

View Projection Animation for Occlusion Reduction

Niklas Elmqvist, Philippas Tsigas
Department of Computer Science & Engineering
Chalmers University of Technology, Göteborg, Sweden
{elm | tsigas}@cs.chalmers.se

ABSTRACT

Inter-object occlusion is inherent to 3D environments and is one of the challenges of using 3D instead of 2D computer graphics for information visualization. In this paper, we examine this occlusion problem by building a theoretical framework of its causes and components. As a result of this analysis, we present an interaction technique for view projection animation that reduces inter-object occlusion in 3D environments without modifying the geometrical properties of the objects themselves. The technique provides smooth on-demand animation between parallel and perspective projection modes as well as online manipulation of view parameters, allowing the user to quickly and easily adapt the view to avoid occlusion. A user study indicates that the technique significantly improves object discovery over normal perspective views. We have also implemented a prototype of the technique in the Blender 3D modeller.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces; I.3 [Computer Graphics]: Three-Dimensional Graphics and Realism

General Terms

Human Factors, Design

Keywords

occlusion reduction, 3D visualization, view projection

1. INTRODUCTION

Three-dimensional computer graphics provides considerable potential for information visualization. However, there is an increased overhead associated with using 3D over conventional 2D graphics for the purposes of orientation and navigation within an environment. In general, 3D graphics impose a high cognitive load on users trying to gain and

maintain an overview of the environment, and often cause disorientation, confusion, and sometimes even nausea. One of the central issues behind this effect is *occlusion*, the phenomenon that nearby objects occlude more distant objects in 3D even if the objects are not overlapping in space.

Why is occlusion a problem in 3D visualization environments? There are two basic issues. First of all, and perhaps most importantly, there is a *discovery* problem if an object is occluded since then the user may never know that it exists. And secondly, even if the user is aware of the existence of an occluded object, there is an *accessibility* problem, since the user will have to move the viewpoint in some nontrivial way in order to retrieve the information encoded in the object.

In this paper, we explore the occlusion problem in more detail, attempting to build a theoretical model for its causes and parameters and also to identify possible solution strategies. Using this model, we develop an interaction technique for view projection animation that aims to reduce inter-object occlusion in 3D environments without modifying the geometrical properties of the environment itself, nor the objects in it. The technique allows for smooth animation between the conventional perspective projection mode, which mimics human vision in the real world and is commonly used for 3D visualizations, and parallel projection mode, where the depth coordinate is ignored and objects are assigned screen space according to their actual geometrical size, regardless of their distance to the viewpoint.

A formal user study conducted on our technique in relation to traditional perspective projection shows a significant improvement of object discovery in 3D environments in our favor. The cost for this increased efficiency is instead significantly longer task completion times; users essentially trade speed for accuracy when using our technique. On the other hand, the results also show that there is no statistically significant difference in completion times between using our projection animation technique and a user-controlled camera. In fact, we believe that our projection animation technique requires less viewpoint position manipulations than normal perspective views, and thus users run a lower risk of becoming disoriented when navigating 3D space.

2. RELATED WORK

While most work on improving the usability of 3D visualizations attacks the higher-level problem of navigation, there also exists a number of papers dealing more directly with object discovery and access in complex 3D environments. The Worlds-in-Miniature technique [11] uses a miniature 3D map of the environment to support both discovery and ac-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AVI '06, May 23–26, 2006, Venezia, Italy.

Copyright 2006 ACM 1-59593-353-0/06/0005 ...\$5.00.

cess, worldlets [4] provide both global overview maps as well as local views optimized for detail, bird’s eye views [5] combine overview and detail views of the world, and balloon force fields [3] inflate to separate occluding objects. None of these make direct use of the view projection to improve perception; however, temporally controlled non-linear projections [10] have been used to great effect in improving navigation and perception of 3D scenes.

