Chapter 7

Knowledge Visualization

Thought is often associated with both language and vision. Research shows that for some thinking may be primarily visual while for others thinking occurs more naturally through language and sound [7-1]. Psychologists have found the ability to form mental images helps us to understand abstract concepts and logical relationships more easily. In many ways, thinking is a combination of both verbal language and visual imagination.

Tufte demonstrates, through clear examples, the ways in which visualization is present in every day life. We find that information visualization, historically speaking, is closely connected to design and communication. The formal qualities of space, shape, density and color are equally present in visualization as in communicative design [7-2]. We might ask: What are the significant differences between visual design, communication and information visualization?

In chapter two, information visualization and knowledge visualization are presented as a difference in approach between large-scale dataset, such as simulations of clouds and water which posses a low data-semantic ratio, and data with a high-semantic content as exemplified by encyclopediae and written text. The data-semantic ratio represents one measure of this relationship. But nothing in the original definition refers to the visual aspect of these fields:

> *Data-Semantic Ratio:* This is the ratio of the number of attributes of each individual element to the number of elements in the whole for some set of information.

How does the visual aspect enter into the study of data and semantics? Both information visualization and knowledge visualization appear to posses some qualities of visual thinking, but the definition above does not require it. This is the primary concern of the present chapter. Through specific examples, relationships between data and semantic content and between linguistic and visual thinking will be investigated in more detail.

7.1. Informational Visualization and Design

One view of scientific and information visualization is that they arise from the historic need for visual understanding of scientific research through human perception. Colin Ware states:

> "On the one hand, we have the human visual system, a flexible pattern finder, coupled with an adaptive decision-making mechanism. On the other hand are the computational power and vast information resources of the computer. Interactive visualizations are increasingly the interface between the two. Improving these interfaces can substantially improve the performance of the entire system." [7-3]

In this paradigm, the human is viewed as equivalent to machine but with a different set of functional abilities. The mention of improved performance most likely refers to scientific understanding in which the chapter is set. With regard to Information Aesthetics, however, this argument avoids the possibility that the communicative goal of visualization does not necessarily need to be scientific. It may also be socio-critical or self-reflective.

Regardless of the goal, however, it seems that a better interface design will improve communication. Tufte shows how this is possible, but also demonstrates that visual design can also mislead the viewer to draw incorrect conclusions. The challenge is the way in which one uses space, color and other visual cues to explore data.

> "Arbitrary, transient, one-sided, fractured, undocumented materials have become the great predicament of image making and processing. How are we to asses the integrity of visual evidence? What ethical standards are to be observed in the production of such images?" [7-4]

Misleading designs need not be intentional. Often, information visualization is used to communicate complex information to an audience unfamiliar with the subject material. In many cases this is likely to be an intractable problem. Nonetheless, one of the compelling aspects of visual communication is its ability, in a single image, to allow the viewer to selectively choose what will be learned. Figure 7.1. shows the a concise view of the phases of the moon, and the planets jupiter and saturn by Alexander Jamieson, 1820.

A central theme in information visualization that enables understanding - distinct from other forms of graphics design - is the concept of a *cognitive map*. Traditionally, a map is a visual representation of a geographic space. There is a one-to-one relationship between space on a traditional map and physical space. However, examples such as the London Underground subway by Harry Beck [7-5] shows that maps need not correlate to physical space. A *cognitive map* allows us to form mental images of non-physical qualities and map them to spatial metaphors [7-6]. Thus a map, speaking abstractly, is not simply a tool but a general principle for visualization. Maps allow us to see non-spatial concepts spatially.

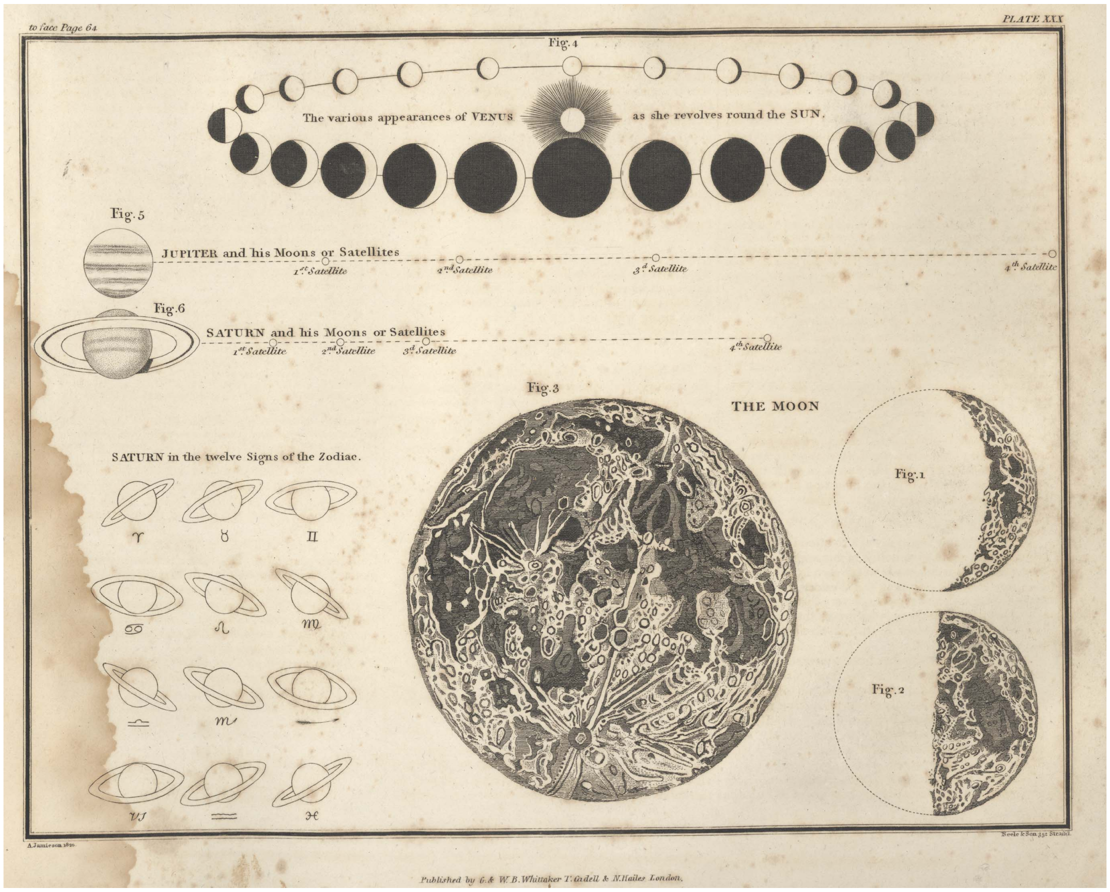


Figure 7.1. Celestial atlas, A. Jamieson, 1820

A specific map, however, is always a map of *something* in some space. In this respect it is formally constrained to the specific concept being portrayed. One of the unique aspects of cognition is our ability to easily jump from one conceptualization to another. While a map of the internet, for example, conveys the *complexity* of the internet as a fixed image it does not reveal the detailed layers of meaning present in it. The only way to explore the meaning present in a rich structure is to be able to navigate quantitative and semantic dimensions simultaneously.

Thought is highly associative. Consider the London Underground Subway of Beck [7-5]. Due to Beck's layout, one can easily identify the sequence of stations that will be arrived at. But who has not wondered how such tracks are laid without intersecting? While Beck eliminates geographic distance in order to more clearly convey useful information, one might like to know why the stops take progressively longer as the Bakerloo line exits the city to the west. Which of the London subway lines was the first to be built? Have any been rebuilt? Can fish near a river tunnel feel when a subway passes? Which fish are more susceptible? These are the questions that often arise while in transit.

Semantic knowledge consists of the complexity of relationships found in the natural associations of thought. To explore the associations above, one must still consult a number of different sources. A map, either cognitive or spatial, is a representation of a single structure within this much larger map of human cognition. This chapter investigates how we might navigate and visualize general associative semantic spaces in a single framework.

7.2. Quantitative and Qualitative Data

In general, studies of the natural world are neither purely quantitative nor purely qualitative but contain aspects of both. Consider the atomic elements as an example. An element is defined by its atomic weight and number (the number of protons in its nucleus). Mendeleev discovered a unique, periodic pattern in the atomic weights of the elements which corresponded to their properties.

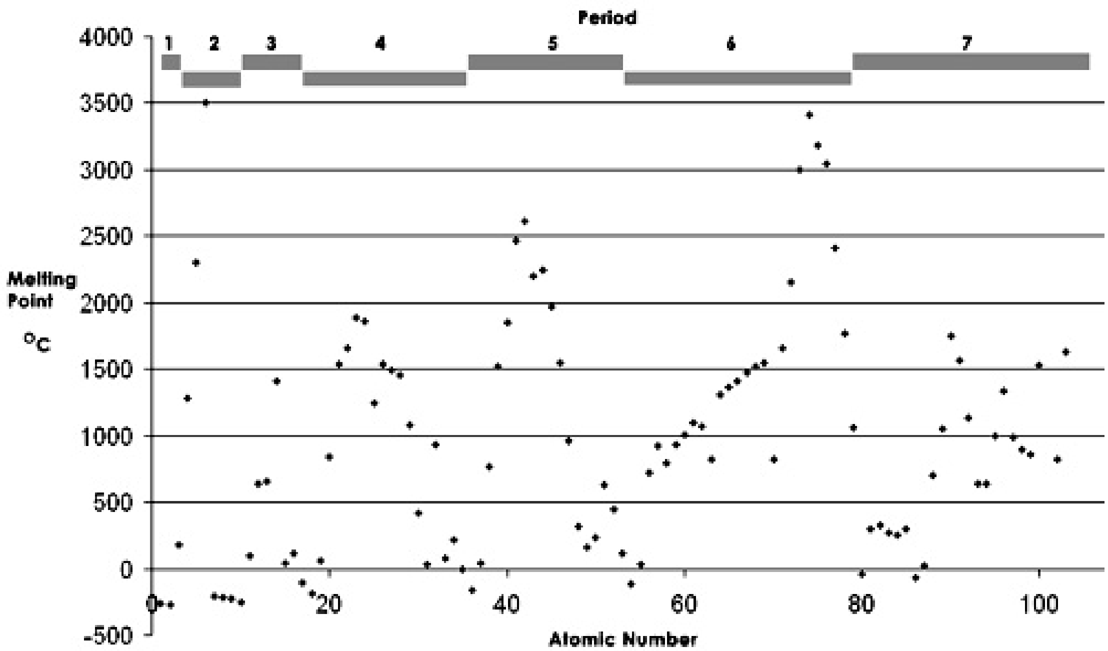
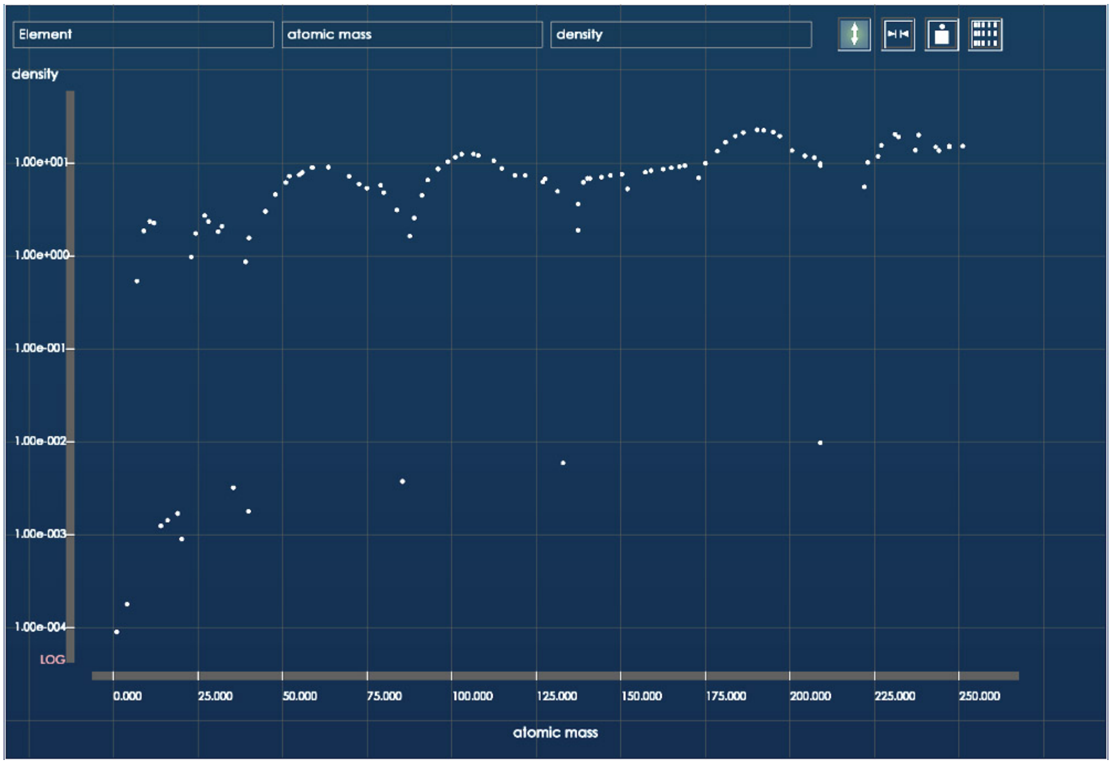


