Visualization Design Patterns for Ultra-Resolution Display Environments

Khairi Reda, Jillian Aurisano, Alessandro Febretti, Jason Leigh, and Andrew E. Johnson

Electronic Visualization Laboratory
University of Illinois at Chicago

***Abstract*—The past 10 years have seen great advances in visualization infrastructure. From large, ultra-resolution displays (URDs) to new interaction devices, modern visualizations technologies hold the promise of brining about a new generation of visual analytic tools that are more capable of tackling big-data challenges. Yet, there is a lack of understanding among visualization designers on how to adapt existing design paradigms to take advantage of these new platforms. In this paper, we look at commonly used visualization design patterns and give guidelines on how to adapt them to URDs. We also describe two example visual analytic tools which leverage our adaptations to enable the analysis of big datasets on URDs.**

***Index Terms*—Ultra-resolution displays; design patterns; visual analytics;**

I. Introduction

Modern visualization practice is predicated on decades-old infrastructure. Small computer monitors with limited resolution and restricted interaction modalities continue to be the principle technological components on which we base our visualization design principles. However, over the last ten years, we have seen great advances in visualization infrastructure marked by the emergence of new technologies, including large, ultra-resolution displays (URD), new interaction devices (e.g. multi-touch surfaces and *Microsoft Kinect*), and low-latency, high-bandwidth computer networks. This was accompanied by a substantial cost decrease, which encouraged users in research labs and industry centers to quickly adopt them. For example URD walls have now become a standard piece of equipment in many research labs, often used to visualize and juxtapose big scientific datasets [1]. The latest generation of these displays provide the ability to seamlessly juxtapose 2D and 3D contents, enabling the creation of hybrid visualizations [2].

Despite these technological developments, the current generation of visualization systems (especially commercial and scientific visualization packages) have not been designed to take advantage of such new infrastructure. For example, most visualization packages are cumbersome to use in URDs as they often employ non-scalable visual representations and rely on interactive features that are intended for use on desktop and laptop screens. Furthermore, many visualization techniques were specifically conceived to work around limitations in display size and resolution. For instance, many visualization tools rely extensively on multi-scale navigation techniques (as epitomized by the *overview first, zoom and filter, then details-on-demand* mantra [3]) to compensate for the limitation in display resolution. Multi-scale navigation, however, can be detrimental to user performance in many common tasks, such as visual comparison [4]. Another alternative is coordinated multiple views, which is commonly employed, but often not pushed to its full potential as it uses up precious screen estate.

Existing design paradigms are also inadequate when it comes to designing new visual analytic tools that leverage the affordances of modern visualization infrastructure. For instance, one of the key features of URDs is their ability to support collaborative analysis by multiple co-located users. Yet, very few visualization systems provide explicit support for collaborative visual analysis.

Given the new technology landscape, we need to adapt our design guidelines to take better advantage of emerging visualization infrastructure. This includes re-examining existing design guidelines, creating new ones, and possibly scrapping some of the existing ones as no longer necessary or applicable.

With the increasingly popularity of URDs, it is time to look at how they reshape the visualization design space. The goal of this short paper is to examine some commonly used visualization design patterns and predict how they might perform on URDs. Importantly, we shall consider the **leverage points** and **challenges** associated with existing designs when they are 'stretched' to a large, high-resolution format. We also describe two exemplifying visual analytic tools that were specifically developed for URDs, and illustrate how we put into practice some of the presented design guidelines. The design patterns mentioned here are by no means complete, but are rather intended as a proposal for how we can systematically investigate and understand the benefits of URDs in the future.

II. Design patterns for ultra-resolution visualizations

Software design patterns are used in the development of all major software systems [9], and have greatly influenced the design of programming languages and programming toolkits (such as Java and Qt). Visualization experts have also begun to identify a number of recurring designs that are applicable in different domains and analysis scenarios. For instance, Visual Thinking Design Patterns (VTDP) is a concept developed by Colin Ware as a framework for structuring the best practices in visualization, based on cognitive and perceptual principles [5]. VTDP present a good way to start when thinking how to adapt visualizations to URDs.