Recent developments in the area also include multiprojection rendering, where several perspectives are combined into one, mainly for artistic purposes. Agrawala *et al.* [1] compose views of multiple cameras, where each object is assigned to a specific camera perspective, allowing for creative scene composition akin to the work of famous painters. Singh [9] uses a similar approach, but smoothly combines the multiple viewpoints into an interpolated virtual camera instead of composing the images of disjoint cameras on an image-level. While only slightly related to our technique, these works give valuable insight into the manipulation of projection transforms. Our technique can also be compared to glances [8], with the perspective view as the base view and the parallel view as the glance.

Finally, the view projection animation technique described in this paper bears close resemblance to the *orthotumble* technique presented by Grossman *et al.* [7] and its predecessor, the animated view transitions described in [6], but the purpose of these are primarily for maintaining and understanding 3D models rather than reducing occlusion. In addition, our approach uses a more correct algorithm for the “dolly-and-zoom” effect, whereas the *orthotumble* algorithm is based on linear matrix interpolation.

3. THE OCCLUSION PROBLEM

The occlusion problem space in 3D environments is defined by the intrinsic *properties* of the environment, their interaction with *human cognition*, the *visual tasks* involved, and the ensuing effects caused by the occlusion. In essence, the environment and its geometrical properties interact negatively with human vision, causing occlusion of objects and leading to loss of correctness and productivity.

3.1 Model

We represent the 3D world U by a Cartesian space $(x, y, z) \in \mathbb{R}^3$. Objects in the set O are volumes within U (i.e. subsets of U). The user’s viewpoint $v = (M, P)$ is represented by the view and projection matrices M and P .

An object o is said to be *occluded* from a viewpoint v if there exists no ray r between v and o such that r intersects no intervening object. Analogously, an object o is said to be *visible* from a viewpoint v if there exists a ray r between v and o such that r intersects no other object. An object o is said to be *partially occluded* from viewpoint v if o is visible, but there exists a ray r between v and o such that r intersects another object.

3.2 Visual Tasks

The occlusion problem typically occurs for (i) *object discovery* (finding all objects $o \in O$ in the environment), and (ii) *object access* (retrieving graphically encoded information associated with each object). There exist many other such *visual tasks*, but in this treatment we only regard the above two as relevant to inter-object occlusion.

We can divide these two tasks into a number of visual

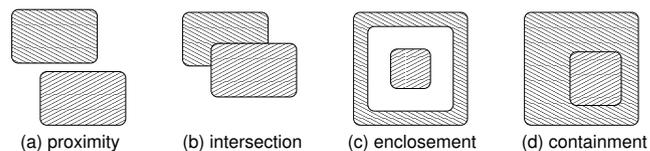


Figure 1: Object interactions causing 3D occlusion.

subtasks. Discovery consists of a combination of visually searching for an unspecified target (unguided visual search), and identifying an object as a target (identification). Access, on the other hand, is again a visual search, but this time for an already known target (guided visual search), followed by the process of perceiving the information encoded in the target (perception).

3.3 Full and Partial Occlusion

Both object discovery as well as object access are severely hampered by the existence of fully occluded objects. More specifically, a fully occluded object will be impossible to discover in an unguided visual search without the use of some occlusion reduction strategy, and identification never becomes an issue. Analogously for object access the visual search will fail (even if the user knows where the object is), and so will perception. As a result, both tasks will affect the efficiency and correctness of users solving tasks using a visualization, but clearly, threats to object discovery are the most serious; if the user is unaware of the existence of an object, she will have no motivation to look for it and access never becomes an issue.

Partial occlusion, on the other hand, has a different effect on these tasks. For object discovery, users may have difficulties distinguishing object identity if too large a portion of the object is occluded. In this situation, the user may either miss the object entirely, count the same object multiple times, or believe different objects are part of the same object. Object access will succeed in the visual search subtask, although the perception subtask may still fail due to important parts of the specific object being occluded.

