Space-Scale Animation: Enhancing Cross-Scale Understanding of Multiscale Structures in Multiple Views

Xiaolong Zhang
University of Arizona
xiaolong@u.arizona.edu

Abstract

Large information or model structures often span many scale levels, and exhibit important features at each scale. Having a coherent and cross-scale understanding of such multiscale structures often requires users to interact with the structures at different scales. While multiple views allow users to see the structures from different locations and at different scales, establishing cross-scale connections between structures in divergent multiple views could be a challenge. This paper proposes a new interactive design, space-scale animation, to visualize the spatial and semantic relationship between structures in different views, and discusses the design and implementation of space-scale animation as a dynamic view transition traversing space and scale. This research not only extends interactive animation techniques by explicitly considering the scale factor, but also argues the necessity to integrate cross-scale semantic information into animation to improve the understanding of complex structures.

Keywords--- **Multiscale, Space-Scale, Animation.**

1. Introduction

The size of information or model structures is increasing. Structures like digital libraries and biological models can easily have thousands or even millions of components, exhibiting different characteristics at different levels. Interacting with such large structures, users often need to examine the structures from various perspectives and at different scales. Tools like multiple views can be helpful.

Multiple views allow users to see different parts of a structure simultaneously, or to build common ground for collaboration [8]. Multiple views usually appear as different windows (Figure 1), or as a view-in-view (Figure 2). With multiple views, users can easily see and interact with objects or structures of interests. However, understanding the relationship between objects in large structures from different views could be a challenge. For example, multiple views in Figure 1 give users two sub-structures of a large biological structure, but users may find it difficult to tie these two structures together simply based on these two static views. Similarly, two views in Figure 2 show geographic information of two places, but they do not tell users how these two places may be related.

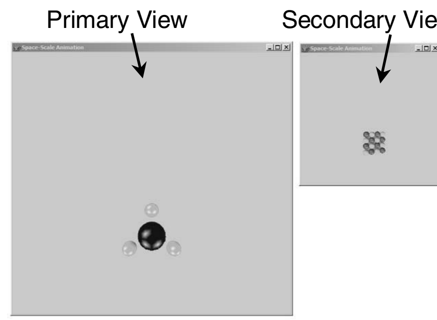


Figure 1: Multiple views to show two different sub-structures in a large biological structure.

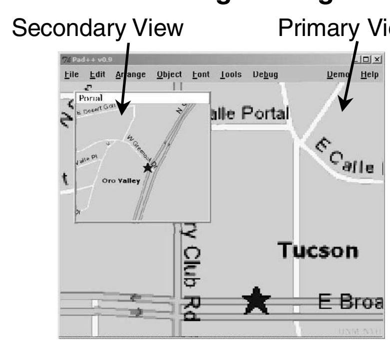


Figure 2: A view-in-view to show two cities. (Maps provided by Mapquest.com)

Usually, users can rely on navigation tools to move around to find out how two views are related. The challenge of this approach is that users may not know to which direction they should navigate and how far they should go. Furthermore, for large structures, two views may be at different scale levels and present scale-dependent characteristics. Such cross-scale representation makes it even harder for users to navigate. For example, observing a protein structure, users may see folded molecules in one view, and atomic lattices in another. To move between these two views, in addition to navigation

Proceedings of the Third International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV'05)

direction and distance, users also need to know which scale they should go. Navigating through both scale and space is a challenge [20].

Users need tools to better understand the cross-scale relationship between structures in different views without being swamped by complicated navigation challenges. Animation has been used in information visualization to help users see how different objects or structures are related. However, traditional interactive animation design largely focuses on view transition in space or time, and scale difference between views and resulting different semantic representations are not often considered. In this paper, we propose a new interactive animation technique, space-scale animation, which concerns cross-scale semantic relationship between structures of interest in different views. This space-scale animation helps users see how structures in different views are related in space and in scale, and provides richer information to tie these structures in a larger semantic context.

This paper is structured as follows. Section two discusses multiscale structures, and a need for tools to facilitate the understanding of them. Section three reviews related research literature. Section four presents the design of scale-space animation. Following it is the description of the algorithm and the implementation of space-scale animation in both 2D and 3D environments. Section six discusses the implication of this research for information visualization and future research agenda.

2. Multiscale Structures

Objects in the physical world and virtual worlds appear differently at different scale levels, presenting different characteristics and patterns. Objects in the physical world can be observed as a cluster of galaxies at the level of 10^{26} meter, or as highly dynamic quarks at the level of 10^{-16} meter [24]. Similarly, structures in virtual worlds also exhibit multiscale characteristics. Large information resources, like digital libraries, are usually organized as hierarchies, which provide information with different levels of abstraction, to support information retrieval. Some structures like file systems use explicit hierarchies to classify documents, while others such as digital maps use implicit semantic structures to bond maps and related geographic information together. In these hierarchies, components are aggregated according to their attributes at different levels, so users can obtain necessary information with different levels of detail.

While multiscale structures span many scale levels, the interaction scale level of human beings is very limited. In the real world, our naturally born capabilities do not allow us to see atoms with our naked eyes, or to walk to a destination thousands of kilometers away easily. Compared with the broad scale range of objects in the physical world from galaxies to quarks, our human normal interaction scale range, which is about from millimeters to hundreds of meters, is very limited. It is a challenge for human beings to interact effectively with the physical world at all scale levels. Similarly, in information worlds, when users work on information structures, objects displayed on a computer screen should be large enough to be distinguishable due to the limited acuity of human eyes. Small screen real estates do not allow complete presentation of large information structures. Users can only see part of a large structure on a single screen. The mismatch in scale makes it difficult or even impossible for people to interact with multiscale objects in both real and virtual worlds.

To deal with multiscale structures, people usually focus on characteristics of objects and structures within certain levels rather than all levels at once. Limiting attention to a limited scale range allows people to break complicated phenomena into small components that are comprehensible and manageable by their limited cognition and interaction capabilities [1]. In selecting observation levels, people often set two scale limits, one upper and one lower. The upper limit, which is called extent, defines the maximum size that is observable, and the lower bound, called grain, defines the minimum size [1]. Limited cognition resources inevitably lead to a competition between a fine grain and a broad extent in the observation of multiscale structures.