TABLE I
ADAPTING WARE'S VISUAL THINKING DESIGN PATTERNS TO ULTRA-RESOLUTION DISPLAYS

Ware's Visual Thinking Design Patterns [5]	Adaptations as we move from small low-resolution monitors to URDs	
	Leverage points	**Challenges**
Visual query: The user scans the display for a target object that matches a visual pattern in his/her working visual memory. The pattern could be previously known or imagined by the user. For instance, a user scanning a weather map looking of a lightening symbol.	• A larger number of items can be displayed and queried on a large display. The increase in time to complete the task is sub-linear relative to the data increase factor, if the target pattern is pre-attentive. • The search can leverage embodied navigation (e.g. head turns, walking up to the display) as opposed to virtual navigation (e.g. zooming and panning). The increased use of physical navigation has been correlated with improved user performance in some tasks [6].	• The search will become serial and will take significantly more time, if the target is not sufficiently distinguishable by one elemental visual feature (e.g. glyph color, shape, or size). • **Workaround:** Leverage pre-attentive processing by designing glyphs to be distinguishable by one elemental visual feature (color, orientation, etc). If complex glyphs are needed to encode multiple attributes, consider adding a second coordinated view and dividing the attributes between the two views.
Visual aggregation: When the information space is big, it cannot be represented all at once on a small display. A hierarchical visual representation is often employed here to aggregate data into clusters, and the user is given the ability to drill down into specific clusters by means of virtual navigation (e.g. zooming with the mouse wheel or clicking on clusters).	• On URDs, the hierarchy can be flattened to show all the data elements at once, reducing the need for virtual navigation. Visual aggregation can be achieved by stepping away from the display to see the 'big picture'	• Achieving visual aggregation by stepping away from the display requires the use of visual glyphs that aggregate [7], which may restrict the choice of visual encodings. • It may be difficult to achieve more than two levels of detail with physical navigation.
Comparison in large information spaces: Often, complex objects (e.g. molecular structures, genomes, galaxies, results of different simulation runs) have to be examined and compared to derive insight. Plumlee et al. suggest the use of persistent magnifying windows instead of zooming interfaces, when the object of comparison is complex enough that it cannot fit in the visual working memory [4]. On a small display, however, only a limited number of windows (typically 4) can fit on the screen.	• An URD allows for a large number of views to juxtaposed. A large set of objects to be displayed in a small-multiples layout, for instance. Complex objects can be compared on multiple criteria by moving one's eye or turning one's head, which eliminates the need to switch between multiple windows.	• Visual comparison may still be difficult, given the complexity of the objects being compared. • **Workaround:** The elements or aspects being compared need to be highlighted or linked. For example, if one is concerned with comparing the center of galaxy discs in an ensemble, then only galaxy centers need to be highlighted and the rest of the image can be dimmed. This allows the user to highlight and focus on the relevant features while still keeping those features in context.
Integration across views: Coordinated multiple views represent a common design strategy in visualization. Coordinated brushing and/or filtering is often employed to facilitate information integration across the different views.	• Using an URD, a larger number of views can be juxtaposed. The views may show heterogeneous data [2], or homogeneous visual representations (i.e. small-multiples) [8].	• There is a potential for information overload here. • Coordination is more challenging, as the user have to comprehend the results of his/her selection and filtering across a larger number of views • **Workaround:** Employ pre-attentive encodings to highlight brushed items. Organize views in a semantically meaningful way. For example, relevant views can be grouped into workspace and given a slight background tint for easy identification.
Lateral exploration: Starting with a particular piece of information, the user 'fans out', laterally exploring related elements and following up a series of links. For instance, exploring a terrorist social network by following the chain of links starting from a suspected terrorist.	• This pattern can be generalized to include the exploration of multiple hypotheses and/or narrative. Using a URD, multiple instances of the same visualization can be spawned, juxtaposed, and explored in parallel on the large display. This can potentially encourage users to consider and contrast multiple hypotheses and narratives before drawing conclusions.	• There is potential to overwhelm the user with too much information.