3.4 Environment Properties

The geometrical properties of the visualization environment are of special interest in this framework because they allow us to characterize the visualization and determine the nature of the occlusion problems that may arise. These properties can also be used to decide which occlusion reduction strategies are applicable for a specific situation.

In this treatment, we identify three main environment properties which interact to cause inter-object occlusion and influences the two basic visual tasks associated with the environment:

- *object interaction* – measure of the spatial interaction of objects in the environment (see Figure 1);
- *object density* – measure of the amount of objects in the environment with regards to its size; and
- *object complexity* – measure of the detail level of individual objects in the environment.

Obviously, these are high-level properties which only generally describe an environment without going into detail.

3.5 Analysis: View Projection Animation

In this paper, we present a specific occlusion reduction technique called view projection animation where the camera projection matrix is smoothly animated from perspective to parallel projection and back. We can categorize this technique as part of a more general solution strategy based on dynamic manipulation of the view projection to favorably present objects and minimize occlusion in the environment.

View projection animation clearly improves object discovery by providing the user with means to avoid nearby objects hiding more distant ones. Toggling between the projection modes yields two different perspectives on the environment as well as intervening views during the smooth animation between them, strongly facilitating unguided visual search and, to some extent, identification, by disambiguating between occluding objects. Object access benefits much less from the technique; previous knowledge of the target’s location is of little use when the view space is non-linearly transformed by the technique.

The applicability of the technique is limited to intersecting objects: since we do not transform the space itself, enclosed and contained objects will remain occluded even after the projection transformation. As will be seen in the user study at the end of this paper, the technique performs well at low to medium-sized object density.

4. VIEW PROJECTION ANIMATION

The idea behind our technique for view projection animation is to combine the major virtues of parallel and perspective projections: that (i) perspective projections offer a realistic view of a 3D environment akin to our perception of the real world, and that (ii) parallel projections offer a more accurate and exact view of the environment. Furthermore, the nature of parallel projection means that inter-object occlusion is reduced in comparison to perspective projection since objects are assigned screen space according to their geometrical size only, regardless of their distance to the camera. Using perspective projection, a tiny object can fill the whole viewport if the viewpoint is located sufficiently close.

By combining these two projection modes into the same interaction technique, we are potentially able to enjoy the best of both worlds: the view defaults to perspective projection when the user is navigating the space normally, but allows for easy switching (glancing) to parallel projection when the user needs to perform object discovery or access. Furthermore, the transition between perspective and parallel projections, and vice versa, is smoothly animated to allow the user to maintain context of the environment and the projection at all times with a minimum of cognitive overhead. Furthermore, the actual transition provides additional information on the structure of the 3D scene.

In addition to transitions back and forth between perspective and parallel projections, we augment our technique with functionality to change the center of projection as well as modify the field-of-view angle in the perspective projection mode. Changing the center of projection gives an additional means for the user to arbitrate between occluding objects, and by gaining control of the field of view, the user can smoothly zoom in and out of the 3D environment at will. See Figure 2 for more details.

For our interaction technique we define three input buttons, labelled PROJ, UTIL, and SHEAR, respectively; these

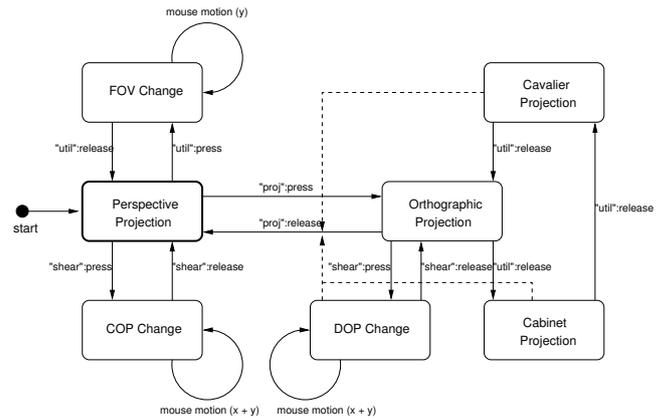


Figure 2: State diagram for the projection animation interaction technique.

