Exploring the Design Space of Multiscale 3D Orientation

James McCrae*   Michael Glueck†   Tovi Grossman‡   Azam Khan§   Karan Singh¶
Autodesk Research   Autodesk Research   Autodesk Research   Autodesk Research   University of Toronto
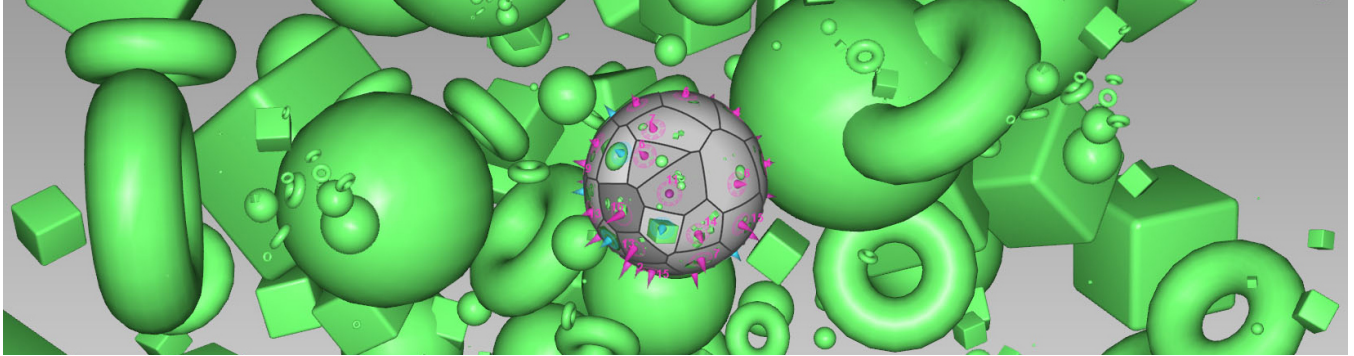University of Toronto

Figure 1: *Users can easily become disoriented in complex 3D scenes containing objects that exist at a number of scales. Advanced visual interfaces, such as the mirror ball (center), may help users understand their position and orientation in the scene.*

Abstract

Recently, research in 3D computer graphics and interaction has started to move beyond the narrow domain of single object authoring and inspection, and has begun to consider complex multiscale objects and environments. This generalization of problem scope calls for more general solutions, which are more akin to information visualization techniques than traditional computer graphics approaches.

We consider the general problem of the users understanding of their position and orientation within a multiscale 3D scene and propose a classification of the design space. To ground this theoretical discussion, we present initial explorations into grouping techniques, visualizations, and interactions to facilitate multiscale 3D orientation.

CR Categories: K.5.2 [User Interfaces]: Graphical User Interfaces (GUI)—3D Graphics;

Keywords: Multiscale, Visualization, 3D Orientation, Design Space

---

*e-mail:mccrae@dgp.toronto.edu
†e-mail:michael.glueck@autodesk.com
‡email:tovi.grossman@autodesk.com
§e-mail:azam.khan@autodesk.com
¶email:karan@dgp.toronto.edu

1 Introduction

Computers and 3D graphics applications are continuously increasing in power, memory, and rendering capabilities, making larger and more complex 3D scenes possible. Domains such as medical visualization, architecture and urban design, geospatial scanning, astrophysics, biochemistry, and abstract data analysis are beginning to consider massive datasets. Many of these datasets contain objects that exist at multiple scales, that is, the objects have meaningful observable properties at scales that are one or more orders of magnitude apart. Even in Figure 1, showing an environment filled with spheres, cones, torii, and cubes of various scales, it is difficult to communicate the fact that there are many tiny sub-pixel objects in the scene as well as several massive objects at a distance.

Multiscale datasets may have a number of inherent structural aspects. For example, a geospatial dataset, as found in Google Earth or Microsoft Virtual Earth, has a large sphere and many small objects on the surface of the sphere. More advanced versions would contain further details inside the buildings on the Earths surface at yet another scale to represent building components such as windows, bricks, or beams. At a certain scale on the surface of the Earth, relative position and orientation would be considered to be two-dimensional, as is used in automotive GPS applications, despite the fact that the dataset is fully three-dimensional. An anatomical dataset would be considered to be more of an immersive solid environment that could, for some purposes, be considered to be hierarchical e.g. in the body, in an organ, in a cell, in a cellular component like the nucleus.

Several scientific fields commonly consider phenomena that cover several orders of magnitude of scale. In the extreme, the visual representation of these scale changes has yet to be tackled by researchers and so, has traditionally been done with an artists hand-drawn illustration. Shown in Figure 2, a hierarchical representation of the human muscle system is drawn. In Figure 3 the structure of the known universe, as seen from Earth, is depicted. Both illustrations use a call-out technique to switch between scales.