In this paper, we will look at Ware's VTDP and consider how they perform when they are 'stretched' to a large, high-resolution display in order to increase the amount of data represented. Our interest is to understand to what degree do Ware's VTDPs address visualization problems in URDs, and to what degree do new design patterns need to be articulated. Each VTDP contain one or more cognitive tasks. It also describes a visual form that can help accomplish the said tasks. Some also include a possible set of interactions (e.g. filtering and highlighting), which can be initiated by the user. Taken together, the components of a VTDP describe a common design for solving recurring problems in visual data analysis.

The description of VTDPs is somewhat decoupled from the technology so as to be as generic as possible. However, it is possible to ask the following question: given a particular VTDP, what benefits and drawbacks can we expect if we were to apply this design pattern to render more data elements on a large, ultra-resolution display? We will therefore look at some of the VTDP identified by Ware et al. and analyze the potential gains and costs that we can expect when the visual layout is stretched to a big display. By stretch we imply re-rendering the visualization to take advantage of the extra resolution afforded by the URD, while still being able to visually resolve individual glyphs (by a human eye with 20/20 vision). The patterns are illustrated in Table I: we list Ware's VTPDs in the first column and describe how they can be adapted to URDs in the second and third columns. The order of presentation starts from low-level patterns, which rely mostly on perceptual processing (e.g. visual query), to higher-level patterns, which apply to cognitive tasks (e.g. lateral exploration).

The advantage of this methodology is that we are starting with known designs that are thought to exemplify the best practices in data visualization. The downside is that we may miss entirely new design patterns that are unique to URDs, which could potentially require a radical departure from existing designs. Nevertheless, given the clear gap in knowledge, it is reasonable to start with the existing design space and see how it can be adapted to modern infrastructure. In this short paper, we limit the discussion to five of Ware's design patterns, which we believe to have the most utility when applied to URD environments.

III. Example applications

To appreciate how the above design patterns can be used to create compelling visualizations for URDs, we can begin by looking at successful examples and try to understand whether and how they leverage some of the presented patterns. Here we describe two such visual analytic applications, which were specifically designed for URD environments, in the domains of behavioral ecology and comparative genomics.

A. Analysis of insect behavior

The study of insect behavior poses many challenges. Insects exhibit stochastic, locally scoped movement patterns that are difficult to characterize case-by-case. To understand their behavior, ecologists resort to collecting a large sample of motion trajectories under varying conditions to tease out general responses. The sheer number of trajectories collected during experimentation makes them extremely difficult to visualize on traditional displays. To address these challenges, we used a 3D URD wall to visualize a large dataset of Seed Harvester ant trajectories collected from video sequences. Figure 1 illustrates the visualization tool. We encode the insect's 2D motion on the *XY* plane (the display's surface) and time using the *Z+* axis (with 3D stereoscopic depth). Figure 2 illustrates this encoding.

We provided two interactive features to enable researchers to explore hypotheses concerning ant behavior, and quickly determine whether those hypotheses are supported by the data. First, a coordinated brushing tool allows the user to brush the background of a single trajectory. This causes the visualization to highlight motion segments in all other trajectories when the insect moves over the brushed area. For example, the researcher could brush the west (left) side of the trajectory to highlight all instances where the ant exited the experimental arena from the west (see Figure 2). Second, a temporal filter lets the user specify a filtering time window. This causes the visualization to display trajectory segments corresponding to insect motion within the specified duration only, such as the first 30 seconds of the experiment.

Design patterns used:

*1) **Comparison in large information spaces**:* We employ a small-multiples layout to allow for visual comparison across a large number of trajectories. In our setup, approximately 500 trajectories can be displayed simultaneously at sufficient resolution. The screen space can be divided into configurable bins that group related trajectories. The bins are also given distinct background tint to easily distinguish them. For example, one bin might show trajectories of ants captured east of the colony's main foraging trail, whereas a second bin might contain ants captured on the trail while carrying a seed. The coordinated brush and temporal filter operate on an entire bin, and all trajectories in a particular bin will react collectively to either operations.

