A Taxonomy of 3D Occlusion Management Techniques

Niklas Elmqvist* Philippas Tsigas†

Department of Computer Science & Engineering
Chalmers University of Technology
SE-412 96 Göteborg, Sweden

ABSTRACT

While an important factor in depth perception, the occlusion effect in 3D environments also has a detrimental impact on tasks involving discovery, access, and spatial relation of objects in a 3D visualization. A number of interactive techniques have been developed in recent years to directly or indirectly deal with this problem using a wide range of different approaches. In this paper, we build on previous work on mapping out the problem space of 3D occlusion by defining a taxonomy of the design space of occlusion management techniques in an effort to formalize a common terminology and theoretical framework for this class of interactions. We classify a total of 25 different techniques for occlusion management using our taxonomy and then go on to analyze the results, deriving a set of five orthogonal design patterns for effective reduction of 3D occlusion. We also discuss the "gaps" in the design space, areas of the taxonomy not yet populated with existing techniques, and use these to suggest future research directions into occlusion management.

CR Categories: H.5.2 [Information Systems]: User Interfaces; I.3 [Computer Methodologies]: Computer Graphics

Keywords: occlusion management, occlusion reduction, taxonomy, design patterns, visual cues, depth perception

1 INTRODUCTION

Human beings employ all manners of visual cues and hints in order to correctly perceive and understand the three-dimensional world surrounding us. Of course, these visual cues can also work against us, fooling our perception into believing things about our environment that are simply not true. In some cases, this is done intentionally through various forms of optical illusions that exploit special characteristics of our minds. A more subtle point, however, is that we can instead choose to directly weaken certain of these visual cues in order to help the human to perceive and understand *more* of her surroundings. In some cases, this selective weakening of visual cues, primarily occlusion, size, and shape, can lead to dramatically increased performance when solving specific tasks in a 3D environment. While this may be difficult to achieve in the real world, it is a perfectly viable approach in a virtual 3D world being visualized on a computer.

In this paper, we explore the design space of interaction techniques that perform *occlusion management* by modifying certain depth cues in order to increase the spatial awareness of the human user and to facilitate special tasks, such as navigating, searching, or understanding the 3D world. More specifically, we present a taxonomy consisting of a small set of dimensions describing important characteristics of these techniques, focusing on the purpose, strength, view paradigm, depth cues, interaction model, and preserved invariances of each technique. We then go on to classify 25 different methods that have been described previously in the literature into the taxonomy. These classifications form a body of data that we can analyze for trends and the existence of clusters; this analysis yields in turn five orthogonal *design patterns* that characterize current work in the field. The patterns are multiple viewports, virtual X-ray tools, tour planners, interactive exploders, and projection distorters, and we describe the typical uses and characteristics of each pattern. More importantly, the pattern identification process also serves to pinpoint the "gaps" in the taxonomy, i.e. as-of-yet undeveloped techniques that could potentially fulfill a useful role in future research.

The purpose of this taxonomy is manifold: (i) to provide a common theoretical framework and vocabulary for occlusion management techniques, giving researchers and practitioners alike a common ground for discussion; (ii) to facilitate qualitative comparison, evaluation and maybe even benchmarking of different methods for occlusion management; (iii) to suggest a small number of archetypes of design suitable as starting points for implementations and prototypes; and (iv) to inform future directions of research within occlusion management and human perception of 3D space.

This paper is organized as follows: We begin by discussing previous work on this subject. We then describe the problem space of occlusion in 3D environments, where the fact that nearby objects occlude more distant ones work against the human perceptual system. This is followed by a presentation of our taxonomy and its dimensions. We also present the full classification of the 25 techniques we have studied in this paper. We then identify and describe the five design patterns, followed by suggestions on future research directions based on unexplored parts of the taxonomy. We finish the paper with some discussions on how to improve and extend the taxonomy and our conclusions on the work.

2 RELATED WORK

No previous taxonomy exists in the literature on the class of occlusion management interaction techniques. More general taxonomies on 3D interaction tend to describe low-level mechanics of manipulative tasks in a morphological fashion, whereas our focus is more on high-level aspects of perceptual tasks related to spatial understanding of the 3D environment. For example, Bowman and Hodges [7] present a general formal framework for 3D interaction in immersive virtual environments (IVEs) based around three tasks: motion control, selection, and manipulation. Bier et al. give a taxonomy of see-through tools [6] for a class of double-handed interaction techniques using transparent sheets called toolglasses that partly inspired this taxonomy. Bowman et al. [8] present a descriptive view of the design space of information display as well as interaction for information visualization within virtual environments.

Although unrelated to the occlusion management area defined here, Pousman and Stasko's taxonomy of ambient visualization [26] inspired the method employed in this paper for deriving the design patterns from the classification data.

*e-mail: elm@cs.chalmers.se
†e-mail: tsigas@cs.chalmers.se

We use our taxonomy as a tool for classifying existing techniques and thus validating its generality, but also as a design space. This allows us to identify holes in the taxonomy, akin to [9].

3 Problem Space

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. The environment and its geometrical properties interact 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) represented by boundary surfaces (typically triangles). The user's viewpoint $v = (M, P)$ is represented by a view matrix M that includes the position and orientation of the user, as well as a projection matrix P that includes view parameters such as viewport dimensions, focal length, far and near clipping plane, etc.