Tools have been invented to extend people's scale range to observe and analyze multiscale structures. In the real world, tools like microscopes, telescopes, space shuttles, and satellites can be used to examine very small or very large objects and structures. The combination of these tools makes a multiscale system that greatly expands the scale range of people's visual perception. Similarly in the virtual worlds, multiscale tools, like zooming, are used to help users see information artifacts at different sizes. With multiscale tools, users can dynamically shift their observation across scale, and manipulate the structures at different levels. By helping users see more and reach further, multiscale tools become important to scientific research and information management, both of which are dealing with increasingly large data sets.

Working on large structures, users may find it a problem to get appropriate content and context information [12]. In traditional environments, the challenge may come from the wide spatial spread of components in a structure, and people need help to better see objects across space. In multiscale space, however, scale becomes another factor that may affect the understanding of structures. Users may fail to obtain appropriate content and context information simply because such information is at different scale levels.

The use of semantic representations in multiscale technology challenges users more in cross-scale understanding of structures. Semantic representations visualize multiscale structures with different representations at different scale levels to let users see different properties of the structures across scale. For example, with semantic representations, a model protein structure will be seen as tangled molecules at one scale level, and as atomic lattices at another. This implies that to provide users with different semantic information about a multiscale structure at different scale, the representation of the structure changes in a non-geometric way along scale. Users will see very different representations at different scales.

Interacting with a multiscale structure, users would need tools to construct a better and more coherent understanding about different representations of the same structure. Without proper help, even though users can see the contents of various representations at different places and scales, it might be difficult for users to see the context that connects them with the structure of interest.

Addressing this context and content problem in cross-scale multiple views could be even more important to collaboration work on multiscale structures. Some tasks in managing multiscale structures may demand more intensive labors, require faster response, or need broader knowledge and expertise across different scale levels. One user may not be capable of doing the whole task [42]. Then, cross-scale collaboration can help to divide their labors, to deal with tasks in parallel, and to combine different knowledge and expertise. One challenge in multiscale collaboration is to share information of interest across scales. Because users tend to be distributed widely in space and scale, their views may differ significantly. Simply sharing the views of others may not be sufficient for collaboration, because cross-scale connections between different views may not be available. Two views in Figure 1 and 2 can be regarded from a collaborative setting with two users. While a user can see what the other is seeing in the secondary view, it is still a challenge to know their collaboration context from these two static views.

What this research proposes is an animated view transition to help build the connection between different views in space and scale. Traditionally, animation design concerns the view point interpolation in space. In this research, we also consider the interpolation of scale, and focus on the presentation of cross-scale semantic connections between structures in animation.

3. Related Research

Relevant research literature fits into three categories: multiscale technology, navigation in large information space, and interactive animation. Multiscale is of interest here because it concerns the understanding and management of large structures. Navigation draws our attention because the difficulty in navigating in multiscale structures is one of primary reasons people need help from tools like space-scale animation. Research on interactive animation lays out technical and cognitive foundations for this research.

3.1. Multiscale Information Visualization

The content and context problem has been studied extensively. Furnas [12] proposed a theoretical framework of generalized fisheye views to balance the content and context in hierarchical structures. This technique has been used to visualize a wide variety of data, including documents [29], 2D spatial data [32], and 3D objects [28]. The degree of interest in observing hierarchical structures can also be facilitated by tools like fractal views [22], hyperbolic views [23], or larger 3D space [30,31,36]. For multiscale structures, this content and context problem is actually an issue concerning the choices of the scale level of interest. Content and context information is relative and dependent on the scale level at which a structure is presented and observed. To choose appropriate content and context is to set up appropriate grains and extents so that users can obtain balanced content and context information.

Multiscale technology is an approach to help people manage grains and extents. In multiscale user interfaces [27,5], users can use zooming tools to manipulate the amount of content and context information displayed on the screen. In addition to manipulating views, multiscale also gives users the capabilities to control their interaction domains and yoke their observation and action parameters together [42].

One technique in multiscale user interfaces is to provide users with semantic information of structures at different scales. This technique is implemented as semantic zooming in 2D multiscale user interfaces [27,5], and renders the same structure with different representations, providing users with appropriate information at different scale levels, such as different appearances [5], different levels of details [2], or information with different densities [40]. Each representation provides users with unique semantic information at a particular scale level, but interpreting the relationship between different representations would require cross-scale contexts.

One technique to help people understand the context of different structures is to provide users with a common view for structures of interest [35]. With a common view, users will see how these structures may be related to each other through some objects that are spatially relevant to these structures. For non-multiscale structures, it might be easy to find common objects as references. For multiscale structures with semantic representations, however, providing common references in a static view might be a challenge, because the structures of interest may be related to each other through other objects that are at different scale levels and may be invisible in a common view. A single static view is not sufficient to let users see such cross-scale context information.

3.2. Navigation in Large Virtual Space

Navigation tools, which allow users to move around and look around, seem to be a good way to help users understand how different views may relate to each other. It has been argued that multiscale navigation can benefit users by providing both context and content information [9,25,33], although the effectiveness of multiscale navigation could be affected by many factors, such as tasks and designs [18].

Navigating in multiscale environments, users need the capabilities to manipulate their observation scale, in addition to the control over view positions and orientations. With so many parameters to control, users may face many challenges in multiscale navigation. One of them is called "*desert fog*" [20], which refers to a situation in which users get no visual information for navigation guidance. The lack of visual guidance could be a problem in any environment. However, the scale factor in multiscale makes it more difficult. While in conventional environments, users can move around to search for visual information, in multiscale environments, users also need to consider which scale they should choose.

Furnas has argued that effective view navigation requires good residues for destinations, and regarded semantic representations as a design to provide effective navigation residues [13]. However, the effectiveness of semantic representations in guiding navigation is on the condition of the knowledge of different representations and their relationships. If users do not know how different structures are related and to what scale each representation belongs, they still have a problem to tell where to navigate.

One way to address this *"desert fog"* issue is to impose some constrains in navigation control, and prevent users from moving away from places with sufficient visual information for navigation guidance [20]. Imposing constraints in navigation is also seen in 3D space, where the 6 degrees of freedom can easily make users get lost. Designs like system-walking tools [16] put some limits on the navigation path to prevent users from being lost. Integrating these designs into a fully interactive system may cause problems sometimes, because users may not be aware of the restrictions on their navigation control, and feel confused when they find their interactive control does not produce expected results.

Teleportation in 3D is a tool to bring users from one place to another without traversing the space in between. It can reduce the chance of getting lost in navigation. However, teleportation only jumps from one view to another, and does not provide sufficient information about how these two views are related to each other spatially and semantically.