Within a reasonable number of scales, interactive data-driven 3D scenes can be created but only recently have techniques been developed to dynamically display such scenes [McCrae et al. 2009].
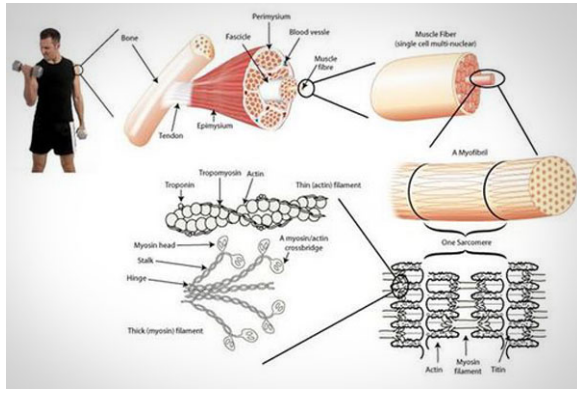
Figure 2: *Artist's rendering of the human muscle structure hierarchy (©University of Waikato).*
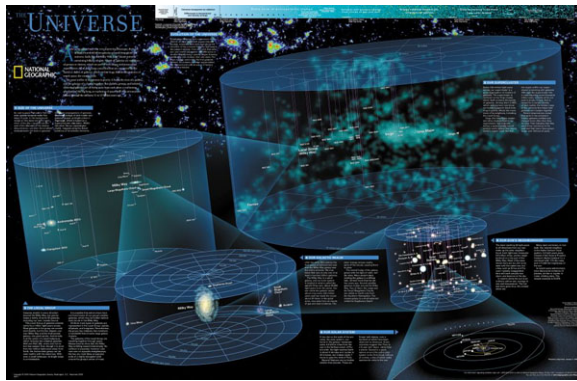

Figure 3: *Artist's rendering of the known universe from our solar system to the most distant super-clusters (©National Geographic).*

Moreover, interactions are limited because current 3D graphics systems and tools were primarily designed only for a single scale environment. As such, we consider a new thread of research pertaining to general interaction with multiscale 3D datasets. In particular, this initial work examines user understanding of position and orientation within such an environment.

2 Related Work

2.1 Multiscale Models

We survey previous work dealing with multiscale models and their navigation. [Zhang 2005] demonstrated the use of animation to enhance user understanding between scales in a multiscale model. [Zhang 2009] stresses the importance of travel speed, and explored different techniques which offered the user control over this scale-dependent parameter. [McCrae et al. 2009] proposed a solution to automatically determine travel speed in a scale-sensitive way using a cubemap technique. In addition, their approach solved other issues relevant to multiscale navigation such as the management of viewing frustum parameters, and was used for collision avoidance. They also presented numerous scale-sensitive interaction techniques for navigation, and demonstrated their effective use within multiscale 3D models.

2.2 Orientation and Awareness of Off-Screen Targets

In the context of traditional 3D applications, the user's understanding and control of their orientation has been studied [Khan et al. 2008] and the resulting ViewCube widget was shown to be effective on typical single-object inspection tasks. However, for multiscale 3D scenes, the global nature of the ViewCube reduces its value. In this case, orientation may be better garnered from relative positioning of local landmarks.

Many video games, both 2D and 3D, have integrated cues to alert players to the location of in-game opponents to increase awareness of off-screen targets. We believe this type of functionality will be central to user orientation in multiscale 3D scenes where it is unlikely that all of the items of interest will be visible within the user's current viewport, or is even possible in any single view of the scene.

This problem has been explained in 2D for handheld devices which have small screen sizes [Burigat et al. 2006]. CityLights [Zellweger et al. 2003], Halo [Baudisch and Rosenholtz 2003], and more recently Wedge [Gustafson et al. 2008] present the user with on-screen proxies as indicators, which help to determine the direction and distance of off-screen targets. Interesting hybrid techniques have also been developed, such as Hop [Irani et al. 2006], which presents users with Halos to indicate the location of off-screen targets, but also allows inspection of a local proxy object without necessitating navigation. Melange [Elmqvist et al. 2008] offers users a folded-space view which guarantees simultaneous visibility of multiple focus regions. It is not clear how such techniques would adapt to 3D scenes, where a user is not limited to planar panning and zooming. Our work builds upon such 2D techniques, but explores potential solutions for 3D environments that are also multiscale in nature.

2.3 Clusters of Targets and Occlusion

When viewing complex 3D scenes, it is possible for many objects to be closely clustered. In such situations, it becomes more difficult to differentiate specific objects from the surrounding distractors, and occlusions can become problematic. BalloonProbe [Elmqvist 2005] is a technique whereby distractor objects in a cluster are pushed to the outside of a spherical force-field to make target objects visible. Subsequently, Elmqvist and Tudoreanu [2006] analyzed the effectiveness of two fish-eye view techniques and two BalloonProbe shapes. Elmqvist and Tsigas [2006] also suggested animating between parallel and perspective view projections to reduce occlusion effects of clustered objects. Later, they also presented a taxonomy of 3D occlusion management techniques [Elmqvist and Tsigas 2008]. Since we are interested in preserving the spatial relationships of objects within the scene, we will explore unobtrusive methods of cueing users to occluded and clustered objects without directly modifying the view projection or displacing scene geometry.

2.4 Object-Based Spatial Partitioning

Another body of research has investigated spatial partitioning schemes to aid in differentiating clustered objects. In particular, Voronoi decomposition has been implemented to calculate discrete regions of space based on the distribution of targets in a scene. Bubble Cursor [Grossman and Balakrishnan 2005] and Starburst [Baudisch et al. 2008] both utilize this method in a 2D environment to cluster space into individual object selection regions. Vanacken et al. [2007] extended the Bubble Cursor to 3D to cluster space in 3D environments. Voronoi spatial partitioning has also been used in 3D as a method of obstacle avoidance in path planning [Pendragon and While 2003]. We are interested in applying spherical Voronoi

decomposition as a method of discretely dividing space around target objects to facilitate the creation of spatial hierarchies, to ease object searching in large environments.