A line segment r is *blocked* by an object o if it intersects any part of o. An object o is said to be *occluded* from a viewpoint v if there exists no line segment r between v and o such that r is not blocked. Analogously, an object o is said to be *visible* from a viewpoint v if there exists a line segment r between v and o such that r is not blocked. An object o is said to be *partially occluded* from viewpoint v if o is visible, but there exists a line segment r between v and o such that r is blocked.

An object can be flagged either as a *target*, an information-carrying entity, or a *distractor*, an object with no intrinsic information value. Importance flags can be dynamically changed. Occluded distractors pose no threat to any analysis tasks performed in the environment, whereas partially or fully occluded targets do, potentially causing decreased performance and correctness.

A set of viewpoints V is said to be *complete* if there exists no object that is occluded in all of the viewpoints v_i.

It is possible to introduce a temporal dimension to this model and discuss concepts like transient occlusion and invariant occlusion. We will ignore this aspect in this treatment, however, and consider only temporally invariant situations. Some of the solutions we discuss will still be applicable to dynamic situations.

3.2 Visual Tasks

The occlusion problem typically occurs in the following three *visual tasks*:

- *object discovery* – finding all targets $t \in O$ in the environment;
- *object access* – retrieving graphically encoded information associated with each target; and
- *spatial relation* – relating the spatial location and orientation of a target with its context.

Other visual tasks that are of relevance include object *creation*, *deletion* and *modification*; in this treatment, however, we consider these to be special cases of discovery and access with regards to inter-object occlusion, and consisting of the same subtasks as these three basic visual tasks.

3.3 Analysis

We can observe that all visual tasks are severely hampered by the existence of fully occluded objects. More specifically, for the purposes of object discovery, a fully occluded object will be impossible to discover without the use of some occlusion management strategy, and identifying whether the object is a target never becomes an issue. Analogously for object access, the visual search will fail, and so will the perception of the object's visual properties. 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, on the other hand, will succeed in the visual search, although the perception of the object may still fail due to important parts of it being occluded.

Spatial relation, necessary for many complex interactions and visualizations, requires overview of the whole world, and is thus severely affected by both partially and fully occluded objects.

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 management strategies are applicable for a specific situation.

In this treatment, we identify three main geometrical properties of the environment that interact to cause inter-object occlusion and influence the three basic visual tasks associated with the environment:

- *object interaction* – spatial interaction of objects in the environment;
- *object density* – amount of objects in the environment with regard to its size; and
- *object complexity* – detail level of individual objects in the environment.

Obviously, these are high-level properties that only generally describe an environment without going into detail on its actual content. Nevertheless, in the following sections we shall see how these property dimensions can serve as powerful reasoning tools for describing a 3D environment and selecting a suitable solution strategy for it.

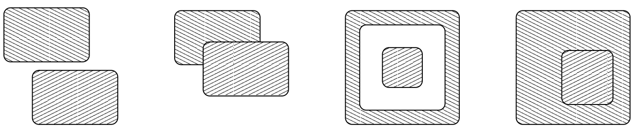
3.4.1 Object Interaction

The object interaction property dimension describes how the individual objects in the environment interact spatially with each other, i.e. whether they touch, intersect or merely reside close to each other. There are five ordinal levels to this parameter (see Figure 1 for a visual overview):

- *none* – no spatial interaction between objects (realistically only applicable for singleton objects);
- *proximity* – objects are placed in such close proximity (without intersecting) that they occlude each other from some viewpoint;

- *intersection* – objects intersect in 3D space (without one fully containing another) such that they occlude each other;
- *enclosement* – one or several objects combine to fully enclose objects (without containing them) such that they are occluded from any viewpoint external to the enclosing objects; and
- *containment* – objects are fully contained in other objects such that they are occluded from any viewpoint.

Examples of these interaction levels exist in all kinds of 3D visualizations: proximity for nodes in 3D node-link diagrams, intersection for visualization of constructive solid geometry (CSG), enclosement for 3D objects placed inside larger objects (i.e. the walls of a virtual house), containment for 3D medical CAT scan data, etc.



(a) proximity (b) intersection (c) enclosement (d) containment

Figure 1: Object interactions that may cause occlusion in 3D environments.

3.4.2 Object Density

The object density is a measure of the number of objects inhabiting the 3D environment; it follows naturally that the more objects per volume unit we are dealing with, the greater the chance and impact of occlusion will be. For environments containing a singleton object, naturally only self-occlusion can occur.

3.4.3 Object Complexity

The third geometrical property with an impact on the occlusion characteristics of an environment is the complexity of the objects in the environment. With complexity, we refer to the detail level of the 3D objects, i.e. typically the number of triangles (or other 3D primitives, such as quads, lines, and points) that make up the object, but we also include attributes such as color, material, and texture in this parameter. It follows that the more complex an object is, the more information it can potentially encode, and the larger the impact occlusion has on identification and perception of the object.

For simplicity, we can often reduce object complexity by splitting objects into smaller (preferably convex) subobjects. Note that this will often result in an increased object interaction and density index. The same mechanism can be used to handle self-occlusion, i.e. when an object occludes parts of itself.