3.3. Browsing Large Structures with Animation

Animation has been regarded as an effective way to support user interaction. Users can benefit from animation both affectively and cognitively [7,30]. Research literature on animation is massive, and addresses a wide variety of issues, such as motion generation, keyframing, motion captures, object deformation and so on, to create smooth animation [17,26,39]. What is relevant to this research is the interactive design of animation, which focuses more on satisfying users' interests and needs in interaction rather than improving system performances in design.

Animation has been used to support the interaction with larger structures. It has been argued the object consistency in animation can help reduce the cognitive load in interacting with large and complicated structures [30]. It is also found that animation can help the construction of a conceptual understanding of information structures without affecting the task performances [3].

Research on animation design in multiscale user interfaces is not often seen. A few designs combined multiscale and animation in visualizing multiscale structures like hierarchies [6,11]. However, most of these designs only animated the dynamic shift between different spatial layouts of the same set of data at the same scale level. These techniques usually do not consider the semantic difference between structures at different scales. In a recent study, van Wijk and Nuij [41] proposed a model to improve the animation of zooming and panning in 2D multiscale user interfaces. However, this model primarily concerns the computational efficiency of animation.

Animation design in 3D usually focuses on the interpolation of view positions and view orientations. Recently, researchers began to integrate the interests and needs of users in animation design. Animated view in flying has been tied with users' travel speed, so that spatial information users get during animation changes with their action scale [37]. Similar approach is also seen in 2D [19]. However, these techniques do not consider semantic representations in visualizing multiscale structures. Instead, they presented the same structure with the same representation, although the representation was differently sized at different scales. The underlying assumption of these techniques is that simply scaling up and down the same representation will provide users with semantic information at different scales. While such techniques may work in some situations, users may find them inadequate in interaction with complex structures, such as protein models, which require the change of representations in a non-geometric way along scale.

4. Space-Scale Animation

Users need tools to help them understand the relationship between different structures at different scales. Structures could be related directly, such as one structure being included in the other, or are related through other intermediate structures. To visualize such relationships, tools should help users see these structures of interest, as well as other structures that bond them together. Furthermore, given the difficulty in navigating through multiscale environment, tools should also help reduce cognitive burdens in navigation control in space and scale.

Space-scale animation is a design targeted to these two goals. It uses system-controlled animation to show users a dynamic view that ties structures at different scales in different views, and it does not require the involvement of users in the selection and control of navigation path. The dynamic view is generated by an interpolating function considering the view difference in both space and scale. In animation, both spatial and semantic information is presented to help users establish the relationship between different structures.

4.1. Concept of Space-Scale Animation

Animation design is to interpolate a series of view frames between two views. For any view frame in a Cartesian space in virtual environments, it can be uniquely determined by its view position, *P,* and view orientation, *O*. In 3D environments, *P* and *O* determine the spatial location and orientation of a view camera. In 2D, where view orientation is fixed, only *P* would be needed to set a special view point, such as the center of a view window.

We can use $V(P, O)$ to represent a view. Thus, to create a view animation to show the transition from View V_a to View V_b is to have an appropriate interpolating function, *f,* which defines all intermediate views:

$$V_i(P_i, O_i) = f(i) \quad (0 \leq i \leq n) \qquad (1)$$

Where: $V_i(P_i, O_i)$ – the *ith* view in the animation
n – the number of view frames in animation
$V_a = V_0 = f(0)$
$V_b = V_n = f(n)$

For multiscale structures, animation also needs to consider viewing scale so that the semantic representations could be displayed effectively and cross-scale relationship between representations can be apprehended properly. For example, let's assume that objects of interest in two views belong to the same hierarchy and their conceptual relationship in the hierarchy is similar to that between Node 1 and 2 in Figure 3. These two nodes are related to each other through nodes A, B and C. To help users understand the relationship between Node 1 and 2, animation may need to include views to their direct parent, Node B and C, as well as their lowest common parent, Node A. A dynamic view transition in a sequence of Node 1, B, A, C, and 2 would visualize how Node 1 and 2 are connected in space and scale. When the objects of interest have a conceptual relationship similar to that between Node 2 and 3 in Figure 3, users would only need an animation that shows a view transition from Node 2, to C, and then to 3. An animation, which includes key frames to show these objects of interest, their lowest common parent, and other parents between them and their lowest common parent, can help users understand semantic relationship of these objects in space and scale.

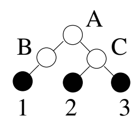


Figure 3: Relationship of structures in views.

Figure 4 compares space-scale animation with traditional space-only animation in a space-scale diagram [14]. Space-scale diagram is a tool to help understand multiscale user interfaces. Figure 4a shows two different representations of a structure at two scales in a workspace. The picture on the top indicates that the structure appears as two objects, a square and a triangle, at a scale of S_A, while the picture on the bottom indicates that users will see the structure as a circle at another scale – S_B. Here, we assume that these three objects are all in the same hierarchy, and the triangle and square are like Node 2 and 3 in Figure 3, and the circle is their lowest parent, like Node C in Figure 3.

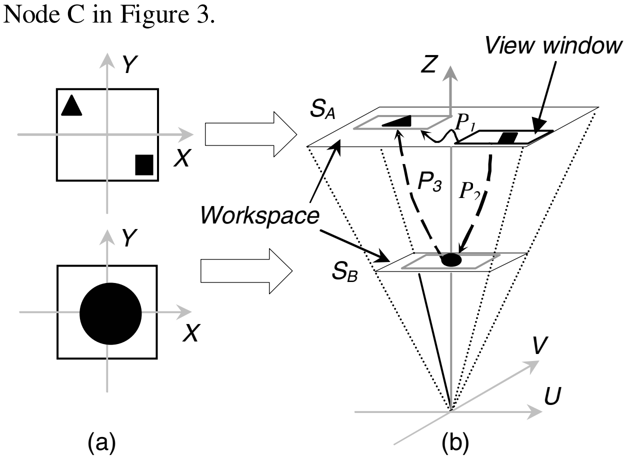


Figure 4: Space-scale diagram of three objects.

Figure 4b is the space-scale diagram of these objects. The vertical axis is the view scale or magnification, and the horizontal axes are mapped to the original spatial dimensions. At different scales, the workspace will look differently, and objects in it will appear with different sizes or even different representations if semantic representations are applied. Stacking these different appearances of the workspace together gives an inverted pyramid. In this diagram, the workspace has both the triangle and square at Scale S_A, but only the circle at S_B.