can be mouse or keyboard buttons. In addition, the technique also captures mouse motion for some parameter changes, notably the field-of-view and center-of-projection (direction-of-projection for parallel mode) modification states. The parallel projection mode has a number of pre-defined oblique projection modes that the user can cycle between: orthographic (head-on parallel projection) versus cavalier and cabinet projections, where the direction of projection is set at fixed values. Note that for all parallel projection modes, the release of the PROJ input will smoothly revert the projection back to the default perspective mode. Reverting to the default state will also reset all view parameters, such as centering the center (or direction) of projection and setting the focal length to the default value.

4.1 Projection Transitions

Transitions between various projection states are performed through simple linear interpolation between the source and destination projection transforms. In the case of the parallel-to-perspective transition (and its inverse), however, a linear interpolation will yield unsatisfactory results due to the non-linear relation between these two projections. For this case, we need to explore the matrix M that relates the two projection matrices P_{par} and P_{per} .

As discussed above, a parallel view transform represents the situation where the focal length of the camera is infinite. The transition from perspective to parallel view can be approximated in a real-life camera by a so-called “zoom and dolly” operation, where the camera is moved backwards at the same time as the focal length is increased (i.e. zoomed in). By keeping these parameters balanced, the focused object in the camera view will maintain the same size and shape, but the rest of the scene will appear to be “flattened”. This is the effect we simulate in our transition between perspective and parallel projection.

4.2 Case Study: Blender Implementation

In order to study the feasibility and flexibility of our projection animation technique, we also implemented it inside the Blender [2] 3D modelling package. Blender is a very powerful and widely used 3D software suite that is freely available as Open Source under the GPL license. Our implementation integrates seamlessly into Blender and allows

for modellers to animate between parallel and perspective projections in different 3D windows; the software naturally already supported these projection modes prior to our modifications, so we changed the projection code to perform a smooth animation between the two matrices. In addition to this, we introduced the capability for users to change the center of projection while in orthographic mode, allowing for an additional way to reduce occlusion.

While a 3D modeller is not the primary target platform for our technique (even though Grossman *et al.* [6, 7] use the effect for this very purpose), this case study shows that the technique can indeed be implemented seamlessly inside existing 3D applications.

5. USER STUDY

We have conducted a formal user study with two main motivations: (i) to empirically investigate the impact of occlusion on object discovery efficiency in 3D environments, and (ii) to study the performance of users given access to our view projection animation technique in comparison to users with a normal perspective view.

5.1 Subjects

We recruited 26 subjects, six of which were female, from the undergraduate engineering programs at our university. No previous experience of 3D applications was required. Ages ranged from 20 to 40 years of age, and all subjects had normal or corrected-to-normal vision.

5.2 Equipment

The experiment was conducted on a Pentium III 1 GHz desktop PC with 512 MB of memory and running the Linux operating system. All tasks were carried out using our prototype implementation. The display was a 19" monitor with the main visualization window fixed at 640×480 size.

5.3 Task

Subjects were asked to perform object discovery in a simple $100 \times 100 \times 100$ environment filled with 3D boxes by counting the number of boxes of a given color. Target colors were restricted to one of the primary RGB colors (i.e. red, green, or blue), and all distracting objects were configured to contain no elements of that color component. Beyond that, each task instance was fully randomized, including the position, orientation, and size of the distractors. At least 1 and at most 10% of the total number of objects were targets. Box dimensions (both targets and distractors) ranged from 1% to 12.5% of the environment dimensions. Intersection but no enclosure or containment was allowed. A simple 20×20 line grid was rendered at the bottom of the environment to facilitate user orientation.

The camera focus point was fixed at the center of the environment and the orientation was randomized within 60° from the horizontal. In addition, the camera position was also randomized and offset sufficiently from the focus point so that all objects in the scene were visible. Field-of-view angle for the perspective view was fixed at 60° . For the dynamic camera, the users could freely orbit the camera around the focus point as well as change the focus distance.

