Displaying Data in Multidimensional Relevance Space with 2D Visualization Maps

Jackie Assa, Daniel Cohen-Or and Tova Milo

Computer Science Department
School of Mathematical Sciences
Tel-Aviv University, Ramat-Aviv 69978, Israel
Email: {jackie,daniel,milo}@math.tau.ac.il

Abstract

This paper introduces a tool for visualizing a multidimensional relevance space. Abstractly, the information to be displayed consists of a large number of objects, a set of features that are likely to be of interest to the user, and some function that measures the relevance level of every object to the various features. The goal is to provide the user with a concise and comprehensible visualization of that information.

For the type of applications we concentrate on, the *exact* relevance measures of the objects are not significant. This enables accuracy to be traded for a clearer display. The idea is to "flatten" the multidimensionality of the feature space into a 2D "relevance map", capturing the inter-relations among the features, without causing too many ambiguous interpretations of the results. To better reflect the nature of the data and to resolve the ambiguity we refine the given set of features and introduce the notion of *composed features*. The layout of the map is then obtained by grading it according to a set of rules and using a simulated annealing algorithm which optimizes the layout with respect to these rules.

The technique we propose here has been implemented and tested, in the context of visualizing the result of a Web search, in the RMAP (Relevance Map) prototype system.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
O1	1	9	1	0	1	0	0	2	7	8
O2	3	5	2	0	4	0	3	9	0	0
O3	2	0	7	0	6	2	5	5	0	3
O4	0	9	0	0	0	0	0	3	0	8
O5	0	5	0	0	0	2	0	4	0	0
O6	6	0	3	0	5	0	3	8	0	5
O7	3	0	4	0	6	0	3	0	0	0
O8	3	0	4	0	5	0	0	6	0	6
O9	3	0	0	0	0	5	6	0	0	6
O10	2	0	2	0	2	0	5	6	7	0
O11	4	0	6	0	6	0	3	0	3	4
O12	5	0	7	0	7	0	0	0	0	0
O13	5	0	0	0	0	0	4	5	0	0
O14	9	0	2	0	2	0	0	0	4	5
O15	8	0	9	0	9	3	0	2	0	4
O16	9	0	9	0	9	8	0	3	4	4
O17	7	0	8	0	9	0	0	0	4	0
O18	2	0	0	0	0	3	4	0	0	0

Figure 1: The objects with their graded features.

1 Introduction

Many information retrieval tasks involve dealing with a large set of objects and require that the relevance of the objects to various features that are likely to be of interest to the user be analyzed [20, 19, 14, 32, 27, 15, 10, 7, 31, 27, 18]). For example, in the data mining context [20, 19], we may be given a database of objects (typically tuples), a set of rules that potentially describes properties of the data, and want to analyze how relevant the rules are to various classes of data objects. As another example, consider a Web search [8, 17, 16, 22, 25]. When posing a query to a Web index server, the answer is often lengthy, much of it not necessarily relevant, and the actual interesting documents are sometimes buried way down the document list. To assist the user in focusing on the relevant documents, it has been suggested that the returning list be analyzed and the documents classified according to their relevance to various features that may interest the user [14, 32, 27, 15, 10, 7, 31, 27].

The question is how to present such object-feature relevance information to the user. What we want is a display that helps to highlight the more interesting classes of objects, filter out uninteresting objects, focus on the important information, understand the significance of various pieces of data, and grasp the relationship among them. Abstractly, the information to be displayed consists of a large number of objects, a set of features, and some function that measures the relevance level of every object to the various features. Each feature can be viewed as a dimension of the information space. The classification of the objects, with respect to the different features, positions the objects in the corresponding "coordinates" of this multi-dimensional information space. The features coordinate indicates its relevance to a given object. A naive representation of such data is shown in Figure 1. The example shows a 10-dimensional space where each row displays the grades (relevance level) of a given object. However, this overloaded table representation is rather confusing and does not reflect the inter-relations between objects and features.

Before we continue, it is important to note that for the type of information retrieval applications described above, the *exact* grades (relevance measures) of the objects are not significant. We are not dealing here with exact scientific measurements, but rather want to enable the user to have, at a glance, a global picture of the data distribution, thereby easing the selection of significant objects. We are therefore willing to trade accuracy for a clear display.