The workspace at S_A is so large that a user cannot see two objects in a single view window. To move from a view of the square to a view of the triangle, the user can pan the view window manually. However, due to the invisibility of the target, the user may not even know which direction to pan the view. The user can have an animation tool that follows a path, P_1, but the animation will only help the user see the spatial difference between two objects. The user cannot obtain information about their cross-scale relationship.

Space-scale animation would show the view transition in both space and scale. The circle is the lowest common parent of the triangle and the square, and a view of it could help users to tie the square and triangle together. A space-scale animation can be created by following the path P_2 and P_3. The animation following P_2 shows the square is gradually being shrunk, until at one point, both the square and the triangle could be visible. In the end of this part of animation, the square and the triangle disappear, and the circle replaces them in the view. The embedment of the square and the triangle in the circle can be clearly depicted in the animation. Following P_3, the animation shows a reversed process, in which the triangle, rather than the square is the focus. Two segments of this animation together help users see the spatial relationship between the square and the triangle, as well as their cross-scale relationship with the circle.

4.2. Space-Scale Animation Design

Traditional animation algorithms usually only consider view position and view orientation, and cannot be directly applied in P_2 and P_3, which require the scale factor. In a multiscale space, a view can be written as *V(P, O, S)*, where *P, O,* and *S* are the view position, view orientation, and view scale respectively. Thus, to create a space-scale animation is to define interpolation functions for view path, view orientation, as well as view scale. Techniques to interpolate view path and view orientation interpolation are mature [26,39]. Thus, the focus of this research is on scale interpolation.

A scale interpolation function could be in any form, linear or logarithmic. In multiscale user interfaces, scaling is usually modeled as a logarithmic function, which gives users a constant relative rate of change in their views. Thus, a logarithmic function is adopted as the basic form of the scale interpolating function.

Figure 5 shows the value of view scale as a function of time (or the view frame number). This function defines a space-scale animation that links two views at two scales of S_0 and S_1 through a view of their lowest common parent, which is also their direct parent in this case, at an intermediate scale of S_C.

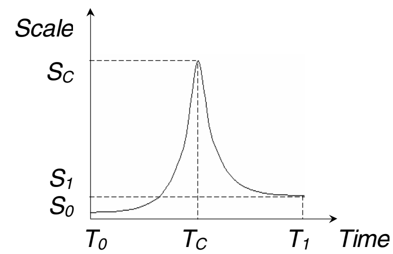


Figure 5: A two-segment interpolation function.

The function has two segments. The first segment interpolates the view scale from S_0 to S_C, and the second segment concerns the scale value from S_C to S_1. Each segment consumes equal time, so T_C is at the mid-point of T_0 and T_1. This figure shows a general case, in which the animation starts and ends at different scales. The function can be written as:

$$\begin{cases} lg(S-S_0)=\alpha(t-T_0)+\gamma & (T_0<t\le T_C) \\ lg(S-S_1)=\beta(T_1-t)+\delta & (T_C\le t<T_1) \end{cases} \qquad (2)$$

Here, S is the interpolated scale value. Parameters of $\alpha, \beta, \gamma,$ and δ are related to S_0, S_1, S_C, T_0, and T_1.

Such a function has one problem: a user will see scale changes very rapidly around S_C. The dramatic change around S_C may make it difficult for users to see the structures around S_C and hurt the understanding of the cross-scale relationship among structures. To address this issue, a modified four-segment function can be used to reduce the change rate of scale value around S_C (Figure 6).

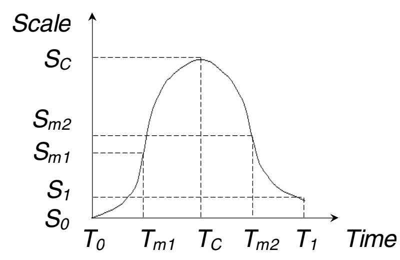


Figure 6: A four-segment interpolation function.

The first segment, from T_0 to T_{m1}, starts slowly from S_0; the second and third segments, from T_{m1} to T_C and from T_C to T_{m2} respectively, make slow transitions around S_C; and the last segment, from T_{m2} to T_1, gradually approaches the destination scale. Each segment uses a quarter of the time of the whole animation.

Two additional intermediate scales, S_{m1} and S_{m2}, are needed to make this four-segment function. Because the scale change is logarithmic, these two scale values should be the geometric averages of S_0 and S_C as well as S_C and S_1. The first and last segments can still follow the function seen in Formula (2). The second and third segments are defined by the following:

$$\begin{cases} lg(S_C - S) = \alpha_1 (T_C - t) + \gamma_1 & (T_{m1} < t \leq T_C) \\ lg(S_C - S) = \beta_1 (t - T_C) + \delta_1 & (T_C \leq t < T_{m2}) \end{cases} \quad (3)$$

Choosing appropriate parameters, these four segments can be made continuous.

The same technique can be applied in situations where parent nodes are found between interested objects and their lowest common parent. Figure 7 shows a scale interpolation function for a space-scale animation which shows three parent nodes. In addition to a view to the lowest common parent of objects of interest at T_C, the user can also see other relevant parent nodes at the scale of S_{P1} and S_{P2}. T_{P1} and T_{P2} are the time these two views occur in space-scale animation, and they are at the first and third quartile of the total animation time.

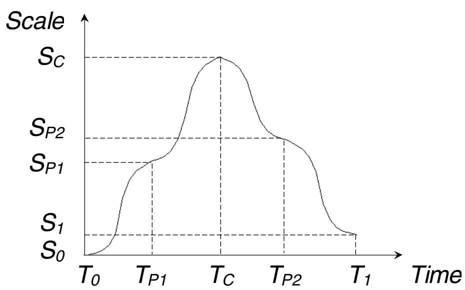


Figure 7: A multi-segment interpolation function.

4.3. Scaling as Moving

In multiscale environments, users can navigate by pure moving or combining moving and scaling [5]. In many situations, users can use only scaling for navigation as long as a proper scaling center can be chosen. This implies that scaling itself alone can change both view position and view scale. Thus, with proper scale values and scaling centers, view position interpolation may not be necessary in space-scale animation.

Any scaling has a "fixed point", a point that stays unchanged during scaling, and around which everything else expands, moving away, or contracts, moving closer. This fixed point is called the "scaling center". The spatial location of the view point also expands or contracts related to the scaling center so that the view will show different objects and structures at different scales.

