, Ltd. 121-129.

CUTTING, J.E. AND VISHTON, P.M. 1995. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth, In *Handbook of perception and cognition, Vol 5; Perception of space and motion*, W. EPSTEIN & S. ROGERS, Eds. Academic Press, San Diego, CA, pp. 69-117.

FITZMAURICE, G., MATEJKA, J., MORDATCH, I., KHAN, A., AND KURTENBACH, G. 2008. Safe 3D navigation. In *Proceedings of the 2008 Symposium on interactive 3D Graphics and Games* (Redwood City, California, Feb. 15 - 17, 2008). SI3D '08. ACM, New York, NY, 7-15.

HAGEN, M. A. 1991. How to make a visually realistic 3D display. *SIGGRAPH Comput. Graph.* 25, 2 (Apr. 1991), 76-83.

HALLER, M., DRAB, S., AND HARTMANN, W. 2003. A real-time shadow approach for an augmented reality application using shadow volumes. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (Osaka, Japan, Oct 01 - 03, 2003). VRST '03. ACM, New York, NY, 56-65.

HERNDON, K. P., ZELEZNIK, R. C., ROBBINS, D. C., CONNER, D. B., SNIBBE, S. S., AND VAN DAM, A. 1992. Interactive shadows. In *Proceedings of the 5th Annual ACM Symposium on User interface Software and Technology* (Monteray, California, United States, Nov. 15 - 18, 1992). UIST '92. ACM, New York, NY, 1-6.

HUBONA, G. S., WHEELER, P. N., SHIRAH, G. W., AND BRANDT, M. 1999. The relative contributions of stereo, lighting, and background scenes in promoting 3D depth visualization. *ACM Trans. Comput.-Hum. Interact.* 6, 3 (Sep. 1999), 214-242.

MURRAY, J. 1994. Some perspectives on visual depth perception. *SIGGRAPH Comput. Graph.* 28, 2 (May. 1994), 155-157.

NAEMURA, T., NITTA, T., MIMURA, A., AND HARASHIMA, H. 2002. Virtual Shadows - Enhanced Interaction in Mixed Reality Environment. In Proceedings of the IEEE Virtual Reality Conference 2002 (Mar. 24 - 28, 2002). VR. IEEE Computer Society, Washington, DC, 293.

NEIDER, J. AND DAVIS, T. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, Release 1. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993.

RITTER, F., SONNET, H., HARTMANN, K., AND STROTHOTTE, T. 2003. Illustrative shadows: integrating 3D and 2D information displays. In *Proceedings of the 8th international Conference on intelligent User interfaces* (Miami, Florida, USA, Jan. 12 - 15, 2003). IUI '03. ACM, New York, NY, 166-173.

TORY, M., KIRKPATRICK, A. E., AND ATKINS, M. S. 2006. Visualization Task Performance with 2D, 3D, and Combination Displays. IEEE Transactions on Visualization and Computer Graphics 12, 1 (Jan. 2006), 2-13.

WANGER, L. 1992. The effect of shadow quality on the perception of spatial relationships in computer generated imagery. In *Proceedings of the 1992 Symposium on interactive 3D Graphics* (Cambridge, Massachusetts, United States). SI3D '92. ACM, New York, NY, 39-42.

WANGER, L. C., FERWERDA, J. A., AND GREENBERG, D. P. 1992. Perceiving Spatial Relationships in Computer-Generated Images. *IEEE Comput. Graph. Appl.* 12, 3 (May. 1992), 44-51, 54-58.

WICKENS, C.D., TODD, S., AND SEIDLER, K. 1989. *Three-dimensional displays: perception, implementation, and applications*. Technical report, University of Illinois. Jan-Oct 1989.

WILLIAMS, L. 1983. Pyramidal parametrics. *SIGGRAPH Comput. Graph.* 17, 3 (Jul. 1983), 1-11

ZHAI, S., BUXTON, W., AND MILGRAM, P. 1996. The partial-occlusion effect: utilizing semitransparency in 3D human-computer interaction. *ACM Trans. Comput.-Hum. Interact.* 3, 3 (Sep. 1996), 254-284.