

Exploratory Visualization using Bracketing

Jonathan C. Roberts
Computing Laboratory
University of Kent,
Canterbury, UK, CT2 7NF
+44 1227 827571
J.C.Roberts@kent.ac.uk

ABSTRACT

There are many tools that provide the user with an abundance of sliders, buttons and options to change; such tools are popular in exploratory visualization. As the user changes the parameters so the display dynamically updates and responds appropriately to changes made. These multiparameter systems can be difficult to use, as the user is often unaware of the outcome of any action before it occurs. Specifically it may be unclear whether to increase or decrease a parameter value to get a desired result. Multiple view systems can help, as the user can try out various scenarios and compare the results side-by-side, although if unrestricted the user may be swamped by numerous and often unnecessary views. In this paper we present the novel idea of 'bracketing', where a principal view is supported with two additional views from slightly different parameterizations. The idea is inspired by exposure bracketing in photography. This provides a middle ground: it offers a way to see adjacent-parameterizations, while allowing yet restraining multiple views. Moreover, we demonstrate how bracketing can be exploited in many applications and used in various ways (within parameter, visual and temporal domains).

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation] User Interfaces – *Graphical user interfaces (GUI), Interaction styles.*
I.3.6 [Computer Graphics] Methodology and Techniques – *Interaction techniques.*

General Terms

Design, Human Factors.

Keywords

Exploratory Visualization, Bracketing, Multiple Views, Coordination.

1. INTRODUCTION

Bracketing is a new idea for depicting parameter changes or balancing space/time tradeoff issues within computer interfaces.

In this paper we apply this idea to challenges in visualization, in particular exploratory visualization. The concept is inspired by bracketing in photography.

When professional photographers take pictures they often make an estimation of the correct light and speed levels, set up the camera with these parameter settings and take multiple pictures with parameterizations either side. In defining bracketing, we quote “[bracketing is a technique where the photographer] takes a series of images of the same scene at a variety of different exposures that *bracket* the metered exposure (or manual exposure)” [1].

This technique can be readily applied to exploratory visualization (EV) by aiding the user as they investigate various parameter settings. In EV the user can try out scenarios, investigate different parameterizations and manipulate and interact with various visual representations the data. Bracketing could be used to support the exploration; the user would designate the parameterization for one view, and the system would automatically depict two additional views alongside that bracket the parameterization. Later in the paper we will explore how these bracketed values can be created. This encourages investigation and aids the user to refine towards an optimum solution.

For example, a user may wish to discover an appropriate threshold value to generate a suitable isosurface. The user would initially guess a value (a decision itself that may be guided by histograms or other statistical visualizations) and then refine the value based on the aesthetic appearance of the resultant visualization. Bracketing could be used to depict three views, one of the chosen threshold (T), with two others depicting slightly different versions of threshold values $T - \delta$ and $T + \delta$. The single view is assisted by two alternatives that provide *context* within the parameter space.

Furthermore, the idea of bracketing can be used to manage the visual space. This is useful, in order to gain a balance between the often restricting fixed view scenario created by dual view systems, to that of a multi-view system utilizing an unrestricted amount of views. Here the bracket can be used to select a reduced number views, from the full list available. It acts as a buffer, where the user can easily delete or move representations in or out of the bracket. These ideas are well suited to be used alongside thumbnail representations. Thumbnails are small, low quality renderings that are under-used in visualization. In particular, thumbnails may be useful to provide rapid exploration of the parameter space [2].

On the surface, bracketing is a simple phenomenon; indeed it may be thought of as coordinating values in the parameter space. But there are various challenges that need addressing: from knowing what to bracket, how to calculate the parameters (that make the

bracketing visualizations), and to work out boundary conditions, for example. The remaining part of this paper expands the bracketing ideology. We detail related work, fundamental principles and illustrate the breadth of the ideas through some demonstrations and examples.

2. BACKGROUND AND RELATED WORK

There are some questions a user may ask when using Exploratory Visualization (EV) tools, such as: Where does the information go? How will the new visualization differ from a previous incarnation when I commit this action? How much do I need to change this parameter to generate a desired result (an action that questions the sensitivity of a parameter)? How to rollback to previous incarnations (exploration history)? Further, the user may wish to find key features or landmarks within the abstract parameter space: the user may be guided towards or explicitly shown these features that have been generated from statistical or automatic analysis of the parameter space.