Because the trajectories are persistently available on the screen, the researcher is able to compare them using combinations of head turns and quick eye movements, eliminating the need for virtual navigation. However, because of the sheer number of trajectories and their relative complexity, finding consistent similarities and differences across the entire layout is still difficult. The coordinated brush and temporal filter serve to alleviate this difficulty by allowing the user to highlight the feature he/she is interested in. For example, if the researcher is interested in the insect's behavior at the beginning of the experiment, the temporal filter can be set to clip the trajectory and show only the first minute and clip the rest. This allows the user to reduce the visual complexity of trajectories and focus on a particular feature or aspect, which simplifies comparison.

*2) **Integration across views**:* Once the relevant trajectories are laid out, visual integration across views must take place. In this particular application, the researcher wanted to identify navigational strategies that are used consistently by Seed Harvester ants under specific conditions. For example, one hypothesized strategy presupposes that ants rely on a sun compass to orient themselves in a direction that would lead them back to the colony's trail. To corroborate this strategy against the data, the researcher created four different bins comprising ants captured east, west, north, and south of the trail. The researcher then employed the coordinated brush to highlight the expected exit side in each bin separately (left for the 'east' group, down for 'north', etc...). The researcher observed that a majority of trajectories in each bin contained a color highlight, indicating a consistent strategy across each group (see Figure 2). The researcher therefore concluded that her hypothesis is indeed supported by the data. An important factor in the design, which helped the user in integrating the patterns across a large number of views, is the use of pre-attentive highlighting. We employed primary colors to highlight brushed feature (red, blue, etc...), which were sufficiently distinct from the trajectory's color (black). This made it possible for the user to easily distinguish the feature of interest, focus on it, and compare that feature in a large number of views. This use also falls under the **'visual query'** design pattern. However, rather than querying for simple features, the user was able to search for and detect complex features, such as loops and zigzag motion patterns, owing to pre-attentive highlighting [8].

*3) **Lateral exploration**:* Our collaborator was interested in exploring a wide variety of hypothesized navigational strategies and behavioral patterns, and seeing whether the data support any of them. The researcher frequently utilized the binning tool, grouping different trajectories that were acquired under different experimental conditions in separate bins. This enabled her to operate on these groups separately, with each bin serving as its own independent visualization. Interestingly, each of the bins served as a placeholder for one behavioral pattern being investigated, while simultaneously encapsulating all the relevant trajectories. This usage scenario falls under the 'lateral exploration' pattern, with the large display serving as a canvas where multiple (and somewhat independent) visualizations can be juxtaposed and explored in parallel. This encouraged the user to perform a lateral exploration of the hypothesis space and switch between different hypotheses [10], which is important but often difficult to do with traditional visual analytic tools. We believe URDs can promote this behavior by enabling the analyst to juxtapose a large number of views, and effortlessly switch between them.

B. Large-scale comparative genomics

Advances in genome sequencing technology have resulted in a substantial decrease in sequencing costs. Consequently, biology researchers are defining new avenues of research around large-scale genomic sequencing. From the 1000 genome project which examines variations in human genomes, to the industrial identification of valuable genes in bacteria, these datasets place new demands on visualization infrastructure. Existing genome visualization approaches are largely designed for the comparison of a few genomes at once on conventional, low-resolution displays. Yet, what scientists need is the ability to simultaneously analyze and compare hundreds or even thousands of sequenced genomes.

