

VENNFS: A Venn-Diagram File Manager

Rosario De Chiara Ugo Erra Vittorio Scarano
Dipartimento di Informatica ed Applicazioni “R.M. Capocelli”
Università di Salerno
84081, Baronissi (Salerno), Italy
{dechiara,ugoerr,vitsca}@dia.unisa.it

Abstract

In this paper we present a prototypal file manager, VENNFS, that is designed to overcome some of the limitations of the current desktop interfaces, that are strongly based on hierarchical file systems. VENNFS allows users to place documents and categories on a plane so that files may belong to multiple categories at once, where proximity on the plane can represent similarity and time filtering is allowed.

1 Introduction

In this paper we present a novel interface that is designed to help the user overcome some of the limitations inherited by the classical hierarchical file system. In fact, common desktop metaphors are based on top of hierarchical file systems that have become, far from being a help, a heavy heritage for any system that manages user’s interactions with a computer.

Hierarchical file systems are, indeed, a natural, efficient and economic way to organize data; nevertheless, their structure restricts possible innovations aimed at making information retrieval as efficient for users as it is for machines.

Historically, when user interfaces moved away from command-line interfaces and began to offer windows and desktop metaphors, users were quite happy because of their proved ability to reduce the time to learn the interface: not a list of, often unpronounceable, words but few, coherent and simple tasks with a pointing device (i.e. a mouse) on a “desktop”. The choice of a desktop as metaphor is particularly effective since it refers to the common everyday work in an office. i.e. something that it is easy to relate to. The huge revolution of personal computing is directly related also to easiness of user interfaces and, therefore, we are not so ingenuous as to renegade the crucial importance of such interface.

Nevertheless, we would like to propose the view that the

desktop metaphor is doomed to be a way to disguise a hierarchical file system that offers to users the capability to classify a document in one, and only one, “folder” (i.e. one position within the hierarchy). Both multiple categories and relationships among categories are ruled out by this kind of interface. In a way, while surpassing in many ways the visionary view of computing provided by Vannevar Bush in 1945 [3], the contemporary revolution of personal computing falls short (in terms of user interfaces) in reaching the smoothness of the interface toward data envisioned by Bush’s Memex.

Our proposal. The prototype we present here, VENNFS, can be considered a first step toward presenting an interface that reflects also the multidisciplinary work and scope that is offered to users by currently available technology. In fact, current storage devices allow personal computers to hold tens of thousands of user files and, at the same time, the “digital era” (i.e. ubiquitous hardware devices like PDAs, cell phones, digital cameras, etc.) makes extremely easy to generate quite a few files, even for moderately technology-oriented users.

VENNFS allows the user to place documents and categories on a plane allowing files to belong to multiple categories at once, by using well-known and intuitive Venn diagrams to represent graphically each category. Placement on the plane allows users to use proximity (between categories and files) to represent closeness and similarity. Finally, VENNFS also offers filtering files by time (i.e. by the date when file was last modified). VENNFS is smoothly interfaced with the underlying (hierarchical) filesystem: it allows exporting to the file system as well as load/save.

Organization of the paper. In the next section we describe the problems generated by hierarchical file systems and some related visualization systems. Then, in Section 3 we present the technique and the prototype and we conclude with Section 4 with some final remarks.

2 Hierarchical file systems

Limitations of the current organization of file systems are well-known [12, 15]. Hierarchical file systems were designed when technology offered storage devices that hold few megabytes of data. In the meantime, disk density has kept a quick growth: until 1988, it was increasing yearly by 29%, to reach a yearly increase of 60% until 1996 [9] and approaching a 100% per year later on [8]. As a matter of fact, this growth looks even faster than the well-known Moore's law [14] for semiconductors.

Some solutions were proposed by software manufacturers. Among the possible solutions, we name the "Object File System", where files are not necessarily hierarchically organized and rely on a layer of data storing borrowed by database technology. Object file systems were considered by Microsoft [16]: an Object File System was part of the "Cairo" operating system (later become Windows 95). It was planned to be based on an SQL layer that would facilitate heterogeneous access to files. The idea was abandoned, probably because of concerns over having different products lines for different file systems. Now, some sneak previews of upcoming releases from Microsoft suggest that such a feature, named WinFS (Windows Future Storage), may be included in future versions.

File systems Visualization. Since number and size of files stored on hard disks has increased so much, research in visualization [19] has shown several attempts to ameliorate the problems raised by the dimension of these (hierarchical) file systems.