3 Design Space

Designing interfaces for 3D environments is a non-trivial task [Bowman et al. 2004], complicated by the difficulties inherent in navigating and exploring 3D scenes [Fitzmaurice et al. 2008; McCrae et al. 2009], as well as the confusing spatial relationships that sometimes exist between scene objects [Elmqvist and Tsigas 2008; Glueck et al. 2009]. To address these issues we structure the design space of multiscale 3D orientation as follows:

- Preserve both egocentric and exocentric user goals.

- Landmark formation via the grouping of objects.

- Environment visualization using indicators and controllers.

3.1 Classification of User Goals

First, it is important to consider whether the user has an egocentric or exocentric mindset while viewing the scene. When exploring a scene egocentrically, the user considers objects relative to their position in 3D space, that is the current camera position. Conversely, an exocentric mindset would lead a user to evaluate spatial relationships relative to a global constant, such as the origin.

Within the framework of user centricity, we classify user motivation in terms of two dimensions: *what* the user is looking for and *where* the user hopes to find the result. Both of these axes are divided into the general cases (*anything* and *anywhere*), and the specific cases (*something* and *somewhere*). These distinctions help us categorize and support users' motivations, summarized in Figure 4.

In the most general case (*anything/anywhere*), a user might be viewing an empty viewport, although she is certain the scene is not empty. This occurs when the user is viewing an empty or seemingly empty viewport, where all geometry lies outside of the viewing frustum. The camera could be facing away from all the scene geometry or the geometry could be too far away that it lies outside the far clipping plane. Distant results may be invisible even if they are in the viewing frustum, if they are rendered in sub-pixel size. Conversely, the camera could be too close to geometry or may even lie inside geometry, such that the geometry fills the entire viewport. This circumstance might motivate her to search for any object, in hopes that finding some geometry will help put her position in virtual 3D space into context.

In the case where content is visible in the viewport, the user may be casually browsing the scene to develop a general understanding of the scene contents and/or structure, that is, to "get their bearings" or understand their orientation. Even when objects are inside the viewport, there are pathological cases caused by objects occluding each other from some or all viewpoints [Elmqvist and Tsigas 2008]. Filtering out objects that are not of interest and rendering them semi-transparently may help [Vanacken et al. 2007], but the remaining items may still occlude one another. Worse yet, one object could be completely inside another object. Even when objects do not occlude one another, closely clustered objects may be hard to differentiate from one another. Our experiences indicate that in practice, complex 3D scenes can contain instances of all of these cases, sometimes occurring in combination with one another.

More commonly the user will be looking for a particular object. If the user is looking for a specific object, regardless of its position, we classify the search task as *something/anywhere*.

When the user is instead looking for any object in a specific location (*anything/somewhere*), they may be doing so egocentrically or exocentrically. For example, in the egocentric case, a user might search for the closest object to the camera, while in the exocentric case, they may search for the closest object relative to a specific object in the scene.

In the most specific case (*something/somewhere*), a user is looking for a specific object at a specific location. Again, there are egocentric and exocentric sub-cases to consider. For example in the former, the user might be viewing a model of a car and be interested only in the lock nuts of the wheel closest to the current camera position. In the latter case, the user is only interested the lock nuts of the front left wheel. In both cases, the user is searching for specific objects in a specific region of 3D space.

Figure 4: *A broad classification of user goals and example cases.*

In all cases, to maintain their sense of orientation, the user must be aware of where objects lie in relation to some position, either the current viewpoint or a fixed point in space. For an orientation technique to be robust under all possible conditions, it needs to represent where objects lie within the scene and where objects lie relative the user's current viewpoint.

3.2 Landmark Formation: Object Grouping

An important consideration before presenting the user with a visualization is the grouping of objects to help form visual landmarks. We first motivate the need for an object grouping approach, and outline two methods that vary according to in-scene context. The variation resulting from context information makes the approaches suitable for grouping objects in multiscale models.

An obvious motivation for grouping objects is that it avoids presenting the user with an overwheling number of object representations at once. In addition, all visualizations will represent each object as a graphical element and this will require some amount of screen-space. Having too many objects, or having representations too densely packed within a local area of the visualization may result in the user not being able to discern a specific object or worse left unable to grasp any understanding of the objects in the scene altogether.

In an interactive system, object grouping is important for providing the user with a limited number of choices at any step. Grouping can be used effectively to impose a spatial hierarchy for all objects in

the 3D scene. An important point to note is that where user interaction occurs within the context of the scene, the user is continuously provided with visual affordances or landmarks. This information aids the user in making traversal decisions from the starting position (or "root node") down the hierarchy to a specific object (or "child node") in the scene.