To address these challenges, we developed a novel visualization tool for the comparative analysis of gene neighborhoods in microbes to identify commonly recurring sets of genes across related species. We employ a 'high-density' visual representation which depicts sequences within a *contig* (a contiguous stretch of assembled sequence data) as arrows. The length of the arrow corresponds to the size of gene in nucleotides and its direction corresponds with the strand of gene and the direction of transcription. This minimalistic encoding represents the needed information in a compact form, with each row representing the entire genome sequence of one microbial species. This in turn allows us to visualize hundreds of genomes at once using an URD (see Figure 3).

Design patterns used:

1) ***Visual aggregation****:* To facilitate comparison across a large number of genomes, we included a number of interactive features that would allow researchers to layout the data in a way that relates to their queries and hypotheses. For example, 'gene neighborhood targeting' allows researchers to select a gene of interest (the target gene), and observe variations in the neighborhood of the target gene across hundreds of genome sequences. This is achieved by moving the targeted gene up to the top of the screen and stacking all contigs containing this gene beneath the target sequence. The targeted gene is centered and given a color. Adjacent genes are colored according to a gradient, which is also applied to orthologous genes in all the other genomes. This effectively creates a custom 'genomic map' which can be used by the researcher to answer his/her specific question. Instead of relying on hierarchical navigation (as often employed in other comparative genomics tools), we rely on physical navigation and exploit the tendency of visual features to aggregate when viewed from a distance. The researcher can step away from the display to look at the entire genomic map, which causes the individual genes to aggregate. Target gene neighborhoods, however, remains prominently visible owing to the color gradient, which allows the researcher to quickly discern variations and similarities in the entire genomic map. Figure 3 depicts gene neighborhood variation in genome sequences of 600 *E. coli* species in one view. Conversely, one can look at a small region of interest in detail by simply walking up to the display.

IV. Conclusions and Future Work

In this paper, we looked at how ultra-resolution display (URD) environments reshape the design space of visualizations. Starting from a set of commonly used visualization design patterns, we attempted to predict how they might perform when 'stretched' to a large, high-resolution format. We also described two example visual analytic tools for URDs and discussed how they put into practice the presented design patterns.

Given the great diversity in the visualization design space, there are surely more patterns to be articulated. Although the patterns discussed here are derived from commonly employed visualization techniques, we believe there are entirely new sets of patterns unique to URDs that are waiting to be discovered.

References

[1] J. Leigh, A. Johnson, L. Renambot, T. Peterka, B. Jeong, D. Sandin, J. Talandis, R. Jagodic, S. Nam, H. Hur *et al.*, "Scalable resolution display walls," *Proceedings of the IEEE*, vol. 101, no. 1, pp. 115–129, 2013.

[2] K. Reda, A. Febretti, A. Knoll, J. Aurisano, J. Leigh, A. Johnson, M. Papka, and M. Hereld, "Visualizing large, heterogeneous data in hybrid-reality environments," *Computer Graphics and Applications*, vol. 33, no. 4, pp. 38–48, 2013.

[3] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, 1996, pp. 336–343.

[4] M. D. Plumlee and C. Ware, "Zooming versus multiple window interfaces: Cognitive costs of visual comparisons," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, no. 2, pp. 179–209, 2006.

[5] C. Ware, W. Wright, and N. Pioch, "Visual thinking design patterns," in *Proceedings of the 19th International Conference Distributed Multimedia Systems (DMS '13)*, 2013.

[6] R. Ball, C. North, and D. Bowman, "Move to improve: promoting physical navigation to increase user performance with large displays," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007, pp. 191–200.

[7] A. Endert, C. Andrews, Y. H. Lee, and C. North, "Visual encodings that support physical navigation on large displays," in *Proceedings of Graphics Interface 2011*, ser. GI '11. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society, 2011, pp. 103–110. [Online]. Available: http://dl.acm.org/citation.cfm?id=1992917.1992935

[8] K. Reda, A. Johnson, V. Mateevitsi, C. Offord, and J. Leigh, "Scalable visual queries for data exploration on large, high-resolution 3d displays," in *Proceedings of the 2012 SC Companion: High Performance Computing, Networking, Storage and Analysis*. IEEE, 2012, pp. 196–205.