Figure 7.2. Atomic number versus melting point for the chemical elements.

Using modern values, it is possible to see the periodic pattern in one of these properties. Melting point versus atomic number for each of the elements is shown in Figure 7.2. The *quantitative* data are the values of the attributes themselves. However, the elements are now known to have many other properties in addition to those Mendeleev examined. One might ask: What other properties show periodic relationships to atomic mass? Are there any properties which do not have periodic relationships? Are their any unusual correlations between any two properties? To answer these questions requires that we use both the values of the data and their qualitative semantics.

A unique graphing system was developed for Quanta in which visualization is directly linked to semantic data sets. There is no need to import data, select, and then plot it. The interface is designed so that the user simply selects the objects to be plotted and the system completes the process of loading data, organizing and visualizing it. The tools are similar in spirit to Hans Rosling's Gapminder software, which allows any two world economic statistics to be plotted against one another [7-7]. Here, similar functions are possible except that the graph is linked to a hypergraph database. This provides a generic structure for knowledge and allows features of any object to be plotted. Connecting to values in the database facilitates interdisciplinary research as objects in different fields can be compared in the same system.



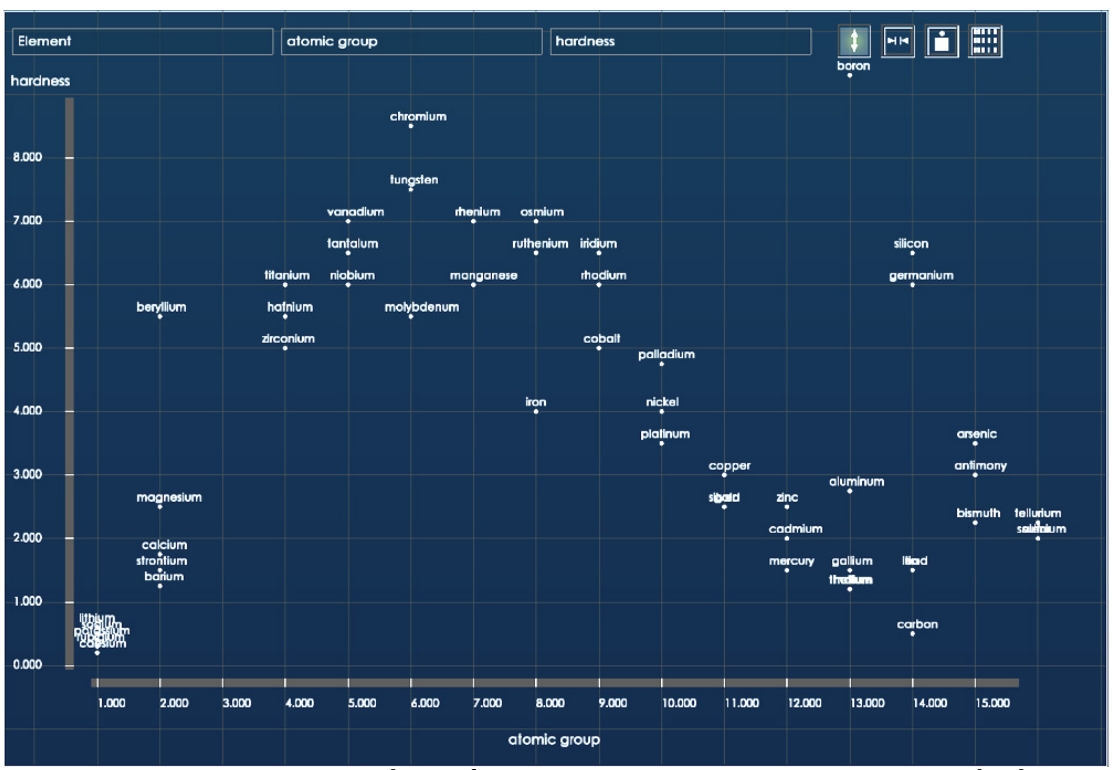
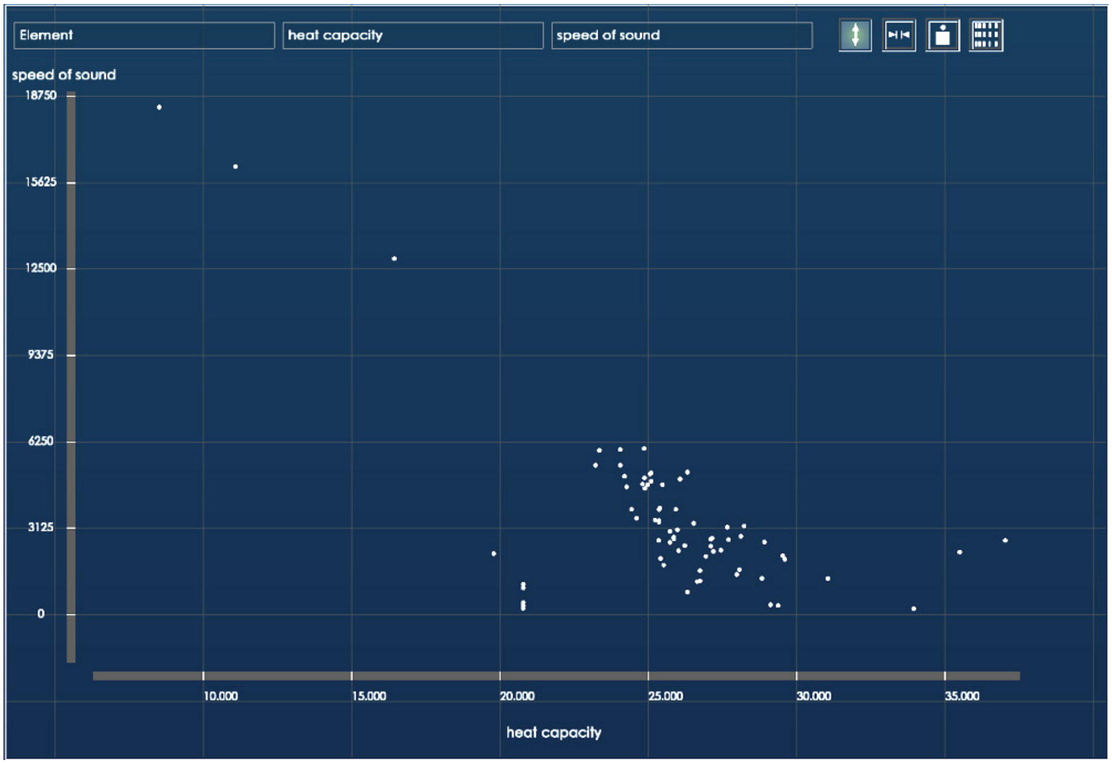


Figure 7.3. Atomic mass versus density for the chemical elements.

Figure 7.4. Atomic group versus hardness for the chemical elements.



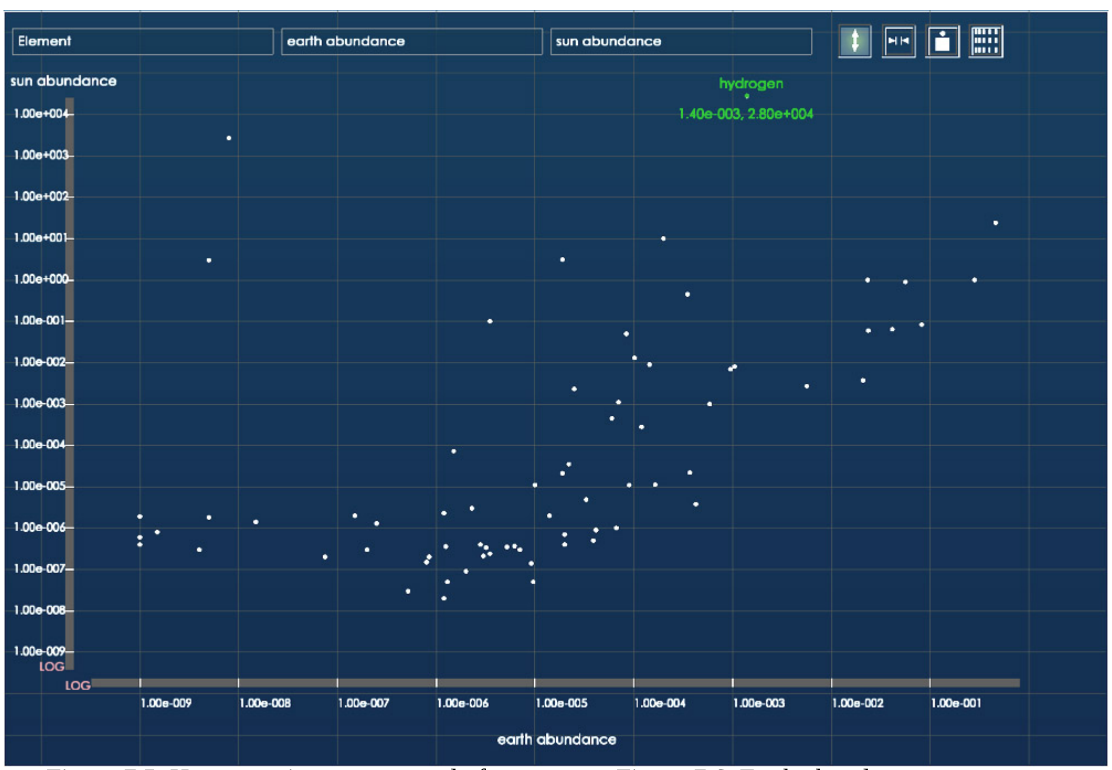


Figure 7.5. Heat capacity versus speed of sound for the chemical elements.