Bracketing techniques can help the user in many of these instances, thus, before detailing bracketing attributes we expand some of these issues.

2.1 Where does the information go?

The user needs to clearly understand what happens to the resultant visual representation [3]. There are three operational models that explain where the information goes; it may (a) replace the current view, (b) display in a new view (replication) or (c) overlay with the current information, (Figure 1). The bracketing concept, taken directly from photography, is obviously ‘replication’; nevertheless it is easy to envisage certain designs and applications that overlay the bracketed (fore/after) realizations in one single view. For illustration, a scatterplot could depict values in black (from a given parameterization), and by red (if a parameter was increased by 10%) or green if decreased by 10%.

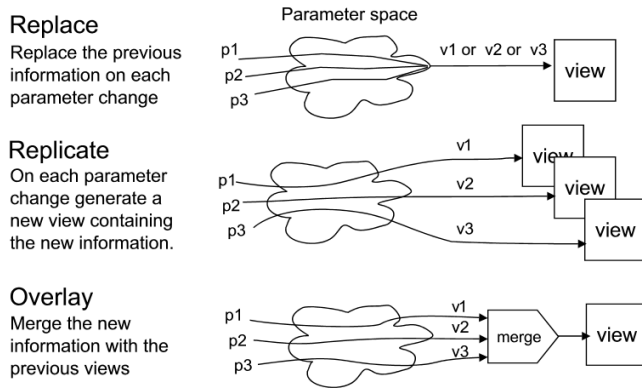


Figure 1. On changing a parameter the new information can replace the old, or produce a new view (replicate) or merge with the existing information (overlay).

2.2 Parameter sensitivity

Any visualization requires setting up multiple parameters. Some parameters may change the filtering, cropping or selection of the image, while others determine the appearance of the visualization.

It may be that some parameters, when altered, make a massive change to the visual depiction, conversely others may only slightly influence the result. Thus, different parameters portray different sensitivities. Moreover, these sensitivities may be dynamic, as they may depend on the settings of other parameters. This sensitivity information can be visualized, which aids the user in their exploration.

For example, Spence and Tweedie [4] in their ‘Influence Explorer’ discuss methods of displaying the extents of the parameters; where results that satisfy the limits are shown in one color, and then those that fail by one limit are shown in another, and so on. This provides the user with some idea of the extent of the parameters: The parameter values are shown in context with additional meta-information. This exploration could be considered as a bracketing method where the information is overlaid in the current view. However, as we shall see the idea of bracketing can have wider implications, and the evaluation of parameter sensitivity can be used to control various attributes of the bracket technique.

Additional context information is also provided through other forms of meta-visualization. Eick in his seminal paper on data-sliders [5] provides this context information by displaying a histogram alongside the parameters so the user can view “selected values in relation to the data and its distribution”. Thus by viewing such related information the user is aided in the control of the system.

Techniques of different views to provide context are also used in video editing where the user can see the before and after frames. In information visualization Nakakoji et al develop this idea for animation as “after-image” and “for-feel” [6] this provides the user with an understanding of what is going to come next. This epitomizes the bracketing technique: as the user is given an idea of the state of the system before and after the chosen value.

2.3 Visualization history

Methods that use multiform and multiple-window techniques allow the user to try out different scenarios and see the results in separate windows, but it is often hard to roll back to a previous scenario. Some visualization systems do overcome the ephemeral nature of the parameterization by storing previous parameter values as a history tree; such as used in GRASPARC [7] and Tioga [8]. It may be interesting (albeit impractical) to simultaneously display every instance in the history tree. This enables side-by-side comparison of various parameterizations. Bracketing may help, where three instances are chosen that represent the history tree: one representing the latest parameterization, with the other two taken from two other instances that delimit

2.4 Automatic searching and automatic visualization

In EV the user is trying to find features of interest as they change and adapt the parameters. Such ‘interesting’ visualizations may be generated by other means. For example, certain rule-based visualization tools automatically generate appropriate visualizations from a list of user-supplied preference-ordered variables [9], or Principal Component Analysis (PCA) techniques can provide ‘optimum’ variable values [10], and data-mining facilities can categorize the data into various groups.