4 Design Space

We characterize the design space of occlusion management techniques using the following primary dimensions:

- **Primary Purpose.** Visual task that the technique is primarily targeting. [discovery, access, relation]
- **Disambiguation Strength.** Maximum object interaction that the technique can handle. [proximity, intersection, enclosement, containment]
- **Depth Cues.** Strength of depth disambiguation cues for the technique. [low, somewhat low, medium, somewhat high, high]
- **View Paradigm.** View method used for the technique, i.e. the arrangement and layout of the visual substrate. [single view, twin separate views, twin integrated views, multiple separate views, multiple integrated views]
- **Interaction Model.** Operational model of user interaction for the technique. [passive, hybrid, active]
- **Target Invariances.** Degree of target invariances preserved using the technique. [0–3 aspects: location, geometry, appearance]

These six dimensions have been identified to be orthogonal, objective, and capture the full expressivity of the design space of these kinds of techniques. In the following sections, we will describe the dimensions in greater detail.

4.1 Primary Purpose

The purpose of an occlusion management technique describes which particular visual task in the problem space that the technique is primarily targeting (see Section 3 for the visual tasks). In other words, this dimension can assume any of the values discovery, access, or spatial relation.

More specifically, an interaction technique designed mainly for discovery focuses on making the user aware of the existence of partially or completely occluded targets, not necessarily making retrieval or relation of information from the objects easier.

A technique designed for access, on the other hand, aims not only to make users aware of an occluded object, but also to allow the user to retrieve the information encoded in the object.

Finally, a technique supporting spatial relation is designed to make not only the object itself but also its surrounding context visible and understandable to the user. This means that it is not possible to simply get rid of the neighboring objects in the interest of seeing the target, since these may carry important information needed to understand the scene (such as the connectivity of a node-link diagram).

Note that a technique may have more than one purpose; this taxonomy dimension captures the **primary** purpose of the technique.

Domain: *discovery, access, spatial relation* (nominal)

Characteristic Techniques:

- *discovery:* image-space dynamic transparency [4]
- *access:* interactive cut-away and break-away views [13], 3D explosion probe [30]
- *spatial relation:* tumbler [27], way-finder [3]

4.2 Disambiguation Strength

Disambiguation strength refers directly to the maximum degree of object interaction that the technique can handle and still fulfill its primary purpose. In other words, this is a measure of how complex object interactions the technique can manage using the terminology from the problem space (see Section 3.4.1). Note that this metric is unrelated to object density, but that very high object density can confound the situation.

The strength of a technique is an ordinal dimension, and it is generally perceived better for a technique to be able to handle high object interaction. On the other hand, strength is related to other factors of the design space, leading to a trade-off between them. For example, virtual X-ray techniques (see Section 5) typically support the highest object interaction (containment), yet are not as scalable as other techniques with more modest strengths.