Figure 7.6. Earth abundance versus sun abundance for the chemical elements.

Several examples are shown in Figure 7.3 thru Figure 7.6. The first shows a log plot of atomic mass versus density (Figure 7.3). The upper group are the solids (higher density) while the lower group are the gases (low density). The periodicity with atomic mass is clearly visible. The second plots show atomic group versus hardness (Figure 7.4). Notice the interesting rise in hardness for the semi-conductors (germanium and silicon). The third graph shows heat capacity versus speed of sounds for the elements (Figure 7.5). To one unfamiliar with chemistry there is a non-obvious inverse correlation here. This can be explained by the degrees of freedom in the atomic structure [7-8].

The last graph is a correlation of the abundance of elements in the sun versus on the earth (Figure 7.6). Once can immediately notice the lighter elements present in both, the heavier elements present only in the earth (due to the Earth's crust), and the rarity of some elements in both (such as gold).

All four graphs were created in less than one minute. At the top left of the graph area the interface includes a text box to enter the object of study, in this case "Elements", and two combo boxes to select the properties for the X and Y axes respectively. Additional buttons allow the user to toggle labels, image icons, and to generate log plots.

This graphing system requires only four mouse clicks to generate a different view of the data. The goal is to develop tools which allow one to move fluidly between various visual representations. Using existing graphing software, it would take several minutes to indicate the data to be graphed, initiate a graph type, select the proper range of values, and annotate the graph.

Algorithmically, when a user selects a new object of study the graph automatically determines the properties that are available to be plotted by scanning the data for numerical attributes. The graph has direct access to the required data. These properties are then included in both X and Y combo boxes as plotable values. The system generates the graph dynamically by pulling out only the relevant data needed for the plot from the much larger semantic network of the Quanta hypergraph.

Graphing is one of the most basic visualization techniques. While the system presented can plot any two semantic qualities it does not yet support multi-dimensional plots, overlaid plots of the same features for different objects, or other graphing formats such as pie or bar charts. These extensions could be easily incorporated.

This graphing system demonstrates that directly linking data to visualization opens new possibilities for how we explore and find meaning in what we study. When semantic meaning and quantitative data are simultaneously present in the same system, it is possible to go from data to visualization immediately. Rather than bring the data to the graph, we can select an object of study and the graph itself investigates the database and *tells us* what is available.

7.3. Trees

Another classical problem in information visualization is the display and navigation of trees [7-5]. Visualizing large trees is difficult because the lower levels may contain a great number of nodes with respect to their parent nodes. One of the most common techniques for displaying a trees structure is the indented list.

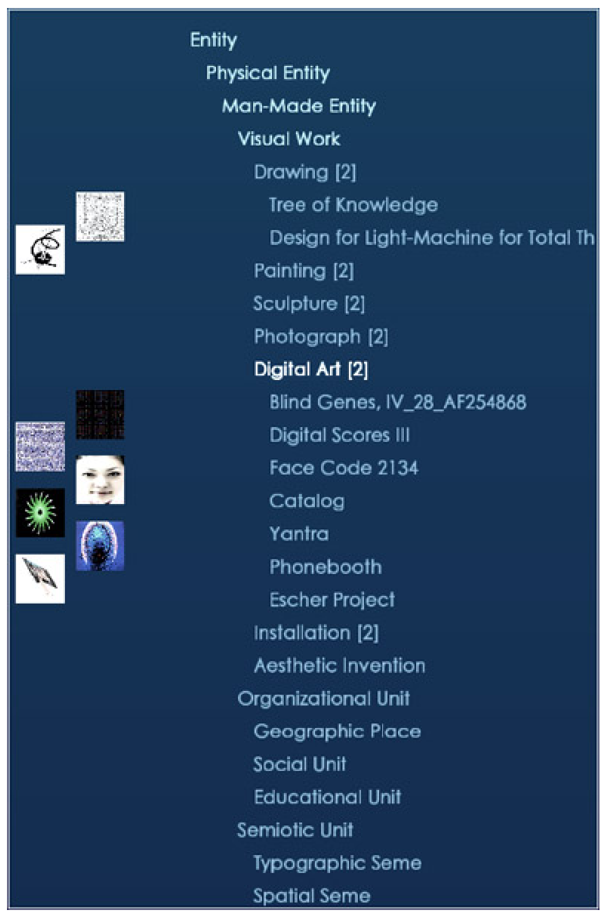


Figure 7.7. Indented lists, with thumbnail icons, in the ontology viewer of Quanta.

An example of an indented list used to display the Quanta generic ontology is shown in Figure 7.7. This list is used in conjunction with a *document view* that allows a user to navigate a set of concepts. This implementation can handle very large trees since only currently open nodes are stored in memory. As the user changes focus, the previous nodes are collapsed, the new focus nodes opened, and the list smoothly pans to the new location.

A list is not necessarily the ideal way to view trees, however. Robertson explores a visualization approach that uses Cone-Trees to provide smooth 3D rotational navigation of a hierarchy [7-9]. Cone-trees, unlike lists, allow the user to see multiple parts of a tree simultaneously (all children of siblings). However, this introduces some problems with cluttering of nodes at a given level. Perhaps related to this, McKenzie *et al.* found that cone trees did not significantly out perform tree lists for some navigational tasks [7-10].

Another alternative, Treemaps, are constructed by recursively subdividing rectangular spaces horizontally and vertically [7-11]. Treemaps have the beneficial property that nodes are assigned areas relative to the size of their subtrees. One critical feature, however, is that treemaps often result in node aspect ratios in which content is difficult to display. Yet a third visualization of trees is hyperbolic layout. Both two-dimensional [7-12] and three-dimensional [7-13] versions are possible. Hyperbolic layout allows the user to see the overall structure of the entire tree, with smooth navigation, but requires spatial distortions to arrange nodes.

A new type of visualization of large hierarchies is introduced in Quanta using the concept of circle packing [1]. The idea is motivated by the observation that circle packing solves the problem of placing the most number of elements in a small space while also preventing overlap. In addition, unlike Treemaps, circles all have identical aspect ratios thus making them better suited to conveying content. Applied hierarchically, each set of children is packed inside the circle of its parent.

[1] A week after the author's implementation of circle packing was completed as described here, similar work by [7-14] was found in SIGCHI 2006. The essential differences are the packing alogrithm (theirs is more efficient and accurate) and the use of dynamic tree construction from an underlying network (present here but not in their implementation).

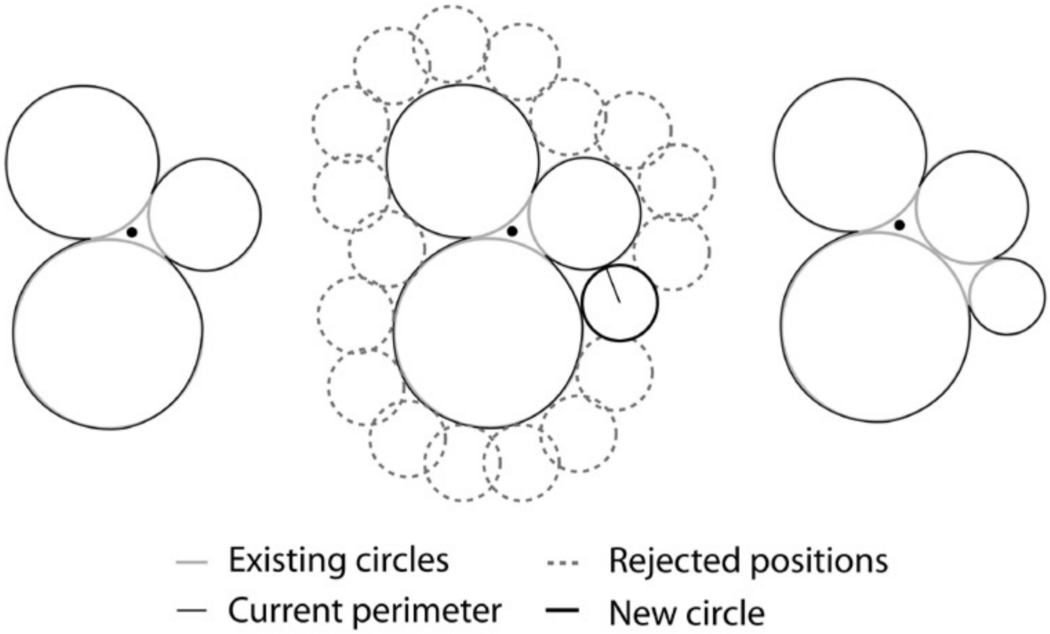


Figure 7.8. Circle packing algorithm. Starting with an existing perimeter (left), a position for a new circle is found along the perimeter that is closest to the center of the group (middle) , resulting in a new perimeter (right). The perimeter function is stored in polar coordinates.

The circle packing algorithm itself consists of several steps. First, a polar perimeter function is defined as the current outer boundary of the packing. A new circle is then located on the boundary by minimizing its distance to origin while being constrained not to collide with other circles. Collisions are detected using a local search of the perimeter function. Finally, the new circle is added to the perimeter function by taking its maximum extents from the origin. This algorithm is shown in Figure 7.8. While it does not produce optimal packings, the algorithm can efficiently handle thousands of nodes in real-time.

An example of the circle packing visualization for the Linnean sub-taxonomy of the Quanta ontology is shown in Figure 7.9. Another example in Figure 7.10 shows image icons placed at circle nodes rather than text.

The primary benefits of circle packing for visualizing trees are its efficient use of space and the same aspect ratio for each node. In addition, like Treemaps, the area of each circle corresponds to information content (or some other desirable property). Sorting of nodes by size before packing allows more critical nodes to be placed at the center of the field of vision. The drawbacks to circle packing are related to the circle being an non-ideal shape for containing textual content and the fact that gaps between circles are not utilized.