There is an infinite space of grouping approaches that can be implemented. However, we can divide this space of algorithms into two classes: those that are variant with respect to in-context data (e.g. camera position, camera orientation), and those that are not. It is recognized that there may be no "best" grouping algorithm, indeed the effectiveness of one method over another is largely scene structure and task dependent.
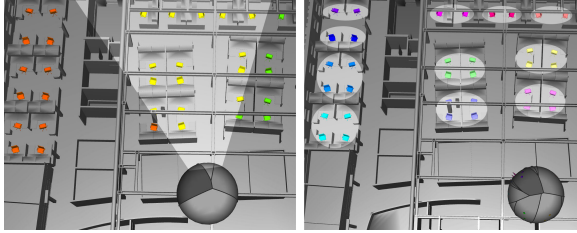

Figure 5: *Top views of the floor of an office which contains chair objects. Left: Object grouping using angular distance with respect to camera position. Right: Object grouping using local distance between objects. Highlighting is used to show the difference between the two grouping approaches.*

We have explored various grouping approaches, in particular those that incorporate in-context data such as positions local to the camera. One approach implemented uses the angular distance between objects relative to the camera (see Figure 5, left), and groups objects together when the angle is below a provided threshold. This technique has the benefit that grouping is based on perspective - two objects at a constant distance from each other are more likely to be grouped together the further from the camera they are.

Another approach we have implemented uses the worldspace distance between objects (see Figure 5, right). The in-context aspect of this approach is that the distance between objects is scaled inversely by the distance from the camera position (when considering grouping two objects/groups, we scale their distance from each other by taking the minimum of the two distances from camera position to object/group centroids). Thus, a pair of objects kept a constant distance from each other will again get grouped together more aggressively if they are further from the camera. A scalar parameter controls the aggressiveness for grouping of distant objects.

Note that the distance and angle-based grouping approaches differ especially in the case of occlusion between objects. Consider two objects A and B, where A is close to the camera and B is relatively distant, and A occludes B when viewed from the camera position. Since A occludes B, the angle between them taken from the camera position is small and so they will be grouped using the angular-based method. In contrast, using the worldspace distance approach does not group them, as they are relatively distant from each other and A is close to the camera.

3.3 Environment Visualization

Given the context of a known user goal and a landmark formulation, we can now consider alternative ways of visualizing the environment than the traditional perspective projection, that may augment the user's understanding of their position and orientation within the

dataset. As multiscale 3D scenes are relatively unstudied, we synthesize previous work into the general notion of an *orientation indicator* together with an *orientation controller* that provides specific representations of the environment for specific user goals. A number of orientation controllers are presented to further explore the design space.

3.3.1 Orientation Indicator: Cones

Motivated by the 2D Wedge technique [Gustafson et al. 2008], our techniques convey orientation information of objects in the form of 3D cones, which indicate the direction and relative distance of each target from the widget. Each cone is oriented such that the tip of the cone points toward a result and its size can be scaled to indicate relative distance amongst all objects, with larger cones representing more distant objects and smaller cones representing closer ones. The cones are rendered opaquely, with smooth-shading and lighting effects, which helps clarify their orientation. The cones incorporate interactive functionality. Hovering over a cone presents a thumbnail preview of the object or group that the cone points towards (see Figure 6). Clicking on a cone moves the user along a path to the target object, providing a direct method of navigating towards objects. An on-screen back button then appears allowing the user to navigate back to the previous camera position.
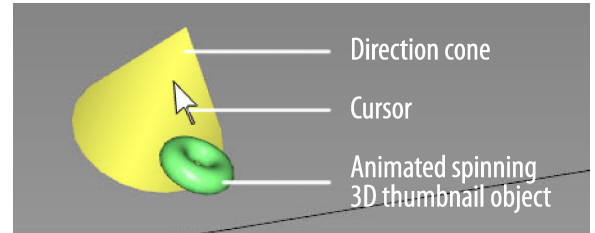

Figure 6: *A 3D thumbnail is displayed when the mouse hovers over a cone, highlighted in yellow.*

Cones can either represent single objects or a group of them. Cones representing single objects are shown in blue, while cones which indicate object groups ("aggregate" cones) are presented in pink with a number indicating how many objects there are in the group. In addition, the number of objects indicated by an aggregate cone may be visually represented by a segmented disk, shown at the base of the cone with one segment per object (see Figure 7).
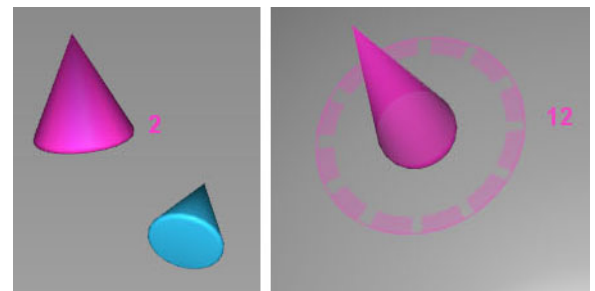

Figure 7: *Example cones. (Left) A standard cone on the right and an aggregated cone on the left. (Right) Detail of an aggregated cone, showing segmentation of the base and the number of objects represented.*

Aggregate cones provide the same interactive functionality as non-aggregate cones. If an aggregate cone is clicked, the camera moves to a child visualization widget that represents this subset of objects.

In this way, a hierarchical spatial partitioning scheme is formed based on objects in the 3D scene.

The cones support our design goals of indicating spatial relationships of the objects, and also helps indicate the presence of objects which are not immediately visible. The cone aggregation also aids our goal of enhancing the visibility of occluded and clustered objects. For example, if only one object is visible but the cone is an aggregate cone indicating two objects, the object which is visible either occludes or contains another object. Also, the thumbnail preview of the objects indicated by a cone animates a rotating viewpoint, which aids in revealing occluded objects.