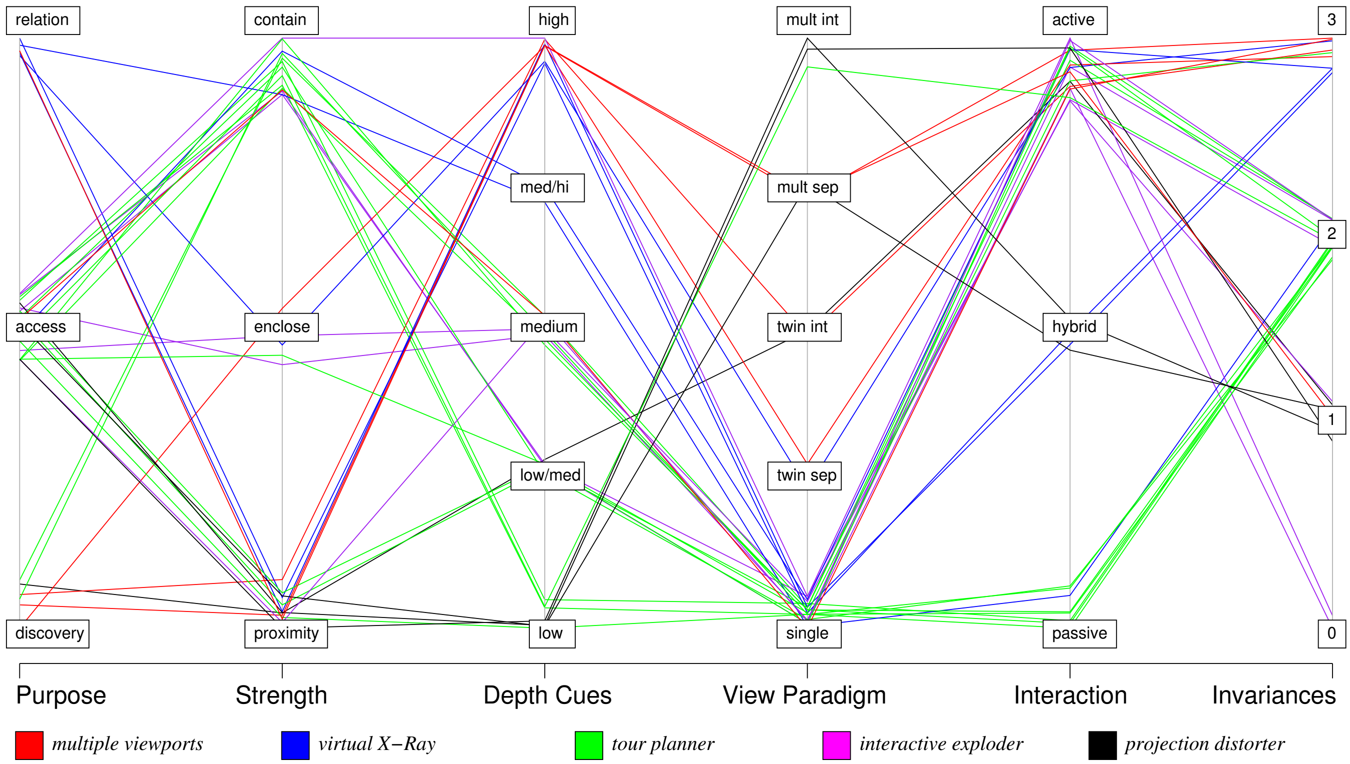


Figure 2: Classification of 25 different occlusion management techniques using the taxonomy (points have been jittered to show distribution).

Domain: *proximity, intersection, enclosement, containment* (ordinal)

Characteristic Techniques:

- *proximity:* none
- *intersection:* view projection morphing [15], worlds-in-miniature [16], bird's eye views [17]
- *enclosement:* worldlets [16], BalloonProbe [14], 3D explosion probe [30]
- *containment:* image-space dynamic transparency [4], importance-driven volume rendering [32], view-dependent transparency [12]

4.3 Depth Cues

As we hinted at earlier in this paper, actually relaxing some of the visual cues humans rely on for spatial perception will most certainly have a negative impact on the user's understanding of his or her surroundings, regardless of any advantages gained from doing this. The perception of depth, i.e. the actual 3D component of our vision system, is most vulnerable to this effect, and thus we define a dimension that captures the degree of depth cues that a technique provides.

Depth cues is an ordinal dimension with a five-value scale ranging from low to high, signifying the amount of depth cues retained by the technique; high would mean that in principle all depth cues are preserved, whereas low means that practically none are.

There are additional visual cues that help humans perceive their environment and that play a role in the classification of occlusion management techniques, some of which we capture in the "target invariances" dimension below.

Domain: *low, somewhat low, medium, somewhat high, high* (ordinal)

Characteristic Techniques:

- *low:* artistic multiprojection [1]
- *somewhat low:* 2D dynamic transparency [19], free-space transparency [21]
- *medium:* BalloonProbe [14], blueprints [25]
- *somewhat high:* image-based exploded view diagrams [22]
- *high:* tumbler [27], worldlets [16]

4.4 View Paradigm

Different occlusion management techniques utilize the view and the view space in different ways; this dimension captures the paradigm employed for managing the visual substrate. Typically, interaction techniques are either based on a single view, twin views, or a large number of views (multiple); similarly, for the case when there are additional views beyond the main one, they may either be separate windows in an overview+detail approach, or integrated in the same image in a focus+context [18] way. The view paradigm dimension is used to classify techniques according to a combination of these two metrics.

The degree of integration can sometimes be tricky to assess—for example, in the case of the worlds-in-miniature (WIM) [31] technique, there is a very obvious second view, i.e. a miniature version of the world, yet since it is a first-class object in the environment, we classify it as being integrated. For bird's eye views [17], on the other hand, the secondary view is in a separate window, and is thus classified as having twin separate views. This factor is also the reason why separating the number of views from their integration is difficult; in the case of Singh's multiprojection techniques [28], the single view actually consists of multiple different cameras, non-linearly combined into one.

Domain: *single view, twin separate views, twin integrated, multiple separate, multiple integrated* (nominal)

Characteristic Techniques:

- *single view:* 3D explosion probe [30]
- *twin separate views:* bird's eye views [17]
- *twin integrated views:* worlds-in-miniature [31], view projection animation [15]
- *multiple separate views:* worldlets [16]
- *multiple integrated views:* virtual multiprojection cameras [28], looking glass (multi-user) [23]

4.5 Interaction Model

We are also interested in capturing the specific interaction model employed by each technique; some rely on active user intervention, exposing occluded content according to direct (free-space transparency [21]) or indirect (temporally controlled non-linear projections [29]) input from the user, whereas others employ a passive interaction approach to present the hidden objects directly to the user with no input necessary (dynamic transparency [4]). A third possible option is a hybrid model, where the technique has two or more distinct modes during which the interaction model changes between active and passive. An example of the latter is the way-finder [3] system, which first calculates a path through the world in an off-line phase, then allows the user to interactively explore the path.

One interesting effect of a passive mode is that a technique employing such an interaction model typically must have prior semantic knowledge about the targets the user considers important (and sometimes even an interest value for each target); an example is the importance-driven volume rendering technique by Viola et al. [32]. Active mode, on the other hand, puts these decisions in the hands of the user, providing for more flexible interaction.

Domain: *passive, hybrid, active* (nominal)

Characteristic Techniques:

- *passive:* image-space dynamic transparency [4], multiblending [5]
- *hybrid:* way-finder [3], path drawing for 3D walkthrough [20]
- *active:* perspective cutouts [11], SDM [10]

4.6 Target Invariances

The sixth and final primary dimension of our taxonomy describes the number of invariances preserved by the technique. A complement to the depth cues parameter above, target invariances describes how many of the following properties of the **targets** (not necessarily distractors) in the environment are retained:

- **Location.** Position and orientation of the target.
- **Geometry.** Shape and size of the target.
- **Appearance.** Color, texture and material of the target.