5.4 Design

The experiment was designed as a repeated-measures factorial ANOVA, with the independent variables “density”

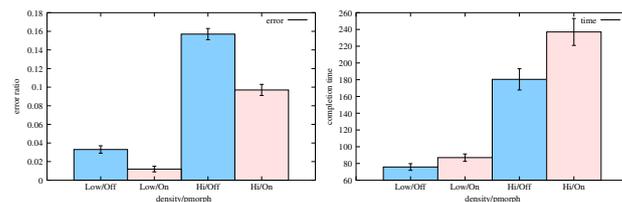


Figure 3: Mean error ratios and completion times (standard deviations shown as error bars).

(two levels, low or high), “camera” (static or dynamic, i.e. a fixed or a user-controlled camera), and “PMorph” (on or off, i.e. whether the projection animation technique was available or not), all of them within-subjects. The dependent variables were the total number of target objects, the number of found targets, and the completion time for each task. Subjects received the “PMorph” and “camera” conditions in randomized order to avoid systematic effects of practice; for the “density” condition, the ordering was low to high.

Users performed the test in sets of 10 tasks for each condition. Each task scenario was completely randomized, with either 50 or 200 total objects in the environment depending on the density, and up to 10% of them being targets.

For each specific condition, subjects were instructed in which features (dynamic or static camera, projection animation on or off) were available to them. Tasks were given automatically by a testing framework implemented in the software and answers were input by the user directly back into the framework, thus requiring no intervention by the test administrator. The software silently recorded the completion time, total target number, and found target number for each task. Trial timing started as soon as each new task was presented, and ended upon the subject giving an answer.

Each session lasted approximately thirty to forty minutes. Subjects were given a training phase of up to five minutes to familiarize themselves with the controls of the application.

With 26 participants and 10 search tasks for each of the 8 conditions, there were 2080 trials recorded in total. After having completed the full test, subjects were asked to respond to a post-test questionnaire.

6. RESULTS

We divide the results from the user study into correctness, completion times, and subjective ranking categories. The error ratio is defined as the cumulative error divided by the sum of the total number of targets for the task set, i.e. the number of errors per target. See Figure 3 for an overview of the results.

The average error ratio for a full task set (10 tasks) using normal perspective projection compared to projection animation was 0.095 (s.d. 0.003) versus 0.055 (s.d. 0.003), respectively. This is a statistically significant difference ($F(1, 25) = 75.757, p < 0.001$). Not surprisingly, density had a significant impact on correctness ($F(1, 25) = 407.290, p < 0.001$); the average error ratio for the low density condition was 0.022 (s.d. 0.002), to contrast with 0.127 (s.d. 0.005) for the high density condition.

The mean completion time of solving a full object discovery task set (10 tasks) using normal perspective projection was 128.093 (s.d. 7.803) seconds, whereas the mean comple-

tion time for projection animation was 162.311 (s.d. 9.697) seconds. This is also a significant difference ($F(1, 25) = 38.752, p < 0.001$).

The subjective rankings given by the participants in the post-test questionnaire are overall positive and in favor of our projection animation technique: 73.1 % ranked our technique over normal perspective projection.

7. DISCUSSION

This paper presents two main contributions: (i) the analysis of the space of the occlusion problem in 3D environments, and (ii) the view projection animation technique used to reduce inter-object occlusion for any 3D visualization. Both of these contributions are validated by the results of the formal user study; we see that increasing object occlusion leads to significantly reduced discovery efficiency, and that the availability of projection animation significantly boosts efficiency in all object density conditions, respectively. In addition, by giving users control over the viewpoint, the impact of the occlusion problem is significantly diminished. On the other hand, this comes at the cost of longer completion times; the participants spent much more time solving tasks when having access to projection animation or a controllable camera, essentially trading speed for accuracy.