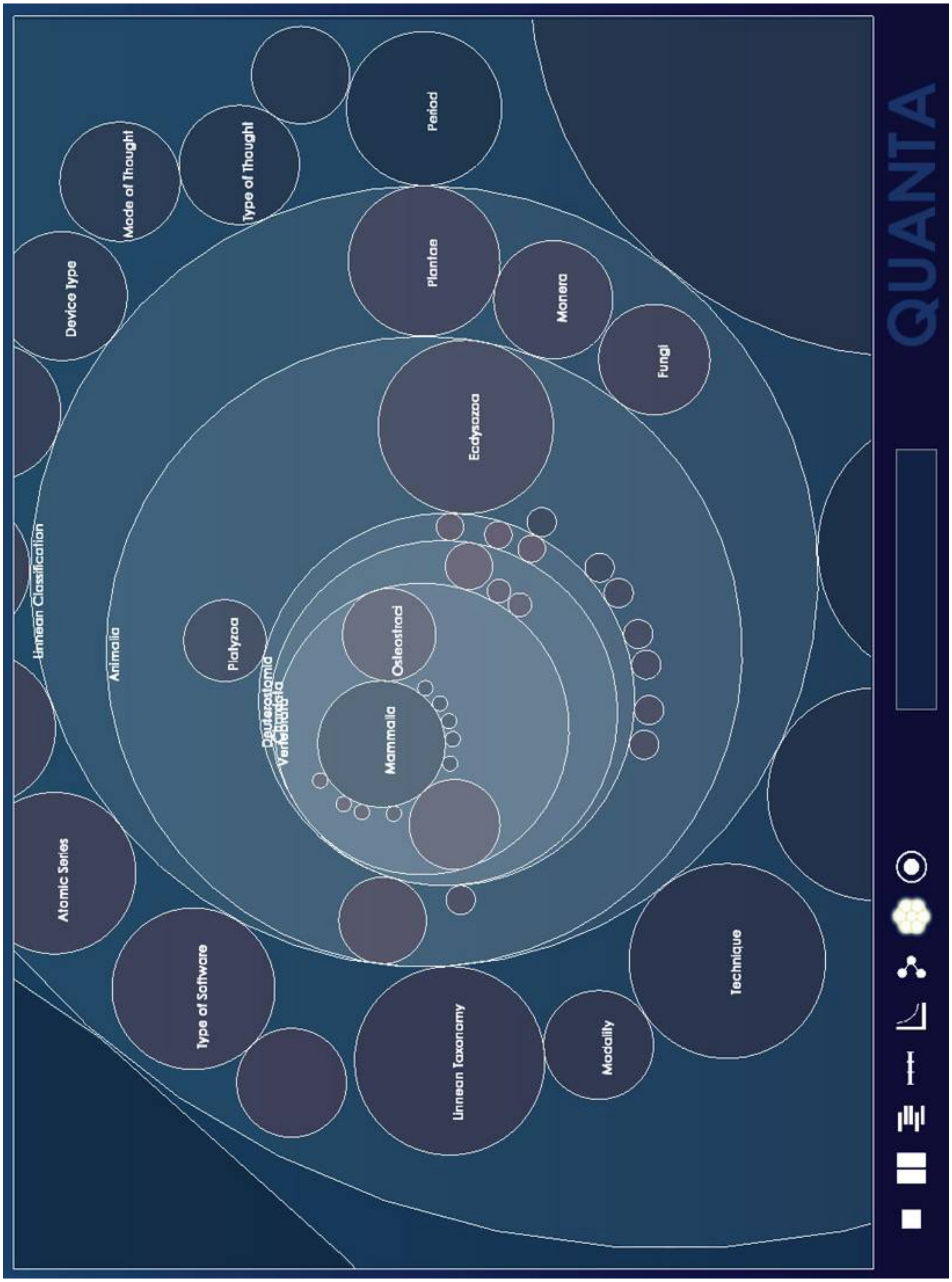


Figure 7.9. Circle packing view of the
Linnean taxonomy of organisms.

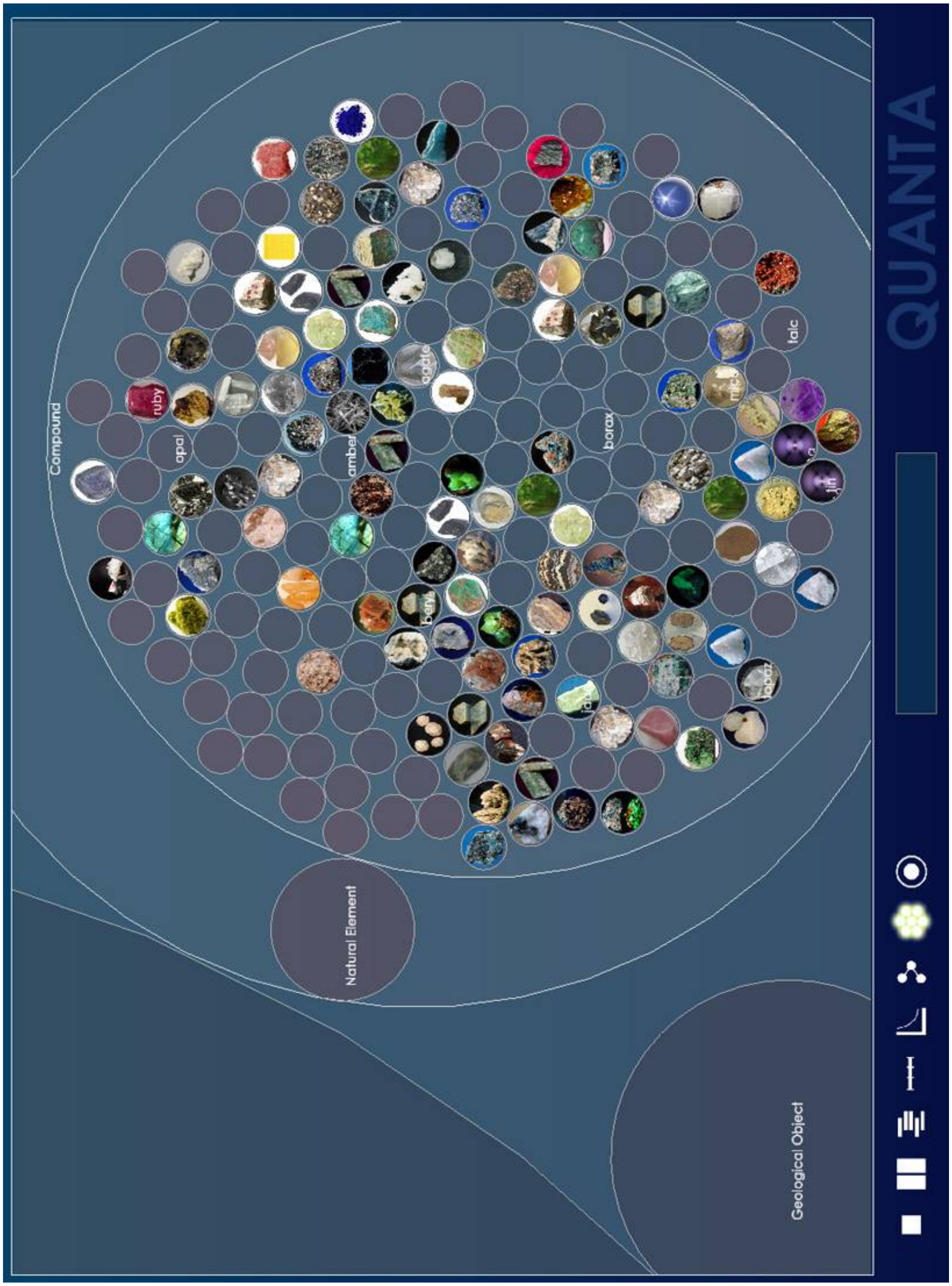


Figure 7.10. Circle packing view of mineral compounds with images.

As with graphs, the techniques for tree visualization are normally applied to data sets that already have a tree-like structure. However, as discussed in the previous chapter (Ontologies), a rich semantic network will contain a number of overlapping hierarchies. A useful visualization of trees would operate not only on fixed structures and relationships but on the many variations of trees embedded in other structures. For example, one might wish to see a circle packing visualization of the Linnean taxonomy in which area was allocated according to actual physical size rather than number of children nodes - or to build a tree according to means of transport rather than phylogenic taxonomy. This would allow an infinite number of trees to be explored from a single semantic network.

The circle packing implementation here is a view-dependent technique which constructs a tree dynamically *while it is being visualized*. Each time the user navigates to a new node the system queries the underlying hypergraph to determine the children to include according to specified criteria. The user is able to freely pan and zoom to focus on any concept. Circle packing of children nodes are performed in real-time when the parent overlaps the center of the screen and its magnified size is above some threshold (i.e. it has focus). When focus shifts, that subtree is collapsed and another opened.

Internally, the system could be easily extended to allow the user to adjust the feature that determines circle size or the tree-branching criteria. However, an interface has not yet been built to modify these attributes.

7.4. Timelines

It is difficult to firmly establish the historical development of linear timelines as distinct from other representations of time [7-15]. The reason for this may be found in one of the first timelines developed by Joseph Priestley in 1765. Rosenberg and Hendrick observe Priestley's need to explain the purpose of spatial representation:

> "[While time is not] the object of any of our senses, and no image can properly be made of it, yet because it has a relation to quantity, and we can say a greater or less space of time, it admits of a natural and easy representation in our minds by the idea of a measurable space, and particularly that of a line." [7-15]

At this point, it was not yet natural to convey temporal information spatially. Priestley found it necessary to explain the purpose of linear arrangement of time through space. However, the timeline is now a common tool for understanding temporal events.

Figure 7.11. Joseph Prestley, *Chart of Biography*, 1765.

Priestley's timeline conveys a biographical overview of 2,000 people from 1200 BC to 1750 AD (Figure 7.11). As with representing geographic space, the problem of density of information is immediately apparent. To investigate how this might be resolved we can look briefly at a modern example in geography that resolves this through navigation.

The Google Maps project allows viewers to zoom in and out of geographic satellite data. As the zoom factor is increased, the system dynamically loads more detailed information. A similar project by Microsoft called Microsoft

Virtual Earth smoothly cross-fades between difference scales to afford a more continuous experience. This idea of dynamic content is an important aspect of large scale Information Visualization.

Development of scalable, dynamic timelines is a current area for research. Huynh compare space-filling methods to linear layouts to develop TimeQuilts [7-15]. These timelines, Figure 7.12, use a "weaving" algorithm to place events with minimal clutter. While full spatial zooming is not possible, the author uses placeholder images to represent clusters to achieve semantic zooming.