Figure 8 shows how scaling can replace moving in 2D multiscale environments. Figure 8a shows a user is working in a workspace with two objects, a triangle and a square. At the given scale in Figure 8a, the user can only see the triangle. The square is not visible because of the limited size of the view window. To see the square, the use can move the view to the right. Another option is to just use scaling. The user can first scale down the world, with the triangle object as the scaling center, and shrink all objects (Figure 8b) until the square becomes visible (Figure 8c). Then, the user can scale up the world by taking the square as the scaling center (Figure 8d). Scaling stops when the square gets a proper size (Figure 8e). As seen, this pure scaling process does not involve any moving. Users do not need to make a decision on navigate direction and distance.

Such scaling as moving could be difficult to use in interactive control, because users still need to know where to place the scaling center appropriately, which scaling direction they should choose to bring all objects of interests visible, and when to stop scaling. If they pick a wrong scaling center, or over-shoot in scaling, they will end in "*desert fog*" [20].

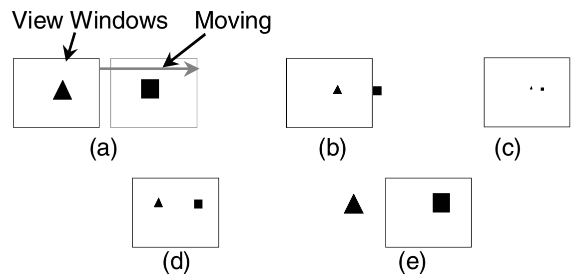


Figure 8: Scaling as moving.

However, these issues are not a concern for system-controlled space-scale animation. With the knowledge about where those objects of interest are located in space and scale, their coordinates can be used as scaling centers so that the user will always see relevant and interested objects in the process of animation. By comparing the view windows size with the rendered size of objects of interest, the system can determine what direction scaling should go and when to stop scaling.

5. Implementation

Space-scale animation has been implemented with Java in both 2D and 3D multiscale environments. Our 2D implementation was built upon the Piccolo toolkit [4], and 3D implementation was in a multiscale virtual environment [42]. Both of these systems provide scaling tools, so we focused on issues concerning the interactive design, such as semantic representations and the process control of space-scale animation.

5.1. Semantic Representations

Semantic representations display different representations at different scales. In our implementation, a semantic representation object wraps an array of geometric representations, a scale-value array defining the scale range for each representation to appear, and a behavior processing module to monitor scaling events in multiscale environments (Figure 9).

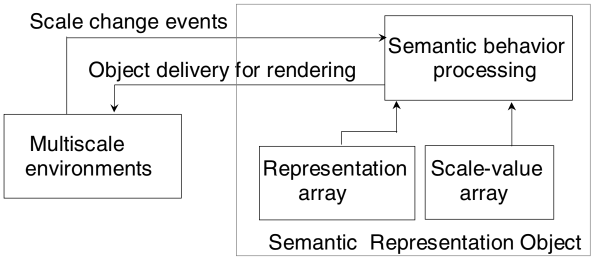


Figure 9: Semantic representation object.

To render a structure semantically, a semantic representation object needs to monitor a user's interaction scale constantly. In our 2D implementation, the semantic representation object obtains the scale value by directly listening to zooming events. Then, the behavior processing module compares this value with the scale-value array, and chooses an appropriate representation for rendering. Our 3D implementation ties a semantic representation object with the transform node that governs the scale of the whole 3D scene. The change of the scale value of this transform node wakes up the semantic representation object, whose behavior processing module then finds a proper representation for rendering.

To visualize cross-scale semantic relationship between objects in space-scale animation, each representation in a semantic representation object should contain structural information describing its relationship with the multiscale structure it belongs to, such as its level and position in a hierarchy. Such information is critical to determining the semantic relationship between two arbitrary objects and finding their relevant parent structures. In some multiscale structures, such structural information of an object can be directly obtained from source data. For example, a map comes with the coordinates of the region it represents and a map scale. Coordinates and scale can be directly used to build a semantic structure to organize relevant maps. The scale of a map sets the level at which this map should be in the structure, and the coordinates determine in which branch this map should be placed. If users are interested in documents in a file system, a semantic structure can be directly constructed by using the file hierarchy, just like what PhotoMesa, a multiscale tool for managing image documents, has done [2]. For 3D structures, modeling data often reflect spatial relationship among objects. For example, 3D models described by a virtual reality modeling language, or VRML, often use the hierarchical structure of VRML to organize objects, and this hierarchy can help to construct a semantic relationship among spatial structures.

5.2. 2D Implementation

The 2D space-scale animation algorithm is as follows:

1. *Identifying two structures of interest in two views (users interactively specify what structures they are interested in);*
2. *Getting the spatial locations of these structures and the scale levels at which they are visible;*
3. *Finding their positions in the multiscale structure;*
4. *Finding their lowest common parent and other parent objects between each structure and their lowest common parent;*
5. *Obtaining the spatial locations of all these parent objects and the view scales at which they appear;*
6. *Obtaining the geometric centers of these two structures of interest and all their relevant parent objects;*
7. *Lining up these geometric centers as scaling centers;*
8. *Generating a multi-segment interpolation function based on the viewing scales of these two structures and all others relevant parent objects;*
9. *Scaling the world by following the interpolation function and choosing appropriate scaling centers (Semantic representations are applied during this step).*

Figure 10 is the screen shot of key frames in a 2D space-scale animation to help users build the connection of two maps seen in Figure 2, Figure 10a and 10g are those two maps exhibiting spatial information of two places at the street level. Figures 10b to 10d show geographic information of the place seen in Figure 10a at the scale levels of city, county, and state. Figures 10d to 10f reveal where the place seen in Figure 10g is located at the levels of state, county, and city. Space-scale

(a) (b) (c) (d) (e) (f) (g)

Figure 10: Key frames in a 2D space-scale animation linking two maps. (Maps provided by Mapquest.com)

Proceedings of the Third International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV'05)
0-7695-2396-X/05 $20.00 © 2005 IEEE

animation that ties these maps together can help users see how close these two places are and how they are located to each other.

5.3. 3D Implementation

The implementation of 3D space-scale animation is slightly more complicated than its 2D version because of the involvement of view orientation. In 3D, two views may have very different view orientations. Difference in view orientation may be a challenge for users, because aligning views by doing mental rotation is difficult for many people [34]. Thus, animation should also interpolate view orientation, and do view rotation.

When both view scale and view orientation are considered in animation, there is an issue about how these two parameters should be integrated. Should each view frame include the change of both view orientation and view scale? Or should only one parameter be allowed to change in two consecutive views? Technically, the difference between these two approaches is minimal. Cognitively, however, these two approaches may affect users very differently. Two successive views that differ from each other in both orientation and scale may demand more cognitive resources from users in interpreting what has happened. Allowing only one kind of change in two views would make it easy to comprehend the view transition. In our implementation, we adopted the second approach, and separated the interpolation of view scale from that of view orientation.