A number of orientation controllers discussed below make use of the cone indicator to help convey a better sense of 3D orientation.

3.3.2 Orientation Controller: Wedge Ring

The Wedge Ring can be seen as a 3D extension of the Wedge technique [Gustafson et al. 2008]. The Wedge Ring offers users insight into the position of off-screen targets in an egocentric manner, relative to the user's current 3D position. Analogous to the Wedge technique, we display peripheral cones as orientation indicators. Unlike the Wedge technique, the cones in our visualization do not repel one another, and some intersection is allowed to occur (see Figure 8). The amount of intersection relates directly to the aggressiveness of object grouping.
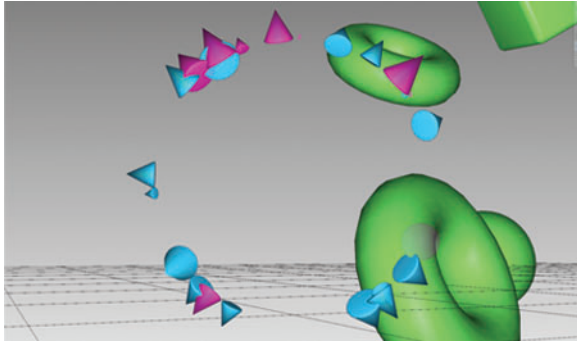

Figure 8: *Wedge Ring: an egocentric visualization, inspired by the 2D Wedge technique [Gustafson et al. 2008].*

As an object moves outside of the camera's field of view due to change in orientation, the cone oriented to indicate that object fades into view. Cones in this visualization are constrained along an invisible circular track within the image plane. The position of each cone on this plane is given by the vector from the camera's eye to the object (or group centroid), projected onto the image plane. Normalizing each projected vector yields the cone position on the circular track. Cones slide around the track as camera orientation changes, and update their orientation and size during navigation.

To more-closely emulate the original Wedge technique, one could normalize the projected vectors using a 1-norm instead of the 2-norm, which would effectively produce positions on a rectangular track. However, we found a rectangular track less desirable as the cones speed up and slow down as they enter and exit the corners. A circular track keeps the cone motion more uniform, but at the expense of being closer to the center of the viewport making it more likely to occlude objects in view.

While providing sufficient information to complete tasks in the *anywhere* axis of search tasks, this egocentric visualization is especially suited for the *somewhere (egocentric)* range. However, the

3D Wedge is not always effective for indicating the presence of targets which are not visible. For example, if a search result is in the viewport but of sub-pixel size, the user may not easily see this target, since the cones fade as objects near the center of the viewport.

3.3.3 Orientation Controller: Wedge Sphere

To contrast the dynamic behavior of the Wedge Ring, we investigated another visualization design which we call the Wedge Sphere. The Wedge Sphere can be thought of as an exocentric implementation of the Wedge Ring. Cones are oriented, scaled, and colored in the same manner as the Wedge Ring; however, they appear in a fixed location in space. The initial position of the Wedge Sphere is determined based on the position and view direction of the camera at the time of invocation. The widget is rendered a fixed distance in front of the camera, centered in the viewport. Rather than being constrained to a two-dimensional track, the cones are instead distributed over the surface of an invisible sphere. The user can control the distance at which they orbit the sphere with a slider control. This gives the user control over how prominently the visualization widget is rendered in the scene. Unlike the Wedge Ring technique, cones do not fade in and out as the objects they indicate enter and exit the viewport (see Figure 9).
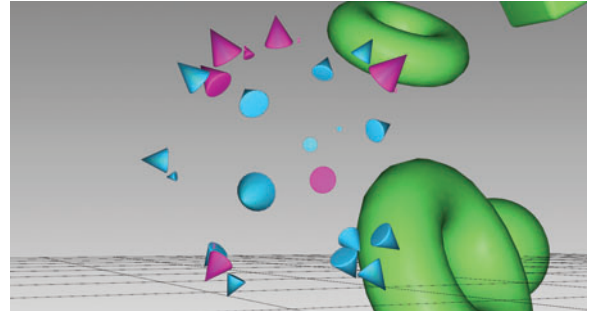

Figure 9: *Wedge Sphere: an exocentric visualization.*

As an exocentric visualization technique, the Wedge Sphere is helpful for user tasks within the *somewhere (exocentric)* range of search tasks. Also, it may be easier for users to track which objects have already been viewed, better facilitating *something/anywhere* tasks than the Wedge Ring.

3.3.4 Orientation Controller: Spherical Radar

We took the Wedge Sphere and reduced the visualization into an egocentric 2D abstraction, which we call the Spherical Radar. Essentially, this is the spherical unwrapping of the coordinates of the objects presented in the Wedge Sphere. The polar coordinates of each cone are used for the horizontal and vertical positions on a 2D grid. To further convey the distance of objects, extended bases are rendered around the cones. Inspired by the design of Position Pegs [Glueck et al. 2009], the bases appear with a thicker shaded region when objects are close, and a thinner shaded region when objects are distant. The visual design of the base has also been extended, explicitly subdividing the shaded region to clearly indicate the number of objects represented by an aggregate cone (see Figure 10).