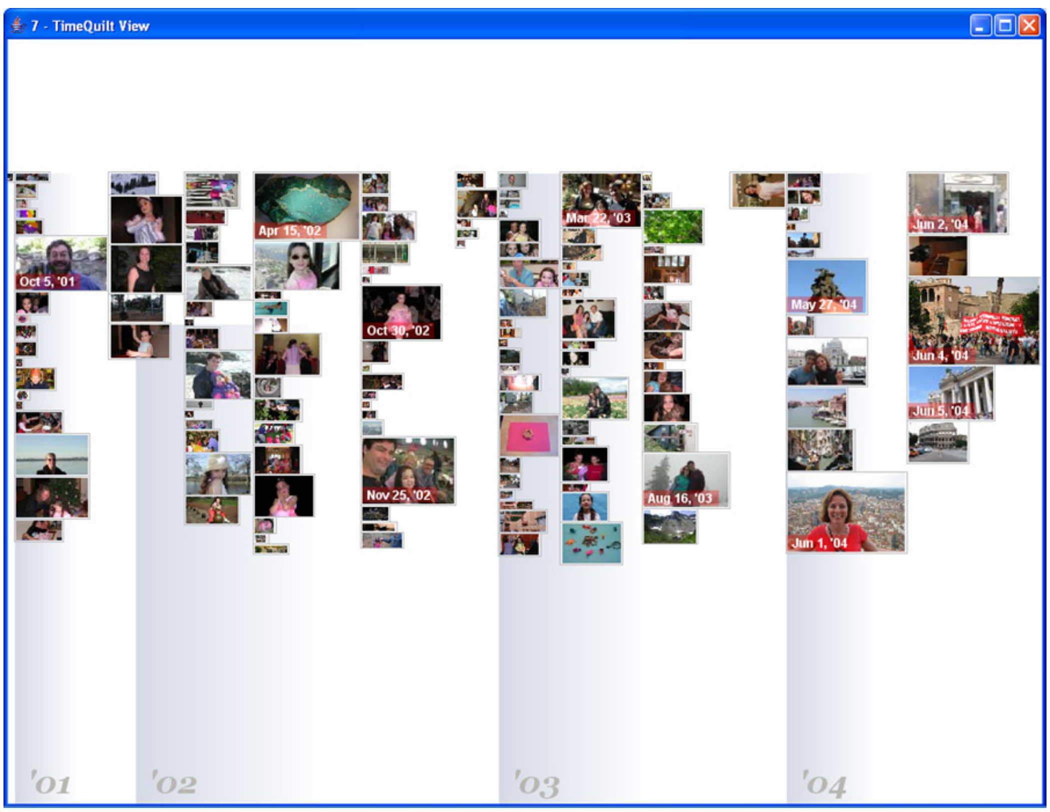


Figure 7.12. TimeQuilts, Huynh et al., 2005

An internet-based Timeline by the same author, developed at the MIT CSAIL, allows smooth horizontal panning across time (the SIMILE system). Rather than allowing zooming on the primary timeline, differences in scale are solved using an *overview* timeline that shows general features at larger timescales. Neither of these solutions provide a fully zoomable timeline.

A comparative, continuously zoomable timeline visualization was developed using the Quanta framework. The difficulty with providing continuous zooming on timelines is due to the fact that, unlike points on a geographic map where any point occupies one X, Y coordinate, events have duration that cover many points along the spatio-temporal axis. In addition, events are discrete and may be large in number. The solution presented is to provide smooth hiding and revealing of information as the user navigates to larger and shorter time scales. In addition, zooming is permitted independently on the X-axis (temporal) and Y-axis (informational) to allow the user to customize the information revealed.

To construct the timeline, a view-dependent collision resolution algorithm is developed. View-dependent techniques, common in computer graphics, perform computations only on information within the current field of view [7-17]. In this case, the collision algorithm attempts to plot all events within the current temporal range at a given zoom level. The algorithm proceeds from left to right, checking if the display of additional events might collide with existing ones. When a temporal gap is large enough, new items are added to the display. This is done on a row-by-row basis from left to right to achieve the maximum possible density without overlap. A diagram of this technique is shown in Figure 7.13.

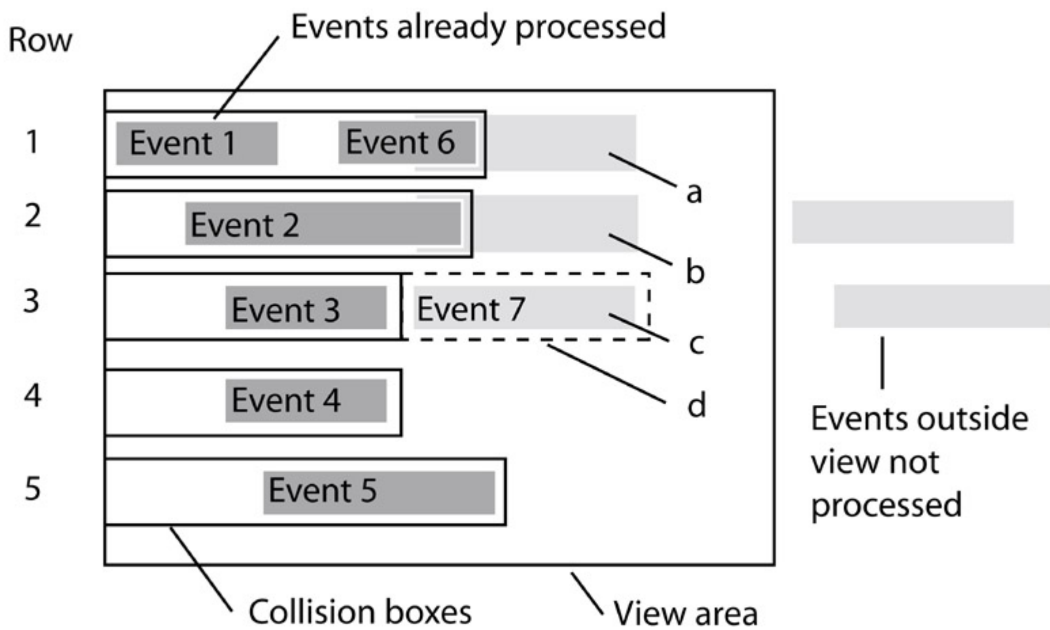


Figure 7.13. Timeline collision algorithm. Starting with currently processed events, collision boxes for each row are used to keep track of free space to the right. Event 7 is being added. In this step, the event it is a) rejected from row 1 because its starting time would collide with Event 6, b) rejected from row 2 because it would collide with Event 2, and finally c) added to row 3 without collision. Had there been no available row for it, Event 7 would have been rejected altogether and Event 8 attempted. The process continues for all events in the view.

By altering the X and Y-axis zoom factors, the user can customize the amount of information revealed. Figure 7.14 and 7.15 show a screenshot of the Quanta Timeline at two different zoom levels. At large time scales, the few items shown are spaced far in time. At shorter times, more items begin to fill in the open gaps. When zooming, there is a point at which the zoom factor is one-to-one with the data such that all events in the given time range are visible.

One of the difficulties of representing many events in a small viewing space (i.e. a computer monitor) is that not all items will be visible. This is true of geographic zoomable maps and also of the collision algorithm presented here. To compensate for this a density plot, shown in Figure 7.16, can be toggled with a button. All events are represented as dots over the same time scale, thus allowing the user to quickly reveal *how much information* is being hidden at any time.

Time is fundamental to life in many ways. Personal events, works, projects, and natural events all occur in time. Therefore, the types of questions one might ask regarding time can be very complex. To give an interdisciplinary example, one might wish to know what paintings and scientific papers between 1970 and 1980 examine the sun as an object of study. How would this compare to a similar query from 1910 to 1920?

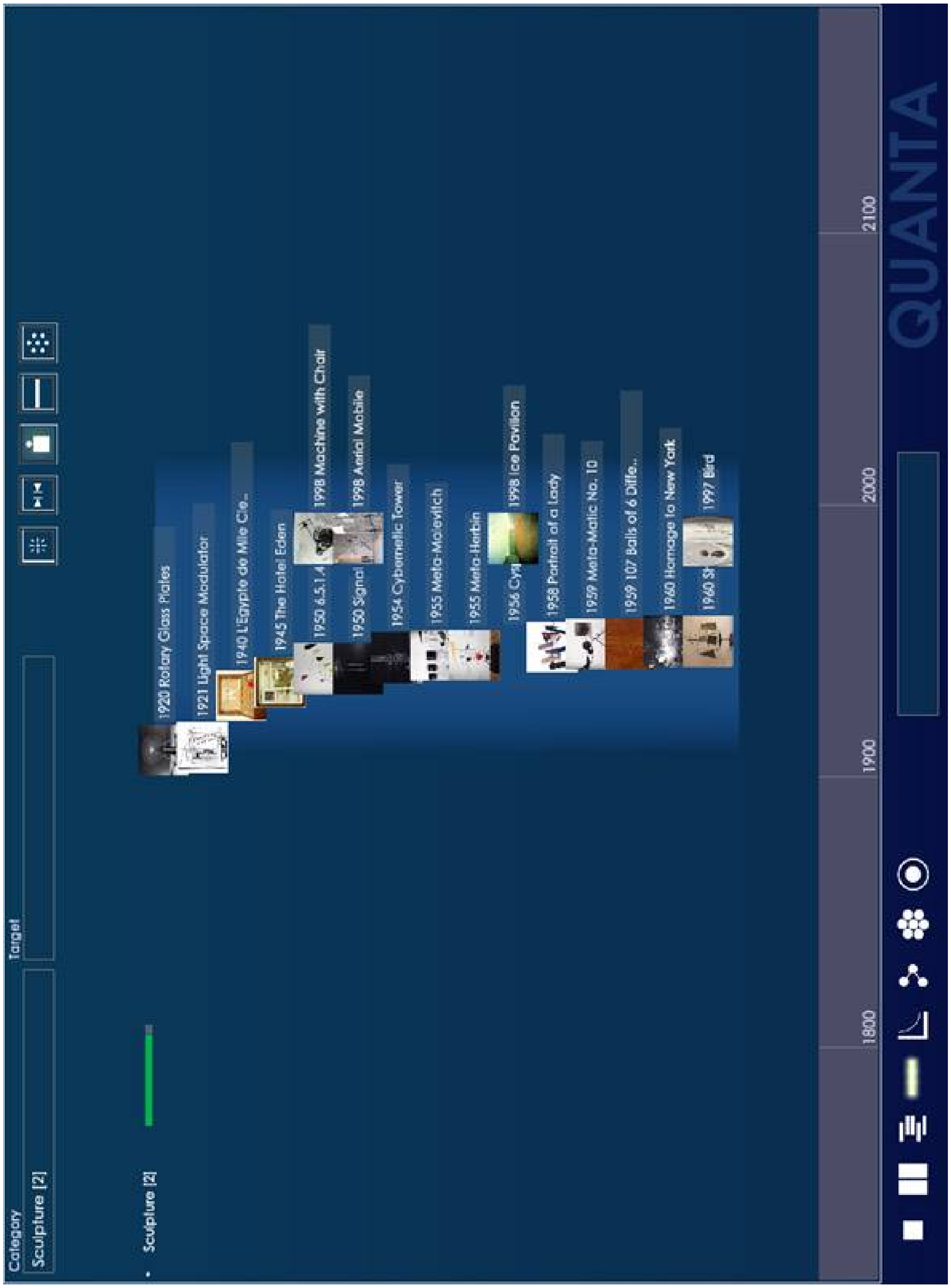


Figure 7.14. Timeline of modern sculpture.
Zoom range from 1700 to 2200 AD.