All of the above properties are all more or less important for visualization applications in 3D environments; for instance, for a simple 3D scatterplot, the location of each data point is vital for the data to be interpreted correctly, so an occlusion management technique designed for use with such data should definitely preserve this property. For color-coded tree hierarchies, such as for a 3D representation of a file system, it might make sense to displace location (as long as connectivity information is retained) but the appearance should not be altered.

The higher number of invariances a technique retains, the better it is, and so this is an ordinal dimension. However, as discussed in the introduction, our normal visual cues are often at an odds with understanding various properties of an environment (e.g. seeing all the targets despite occlusion), and thus this is an example of a classical trade-off decision specific to each technique. Often, designers can gain certain attractive properties by relaxing others, all depending on the particular application area of the technique.

Domain: *0-3: location, geometry, appearance* (ordinal)

Characteristic Techniques:

- *location:* preserve: interactive cut-away and break-away views [13]; discard: 3D explosion probe [30]
- *geometry:* preserve: worlds-in-miniature [31]; discard: non-linear projection [1, 29]
- *appearance:* preserve: BalloonProbe [14]; discard: view-dependent transparency [12]

5 Design Patterns

We have classified the 25 techniques involved in our survey using our taxonomy; the result can be summarized in the parallel coordinate plot in Figure 2. We then study this body of classifications to see patterns and trends, using for instance hierarchical clustering mechanisms. This analysis yields five distinct and orthogonal archetypes of design, or *design patterns* [2], i.e. a generic and reusable solution to a commonly occurring problem within a specific context. The five patterns we have identified we call Multiple Viewports, Virtual X-Ray, Tour Planner, Interactive Exploder, and Projection Distorter. We will describe these in the following sections.

According to pattern lore, a design pattern has four essential elements: a name, a problem (already given), a solution, and the consequences of using the pattern. We use these elements in our discussion of each pattern. We also show the distribution of techniques implementing the pattern on the design space.

5.1 Multiple Viewports

The Multiple Viewports pattern (red in Figure 2) is characterized by a view paradigm based on two or more separate views, resulting in an overview+detail kind of layout. Instances of this pattern also tend to preserve most, if not all, invariances—the trick lies in the placement of the additional cameras, not manipulating the image seen from them. It is most effective for 3D environments that lend themselves to overviews, such as landscapes and structured buildings. Furthermore, the interaction model tends to be active; no existing technique performs the automatic placement of cameras that would be necessary for passive interaction.

Solution: Manage discovery and access of targets by providing several alternate (often separate) viewports of the 3D environment. Typically, one viewport is designated as the main viewport, with the other viewports as secondary and generally smaller. Accordingly, the main viewport is often used for detail or first-person views, whereas the alternate views give either static or dynamic overviews of the environment (such as an overhead map).

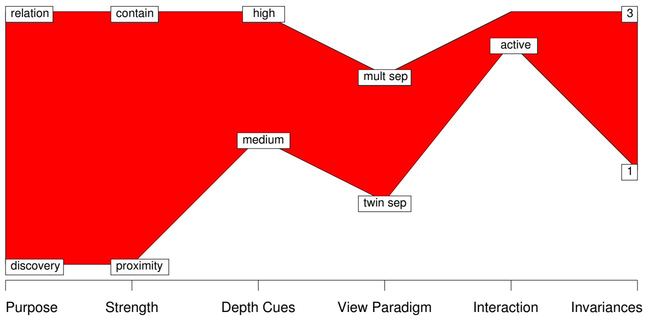


Figure 3: Classification distribution of multiple viewports techniques.

Consequences: The use of the Multiple Viewports pattern trades screen estate and user attention for increased discovery and access; the user will have a smaller main visualization window than otherwise, and may have to split his or her attention across all of the viewports. Furthermore, in some situations, it is not clear what constitutes an overview, and thus introducing additional viewports may have diminishing returns. However, this is a very powerful approach for suitable environments.

Examples: Tumbler [27], worlds-in-miniature [31], worldlets [16], bird's eye views [17].

5.2 Virtual X-Ray

The Virtual X-Ray pattern (green in Figure 2) is based on an image-space approach where occlusion can be easily detected and sometimes even delegated to programmable fragment shaders. The pattern is not limited to 3D—the same idea permeates dynamic transparency techniques for 2D windowing systems, such as the free-space transparency [21] and multiblending [5] techniques. Typically, example techniques have very high disambiguation strength. Furthermore, there is a clear division between two types of Virtual X-Ray techniques; active ones, where the user controls a "searchlight" on the 2D view, and passive ones, where semantic information allows the system to automatically uncover targets.

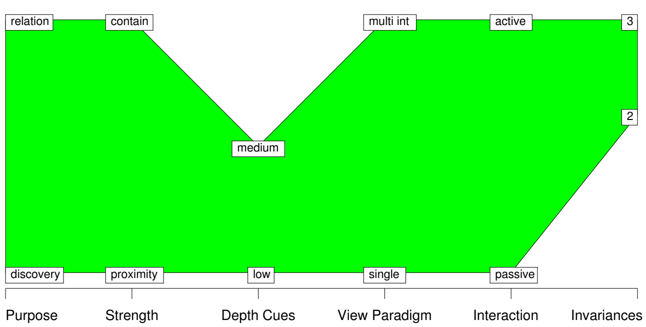


Figure 4: Classification distribution of virtual X-Ray techniques.