Figure 11 is a 2D illustration of our 3D implementation. Figure 11a shows the ultimate goal of the view transition, which is to create an animation showing how View *A* and View *B*, which are at different locations and with different orientations, are related. These two views focus on different objects: a triangle in *A* and a square in *B*. Traditionally, animation algorithms create a path *T* for view position and orientation between View *A* and *B*. However, this approach does not help the understanding of the semantic relationship between these two objects of interest and their parent, *P,* a circle.

Space-scale animation first rotates the view so that the viewing direction is perpendicular to the line linking two objects of interest (Figure 11b). This allows a better view of both objects and their parents in the animation process. Then, the whole world is scaled down with the triangle as the scaling center (Figure 11c). During this scaling, the user may see different representations because of semantic representations, as seen in Figure 11d, where shrinking the triangle and the square more would make them disappear and reveal their parent object, the circle. When the scale reaches the level at which the parent node is totally visible, a view rotation process is initiated so that the square will be in the center of the view (Figure 11e). Although the square might not be visible at this point because of semantic representations, its coordinates are known by the system. Thus, a rotation process can be initiated and executed automatically. Next, the whole world will be scaled up by taking the square as the scaling center, with the involvement of semantic representations (Figure 11f). Scaling stops when the target scale is reached, and then the view will be rotated appropriately to the final view position (Figure 11g).

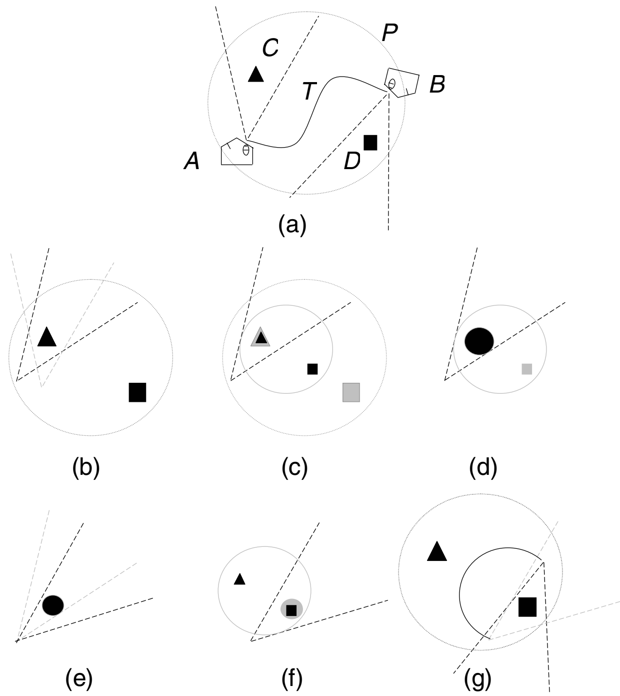


Figure 11: Rotation-involved space-scale animation.

The algorithm for 3D space-scale animation is similar to that for 2D, except that a rotating process is initiated first whenever a new scaling center is chosen so that the scaling center is always positioned in the middle of the view during scaling. For rotation processes, spherical linear interpolation functions are used to achieve a constant angular velocity.

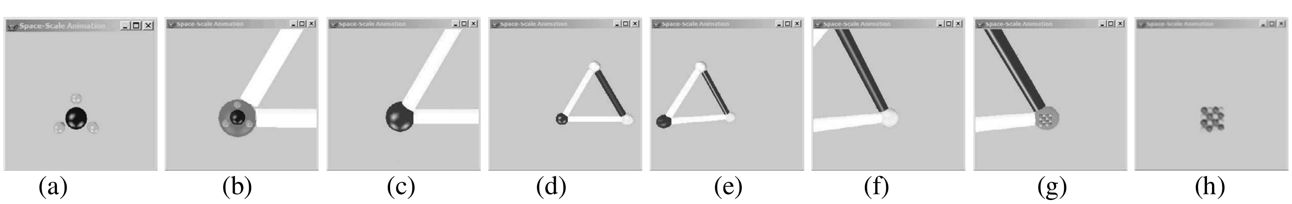


Figure 12: Key frames in a 3D space-scale animation showing the relationship of atomic structures.

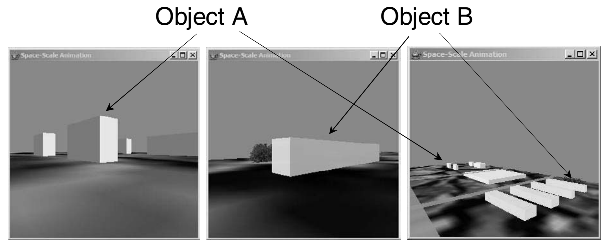
Figure 12 is the screen shot of key frames in a 3D space-scale animation connecting two atomic structures seen in Figure 1. Figure 12a and 12h are the two divergent views. Scaling down the world will reveal a new structure (Figure 12b), which includes the parent object of the structure seen in Figure 12a. At this scale, the embedment is clearly presented with the parent object rendered as transparent. Scaling down the world more, the parent is no longer transparent, and is presented with its real appearance (Figures 12c). Further scaling down the world, users see the structure in which the parent object resides (Figure 12d). Then, the view is rotated to put the next scaling center in the middle of the view (Figure 12e). Scaling up the world (Figure 12f), the relationship between the target object and its direct parent becomes visible (Figure 12g). Eventually, the world is scaled up to the point at which the view only shows the target structure. This space-scale animation not only helps users see how two structures are spatially related, but also informs users what a cross-scale chemical structure ties them together.

5.4. General Space-Scale Animation

This space-scale animation technique can also be useful in the understanding of structures in very large space, which may not have explicit multiscale structures. For example, many 3D virtual environments are large, and users may need to go to very different places to collect information. It could be hard for users to tie objects from different places together and have a coherent understanding of the whole world, given that finding common reference objects for distributed objects could be a challenge. Even if people can work collaboratively in such environments, they may still find it difficult to build common ground for collaboration because of their distributed views and interests across large space. In these situations, space-scale animation can be used to narrow down the difference among divergent views by presenting the world at different scales.