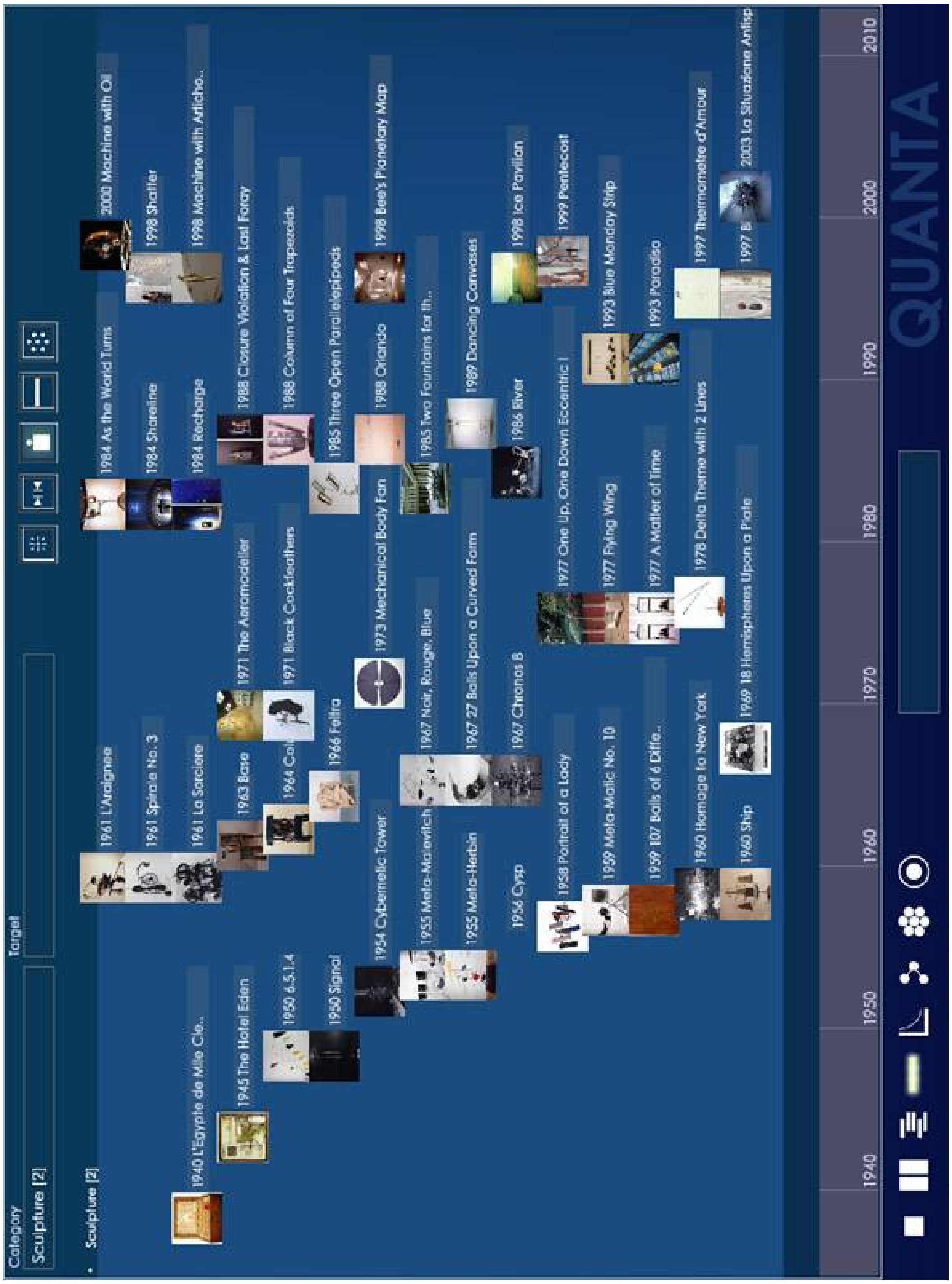


Figure 7.15. Timeline of modern sculpture.
Zoomed range is from 1930 to 2010 AD.

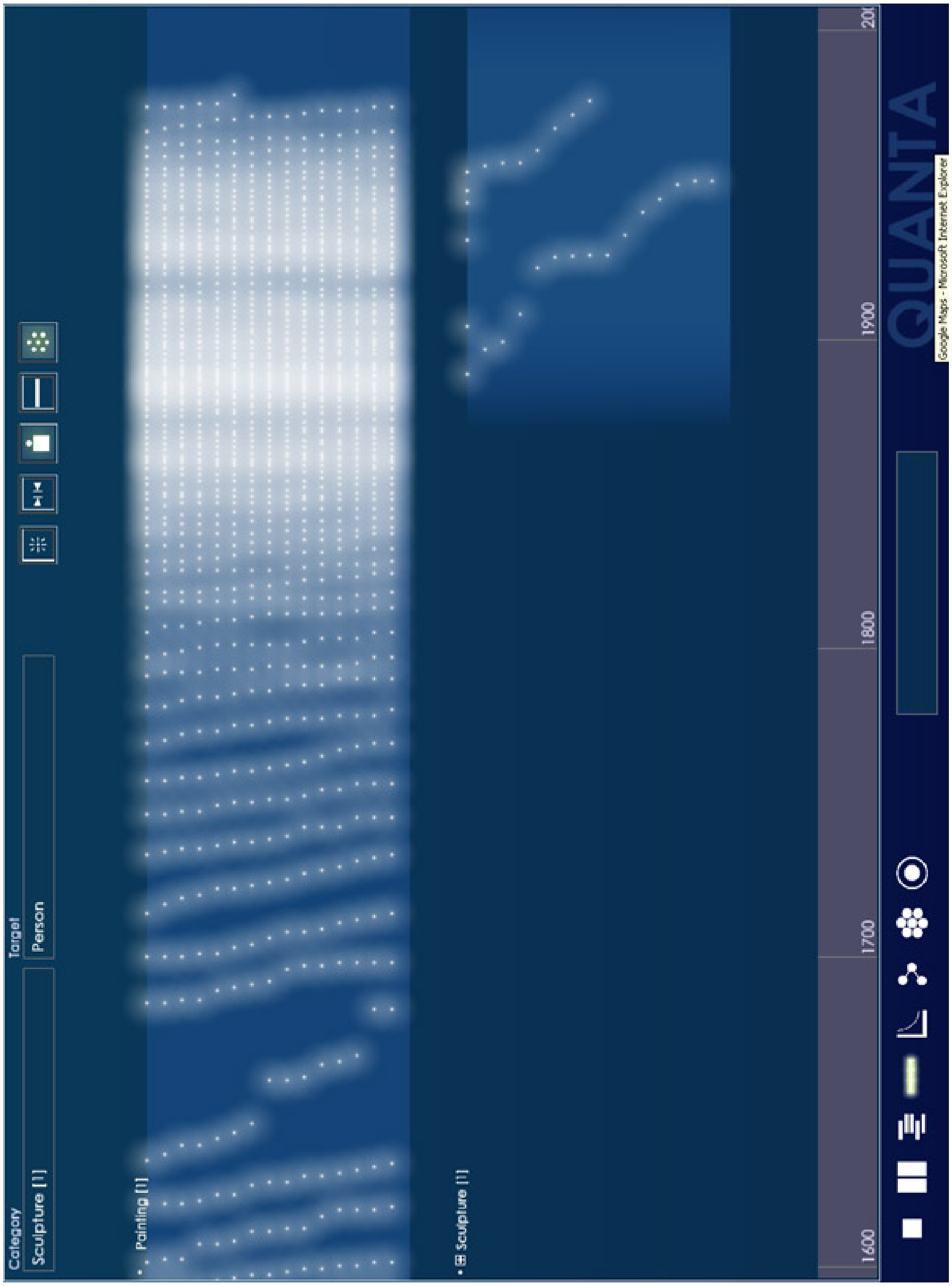


Figure 7.16. Timeline density plot of painters and sculptors from 1600 to 2000. Each point represents an individual, reflecting the amount of information available in the current database.

The Quanta timeline is also novel in that it can dynamically generate *comparative* timelines. This is motivated by interdisciplinary questions related to simultaneous events in different fields. How much correlation is there between research in one field verses another? Is there a difference when looking at nearby disciplines, such as engineering and computer science, versus distant disciplines such as physics and the visual arts.

To investigate such questions, the comparative timeline utilizes the vertical axis to allow any number of subject areas to be compared. The horizontal axis is always used for time. Two combo boxes specify the *category* and *target* for the addition of new sub-timelines. The *category*, such as painting, physics or engineering, specifies the semantic space from which events will be drawn. The *target*, such as people, works of art, or research papers, specifies the types of events that will be plotted. When a new sub-timeline is requested the system initiates a background process, simultaneous with the visualization, that begins to scan the underlying database for suitable events that match the given criteria. The *category* and *target* of various sub-timelines need not match. As shown in Figure 7.17, it is possible to compare Visual Works (target) in the field of Sculpture (category) to Research Papers (target) in the field of Physically-Based Modeling (category).