Visualization of a hierarchical file system was a motivating factor for Treemaps [18] that in the last decade was used in different contexts. The technique visualizes recursively a hierarchy by splitting a fixed amount of screen space into rectangles and alternating horizontal and vertical directions. Information Slices [2] offer a series of cascading semicircular disks. Each disk represent several levels of the hierarchy and the user can have two disks shown at the same time.

Several 3D visual metaphors were also proposed. With Information Pyramids [1], subdirectories are small plateaus visualized above a larger one (the common parent directory) with side that is proportional to the number of files or to the files size. The overall impression is that the pyramids grow as the hierarchy is descended. As an application for 3WPS, a 3D Web-based Process Visualization framework, a 3D filesystem browser was built [13]. The 3D scene is built on-the-fly in VRML and can be navigated remotely with a VRML browser.

In our opinion the main limitation of these tools is that they provide an alternative, sometimes nifty, way of visualizing a hierarchical file system without providing extended capabilities to user.

One of the few attempts to move away the desktop metaphor by providing extended capabilities is LifeStreams technology [6], used by commercial products like ScopeWare [17]. The main idea is to leverage on the sense of time of computer users: the most important (and searched for) documents are the most recent ones. In a way, the desktop is arranged more like a diary of activities and related documents, by offering also textual search within the documents. While aimed at the same problems we are tackling, ScopeWare does not fully address the issue of multiple categories and relationships between documents.

3 VENNFS

Our goal is to propose a technique and a prototype to graphically organize files on a plane in such a way that (1) each files can belong to multiple categories, (2) proximity on the plane of files/categories can be used by the user to relate information and (3) filtering by time is possible. In our opinion, by offering to users these 3 characteristics, VENNFS can extend the traditional desktop metaphor based on hierarchical file systems.

3.1 Main features

Multiple Categories. Our goal is to solve the problem of documents that belong to several aspects of our work and that, therefore, should be placed into several folders at once. The technique we used is based on Venn diagrams [21]. Let C be a collection of n simple closed curves drawn in the plane: then C is an *independent family* if the intersection of any combination of the interior or the exterior of each curve is non empty. If each intersection is connected and there are finitely many intersections, then C is a Venn diagram.

Visualization of Venn Diagrams is simple and intuitive up to three sets: the construction outlined by Venn in his paper for higher number of sets is quite not intuitive, though more symmetric drawings were found more recently [5]. In general, anyway, as the number of sets increases, so does the visual complexity of the figure.

Therefore, we choose to restrict the visualization technique only to ellipses that represent categories (i.e. sets)¹. With ellipses, it is possible to represent all the intersections of up to 5 sets but construction for 4 and 5 sets are rather complicated [7]. We argument that this restriction does not represent a noticeable limitation to the average user but makes the tool quite simpler and immediate to use.

A final remark is needed: symbolic links offered by file systems do not represent a suitable solution to our problem since they represent a solution for a single document and the approach is neither systematic nor visual (i.e. intuitive).

¹In this version of the prototype, ellipses cannot be rotated. Rotation will be introduced in the next version.

Moreover, links are difficult to trace back from the original document and therefore it is complex to change the folder which the document belongs to: all the links pointing at the document should be changed at once.

Relating by proximity. By placing the documents on an easily navigable plane (see next subsection), we facilitate the user to set data relationships as spatial relations by representing closeness of argument (between categories as well as between documents) by means of proximity in the plane provided by VENNFS. In a way, we encourage the user to leverage on his own spatial abilities, by placing related categories close to each other. An interesting consequence of using distance to relate documents is that filtering by arguments is implicitly obtained by zooming on the region and zooming in/out to get at the desired level.

We may say that, by using VENNFS, the user can draw the environment represented by the (unstructured) corpus of his/her own documents, place documents into (possibly multiple) categories and relate categories by proximity: the task of information retrieval for the user is made easier by building a cognitive map [20] of this environment. Since the environment is created by the user himself, the internalized analogy in the human mind of the physical layout (created by using VENNFS) becomes easier to grasp. In a sense, one may see VENNFS as a way to make a cognitive map of its documents explicit and browseable.

Since it is known that the identification of landmarks helps the navigation [4] as well as learning and memorizing [10], VENNFS also allows to place landmarks (diamond symbols \diamond that do not correspond to files) on the plane. Navigation aid is provided by the list of landmarks (shown on the left panel) that allows the user to reach quickly each landmark by double-clicking on landmark's label.