Figure 13 shows a scenario in which two city planners are working together on a planning project, but they get totally different views, Figure 13a and 13b respectively, because of the different scale levels at which they are working. These two planners can use space-scale animation tools to construct the context of their divergent views through an overview of the city, such as Figure 13c. With a dynamic view transition that ties two views with the overview, planners will know where their partner is working and how their objects of interest, Object A and B, are spatially related.

One major concern with the use of space-scale animation to visualize structures that are not organized in a multiscale way is that it may be difficult to find an object that can work as the lowest common parent to tie two arbitrary objects together. Then, choosing the intermediate scale for animation becomes a challenge.



(a) (b) (c)

Figure 13: Two divergent views in collaboration and an overview showing their relationship.

If spatial structures are what users are interested in, one simple way to address this issue is to use the spatial bounding boxes of objects. A bounding box can be generated dynamically as long as the coordinates and spatial dimensions of an object are known. For two objects distributed widely, a bounding box can always be found to include their bounding boxes, and then the view to this parent bounding box can be treated as the parent object of these two objects. The location, spatial dimension, and the viewing scale of this parent object can be used to generate space-scale animation. Figure 13c is actually obtained in this way by first creating a bounding box that includes both Object A and B and then setting up the viewpoint and viewing scale to have this bounding box within the view window. Because this bounding box is only important for the system in creating the animation, not for users' perception and understanding of the environment, it does not have to appear in the view. Figure 14 shows what this bounding box would look like if it could be displayed.

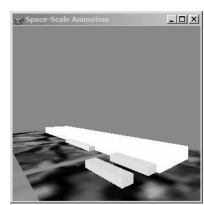


Figure 14: The bounding box used to generate the overview of Figure 13c.

Currently, our system allows the direct import of VRML models, and supports the examination of these models with space-scale animation by the bounding box method if semantic structural information is not provided in VRML model. The same technique can easily be applied in 2D scenarios, in which the spatial relationship among objects is a concern.

Often, users' interests may go beyond spatial relationship among objects. Then, space-scale animation should include other semantic information that users may want in the dynamic view. For example, to examine a 3D model protein structure, users may need to see biological connections between atoms and molecules, rather than just their spatial relationships. Presenting such connections in animation would require information about how components in the structures of interest are related to each other in non-spatial contexts. The development of modeling and description methods makes it possible to embed semantic information into modeling data. For example, using XML for 3D description allows the integration of semantic knowledge structures with spatial structures [10,21]. This provides a good opportunity to use multiscale technology and space-scale animation to support the visualization and interaction with large and complicated structures.

If knowledge structures are not available in original data sources, users would need help in creating such structures. Advanced technologies like data mining or ontology mapping could be useful to provide users with potential organization schemes, although it remains a question of whether system-generated schemes are what users may want. Of course, users can have tools to manually label individual objects to create their own semantic descriptions. However, it will be a daunting task to handle large structures with thousands or even millions of components by using this approach. Apparently, this is an issue concerning the creation of computerized classification schemes for massive data and information, a challenge far beyond the scope of this research.

6. Conclusions

This research has proposed a new interactive animation method to help users establish the connections between multiple views, which focus on objects and structures distributed in different places and at different scales. Space-scale animation traverses space and scale, and provides users with a proper context for different structures in both space and scale dimensions. For large and complicated structures, showing cross-scale context information could be important to the understanding of structures in different views, in particular when these structures are related to each other through structures at other scale levels, as seen in Figure 10 and 12.

The focus of the research has been on the interpolation function and semantic representations of structures in designing space-scale animation. Because of the unique feature of scaling-as-traveling in multiscale environments, the interpolation of spatial position often becomes unnecessary when proper scale interpolation functions and scaling centers can be chosen. For semantic representations, our efforts have focused on the interactive design that presents users with appropriate representations at different scales in animation.

An issue in our 3D implementation of space-scale animation is that the interpolation function does not consider whether there are objects on animation trajectories which may block a user's view to those structures of interest and prevent the user from seeing clearly how different structures are related in animation. It is our interest to address this issue in the future by integrating more complicated trajectory functions that allow the avoidance of obstacle objects [38] into design.

We are also interested in knowing how effective this space-scale animation technique is in supporting cross-scale understanding. It has been argued that the effectiveness of animation techniques in supporting user interfaces is related to many factors, such task domains, user experience, image quality, transition, and interactivity [15]. We are interested in studying user performances in using space-scale animation in managing geographic information and complex model structures. In particular, we would like to use lab experiments to study the effectiveness of the approach of using the lowest common parent to tie objects that are distributed in space and scale. Also, we will examine how this space-scale animation may help cross-scale collaborative information sharing by comparing it with other techniques, such as verbal communication.

Acknowledgements

This research is supported by a grant from the Office of the Vice President for Research at the University of Arizona. The author also thanks the anonymous reviewers for their constructive remarks.

References

[1] Ahl, V., & Allen, T. F. (1996). Hierarchy Theory: A Vision, Vocabulary and Epistemology. Columbia University Press.

[2] Bederson, B. B. (2001). PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. In Proceedings of Symposium of User Interface Software and Technology (UIST' 2001), pp. 71-80.

[3] Bederson, B. B., & Boltman, A. (1999). Does Animation Help Users Build Mental Maps of Spatial Information? Proceedings of IEEE Information Visualization Symposium (InfoVis'99), pp. 28-35.

[4] Bederson, B., Grosjean, J., & Meyer, J.(2004). Toolkit Design for Interactive Structured Graphics. IEEE Transactions on Software Engineering, 30(8), pp. 535-546.

[5] Bederson, B., & Hollan J. (1994). Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. Proceedings of Symposium of User Interface Software and Technology (UIST'94), pp.17-26.

[6] Björk S. (2000). Hierarchical Flip Zooming: Enabling Parallel Exploration of Hierarchical Visualizations. In Proceedings of AVI 2000, pp. 232– 237.

[7] Chang, B., & Ungar, D. (1993). Animation: From Cartoons to the User Interface. Proceedings of Symposium of User Interface Software and Technology (UIST'93), pp. 45-55.

[8] Clark, H. H., & Brennan, S. E. (1991). Grounding in Communication. In Resnick, L. B., Levine, J., & Teasley, S. D. (Eds.), Perspectives on Socially Shared Cognition.