Unlike the above methods, the Spherical Radar technique only preserves context indirectly and is not presented in-scene. Instead we remove the user from the direct context of the 3D scene and offer the ability to view all objects simultaneously, regardless of position. The user can navigate through all the objects using pan and zoom operations, guided by 3D thumbnail previews given by the cones
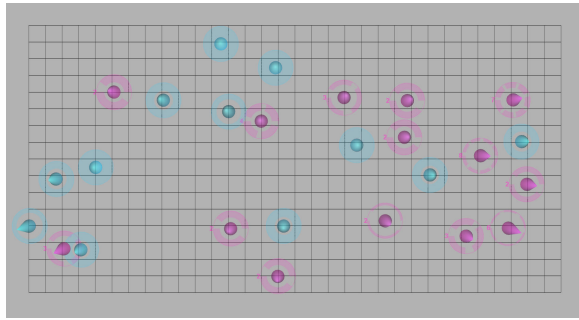
Figure 10: *Spherical Radar: an egocentric 2D abstraction.*

Figure 11: *Close-up of Mirror Ball: an exocentric visualization based on a spherical Voronoi diagram.*

(see Figure 7). As the user zooms in on an aggregate cone, it subdivides into separate cones representing either objects or sub-groups.

This technique provides an egocentric visualization that is more spatially consistent than the Wedge Ring, at the cost of sacrificing the direct in-scene context of the results. Egocentric comparisons such as determining the closest target to the user become easier, while exocentric tasks such as comparing the spatial location of two objects in the scene become more difficult.

3.3.5 Orientation Controller: Mirror Ball

The Mirror Ball is an exocentric visualization that expresses the spatial distribution of objects by showing Voronoi region boundaries on the surface of a sphere. Visually, the Mirror Ball is like the rear-view mirror in a car, and combines the spherical presentation of cones from the Wedge Sphere and the augmented cone design from the Spherical Radar. Objects represented are rendered to a texture and applied to each region of the surface of the Mirror Ball, creating the mirroring effect. Each rendering is performed using a unique camera position which properly frames the object/group, and the camera orientation comes from the direction to the object/group from the Mirror Ball. Because of this, objects which are very distant or small will be clearly visible on the surface of the Mirror Ball. As such, objects may be closer or further than they appear. This makes it ideal for visualizing multiscale objects.

To minimize the occluding effect of the cone on the projected image, we keep the cone slim and provide a relative distance cue by scaling only the height of the cone (see Figure 11). The shading of the background of each Voronoi region represents the relative distance to the object/group, where "brighter" corresponds to "closer".

Aggregate cone behavior is also enhanced through child Mirror Balls to navigate the grouped objects. Parent Mirror Balls remain visible in the 3D scene and a dashed line is drawn from the parent to the child to indicate the relationship. Users are thus able to navigate through groups hierarchically in the scene (see Figure 12).

For a particular incident viewing direction, the surface of the Mirror Ball conveys information about objects in the half-space behind the camera. In the pathological case where an object is too small to be represented at the current viewing scale, that object will, almost contradictorily, become highly visible to the user when it moves behind the camera.

The Mirror Ball, like the Wedge Sphere, is also an exocentric visualization and we see it supporting similar user tasks, within the *somewhere (exocentric)* and *something/anywhere* ranges.
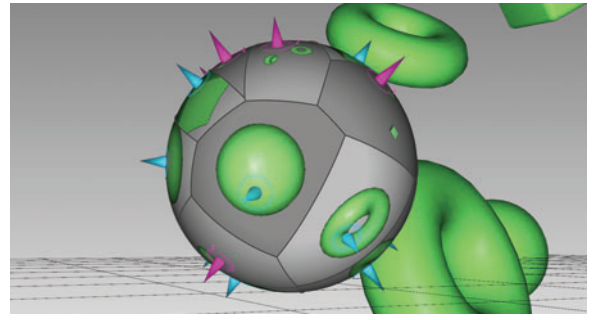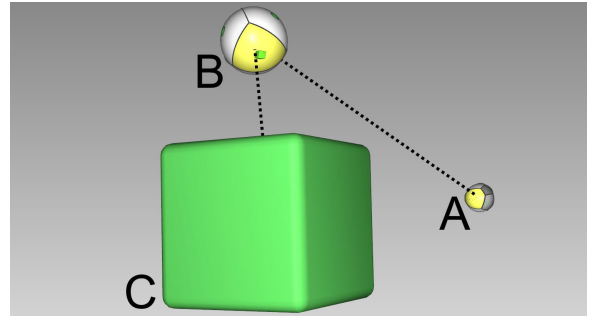

Figure 12: *A dashed path connects a parent widget (A), child widget (B), and target object (C).*

3.3.6 Orientation Controller: Anamorphic Lens

The Anamorphic Lens view, shown at the center of the viewport, extends the camera's field of view to nearly 180 degrees both horizontally and vertically (see Figure 13). For instance, the lens conveys a purple sphere will be visible if the user rotates the camera upward and to the right. For each region, the background colour conveys a relative distance to the centroid of its object cluster, where whiter shades are closer. So, for example, we can see that the central red cluster is further away than the purple sphere (even though the purple sphere is not visible in the viewport) due to the darker background of the central red cluster's region. As with the Mirror Ball, each region is textured with a perspective of the object/group that is at the proper distance to frame it. To see an example of this, observe the blue dot toward the top left of the figure, in front of the orange object cluster. Hardly visible, its corresponding region on the Anamorphic Lens shows it to be somewhat closer than the orange cluster, and that it is in fact a torus.