Solution: Make targets visible through intervening distractors by turning occluding surfaces invisible or semi-transparent. The method for distractor removal is characteristic: some techniques are view-dependent (breakaway) whereas others are static (cutaway); some eliminate distractors (or parts of distractors), others merely make distractors semi-transparent. Active interaction facilitates exploration whereas passive interaction requires target information but yields a potentially higher correctness.

Consequences: The Virtual X-Ray pattern makes discovery trivial and facilitates access by selectively removing distractors occluding the targets. However, this is a direct weakening of occlusion depth cues, causing a decrease in depth perception and making spatial relation more difficult. The use of semi-transparency also results in high visual complexity and imposes a high cognitive load on the user. Finally, Virtual X-Ray can make visibility computations for rendering optimization useless.

Examples: Perspective cutouts (active) [11], image-space dynamic transparency (passive) [4].

5.3 Tour Planner

The family of Tour Planner techniques (blue in Figure 2) is characterized by a hybrid interaction model consisting of an offline and an online phase where first the path is defined or computed and then interactively shown in the environment itself. Typically no distortion is imposed on the view (a temporal canvas is used), so all invariances are usually retained.

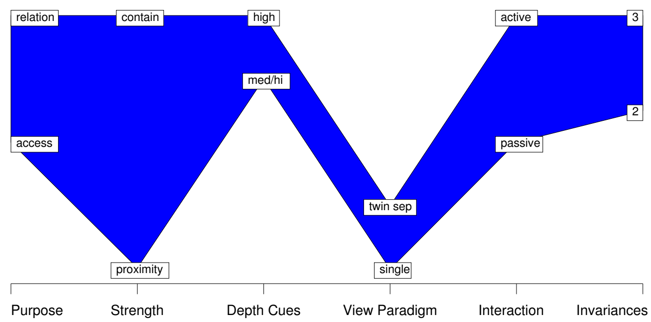


Figure 5: Classification distribution of tour planner techniques.

Solution: Present all targets in an environment by constructing a complete (i.e. all targets are visible in at least one point) path through it. It should also conform to a number of additional constraints (such as short or optimal length, closed, uniform visual complexity, etc). Often realized in an offline precomputation or specification step followed by an interactive exploration phase where the user is guided by the computed path.

Consequences: The Tour Planner pattern is non-invasive and thus will not modify the environment itself and will typically retain all invariances. This however means that the pattern's disambiguation strength is generally low. The path computation step can sometimes be costly in terms of computation time, and intractable to dynamically changing situations.

Examples: Way-finder [3].

5.4 Interactive Exploder

Interactive Exploders (purple in Figure 2) manage occlusion in the object space through active user interaction in a direct manipulation approach. The exploding metaphor means that target location is rarely retained, although most other invariances typically are.

Solution: Provide a user-controlled distortion probe that locally displaces objects to manage occlusion. The approach is based either on (i) removing distractors or (ii) separating targets; in the former case, we want to eliminate objects that get in the way, whereas in the latter, we instead want to disambiguate between several targets who share the same space. The effect is similar to that of exploding diagrams used for technical illustration, but

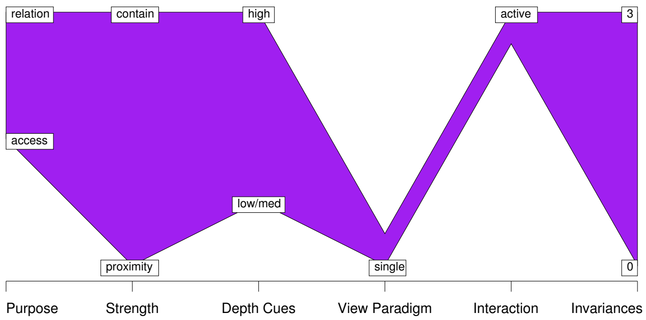


Figure 6: Classification distribution of exploder techniques.

here the interaction is active and under direct user control in the object space. Implementations often provide some kind of visual cue of the original positions of displaced objects, typically using wireframe or ghosting (transparency).

Consequences: Using an Interactive Exploder can help disambiguate even very difficult situations, but the very nature of the pattern means that at the very least location is not preserved. The pattern is best suited for discovery. The local influence model means that there may be a problem of reach in a virtual environment.

Examples: 3D explosion probe [30], deformation-based volume explosion [24].

5.5 Projection Distorter

This pattern (black in Figure 2) is signified by a view-space approach presented using two or more integrated views. Since non-linear projections are typically employed to pack as many of the targets as possible into a single view, few invariances are retained. Thus, this pattern is often best used for discovery, rarely for access, and almost never for relation.

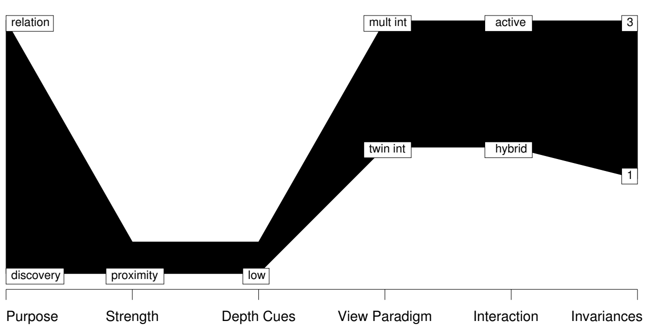


Figure 7: Classification distribution of projection distorter techniques.