An interesting characteristic is that the 2-dimensional plane offered by VENNFS looks (to the user) "infinite" and without clear boundaries (i.e. it stretches as far as the user wants) but, at the same time, by using the *overview* function (described later), it is easy to move to a viewpoint that allow to encompass all of it within one sight.

Notice that, unlike other techniques (see for example [18]), no space-filling strategy is used since we rely on the distance as an additional instrument for the user to relate between categories/documents.

Filtering on file time. Since accessing data is usually performed by task, usually documents accessed recently are probably going to be accessed again soon. The importance of the "time" coordinate is well recognized in commercial products (see Scopeware [17]).

Therefore, VENNFS, first, provides an indication of how recent is the file, by using an easily interpretable metaphor: recent (i.e. recently accessed) files are "hot" (i.e. red) while

old files are "cold" (i.e. blue) with intermediate colors to represent intermediate age.

Then, it is important to allow filtering over the time by showing only the files whose last modification date is below a given date. The filtering is obtained by moving a slider between the date of the oldest file and the date of the newest one.

3.2 A presentation of the prototype

Several important functionalities are briefly described here (for more details, see VENNFS website <http://isis.dia.unisa.it/projects/vennfs>).

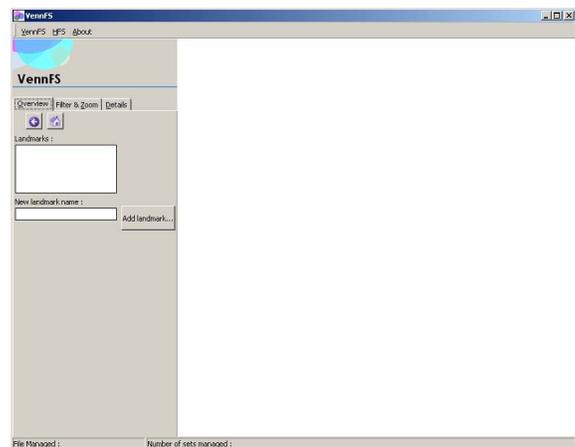


Figure 1. VENNFS's start-up.

As VENNFS starts, after the splash screen, the initial screen shows an empty "desktop" on the right, a left pane (with three tabbed panes) and few simple menus (see Fig. 1). The tabbed panes on the left are "Overview", that allows navigation and placing/jumping to landmarks, "Filter&Zoom", that offers filtering by time, and "Details" that offer information (as well as editing) about categories and documents. These three tasks obey to Shneiderman task by data type taxonomy [19], as well as fully reflecting his Visual Information Seeking Mantra "*Overview first, zoom and filter, then details-on-demand*". The menus offer setting preferences ("VennFS") and exporting to a hierarchical file system ("HFS").

Creating categories and adding files. Creating a category is done by tracing on the plane with the right button. The result is an ellipse with default color and name: the color and the names of the category that is selected (showing a small movement by a smooth animation) can be changed in the "Detail" tabbed pane (see Fig. 2) and pushing the "Apply" button.

Files (as well as directories) are added by simply drag-and-dropping them on the corresponding set. Each file is

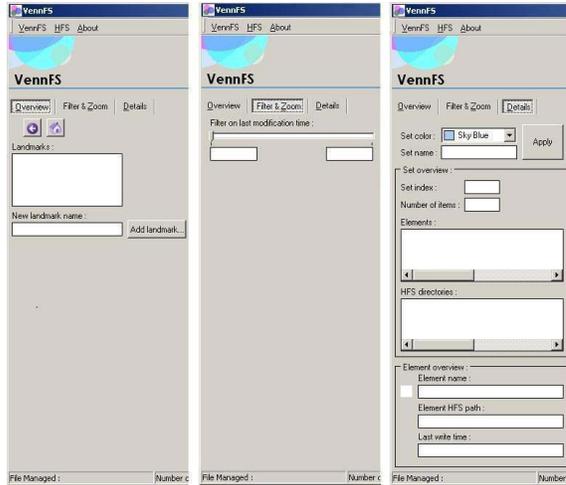


Figure 2. The three tabbed panes: Overview (left), Filter (center) and Details (right).

represented by a small square, that is colored depending on its relative age, among the files currently inserted. Both for category creation and file adding, VENNFS asks for confirmation with Accept/Cancel buttons.

Navigation: zooming, overview, landmarks. The categories are placed on a plane and the camera (i.e. user's point of view) can move along the plane but also along its orthogonal to allow an overview of a larger part of the plane.