At the bottom left we simultaneously show the Mirror Ball to visualize objects behind the camera. In Figure 13, we can observe for example that there are no objects directly behind the camera, but that there is a green object group which contains a cube directly above, that is outside of the current view. Combining these two techniques together gives a visual representation for just about all objects in the scene (except when the Anamorphic Lens/Mirror Ball represent an object very obliquely).

3.3.7 Orientation Controller: Flower Garden

While the above technique allows the user to see most objects in the scene simultaneously, we present a technique that explicitly shows all objects and groups. In a technique we call the Flower Garden, we extend the Mirror Ball by including the option to unfold it into a
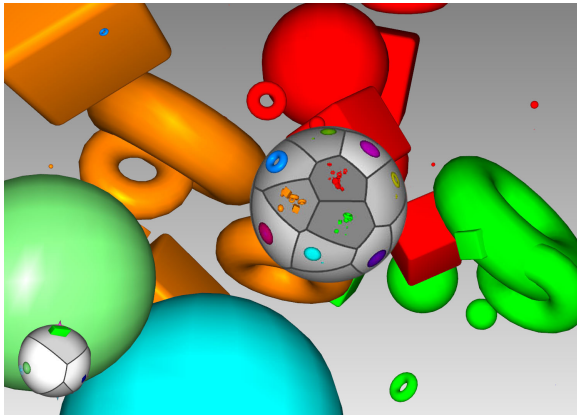
Figure 13: *Our anamorphic lens visualization. Object clusters are uniqeuly coloured to convey the correspondence between their in-scene and visualization representations. Combined with the Mirror Ball shown at bottom left, this method visualizes objects both in front of and behind the camera simultaneously.*

planar visualization that displays all objects at the same time. Similar to the Spherical Radar, we remove the user from the direct context when displaying the Flower Garden (see Figure 14).
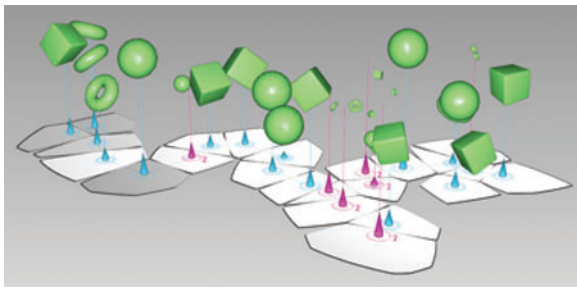

Figure 14: *Flower Garden: a 2D abstraction, based on an "unfolded" Mirror Ball. Adjacent regions signify little change in camera orientation is necessary to go from viewing one object to another.*

The motivation behind this visualization is that less active inspection would be required if information for all objects could be viewed simultaneously. Using the analogy of a flower garden, each cone has a stem rising out of it, at the top of which a rendering of the object/group is rendered, like the head of a flower. The relative height of the renderings visualizes the relative distance of objects from the camera, and also helps to minimize their interference.

This technique extends and complements the Mirror Ball visualization. We see the two being used in conjunction to allow the user to choose how to explore objects, in-scene and in-context, or in an abstract manner. Depending on the user task, we obtain some benefits of both the Mirror Ball and the Spherical Radar techniques.

4   Discussion and Future Work

During initial pilot studies, we collected impressions of our techniques by allowing participants to explore a randomly generated scene of graphics primitives. In general, participants found the Wedge Ring to be the most confusing due to its dynamic nature. The Spherical Radar was seen as too abstracted from the scene to be useful in the general case. Displaying thumbnails in the Mirror

Ball was preferred over hovering to invoke them. Overall, the most promising technique was the Mirror Ball and, by association, the Flower Garden.

Based on our design space structure, we feel that improvements in landmark formation will benefit all the visualization techniques. In particular, adding metadata into the visualization widget or into the scene may be helpful. Scented Widgets [Willett 2007] integrated auxiliary metadata into standard 2D input widgets, which indicated the behavior of other users' interaction. In combination with providing users with a sense of spatial history, this could provide new users with a roadmap to key scene features, or even provide more advanced users with richer data such as the number of times an object was viewed. These types of schemes could facilitate an improved sense of orientation over a longer period of usage.

We have also focused exclusively on scenes where content is static. There are many scenes in 3D, such as in animation and simulation, where content is potentially dynamic in nature. A visualization that is capable of responding to changes within a scene, either objects moving, or changing in shape or size is a challenging and interesting problem. In general, as orientation tasks in specific domains are attempted, new dimensions in the design space presented may become clear.

5   Conclusions

Our work serves as an initial exploration into the design space of user orientation within multiscale 3D environments and potential design solutions to basic user navigation tasks. Moving beyond the scales considered by typical 3D applications, we have discovered unique problems and solutions and have presented a design space that considers egocentric and exocentric user goals, landmark formation rules, and indicators and controllers for environment visualization. While creating advanced visual interfaces for multiscale 3D scenes is challenging along every step of the way, we feel that domains such as biochemistry, nanotechnology, and medical visualization will be greatly enhanced by improved interactions. Given that our work serves as an entry point into this topic, there are a number of interesting areas open to future work.

References

BALZER, M., AND DEUSSEN, O. 2005. Voronoi treemaps. In *Proc. INFOVIS '05*, 7.

BAUDISCH, P., AND ROSENHOLTZ, R. 2003. Halo: a technique for visualizing off-screen objects. In *Proc. CHI '03*, 481–488.