Abstract Parameter Space

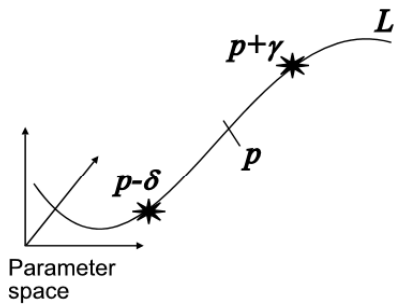


Figure 2. Schematic showing abstract parameter space, with p representing the current parameterization, L the line of similarity, the values $p-\delta$ and $p+\gamma$ represent the values that bracket the current choice.

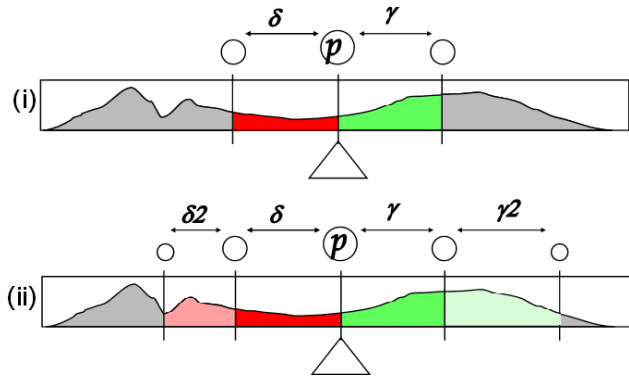


Figure 3. Bracketing quantities along a particular slider. Picture (i) depicts the traditional bracketing scenario, (ii) shows multiple brackets where δ_2 is dependent on δ and γ_2 is dependent on γ .

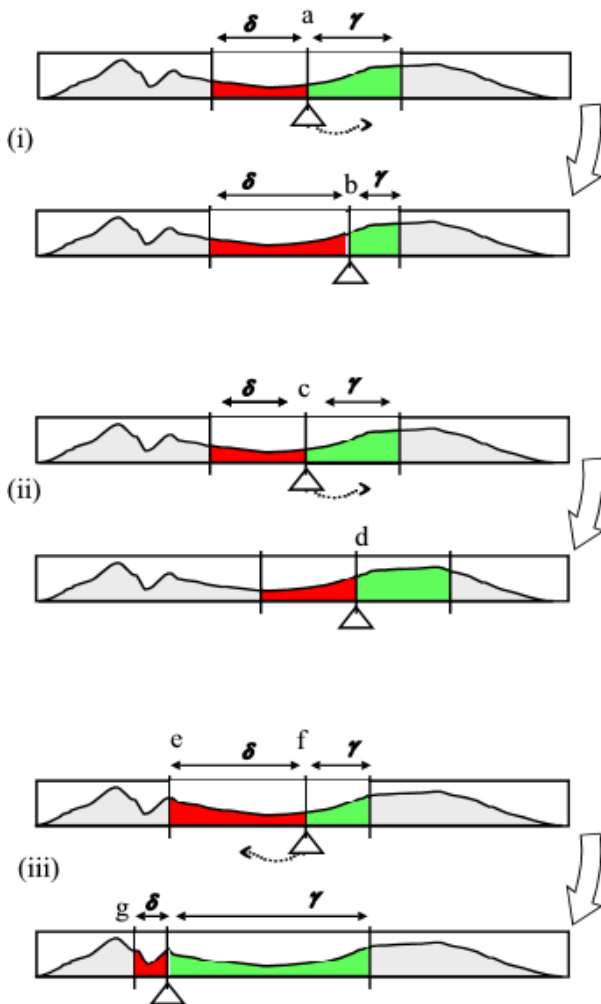


Figure 4. (i) Static bracketing, where the inner parameter can change (constrained by the outer two). (ii) Coordinated bracketing, where all values increase synchronously. (iii) Dynamic bracketing, where δ and γ change based on an associated function (such as features of the underlying data).

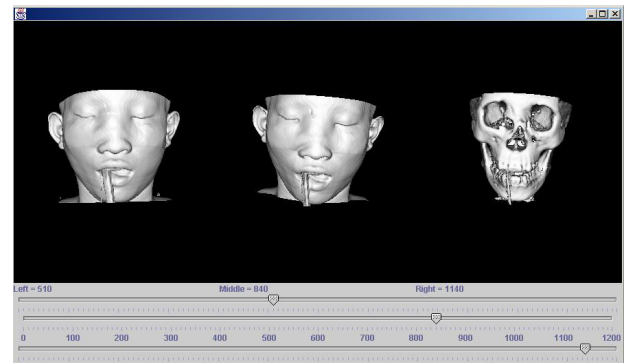


Figure 5. A screen shot of our bracketing demonstrator (written in VTK) that shows the isosurface of volumetric data, where the user sets up initial parameters to generate the centre view, and two additional bracketed views are automatically generated. The user can change the initial settings and the other parameters are coordinated to update the bracketed views appropriately.