A better visualization paradigm is to provide the user with a *relevance map* that summarizes the given information in a more intuitive way. A simple version of such a map visualizes the features as gravitation nodes, and the objects displayed as points centered around the features that are most relevant to the object. Figure 2 shows the relevance map constructed for the information presented in Figure 1. The labeled circles represent features, and the dots

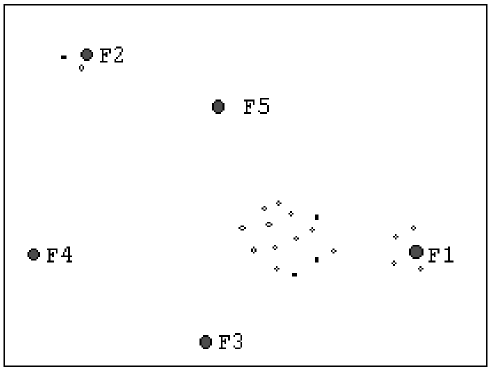


Figure 2: The features as gravitation nodes, and the objects are displayed as points centered around the features that are most relevant to the object.

represent objects. With this visualization, the user can immediately see that there are some objects relevant only to the feature F_1, few to feature F_2, whereas all the other objects are relevant to several features. However, mapping a hyperspace down onto a 2D map is impossible without introducing ambiguities and conflicts. For example, in the above map it is not clear if the objects in the center are relevant to *all* the features or just to some of them. Some of those objects may be relevant to only two diagonal features, say F_4 and F_1.

The solution proposed in this paper is based on a layout method that adapts the feature set to obtain a more comprehensible map. The idea is to identify the potentially ambiguous relations among objects and features, and construct new *composed features* to resolve them. The relevance map is then built with respect to this extended set of features, and thus provides a clearer display of the information. For example, consider the map in Figure 3. It enables one to distinguish between objects relevant only to the F_4 and F_5 features (they are located near the *composed node* labeled with the pair F_4, F_5), and documents that are relevant only to $F_1 - F_3$ or to $F_1 - F_4$.

Unlike previous works, where the layout technique assumed that the features being displayed were almost independent, we have to deal here with interdependencies among the features (i.e. the dependencies between composed and basic features). The dependencies have to be reflected in the data representation. The layout of the extended set of features becomes an optimization problem, which we solve using simulated annealing [9].

The visualization technique we propose here has been tested in the context of a Web navigation aid tool. The RMAP (Relevance Map) prototype system was built to experiment and demonstrate the effectiveness of relevance maps as a tool by visualizing the results of Web searches.

The rest of this paper is organized as follows. In Section 2 we illustrate the notion of a relevance map as a mapping mechanism from hyperspaces into a 2D map. In Section 3 we discuss the details of the simulated annealing algorithms. In Section 4 we describe a specific application - a Web search system - where the above technique was used. Section 5 describes related work in this field, and finally, 6 is the conclusion.

2 The relevance map

As mentioned in the introduction the layout of the objects in a 2D map should reflect their inter-relation with respect to the selected features. Since the number of features is usually much greater than two, it is a multidimensional layout problem [27, 7, 14, 9]. Projecting a multidimensional vector space onto a 2D map, yields conflicts and ambiguities. In this section we present a method that lays out nodes (features) and points (objects) with their relative relevance (grades) on a 2D map, and alleviates the ambiguity problem by introducing new feature nodes.

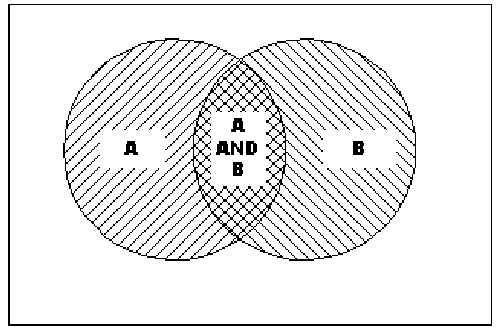


Figure 5: The Venn-diagram regions of two features

Each feature defines a dimension or an axis along which the object-point can be placed such that it reflects its relevance to that feature. Placing two orthogonal dimensions generates a planar 2D map, which represents the information space of the two features. Similarly, the method can be generalized to an n-dimensional feature space. However, the visualization, orientation and understanding of n-dimensional spaces is known to be difficult [2, 18]. The proposed relevance map represents n-dimensional object-points on a 2D map such that their location reflects their relevance to the features. The relevance map is a 2D space where a distance function is defined but not a metric. That is, the proximity of two points means that they are likely to have similar data, but the distance between them cannot be compared to a third point.