Solution: Integrate several different views of targets into a single view in order to maximize discovery. The solution is then often reminiscent of a focus+context technique with one focus per view. Individual view selection is often actively controlled by the user in an online or offline manner. In one case, a hybrid approach is employed where target semantic information is extracted from previous user explorations using data mining techniques and then used to inform the technique [29].

Consequences: The use of the Projection Distorter pattern affects only the view projection code of an application and is thus relatively easy to integrate into existing code. On the other hand, the resulting visual displays can often become disconcerting and disorienting to the user. Few object properties are retained.

Examples: Artistic multiprojection [1], view projection animation [15].

6 Future Research Directions

Besides identifying existing design archetypes in the literature, we can also extract possible future research opportunities from our taxonomy by studying the as-of-yet unexplored parts of the design space. One such observation is the need for techniques that make users aware of occluded content without compromising visual quality and imposing a high cognitive load on the user. Retaining a high degree of depth cues is important for complex visual tasks such as spatial relation. Another interesting area to explore is hybrid-interaction methods where the user's own actions are used to inform the target selection. This approach may help solve the trade-off between the precision that a passive interaction model provides as opposed to the more general nature of active user interaction.

Combinations of patterns could be profitable ways of utilizing the strong points of two different methods while at the same time making up for the weak ones. For example, a multiple-viewport technique could be augmented with virtual X-Ray support in one or several of the views. A tour planner could be paired with an interactive exploder to help disambiguate in difficult situations of locally high target congestion.

General 3D navigation has been shown to be a task with a very high cognitive load; for every traveled world unit, the user runs the risk of becoming disoriented, totally lost, or even nauseous. For the longer term, it can be noted that the ultimate goal of occlusion management techniques should be to help minimize the need for 3D navigation in general. Perhaps the class of interaction techniques described in this paper can help short-circuit excessive navigation in the first place.

7 Discussion

The taxonomy presented in this paper has been designed to be orthogonal and objective, with no dimension being reducible to another and having a minimum of coupling to the other dimensions. Regardless, it is always possible to debate the inclusion or exclusion of specific property dimensions to a taxonomy. We believe this to be a valid one, and the successful classification of 25 different techniques using it confirms this claim.

Nevertheless, property dimensions that were excluded for various reasons include scalability (the amount of object density the technique can handle), influence level (i.e. whether the interaction technique operates on a local, regional, or global level), and dimensionality (2D, 2.5D, 3D, etc). Classification using these and other dimensions is left as an exercise to the reader.

Despite the lofty goal mentioned in the previous section of bypassing the need for navigation, a number of 3D navigation techniques were indeed included in the classification in this paper. These were selected due to them being on the borderline of what constitutes an occlusion management technique, or representative for a specific class of techniques. Many other 3D navigation techniques in the literature were excluded from this classification; the line had to be drawn somewhere.

Some notable occlusion management techniques—such as cutting planes, filtering, and spatial indices—have been left out of this taxonomy. They can easily be added without too much trouble.

An interesting observation on the design patterns identified in this paper is that the separating feature between most patterns is the approach taken to visualize occlusion (the canvas used, if you will). For instance, virtual X-Ray techniques use image space, projection distorters use view space, interactive exploders use object space, and tour planners utilize temporal space.

8 Conclusions

Occlusion management is a subset of 3D interaction techniques concerned with improving human perception for specialized visual tasks through manipulation of visual cues such as occlusion, size, and shape. In this paper, we have presented five archetypical design patterns for occlusion management based on a classification of existing interaction techniques. The patterns include multiple viewports, virtual X-ray, tour planners, interactive exploders, and projection distorters. The underlying taxonomy used for this classification is based on six characteristic properties of occlusion management techniques. Analysis of this taxonomy also yields additional missing patterns, such as primarily techniques for target awareness and hybrid-interaction approaches with an emphasis on retaining a high degree of depth cues and supporting spatial relation.

References

[1] Maneesh Agrawala, Denis Zorin, and Tamara Munzner. Artistic multiprojection rendering. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 125–136, 2000.

[2] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*. Oxford University Press, 1977.

[3] Carlos Andújar, Pere-Pau Vázquez, and Marta Fairén. Way-finder: guided tours through complex walkthrough models. In *Proceedings of EUROGRAPHICS 2004*, pages 499–508, 2004.

[4] Ulf Assarsson, Niklas Elmqvist, and Philippas Tsigas. Image-space dynamic transparency for improved 3D object discovery. Technical Report 2006:10, Chalmers University of Technology, 2006.

[5] Patrick Baudisch and Carl Gutwin. Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. In *Proceedings of the ACM CHI 2004 Conference on Human Factors in Computing Systems*, pages 367–374, 2004.

[6] Eric Bier, Maureen Stone, Ken Fishkin, William Buxton, and Thomas Baudel. A taxonomy of see-through tools. In *Proceedings of the ACM CHI'94 Conference on Human Factors in Computing Systems*, pages 358–364, 1994.

[7] Doug A. Bowman and Larry F. Hodges. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages and Computing*, 10(1):37–53, 1999.