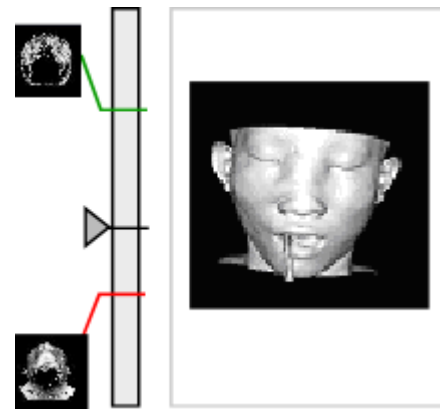


Figure 6. A mockup showing how bracketing could be used with thumbnails and popups. The popups would appear when the user operates the slider, this provides additional information to the user as they move the slider.

3. THE IDEOLOGY OF BRACKETING

In this section we detail various important bracketing principles and attributes.

3.1 Parameter Space

It is feasible to imagine that every solution in a multiparameter system could be plotted: a dual-variable system the plot would fill a two-dimensional xy-plot, a tri-variable system would be plotted in a three-dimensional plot and so on. Thus, a point p in this abstract space would represent a solution (a particular visualization); moreover, p also represents a parameterization. This is depicted in Figure 2.

3.2 The Bracketing values δ and γ

Given a current parameterization p it is possible to describe values that bracket it. For instance, it may be useful to bracket a threshold T plus and minus a constant (c), (given by $T+c$; $T-c$). These constants, that bracket the parameterization as fore and after values, may in fact be different quantities. Thus, in general we define these *before* and *after* values $p-\delta$ and $p+\gamma$ respectively. In particular, it is often useful to designate which variable is being bracketed upon, hence we can explicitly write $p(T-\delta)$ and $p(T+\gamma)$.

3.3 Similarity

Lines L through parameter space (and also the before and after values) signify some form of similarity. By moving along this line each of the generated representations are similar by some means. For example, the user may wish to explore different isosurface thresholds, and so the line could signify changes made to this specific parameter. The notion of similarity itself may depend upon the specific use of the bracket. Moving along this line is like looking at a series of frames from a photographic film; indeed, Kuederle et al [11] use this idea along with thumbnails to evaluate multiple MRI images. The thumbnails provide the context.

This concept can be much broader than merely increasing or decreasing one parameter value. The line L itself may be an abstract concept that non-linearly cuts through the parameter space. For example, in facial reconstruction the exploratory system has many parameters that change (say) the width and size of a mouth, nose or other part of the face. It would be useful to change to an older looking face or one that is ‘more female’ in appearance [10]. Line L could represent femininity and thus changes along L would change the masculine appearance of the face (resulting in the dynamic change of many other parameter values). Hanson [12] in his work describes constraints for animation; movements along L may be described as constraints along this abstract space.

A comment should be made about continuous and non-continuous variables. We need not assume that the abstract parameter space (and even the line within) is a continuous notion, indeed, the representation through parameter space could represent discrete changes, and then we may perceive that movements along line L maybe snapped at some discrete distance.

3.4 Bracketing quantities

Currently we have assumed single ‘before’ and ‘after’ values, which generates three views, Figure 3i. This will be sufficient for most applications. However, it is easy to conceive n-bracketing quantities, where the outer values depend on the inner ones, Figure 3ii.

It is also conceivable (and potentially useful) to utilize a number of neighbourly parameterizations that are *near* a particular point (p) rather than merely two that bracket or are similar (along L). For example, Solomon et al [10] provide the user with a matrix of visual depictions of the faces, in their face reconstruction system; each one is similar to the principal picture. This would be extremely useful in volume rendering, which utilizes a transfer function to map various graphical attributes to the data (such as colour, shading and transparency). This is often hard to setup. The user could provide a ‘best guess’ and the system automatically generates multiple visualizations that bracket and surround the initial parameter state. Moreover, Solomon et al utilize a learning algorithm to generate the next set of parameterizations from a subset of visualizations that the user believes best-fits their requirements. Taken to the extreme, numerous visualizations from parameterizations spread throughout the abstract space could be generated in separate views. This is similar to [11] who populate the space with many thumbnail representations.