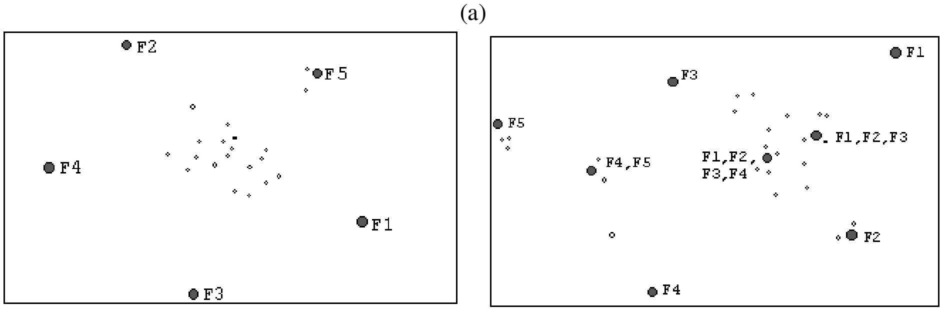
A feature dimension and an associated relevance threshold partitions a 2D map into two regions, one that is "more relevant" to the feature and the other "less relevant". The relative distance of a point from the feature node expresses the relevance of the point to the feature (see Figure 4(a)). Placing two feature nodes in the 2D plane, partitions the map into four regions where the location of a point reflects its relevance to the two features. A high relevance to one feature places the point close to the feature node. A point with high relevance to both features is placed between them (see Figure 4(b)). Weak relevancy to either features places the point far from both. The features partition the plane into topological regions like Venn diagrams (see Figure 5). This scheme shows the mutual relevancy of a point to the two features, and it is a stronger expression than just the distance (grade) to a feature. Unlike Venn diagrams where points are either in or out of a region, here the position of a point in the map expresses its relevancy to multiple features.

Venn diagrams deal well with up to three dimensions. There are Venn diagrams for more dimensions [29], but, they do not have a structure for the topological regions and do not give an intuitive view. This problem is referred to as inconsistency, and is likely to occur when points have high grades for several features and need to be placed simultaneously in several regions. The inconsistency might be resolved by either placing multiple copies of a point in all the regions while maintaining a visual link between them, or by placing it once in the most relevant region.

Since the location of the point is a rough estimate of the relevance degree to the features, we can measure the "degree of inconsistency". That is, a point which can be placed in more than one region has a degree of irrelevance to the features. By neglecting minor inconsistencies we overcome many possible conflicts and generate a map representing a large number of dimensions. Note that the relevance map is not used for scientific measurements, but

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
O1	2	4	3	5	0	0	0	0	0	0
O2	1	4	3	2	0	0	0	9	1	2
O3	0	0	0	8	8	0	0	0	0	2
O4	4	3	3	0	0	0	3	9	0	0
O5	0	0	0	0	7	0	0	0	0	2
O6	4	4	5	5	0	0	0	0	1	3
O7	0	0	1	0	0	0	0	0	0	4
O8	0	0	0	0	8	0	0	0	0	5
O9	0	0	0	1	9	1	0	0	0	0
O10	0	0	0	5	5	0	4	0	2	1
O11	6	7	5	6	0	1	0	0	2	0
O12	2	2	5	7	3	0	0	0	0	0
O13	3	3	6	0	0	0	0	0	1	0
O14	0	0	1	1	0	0	4	0	1	5
O15	5	5	7	2	0	1	4	8	2	1
O16	6	9	5	6	0	0	0	7	1	2
O17	0	3	0	0	0	0	0	0	0	0
O18	0	4	0	0	0	0	0	0	1	2

(a)



(b) (c)

Figure 3: *The need for composed nodes. (a) Multidimensional relevance data, (b) a relevance-map with no composed nodes, and (c) a relevance-map with multi-featured (composed) nodes.*

rather to increase user intuition of its information space.

Although using topological regions formed by more than three or four features seems neither intuitive nor mathematically inconsistent, the method can be extended to more features. Our framework supposition is that the relevance map represents the relevance of points to the feature nodes. So most of the points are placed in a way that best expresses their characteristics while maintaining a low degree of inconsistency. This supposition translates the problem to an optimization problem in which the placement is a function of the given points and feature nodes.

An example of a five-feature placement is illustrated in Figure 3(c). This example shows the five features F1 to F5, and three more nodes. The “F1,F2,F3,F4” node is for points which are highly relevant to all those four features, and the “F1,F2,F3” node is for points which are highly relevant especially to F1, F2 and F3. Similarly, node “F4,F5” is for points relevant to both features F4 and F5. The additional three *composed* nodes are needed because a significant portion of the points are categorized to these three groups, and at the same time are not creating too many inconsistencies in the map. Note that the selection of additional nodes is driven by the characteristics of the population points and thus the map adapts itself to a “clear” display of the relevant points. Note that the map in Figure 3(b) does not provide any effective insight. Clearly, the map in Figure 3(c) is more effective as it emphasizes the different dependencies among the objects with respect to the features.

3 Placement algorithm

The point placement around the feature nodes (inside the topological regions) can be perceived as celestial bodies influenced by gravitational forces. Each node affects its surrounding environment, pulling strongly on the relevant points and weakly on the irrelevant ones.