It is particularly interesting to study whether a user-controlled camera is sufficient to negate the occlusion problem, and whether the projection animation technique presented here is really necessary. There is no clear benefit of projection animation over a traditional dynamic camera. However, we claim that projection animation is orthogonal to controllable cameras, and that they complement each other. Furthermore, our informal observations during the user study indicated that users with access only to a controllable camera performed significantly more view changes than when having access to both a controllable camera and projection animation. All 3D view changes incur a risk of loss of context and orientation, especially for high object densities, and so it is in our best interest to keep the amount of such changes low. We suggest that a combination of the two conditions will work best for practical applications.

Parallel projection assigns screen space to objects proportional to their geometrical size regardless of the distance to the camera, but the drawback is that the viewing volume is a box instead of a pyramidal frustum as for perspective projection. This means that peripheral objects will be lost in parallel mode, essentially rendering these objects impossible to discover. By smoothly combining *both* parallel and perspective projection into a single interaction technique, we are able to sidestep this problem and get the best of both worlds from the two projections.

A potential drawback of the technique is that the use of parallel projection leads to loss of depth cues in a 2D image of the environment. However, the spring-loaded nature of the interaction allows users to easily switch back and forth between projection modes to disambiguate potential depth conflicts between objects.

8. CONCLUSIONS AND FUTURE WORK

We have presented an interaction technique for the seamless integration of perspective and parallel projection modes, allowing users to combine realism with accuracy, as well as reducing inter-object occlusion in 3D environment views.

We also presented a general theoretical framework for the occlusion problem as a whole. Results from a user study conducted on a prototype version of the technique show that occlusion in 3D environments has a major impact on efficiency, but that our technique allows for significant improvements in both object discovery and object access.

In the future, we intend to pursue additional means to reduce occlusion in 3D environments. This includes the use of overviews and maps as well as dynamic transparency and walkthrough generation.

Acknowledgments

Many thanks to the developers of the Blender project for their help on integrating the technique into the Blender3D modeller.

9. REFERENCES

- [1] M. Agrawala, D. Zorin, and T. Munzner. Artistic multiprojection rendering. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 125–136, 2000.
- [2] Blender, Mar. 2005. See <http://www.blender3d.org>.
- [3] N. Elmqvist. BalloonProbe: Reducing occlusion in 3D using interactive space distortion. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 134–137, 2005.
- [4] T. T. Elvins, D. R. Nadeau, and D. Kirsh. Worldlets – 3D thumbnails for wayfinding in virtual environments. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 21–30, 1997.
- [5] S. Fukatsu, Y. Kitamura, T. Masaki, and F. Kishino. Intuitive control of “bird’s eye” overview images for navigation in an enormous virtual environment. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 67–76, 1998.
- [6] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. W. Fitzmaurice, A. Khan, and W. Buxton. Interaction techniques for 3D modeling on large displays. In *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pages 17–23, 2001.
- [7] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. W. Fitzmaurice, A. Khan, and W. Buxton. Creating principal 3D curves with digital tape drawing. In *Proceedings of the ACM CHI 2002 Conference on Human Factors in Computing Systems*, pages 121–128, 2002.
- [8] J. S. Pierce, M. Conway, M. van Dantzych, and G. Robertson. Tool spaces and glances: Storing, accessing, and retrieving objects in 3D desktop applications. In *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pages 163–168, 1999.
- [9] K. Singh. A fresh perspective. In *Proceedings of Graphics Interface 2002*, pages 17–24, 2002.
- [10] K. Singh and R. Balakrishnan. Visualizing 3D scenes using non-linear projections and data mining of previous camera movements. In *Proceedings of AFRIGRAPH 2004*, pages 41–48, 2004.
- [11] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: Interactive worlds in miniature. In *Proceedings of the ACM CHI’95 Conference on Human Factors in Computing Systems*, pages 265–272, 1995.