[9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: Abstraction and reuse of object-oriented design*. Springer, 1993.

[10] D. Klahr and K. Dunbar, "Dual space search during scientific reasoning," *Cognitive science*, vol. 12, no. 1, pp. 1–48, 1988.

Fig. 1. Analysis of insect motion patterns on a 3D ultra-resolution display wall with a resolution of 19 Megapixels. Trajectories are juxtaposed in a small-multiples layout and grouped into bins depending on their associated meta data. The bins are given distinct background tint to distinguish them easily. The use of independent bins allows the user to explore multiple hypotheses in parallel.

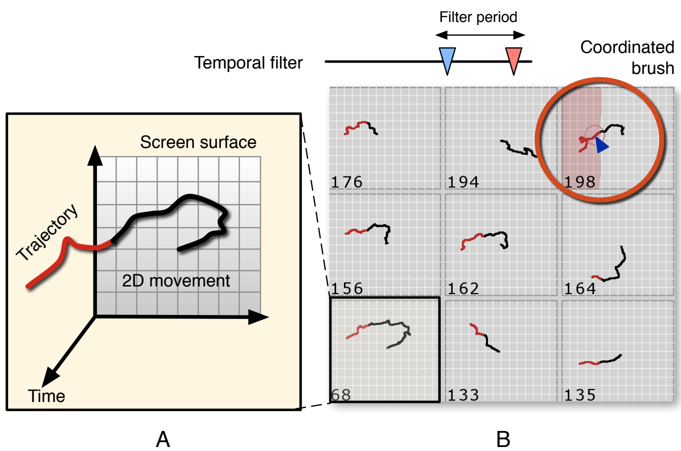

Fig. 2. **(A)** Visual encoding of a single trajectory. Stereoscopic depth cues are used to convey time. **(B)** The coordinated brush tool (top right) along with the temporal filter (top center) can be used to visually test hypotheses corresponding to spatio-temporal behavioral patterns. In this example, the researcher checks whether "Ants that were captured east of the colony's foraging trail will exit the experimental arena from the west side." A red highlight in majority of trajectories indicates the hypothesis is supported.

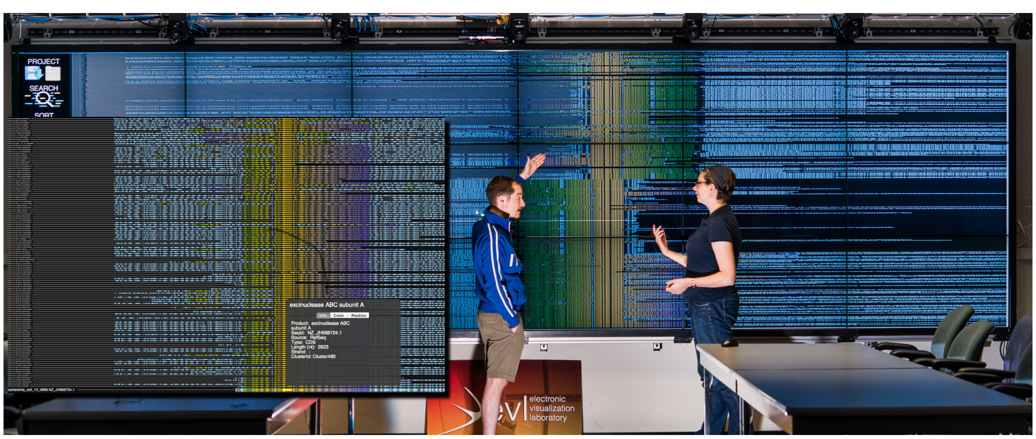

Fig. 3. Comparative analysis of a targeted pathogenicity gene across 600 *E. coli* species. The application of color gradient allows users to quickly discern variations and similarities across hundreds of genomes by stepping back from the display to see the big picture. The inset shows a closeup view of the visualization and illustrates the 'high-density' representation, which exploits the tendency of small visual features to visually aggregate, when viewed from distance.