[9] Combs, T., & Bederson, B. B. (1999). Does Zooming Improve Image Browsing? Proceeding of ACM Digital

Library (DL'99), pp. 130–137.
[10] Dachselt, R. Hinz, M., & Meißner, K. (2003). Contigra: An XML-Based Architecture for Component-Oriented 3D Applications. Proceedings of ACM Conference on 3D Web Technology, pp. 155-163.
[11] Dill, J., Bartram, L., Ho, A., & Henigman, F. (1994). A Continuously Variable Zoom for Navigating Large Hierarchical Networks. Proceedings of IEEE Conference on Systems, Man and Cybernetics, pp. 386-390.
[12] Furnas, G. W. (1986). Generalized Fisheye Views. Proceedings of ACM Conference on Human Factors in Computing System (CHI'86), pp. 16-23.
[13] Furnas, G. W. (1997). Effective View Navigation. Proceedings of ACM Conference on Human Factors in Computing System (CHI'97), pp. 367-374.
[14] Furnas, G. W., & Bederson B. B. (1995). Space-Scale Diagrams: Understanding Multiscale Interfaces. Proceedings of ACM Conference on Human Factors in Computing System (CHI'95), pp. 234-241.
[15] Gonzalez, C. (1997). Does Animation in User Interfaces Improve Decision Making? Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'96), pp. 27-34.
[16] Hanson, A., Wernert, E., & Hughes, S. (1997). Constrained Navigation Environments. Scientific Visualization: Dagstuhl'97 Proceedings, pp. 95-104.
[17] Hodgins, J., O'Brien, J., & Bodenheimer, R. (1999). In Webster, J. (ed.), Encyclopedia of Electrical and Electronics Engineering, Computer Animation, 3, pp. 686-690.
[18] Hornbaek, K., Bederson, B. B., & Plaisant, C. (2002). Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview. Transactions on Computer-Human Interation. vol. 9, no. 4, pp. 362-389.
[19] Igarashi, T., & Hinckley, K. (2000). Speed-dependent Automatic Zooming for Browsing Large Documents, Proceedings of ACM Symposium of User Interface Software and Technology (UIST'2000), pp. 139-148.
[20] Jul, S., & Furnas, G. W. (1998). Critical Zones in Desert Fog: Aids to Multiscale. Proceedings of ACM Symposium of User Interface Software and Technology, pp. 97-106.
[21] Kim, T. & Fishwick, P. A. (2002). A 3D XML-Based Customized Framework for Dynamic Models. Proceedings of ACM Conference on 3D Web Technology, pp. 203-109.
[22] Koike, H., & Yoshihara, H. (1993). Fractal Approaches for Visualizing Huge Hierarchies. Proceedings of IEEE Symposium on Visual Languages (VL'93), pp. 55-60.
[23] Lamping, J., Rao, R., & Pirolli, P. (1995). A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. Proceedings of ACM Conference on Human Factors in Computing System (CHI'95), pp. 401-408.
[24] Morrison, P, & Morrison, P. (1994). Powers of Ten: About the Relative Size of Things in the Universe, and the Effect of Adding Another Zero. W. H. Freeman & Co.
[25] Páez, L. B., da Silva-Fh., J. B., & Marchionini, G. (1996). Disorientation in Electronic Environments: A Study of Hypertext and Continuous Zooming Interfaces. Proceedings of the American Society for Information Science (ASIS'96), pp. 58–66.
[26] Parent, R. (2001). Computer Animation: Algorithms and Techniques. San Francisco: Morgan-Kaufmann.
[27] Perlin, K., & Fox D. (1993). Pad: An Alternative Approach to the Computer Interface. Proceedings of ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'93), pp. 57-64.
[28] Raab, A., & Ruger, M. (1996). 3D-Zoom: Interactive Visualization of Structures and Relations in Complex Graphics. In Girod, B., Nieman, H., & Seidel, H. (Eds.) 3D Image Analysis and Synthesis, pp. 125-132.
[29] Robertson, G., & Mackinlay J. (1993). The Document Lens. Proceedings of ACM Symposium of User Interface Software and Technology (UIST'93), pp. 101-110.
[30] Robertson, G., Card, S., & Mackinlay, J. (1991). Information Visualization Using 3D Interactive Animation. Proceedings of ACM Conference on Human Factors in Computing System (CHI'91), pp. 461-462.
[31] Robertson, G., Mackinlay, J., & Card, S. (1991). Cone Trees: Animated 3D Visualizations of Hierarchical Information. Communications of ACM, 34(2), pp. 89-194.
[32] Sarkar, M., & Brown, M. H. (1992). Graphical Fisheye Views of Graphs. Proceedings of ACM Conference on Human Factors in Computing System (CHI'92), pp. 83-91.
[33] Schaffer, D., Zuo, Z., Greenberg, S., Bartram L., Dill J., Dubs, S., and Roseman, M. (1996). Navigating Hierarchically Clustered Networks through Fisheye and Full-zoom Methods. Transaction on Computer Human Interaction, vol. 3, no. 2, pp. 162-188.
[34] Shepard, R.N., & Metzler, J. (1971). Metal Rotation of Three-Dimensional Objects. Science, 171, pp. 701-703.
[35] Snowdon, D., & Jää-Aro, K. (1997). A Subjective Virtual Environment for Collaborative Information Visualization. Virtual Reality Universe'97.
[36] Strasnick, S. & Tesler, J. (1996). Method and Apparatus for Displaying Data within a 3D Information Landscape. US Patent 5528735, Silicon Graphics, Inc. Filed 23rd March 1993, Issued 18th June 1996.
[37] Tan, D., Robertson, G., & Czerwinski, M. (2001). Exploring 3D Navigation: Combining Speed-Coupled Flying with Orbiting. Proceedings of ACM Conference on Human Factors in Computing System, pp. 418-425.
[38] Warren, W. H., Di, S., & Fajen, B. R. (2003). Behavioral Dynamics of Avoiding a Moving Obstacle. Journal of Vision, 3(9), 134a.
[39] Watt, A., & Policarpo, F. (2001). 3D Games: Animation and Advanced Real-time Rendering. Addison-Wesley.
[40] Woodruff A., Landay, J., & Stonebraker, M. (1998). Constant Density Visualizations of Non-Uniform Distributions of Data Visualization. Proceedings of ACM Symposium of User Interface Software and Technology (UIST'98), 1998, pp. 19-28.
[41] van Wijk J., & Nuijj, W. A. (2003). Smooth and Efficient Zooming and Panning. Proceedings of IEEE Symposium on Information Visualization (IV'03), pp. 15-22.
[42] Zhang, X., & Furnas, G. W. (2005). mCVEs: Using Cross-Scale Collaboration to Support User Interaction with Multiscale Structures. Presence: Teleoperators and Virtual Environments, vol. 14 no. 1, pp. 31-46.