BAUDISCH, P., ZOTOV, A., CUTRELL, E., AND HINCKLEY, K. 2008. Starburst: a target expansion algorithm for non-uniform target distributions. In *Proc. AVI '08*, 129–137.

BOWMAN, D. A., KRUIJFF, E., LAVIOLA, J. J., AND POUPYREV, I. 2004. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

BURIGAT, S., CHITTARO, L., AND GABRIELLI, S. 2006. Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In *Proc. MobileHCI '06*, 239–246.

ELMQVIST, N., AND TSIGAS, P. 2006. View projection animation for occlusion reduction. In *Proc. AVI '06*, 471–475.

ELMQVIST, N., AND TSIGAS, P. 2008. A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 5, 1095–1109.

ELMQVIST, N., AND TUDOREANU, M. E. 2006. Evaluating the effectiveness of occlusion reduction techniques for 3d virtual environments. In *Proc. VRST '06*, 9–18.

ELMQVIST, N., HENRY, N., RICHE, Y., AND FEKETE, J.-D. 2008. Melange: space folding for multi-focus interaction. In *Proc. CHI '08*, 1333–1342.

ELMQVIST, N. 2005. Balloonprobe: reducing occlusion in 3d using interactive space distortion. In *Proc. VRST '05*, 134–137.

FITZMAURICE, G., MATEJKA, J., MORDATCH, I., KHAN, A., AND KURTENBACH, G. 2008. Safe 3d navigation. In *Proc. I3D '08*, 7–15.

FUNKHOUSER, T., MIN, P., KAZHDAN, M., CHEN, J., HALDERMAN, A., DOBKIN, D., AND JACOBS, D. 2003. A search engine for 3d models. *ACM Trans. Graph. 22*, 1, 83–105.

GLUECK, M., CRANE, K., ANDERSON, S., RUTNIK, A., AND KHAN, A. 2009. Multiscale 3d reference visualization. In *Proc. I3D '09*, 225–232.

GROSSMAN, T., AND BALAKRISHNAN, R. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. CHI '05*, 281–290.

GUSTAFSON, S., BAUDISCH, P., GUTWIN, C., AND IRANI, P. 2008. Wedge: clutter-free visualization of off-screen locations. In *Proc. CHI '08*, 787–796.

IRANI, P., GUTWIN, C., AND YANG, X. D. 2006. Improving selection of off-screen targets with hopping. In *Proc. CHI '06*, 299–308.

IYER, N., KALYANARAMAN, Y., LOU, K., JAYANTI, S., AND RAMANI, K. 2003. A reconfigurable 3d engineering shape search system part i: shape representation. In *Proc. ASME DETC 2003*.

KHAN, A., MORDATCH, I., FITZMAURICE, G., MATEJKA, J., AND KURTENBACH, G. 2008. Viewcube: a 3d orientation indicator and controller. In *I3D '08: Proceedings of the 2008 symposium on Interactive 3D graphics and games*, 17–25.

LAMPING, J., AND RAO, R. 1994. Laying out and visualizing large trees using a hyperbolic space. In *Proc. UIST '94*, 13–14.

LOU, K., JANYANTI, S., IYER, N., KALYANARAMAN, Y., PRABHAKAR, S., AND RAMANI, K. 2003. A reconfigurable 3d engineering shape search system part ii: database indexing, retrieval and clustering. In *Proc. ASME DETC 2003*.

MCCRAE, J., MORDATCH, I., GLUECK, M., AND KHAN, A. 2009. Multiscale 3d navigation. In *Proc. I3D '09*, 7–14.

NACENTA, M. A., ALIAKSEYEU, D., SUBRAMANIAN, S., AND GUTWIN, C. 2005. A comparison of techniques for multi-display reaching. In *Proc. CHI '05*, 371–380.

PENDRAGON, T., AND WHILE, L. 2003. Path-planning by tessellation of obstacles. In *Proc. ACSC '03*, 3–9.

SEBRECHTS, M. M., CUGINI, J. V., LASKOWSKI, S. J., VASILAKIS, J., AND MILLER, M. S. 1999. Visualization of search results: a comparative evaluation of text, 2d, and 3d interfaces. In *Proc. SIGIR '99*, 3–10.

TANGELDER, J. W., AND VELTKAMP, R. C. 2008. A survey of content based 3d shape retrieval methods. *Multimedia Tools Appl. 39*, 3, 441–471.

VANACKEN, L., GROSSMAN, T., AND CONINX, K. 2007. Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. *IEEE Symposium on 3D User Interfaces*, 117–124.

WILLETT, W. 2007. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics 13*, 6, 1129–1136. Student Member-Heer, Jeffrey and Member-Agrawala, Maneesh.

ZELLWEGER, P. T., MACKINLAY, J. D., GOOD, L., STEFIK, M., AND BAUDISCH, P. 2003. City lights: contextual views in minimal space. In *Proc. CHI '03*, 838–839.

ZHANG, X. 2005. Space-scale animation: Enhancing cross-scale understanding of multiscale structures in multiple views. In *CMV '05: Proceedings of the Coordinated and Multiple Views in Exploratory Visualization*, 109–120.

ZHANG, X. L. 2009. Multiscale traveling: crossing the boundary between space and scale. *Virtual Real.*, 101–115.