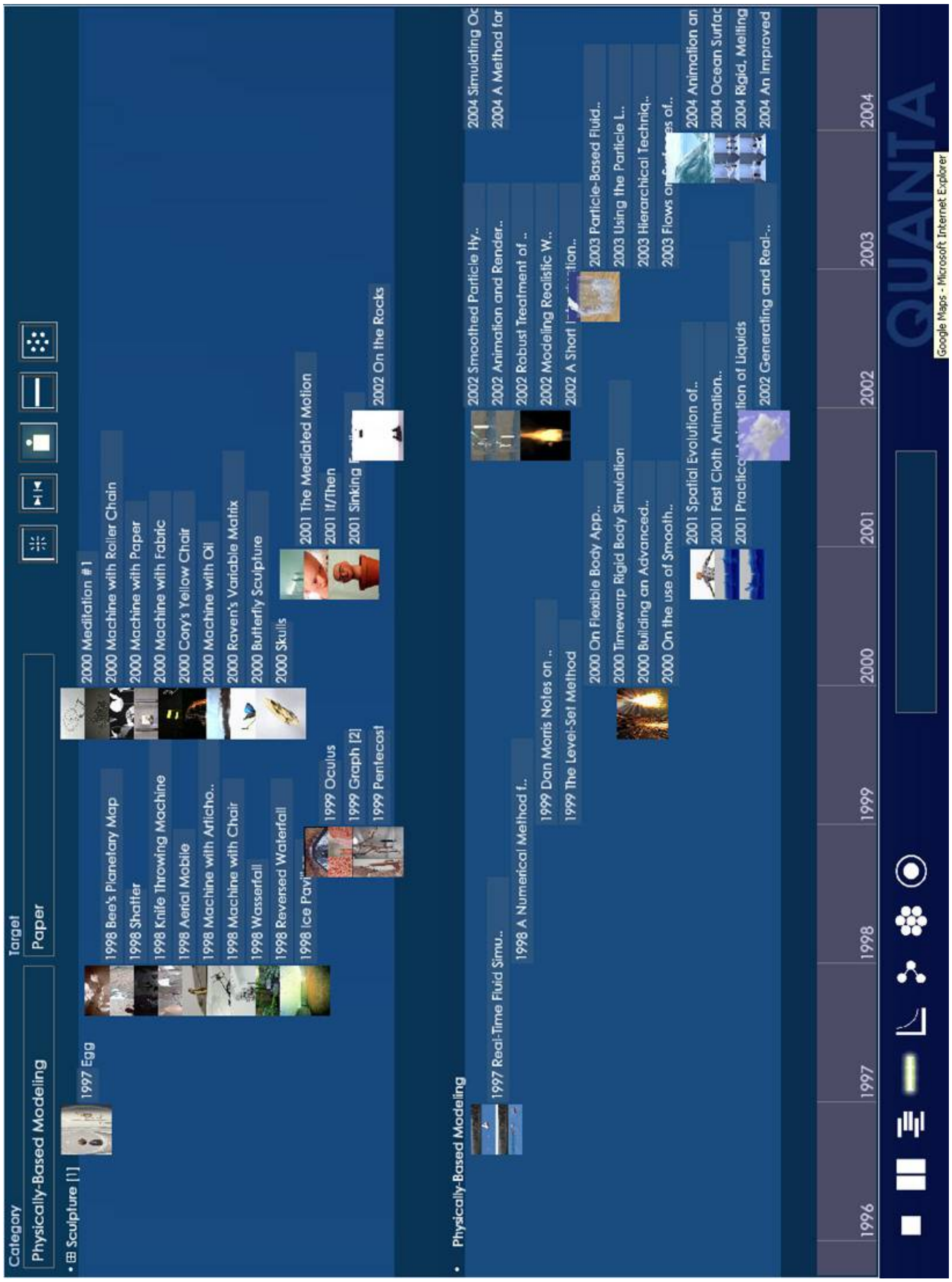


Figure 7.17. Comparative timeline of sculptures and papers on the topic of physically-based modeling (subfield of computer graphics) from 1996 to 2004.

7.5. Networks

The term network can have many meanings. It may refer to a simple graph of nodes and edges, it may refer to a physical infrastructure such as the internet, or it may refer to a semantic network of concepts. The development of graph-based visualization was founded on the need to understand systems of connections.

Visualization of network structures is made difficult by the fact that many cross-connections cannot be easily represented in two or three-dimensional space [7-5]. However, several research directions have made it possible to explore complex networks. The first, spring-embedder models, allow a graph to iteratively achieve a less cluttered layout of nodes. In this case, the criteria for placement include uniform length of edges and symmetry [7-18]. A physically-based spring system allows nodes to dynamically achieve near optimal positions. Kamada and Kawai extend this minimize the number of edge crossings and to achieve uniform node distribution [7-19].

Node placement techniques are not always successful as the number of interconnections in a network may be such that no flat representation without overlapping lines is possible. Thus, for most applications, any graph-based visualization will contain overlaps. Tools can be provided to selectively filter edges based on various criteria [7-20].

Another approach is to include semantic information in the node layout criteria. A successful application of this is the Netmap which groups generic concepts in a networks into radial segments [7-21]. The network is drawn inside the circle thus formed, with connections made to the segments. In this way, it is possible to see patterns in connectivity related to the concepts shown. Other variations place nodes in columns rather than radially [7-22]. Finally, it is possible to represent networks in the third dimensions as well. An early example, Narcissus, uses a 3D spring-embedder model to place spherical nodes in space [7-23].

Since the late 1990s, and due to the relative ease in programming them, the number of graph-based visualizations developed has increased dramatically. This is apparent from a website titled Information Aesthetics which collects new examples of these types of visualizations daily.[2] While many were developed as information visualization tools, others were developed for aesthetic reasons. While aesthetic pursuits are valuable to push creative limits, with the quantity of repetitive work one wonders what social processes might enable better collaboration and reuse to develop integrated tools.

[2] Source: http://infosthetics.com/

An important distinction should be made between aesthetic and functional approaches to network visualizations in particular. Due to their complexity, or perhaps something inherent about networks themselves, visualizations of graphs have the unique property that their visual representation often conveys the *feeling of complexity* without functional utility. This may be a goal in itself.

However, if the goal is to find or retrieve meaningful knowledge it is often necessary to supplement a purely graph-based visualization with other semantic attributes - as was done with the Netmap. This is especially important when dealing with semantic networks as simplifying grammatic structures with bidirectional links results in a loss of meaning present in the connections. If the goal is functional, one must remain aware of what is lost when reducing meaningful networks to graphs. In this case, it is often better to provide multiple visualizations of different types to navigate the data. The Quanta network, for example, contains many overlapping hierarchies (existential taxonomy, classification taxonomies) which are not visible in a purely graph-based layout.

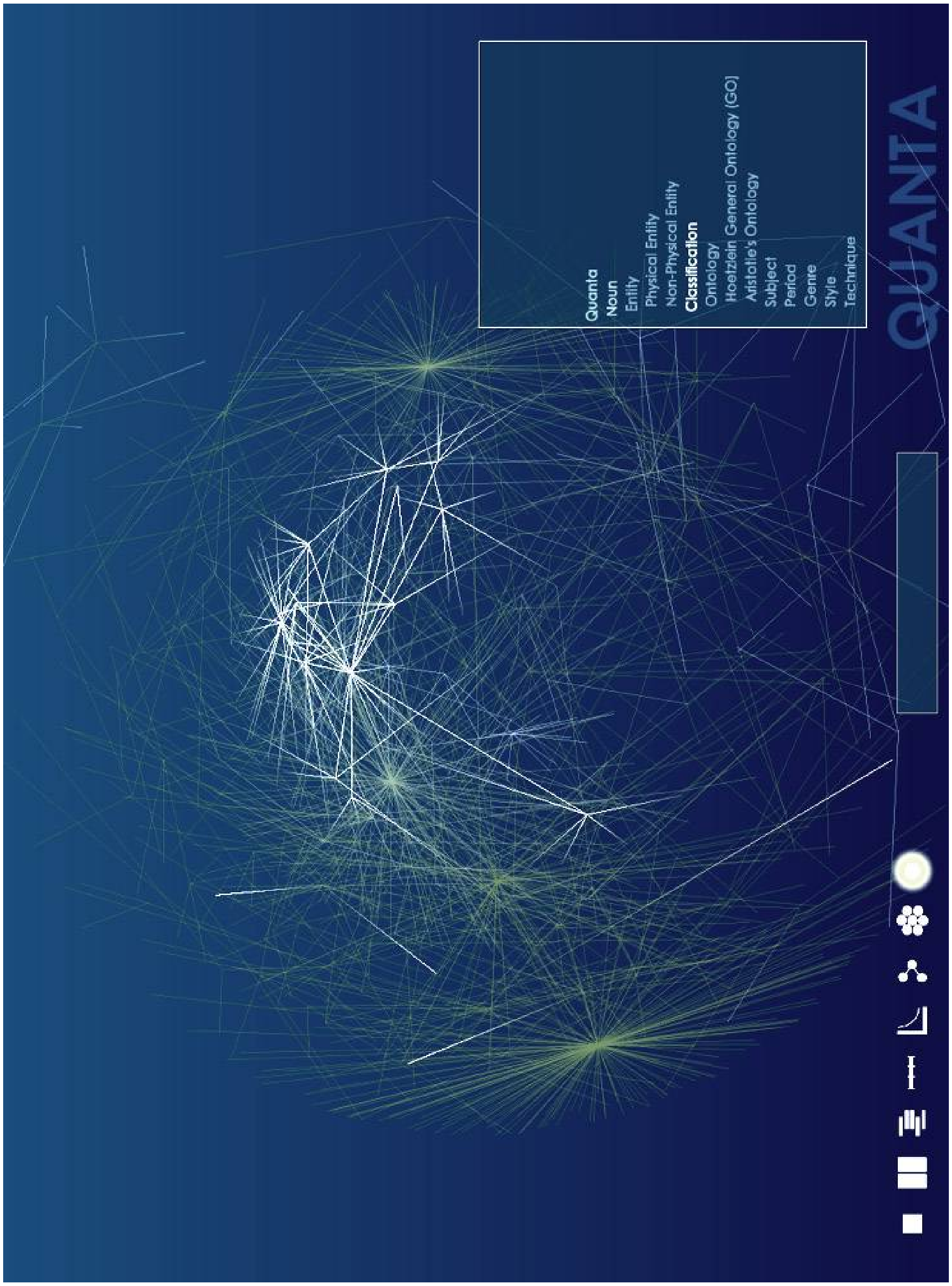
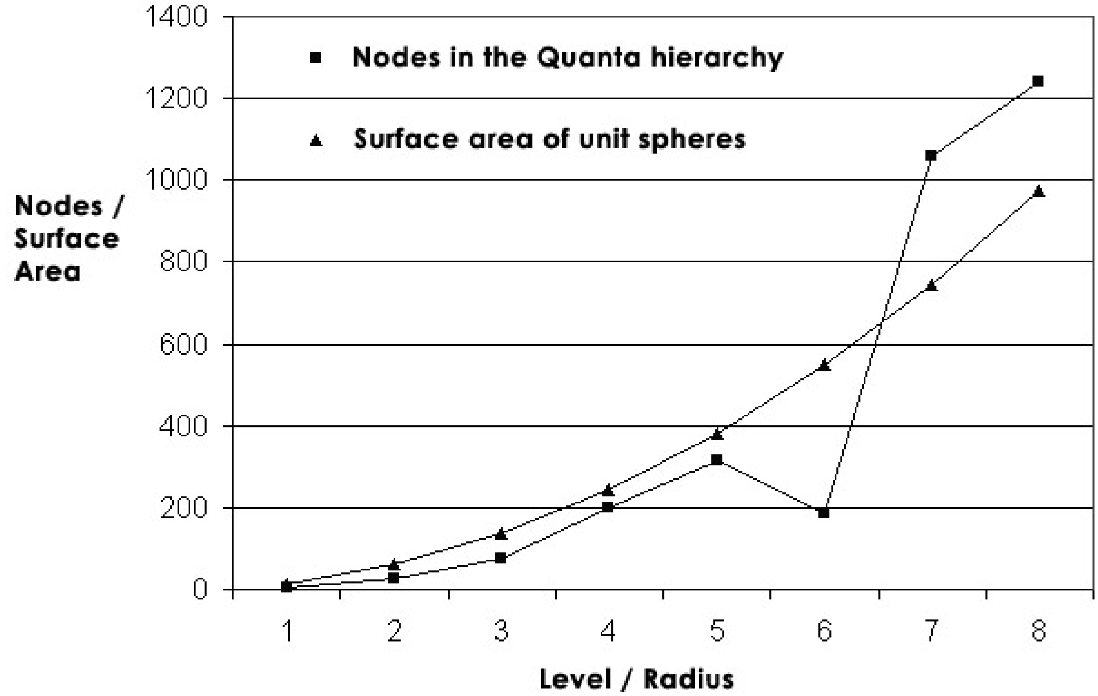


Figure 7.18. Network visualization on nested spheres

The final visualization in this thesis conveys a majority of the Quanta semantic database using nested spheres similar to phyllotactic trees in which nodes are iteratively arranged based on biological distribution patterns [7-24]. Due to the complexity of the network, and following the previous discussion, rather than taking a function approach this visualization was developed more for aesthetic reasons. As such no labels are shown. Displayed in Figure 7.18, the visualization uses the first eight thousand nodes of the Quanta ontology. The levels of the existential taxonomy of Quanta are placed on each sphere. Thus, the inner most sphere holds basic linguistic concepts: Noun, Verb, Adverb. The next sphere holds Physical and Non-Physical entities. The outer most sphere holds individual works of art, sculpture, or living organisms - specific instances of more abstract concepts.