3.5 Manipulations of p , δ and γ

User interaction and manipulation of these parameters need to be considered. There are three main scenarios: Static, Coordinated, and Dynamic.

Static describes a state where the initial parameter values are made, the system generates two additional values that bracket, and the user can alter the initial parameter guarded by the two bracketed values. The bracketed values themselves do not move. Figure 4i.

In the coordinated version, the bracketed values are setup as before, but as the user changes the principal value the bracketed values simultaneously alter. (This is coupled parameter manipulation in the layered model of Roberts [13]). This can be directly applied to slider bars, see Figure 4ii.

The dynamic strategy details how the values of δ and γ can dynamically change, as the user alters p . For instance, it may be useful that δ and γ snap to certain features of L based on the underlying data. The values may snap to the next highest peak in a histogram or move to another statistically significant points (such as a percentile) or proportionally to some other function.

Obviously with these strategies there are various boundary conditions to organise. One solution would be to restrict the values to the limit when the extremities are confronted. If there are multiple brackets as in Figure 3ii, then one of two methods could be employed [14]. First, the inner ones can move up to the limit exposed by the outer brackets. Second, the user may grab each inner bracket (moving it within the restriction of the width of the encapsulating bracket) until they attempt to *push* past the boundary of the next bracket when that bracket width changes and the outer ones move as well.

4. APPLICATION

The idea of bracketing could be applied to many domains (parameter, visual or temporal) and built-into many designs. We have experimented with the bracketing idea in several forms. Figure 5 shows a screen shot from our bracketing demonstrator tool that depicts an isosurface value being altered. This tool explores parameter bracketing and has been developed using VTK [15]. The user can set the initial parameterization (p), which corresponds to a threshold, and choose values for δ and γ to

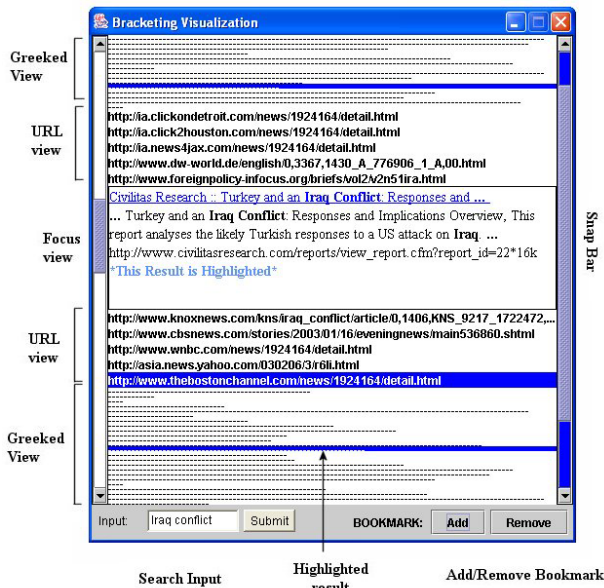


Figure 7. Screen shot of the web-search-result bracketing visualization. The focus view shows the full detail of the “search result elements”, with various views of lowering detail in the adjacent brackets.

determine two additional values that bracket the initial threshold. As the user changes the current parameterization p so the bracketed values (and hence the visualizations) synchronously alter; this demonstrates coordinated manipulation (as shown in Figure 4ii).

Moreover, we have explored this idea in the visual domain, utilizing the technique to provide detail in context. We have developed a tool for detail-in-context web search result visualization [14], Figure 7. This design is similar to the perspective wall technique [16], where the user focuses on the information in the centre wall, with the sides giving the context, the user can scroll the wall to move different information into the centre. The difference with our method is that the user may alter the bracketing sizes and quantities (in this example we use two additional brackets) and that displayed information can come from various parameterizations, indeed, in Figure 7 we employ a semantic level change – for instance, the greeked text shows the size of the web page.

Finally, we foresee that bracketing can be used in many applications. For example, the technique of bracketing could be readily utilized along with popups and thumbnails, such that thumbnail views popup when required and move as the slider is operated. These popup visualizations can be easily generated to display various fore and after parameterizations, Figure 6. For instance, the thumbnails could represent low-quality visualizations of the bracketed values, thus allowing the user to observe and make better judgements as to which direction they should move the slider.