Navigation is possible with several tools. First of all, by "grab-and-move" with mouse middle button, one can move in any direction. Then, a category can be selected by double-click and the point of view is moved so that the category is in the middle of the screen and the camera is at the same distance from the plane as when the category was created, in order to preserve the context that was shown when the user created the category. Zooming is allowed by using the mouse wheel. A particularly useful tool is the zooming-by-context: the user can select a rectangular area on the plane and the camera is moved so that the area is shown at full size.

Then, an overview button is provided with a home icon (shown in Fig. 2 (left)) that allows to move the point of view so that all the categories can be viewed together on the screen. To allow a (primitive) focus+context aid, we added a back button (that, in this version, works only after an overview) to go back at the camera position before the "Overview" was requested.

Landmarks can be placed anywhere in the plane: they do not correspond to files and have a label. By typing the label in the "New Landmark name" textbox (in the "Overview" tabbed pane, in Fig. 2 (left)) and pushing "Add landmark",

the user can, then, click on the plane to place the landmark and its label is shown in the "Landmarks list". It is possible to quickly navigate to the landmark by double-clicking on the landmark name in the list.

Filtering. By the "Filter&Zoom" tabbed pane (Fig. 2 (center)), files can be filtered on their last modification time, by using the slider whose extremes show the oldest modification time and the most recent one among the files inserted in all the categories. Filtering makes the little square that represents each file fade away.

Exporting to the file system. VENNFS allows a certain degree of integration with the underlying file system. In fact, it allows exporting the current categories and files on an ordinary (i.e. hierarchical) file system: categories are replaced by directories, files are links to the actual file and files belonging to multiple categories are present in each category (as links). It is also possible to save/load the current diagram. The export operation occur in the directory that can be set in the "Settings" panel by the VennFS menu. In the same panel, the user can set the length of the file-names written on the plane.

3.3 Implementation details

VENNFS was developed and tested in Microsoft Windows 2000/XP on an ordinary PC with an OpenGL 1.3 accelerated graphic board (such as ATI, NVidia, etc.).

In Fig. 3 we show an example of VENNFS. Several categories, meaningful for a PhD student in Computer Graphics, are shown with few files. Notice that ellipses representing the categories are smaller than they would be in reality, since we wanted to present all of them in one screenshot. In Fig. 4 we show how the details for categories and files are shown.

Graphic boards and OpenGL. Graphics boards currently offer a substantial amount of machine cycles to perform tasks that used to be performed by the processor itself. Moreover, some effects (like alpha-blending) that were relegated to special-purpose software (entertainment, CAD) are now available (for free) on our everyday personal computer via a graphic board.

In VENNFS we exploited such new features (alpha-blending, scaling, camera effects, smooth transitions etc. . .) so that the interface offered is more dynamic and supportive toward the users. As an example we can cite VENNFS's "smooth zooming" effect [19] used to change context (switching among categories) without disorienting the user. Also "category selection" is visualized by using a smooth animation (offered by the graphics board).

Exploiting graphics hardware also allows to solve (by hardware) classical geometrical problems [11]. In VENNFS