[8] Doug A. Bowman, Chris North, Jian Chen, Nicholas F. Polys, Pardha S. Pyla, and Umur Yilmaz. Information-rich virtual environments: theory, tools, and research agenda. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology 2003*, pages 81–90, 2003.

[9] Stuart Card, Jock Mackinlay, and George Robertson. The design space of input devices. In *Proceedings of the ACM CHI'90 Conference on Human Factors in Computing Systems*, pages 117–124, 1990.

[10] Mei C. Chuah, Steven F. Roth, Joe Mattis, and John Kolojejchick. SDM: Selective dynamic manipulation of visualizations. In *Proceedings of the ACM Symposium on User Interface Software and Technology 1995*, pages 61–70, 1995.

[11] Chris Coffin and Tobias Höllerer. Interactive perspective cut-away views for general 3D scenes. In *Proceedings of the IEEE Symposium on 3D User Interfaces 2006*, pages 25–28, 2006.

[12] Joachim Diepstraten, Daniel Weiskopf, and Thomas Ertl. Transparency in interactive technical illustrations. *Computer Graphics Forum*, 21(3):317–325, 2002.

[13] Joachim Diepstraten, Daniel Weiskopf, and Thomas Ertl. Interactive cutaway rendering. In *Proceedings of EUROGRAPHICS 2003*, pages 523–532, 2003.

[14] Niklas Elmqvist. BalloonProbe: Reducing occlusion in 3D using interactive space distortion. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology 2005*, pages 134–137, 2005.

[15] Niklas Elmqvist and Philippas Tsigas. View projection animation for occlusion reduction. In *Proceedings of the ACM Conference on Advanced Visual Interfaces 2006*, pages 471–475, 2006.

[16] T. Todd Elvins, David R. Nadeau, and David Kirsh. Worldlets – 3D thumbnails for wayfinding in virtual environments. In *Proceedings of the ACM Symposium on User Interface Software and Technology 1997*, pages 21–30, 1997.

[17] Shinji Fukatsu, Yoshifumi Kitamura, Toshihiro Masaki, and Fumio 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 1998*, pages 67–76, 1998.

[18] George W. Furnas. Generalized fisheye views. In *Proceedings of the ACM CHI'86 Conference on Human Factors in Computer Systems*, pages 16–23, 1986.

[19] Carl Gutwin, Jeff Dyck, and Chris Fedak. The effects of dynamic transparency on targeting performance. In *Proceedings of Graphics Interface 2003*, pages 105–112, 2003.

[20] Takeo Igarashi, Rieko Kadobayashi, Kenji Mase, and Hidehiko Tanaka. Path drawing for 3D walkthrough. In *Proceedings of the ACM Symposium on User Interface Software and Technology 1998*, pages 173–174, 1998.

[21] Edward W. Ishak and Steven K. Feiner. Interacting with hidden content using content-aware free-space transparency. In *Proceedings of the ACM Symposium on User Interface Software and Technology 2004*, pages 189–192, 2004.

[22] Wilmot Li, Maneesh Agrawala, and David Salesin. Interactive image-based exploded view diagrams. In *Proceedings of Graphics Interface 2004*, pages 203–212, 2004.

[23] Julian Looser, Mark Billinghurst, and Andy Cockburn. Through the looking glass: the use of lenses as an interface tool for augmented reality interfaces. In *Proceedings of GRAPHITE 2004*, pages 204–211, 2004.

[24] Michael J. McGuffin, Liviu Tancau, and Ravin Balakrishnan. Using deformations for browsing volumetric data. In *Proceedings of the IEEE Conference on Visualization 2003*, pages 401–408, 2003.

[25] Marc Nienhaus and Jürgen Döllner. Blueprints: Illustrating architecture and technical parts using hardware-accelerated non-photorealistic rendering. In *Proceedings of Graphics Interface 2004*, pages 49–56, 2004.

[26] Zachary Pousman and John Stasko. A taxonomy of ambient information systems: Four patterns of design. In *Proceedings of the ACM Conference on Advanced Visual Interfaces 2006*, pages 67–74, 2006.

[27] Gonzalo Ramos, George Robertson, Mary Czerwinski, Desney Tan, Patrick Baudisch, Ken Hinckley, and Maneesh Agrawala. Tumble! Splat! Helping users access and manipulate occluded content in 2D drawings. In *Proceedings of the ACM Conference on Advanced Visual Interfaces 2006*, pages 428–435, 2006.

[28] Karan Singh. A fresh perspective. In *Proceedings of Graphics Interface 2002*, pages 17–24, 2002.

[29] Karan Singh and Ravin Balakrishnan. Visualizing 3D scenes using non-linear projections and data mining of previous camera movements. In *Proceedings of AFRIGRAPH 2004*, pages 41–48, 2004.

[30] Henry Sonnet, M. Sheelagh T. Carpendale, and Thomas Strothotte. Integrating expanding annotations with a 3D explosion probe. In *Proceedings of the ACM Conference on Advanced Visual Interfaces 2004*, pages 63–70, 2004.

[31] Richard Stoakley, Matthew J. Conway, and Pausch 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.

[32] Ivan Viola, Armin Kanitsar, and Eduard Gröller. Importance-driven volume rendering. In *Proceedings of the IEEE Conference on Visualization 2004*, pages 139–145, 2004.