The goal of this visualization was to communicate the richness of semantic data and to express the similarity between cognition and dynamic data systems using a neurological metaphor. To explore this, an additional list view is introduced whereby the viewer may navigate through various concepts. As a concept is selected, all the semantic links which it touches are highlighted in the network. After a gradual decay, this "lighting" of the network fades away. Interestingly, for more abstract terms a larger portion of the entire network is lit up as the number of connected concepts is greater. The analogy to human cognition is clear, but is this because human thought is similar to dynamic structures found in a machine? Or is it because we, as humans, have a tendency to mimic our current understand of ourselves in the mechanical systems we construct? Perhaps just the visual effect of blinking nodes in a network is reminiscent of neurological functioning. Yet we must wonder what specific relationships this has to reality.



Tablee 7.1. Correlation between growth in data nodes for the current Quanta taxonomy and unit sphere surface areas (scaled by 1.1)

Level (L)	Quanta Nodes (N)	Sphere Area (SA) R = 1.1 * L	Ratio (SA / N)	Cumulative Nodes	Cumulative Node Density
1	3	15	5.07	3	1.858
2	26	61	2.34	29	1.538
3	74	137	1.85	103	1.461
4	200	243	1.22	303	1.177
5	316	547	1.20	619	1.125
6	186	745	2.94	805	1.495
7	1058	973	0.70	1863	1.026

While developing this visualization, a unique correlation was found between number of nodes in the existential hierarchy and the surface area of each sphere, as shown in Table 7.1. This suggests that for certain types of data a spherical arrangement could allow 3D navigation of complex data with *equal node density* throughout the space. By combining a proportional surface placement technique such as circle packing with nested spheres, a constant average distance between nodes could be maintained despite the fact that the data sets grow exponentially at each level. Such an arrangement would allow clutter-free navigation of networks in a three-dimensional space. This technique was not attempted here, but could be an interesting direction for future research .[3]

7.6. Knowledge Visualization and Cognition

The visualization system developed here is built on basic principles of information visualization and human-computer interaction. There are the obvious affordances such as smooth navigation and zooming available in each visualization. In addition, buttons and combo boxes allow selection and toggling of important features to further increase usability. Color, images, and other feedback is designed to make the system easier to use.

[3] There are two criteria for such a technique. First, the nodes should be placed on nested spheres with a radius modified by a scaling constant: $R = kL$ (where L = level). Second, on each sphere the nodes should be equally distributed. To achieve this, while also maintaining alignment with parent nodes, a suitable placement algorithm similar to treemaps or circle packing would need to be used but which was modified to function on the surface of a sphere. The current network visualization only meets the first criteria. Presently nodes at a given level are not distributed across the entire sphere surface.

This visualization framework also contains several novel aspects. Each visualization is dynamically connected to the hypergraph network of Quanta, thus giving access to rich semantic data. Queries for data, such as the display of sculptures (as a target) in the Timeline visualization, allow the system to limit the scope of the data. This process, visualizing selective data through database queries, is not novel [7-25]. However, each visualization may filter and restructure the data through the semantics of the hypergraph. For example in addition to target, which specifies the base dataset, the Timeline visualization loads events that fall with in a certain *category*. That is, events must match the semantics:

X has Subject (category)

X is a (target)

Similarly, the Graph visualization uses semantic features present in the hypergraph to automatically determine which properties may be graphed. The visualizations are built on top of the hypergraph database. Therefore, each new visualization will have access to these semantic relationships. In this way, visualizations can allow for navigation beyond that provided by simple database queries.

An important theory of visual cognition is that visual short-term memory is used to hold temporary models of concepts which are too dense or complex to be understood in their entirety. In an interesting experiment, Philips and Baddeley show two large matrices of numbers to viewers for only a brief time [7-26]. The viewers are then asked to determine if any of the elements were different in the sets shown. The experimenters found that the viewers were able to determine missing elements at a rate higher than would be caused by chance. Since the participants did not have time to examine each element, they conclude that a form of *visual short-term memory* (VSTM) is used to create a temporary, abstract model of the data.

The Quanta framework for visualizing semantic data is based on a similar model. A visualization in Quanta consists of temporary, short-term structures that correspond to the fundamental structural units in information visualization: graphs, trees and networks. Thus an individual visualization is analogous to visual short-term memory. For each query, a visualization dynamically rebuilds its own internal structures from the larger, more permanent information in the hypergraph database. The database in this case is analogous to objects in the real world. These ideas are shown in Figure 7.19

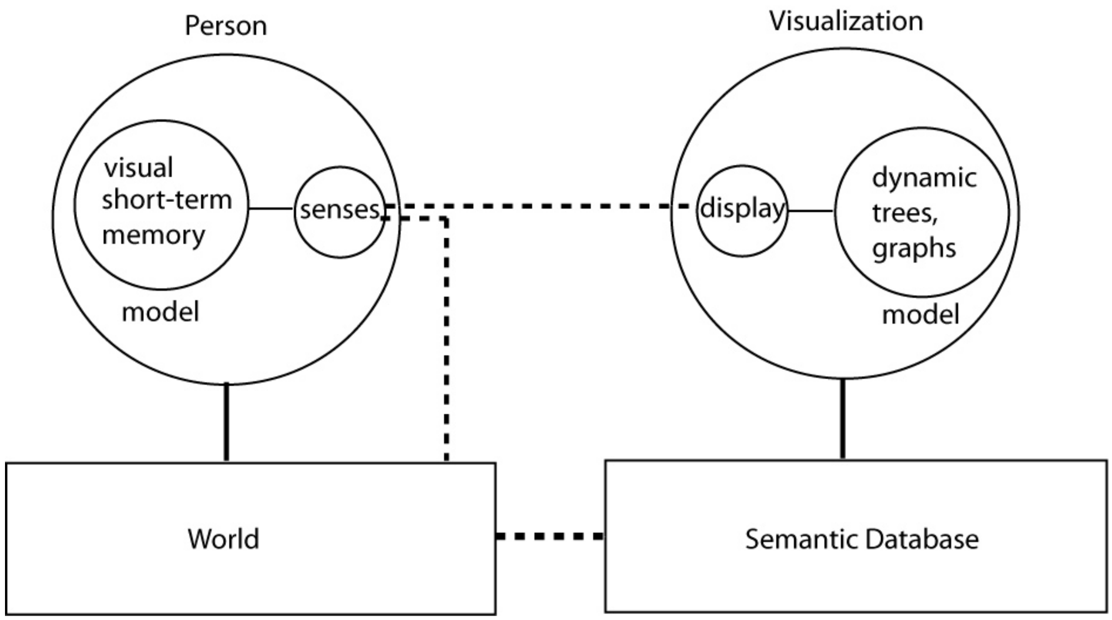


Figure 7.19. Analogies between human cognition and visualization systems. Dynamic structures and visual short-term memory are similar in that they provide temporary models of deeper semantics.

While datasets often connote simple structures, human knowledge consists of trees within trees, overlapping networks, quantitative and qualitative data, and temporal and spatial information all present simultaneously. The goal of the visual aspect of this thesis was to develop a system in which established design units such as treemaps, timelines, graphs and hyperbolic and network layouts become interchangeable building blocks that operate on and freely navigate deeper semantic structures. By developing a system that more closely resembles human cognition, it should be possible to build a knowledge resource that more closely matches the flexible of human thought.

One way in which we might distinguish *knowledge visualization* from *information visualization* is through abstraction away from the data structures that currently define the field. While the primary object in information visualization is the structure, whether it be a list, a tree, or a graph, the primary object in knowledge visualization is the relationship (semantic link). Out of these links any number of trees and graphs might arise. These simpler structures provide momentary, multifacted understanding of the deeper meaning found in complex systems.

Another way to define the distinction between knowledge visualization and information visualization is in terms of the *data-semantic ratio*. Visually a tree of living organisms shown via circle packing can convey only a single attribute in each circle: its name for example. There is little visual space for much more. Thus, the data as shown in the visualization has a low DSR (N elements with 1 attribute each). However, each organism is richly connected to many other concepts according to its morphology, geographic place, relationship in the food chain, and behavior. The actual data has a very high DSR (N elements with M attributes each). In developing general knowledge systems, a single visualization should be one among many temporary constructs that allow us to perceive complex data.

Table 7.2. Scientific, Information and Knowledge Visualization according to the features of structure, data-semantic ratio and goal.

	Structure	**DSR (4)**	**Modality**	**Goal**
Sonofication	structures	low	aural	functional
Scientific Visualization	particulate (1)	low	visual	functional
Information Visualization	structural (2)	low	visual	functional
Knowledge Visualization	relational (3)	high	visual	functional (5)
Information Aesthetics	structural	low	visual	aesthetic (6)
Knowledge Aesthetics	relational	high	visual	aesthetic

1) Particulate: Datasets that contain discrete elements that map directly to spatial locations. Example: Cloud simulation, molecular structure.

2) Structural: Refers to abstract yet specific data structures such as lines, trees, graphs and networks. Examples: Cone-trees, Treemaps, Hyperbolic network layout

3) Relational: Refers to content which is highly relational. May all be contained in a semantic network but new techniques are needed to navigate patterns. Examples: encyclopedia articles, linguistic statements, thought.

4) The Data-semantic Ratio (DSR) characterizes the degree to which relationships are present in the data. See Chapter 2.

5) Functional: Communicative goal is to provide a clearer understanding of the object of study. The inform the view of that particular object, i.e. to education or enable research.

6) Aesthetic: Communicative goal may go beyond explaining the data to other artistic purposes (social, reflective, critical)

Table 7.2 summarizes this discussion on the various fields of information visualization. Sonofication is included as a reminder that visual interfaces are but one modality. While the discussion here has been primarily about functional tools, Information and Knowledge Aesthetics are distinguished from visualization in that their communicative goals may differ.

The purpose of this chapter has been to investigate the field of information visualization and to present novel visualization techniques for semantically connected data. Several novel visualizations have been demonstrated including view-dependent comparative timelines, view-dependent circle packing, automatically generated graphs and an active network of semantic data on nested spheres. A general analysis of human cognition suggests that multiple, temporary visual structures are an important aspect of knowledge visualization. Finally, providing these visual systems with the data of rich semantic databases allows for more flexible and precise navigation of complex ideas, and further enables interdisciplinary research.