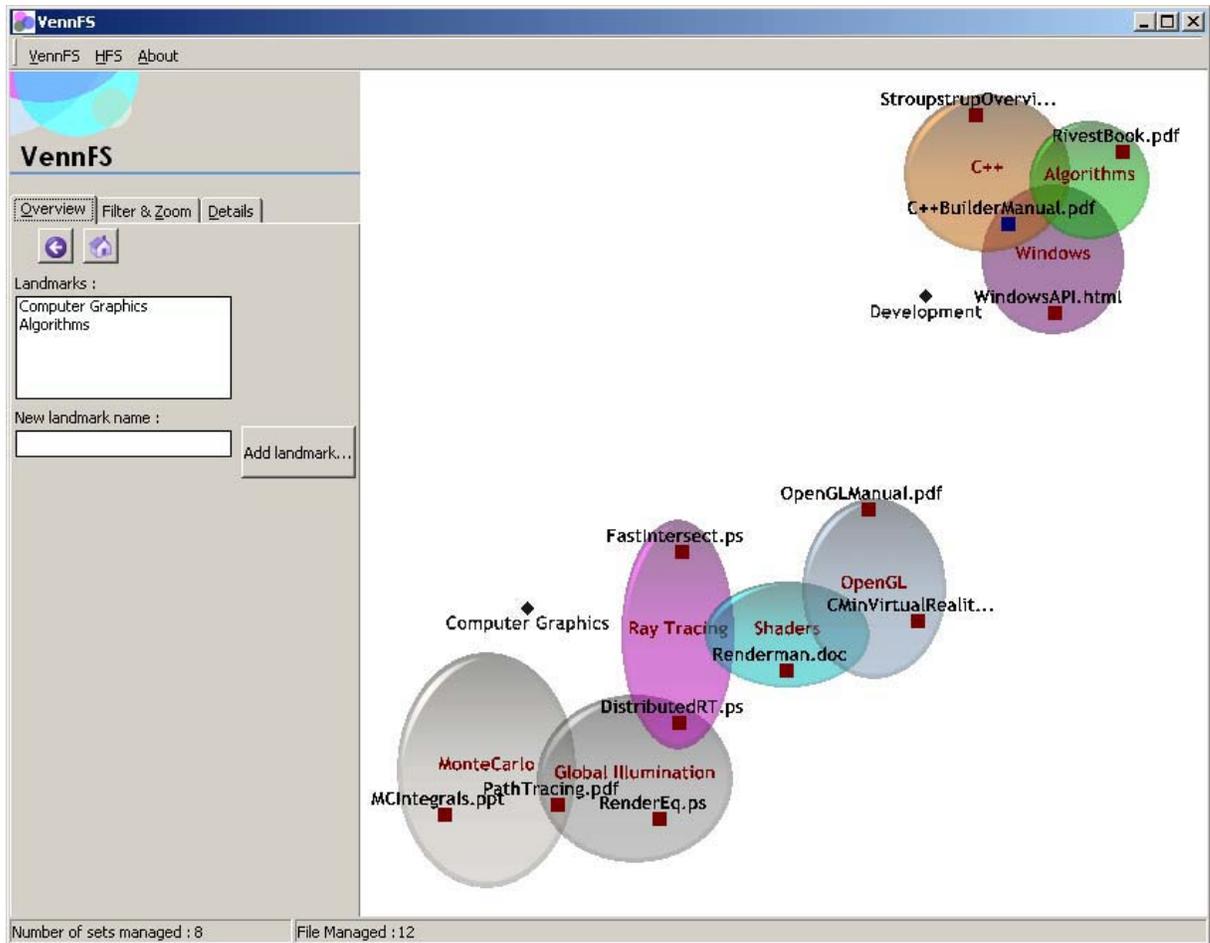


Figure 3. Several categories (with a certain overlap) are built around the landmark “Computer Graphics”. A paper on Path Tracing (PathTracing.pdf) belongs to two categories: MonteCarlo e Global Illumination. Also notice how files that belong to OpenGL category are “closer” to the “Development” landmark, that is also close to files relevant to programming.

we used graphics hardware to perform the “category selection” without using additional data structures: picking calls (like `glPickMatrix()`) offered by OpenGL [22] allow our program to get all the sets under the point clicked by the user.

4 Conclusions

We may say that, as users left the command line to embrace a WIMP (*Windows, Icons, Menus, Pointing devices*) interface, they were given the capability of accessing a hierarchical file system from multiple windows at once. Now, our VENNFS extends and facilitates management of files that belong to multiple categories, helps user orientation among numerous documents and allows filtering on de-

mand.

Being superimposed (as an interface) to a hierarchical file system our tool allows the user to place in each category only those documents that are really meaningful i.e. that contain information and not only data. In fact, often several auxiliary files (like log files, backup copies, etc.) are automatically created with the “master” document and placed into the same folder: their explicit presence can only distract the user from his/her task since their role is determined/needed by the software and not by the user.

Future work. The prototype VENNFS is currently being modified in order to offer smoother interactions with users. Version 1.0, described here, as well as any new version, some documentation and examples are available on

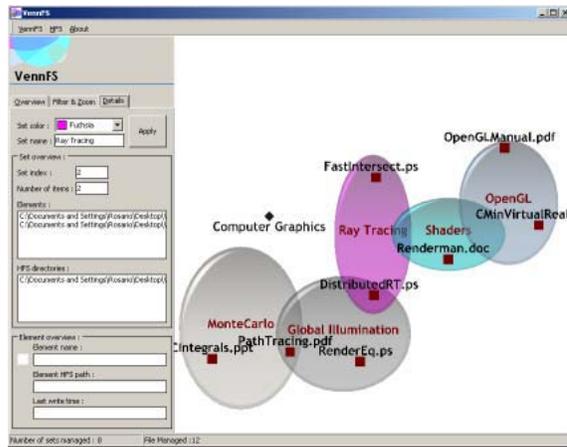


Figure 4. The details shown for a category and for a file selected on the list.