5. CONCLUSION

There are many challenges to overcome in exploratory visualization. We believe the idea of bracketing is another tool that can aid the user in their exploration task. It has the potential

to automatically generate visualizations that support the exploration task and provide context information to aid the user hone onto a desired solution.

We have described the ideology of bracketing; covering many aspects and illustrating the breadth of where these techniques could be applied. In particular the concept of dynamic bracketing, where the bracketed values change depending on underlying principles and associated functions, has much potential, and could aid the user to more quickly focus on pertinent features of interest.

6. ACKNOWLEDGEMENTS

I thank my colleagues of the Kent Visualization Group for their input and comments. Specifically, I acknowledge Edward Suvanaphen and Lei Tan for their help with various bracketing demonstrations.

REFERENCES:

- [1] Askey,P. http://www.dpreview.com/learn/Glossary/Exposure/Auto_Bracketing_01.htm. Visited 1/03/04.
- [2] Ogden,W.C, Davis and Rice,S. “Document Thumbnail Visualization for Rapid Relevance Judgments: When do They Pay Off?” Text Retrieval Conference, pages 528-534, 1998.
- [3] Roberts, J.C. “Multiple-View and Multiform Visualization”. Visual Data Exploration and Analysis VII. Proceedings of SPIE. Vol. 3960, pages 176-185. January 2000.
- [4] Tweedie, L., Spence, R., Dawkes, H. and Su, H. “Externalising Abstract Mathematical Models”, ACM, Proceedings CHI’96, pages 406-412. 1996
- [5] Eick, S.G. “Data visualization sliders”. In UIST ’94 Conference Proceedings, pages 119-120, Monterey, California, 1994.
- [6] Nakakoji,K and Takashima,A and Yamamoto,Y. “Cognitive Effects of Animated Visualization in Exploratory Visual Data Analysis”, Proceedings Information Visualization 2001. IEEE, pp.77-84. 2001.
- [7] Brodlie,K, Poon,A, Wright,H, Brankin,L, Banecki,G and Gay,A. “GRASPARC - A Problem Solving Environment Integrating Computation and Visualization”. Proc. Visualization ’93. Pages 102-109. 1993.
- [8] Brittain,D,L, Aller,J, Wilson,M, and Wang S-L.C.. “Design of an end-user data visualization system”. In Proceedings Visualization ’90. IEEE Computer Society, pages 323-328. 1990.
- [9] Stonebraker,M, Chen,J, Nathan,N, Paxson,C, Su,A, Wu,J. “Tioga: A Database-Oriented Visualization Tool”. In Proc. Visualization ’93, IEEE Computer Society Press, Pages 86-93. 1993.
- [10] Solomon,C, Gibson,S and Pallares-Bejarano,A. “The detection and manipulation of human facial attributes using a parametric appearance model”, The 2nd International workshop for Biologically Motivated Computer Vision, Tubingen, Germany, Nov 2002.
- [11] Kuederle,O, Inkpen,K, Atkins,S, Carpendale,S.T “Interacting with Image Sequences: Detail-in-Context and Thumbnails”. Graphics Interface 2001. Pages 111-118
- [12] Hanson,A and Wernet,E. “Constrained 3D Navigation with 2D Controllers”, Proceedings IEEE Visualization ’97. Pp.175-182. 1997
- [13] Roberts,J.C. “Exploratory Visualization with Multiple Linked Views”. In Exploring Geovisualization (Eds, J. Dykes, A. MacEachren and M.-J. Kraak), Oxford: Elsevier. 2004
- [14] Roberts,J.C and Suvanaphen,E “Visual bracketing for web search result visualization”. In Ebad Banissi et al, editor, Proceedings Information Visualization (IV03), pages 264-269. IEEE Computer Society, July 2003
- [15] Schroeder,W, Martin,K, Lorensen,W “The Visualization Toolkit: An Object Oriented Approach to 3D Graphics 3rd Edition”, Kitware, Inc. February 2003.
- [16] Mackinlay,J.D, Robertson,G.G, and Card,S.K, “The perspective wall: detail and context smoothly integrated”, Proceedings of SIGCHI. New Orleans, USA. 1991