<http://isis.dia.unisa.it/projects/vennfs>.

Among the characteristics currently under development, we are working to allow cut&paste facilities, files/categories moving from one place to another (as well as changing the size of categories), a fully fledged history and more intuitive details-on-demand on a file by interacting on its graphical representation.

While polishing up the prototype, we will be considering further extensions of VENNFS in three dimensions and by using arbitrary shapes (not only ellipses) for the categories. Another possible extension could be to include evocative images in the background, that can help orientating the user in placing categories and files. Several appealing perspectives can be foreseen along these directions, as well as potential problems in the complexity and usability of the interface. It can be also interesting to provide VENNFS functionalities in a cooperative setting, allowing to create a shared repository of files that can be accessed from a remote location, maybe through a standard web browser.

References

- [1] K. Andrews. Visual exploration of large hierarchies with information pyramids. In *Proc. of Sixth IEEE International Conf. on Information Visualisation (IV'02)*, London (England), July 10-12, 2002.
- [2] K. Andrews and H. Heidegger. Information slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs. In *Proc. of IEEE Symp. on Information Visualization (InfoVis '98)*, North Carolina (USA), Oct. 1998.
- [3] V. Bush. As We May Think. *Atlantic Monthly*, 1945.
- [4] A. Dillon, J. Richardson, and C. McKnight. Navigation in hypertext: a critical review of the concept. In *Diaper, D.,*

- Gilmore, D., Cockton, G., and Shackel, B. (Eds) INTERACT '90*. North Holland, Amsterdam, 1990.
- [5] A. W. F. Edwards. Venn diagrams for many sets. *New Scientist*, 7 January 1989.
- [6] E. Freeman and S. Fertig. Lifestreams: Organizing Your Electronic Life. In *Proceedings of 1995 AAAI Fall Symposium*. AAAI, 1995.
- [7] B. Grünbaum. The construction of venn diagrams. *The College Mathematics Journal*, 15:238, 1984.
- [8] J. R. Harrow. Moore's law expanding?, Aug. 2nd 1999. <http://h18000.www1.hp.com/rcfoc/19990802.html>.
- [9] J. Hennessy and D. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufman, San Francisco, 1996.
- [10] C. Johns and E. Blake. Cognitive maps in virtual environments: Facilitation of learning through the use of innate spatial abilities. In *Proc. of 1st International Conf. on Computer Graphics, Virtual Reality and Visualisation in Africa, Cape Town, South Africa. 2001*, 2001.
- [11] I. Kenneth E. Hoff, J. Keyser, M. Lin, D. Manocha, and T. Culver. Fast computation of generalized Voronoi diagrams using graphics hardware. In *Proc. of the 26th annual Conf. on Computer graphics and interactive techniques*, pages 277–286. ACM Press/Addison-Wesley Publishing Co., 1999.
- [12] T. H. Kim. *Frigate: An Object-Oriented File System*. PhD thesis, University of California at Los Angeles, 1998. <http://fmg-www.cs.ucla.edu/ficus-members/tek/papers/diss.ps.Z>.
- [13] D. Malandrino, G. Meo, G. Palmieri, and V. Scarano. 3WPS: A 3D Web-based Process Visualization Framework. In *Proc of 1st IEEE International Symp. on 3D Data Processing Visualization and Transmission (3DPVT02), Padova (Italy)*, June 19-21, 2002.
- [14] G. E. Moore. Cramping more components onto integrated circuits. *Electronics*, 38(8), 1965.
- [15] J. Nielsen. The Death of File System, February 1996. <http://www.useit.com/papers/filedeath.html>.
- [16] M. Ricciuti. New windows could solve age-old format puzzle—at a price, March 13, 2002. CNET News.com <http://news.com.com/2009-1017-857509.html>.
- [17] Scopeware, by Mirror World Technologies Inc. http://www.scopeware.com/home/scopeware_home.html.
- [18] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1), January 1992.
- [19] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of 1996 IEEE Symp. on Visual Languages, pages 336-343, Boulder, CO, USA, Sept 1996*.
- [20] E. C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55:189.
- [21] J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 9, 1880.
- [22] M. Woo, J. Neider, T. Davis, and OpenGL Architecture Review Board. *OpenGL programming guide: the official guide to learning OpenGL, version 1.1*. Addison-Wesley, Reading, MA, USA, second edition, 1997.