The layout process and generation of the relevance map consists of two stages. First, the feature-nodes are placed and then the object-points. The feature placement consists of two sub-stages:

1. Creation of composed nodes. The system examines the classification of points and if a significant portion of the population is found to correlate closely to more than one feature, a new node is generated, representing a composition of features. The newly created nodes are referred to as composed nodes. The system locates an n-tuple of features with enough corresponding points which are highly relevant to all of the n features.

2. Placement of features. To place the nodes on the map, an undirected graph is constructed where edges are defined between composed nodes and source basic nodes. For example, see the graph shown in Figure 6 where the edges are drawn in light grey. The edges emanate from the basic feature-nodes F1-F5 to the two composed nodes.

This graph is used as input for the simulated annealing algorithm, which positions the graph nodes “nicely” on the 2D map [9], according to some heuristic criteria.

Simulated annealing is a flexible optimization method, suited for

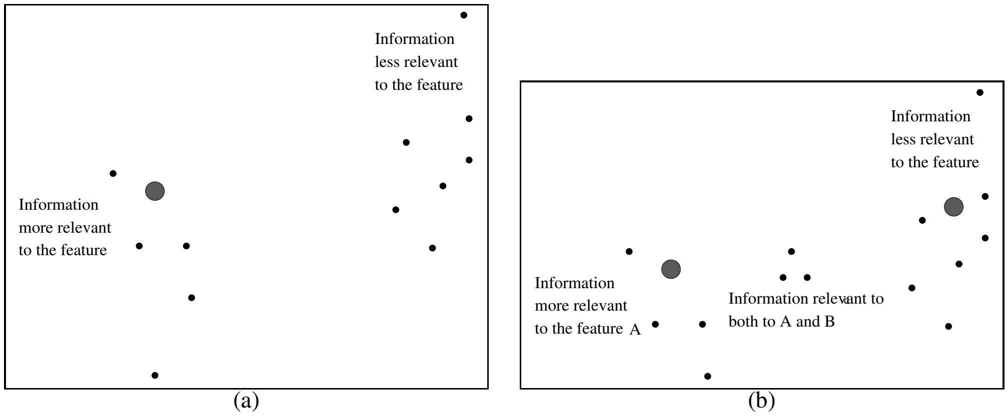


Figure 4: *(a) A relevance-map of a single feature, and (b) a relevance-map of two features.*

large scale combinatorial problems. This method differs from standard iterative improvement methods by allowing "uphill" moves - moves that degrade, rather than improve, the temporary solution. This provides more flexibility enable escapes from local minima solutions to the rather desired global minimum. Simulated annealing derives from its similarity to the process in which liquids are cooled to a crystalline form, a process called annealing. The annealing algorithm is based upon grading a given graph situation according to several simple factors, and allowing the annealing process to try and locate the global minimum of these rules.

1. Nodes are placed inside the viewing frame.
2. Nodes are distributed evenly over the map.
3. Basic nodes are distributed evenly around their composed nodes with even edge lengths.
4. Edges do not cross each other, and are away from the other nodes.

Rule 1 states that nodes should be placed far from the viewing window frame and inside it. An example of the effect this rule may have on a seven-feature node is demonstrated in Figure 6(a).

Rule 2 states that nodes should be distributed evenly over the map, and farther away from each other. This rule checks the distance between each pair of nodes in the graph, and gives a lower cost for graphs which have nodes separated as far from each other as possible. As an example, applying this rule on top of the first rule, produces the result demonstrated in Figure 6(b).

Rule 3 states that a composed node should have its basic generating nodes around it, and at equal distances from it. This rule is implied by the desire to see the composed node in the center of its components, and is calculated by comparing the distance between the weight point of the basic nodes and the location of the composed node, and by measuring the variance of edge lengths. A simple example of how this rule contributes to the understanding of the map, is shown in Figure 6(c).

The last rule states that edges should be as far away from other edges and nodes as much as possible. Although this rule can create conflicting results in non-planar graphs, its role is mainly to minimize crossing edges, and trying to space the graph placement. For example, adding this rule , produces the graph shown in Figure 6(d).

Each graph position has a value which reflects its accommodation to the above criteria. The simulated annealing algorithm iteratively searches for the position which minimizes the graph cost. The convergence to the final position is quite fast, for example on a 15 nodes graph it finishes the placement of the graph nodes on a HP 715/75 workstation in less than 2 seconds. Starting from an arbitrary position (Figure 7(a)), after 35 iterative steps the graph converges to its final position (Figures 7(b-d) display stages 5, 30 and 33 respectively).

Once the feature-nodes have been placed, the object points have to be positioned. Their initial location is around their top-graded feature-nodes. The other nodes influence the position of the points by pulling them by "forces" determined with respect to their grades. Points which have no nodes influencing them, are given a distinct position on the upper left-hand corner of the screen.

The final position of the objects is determined by a relaxation process, where the points are perturbed to be evenly distributed across the map. This is especially important in crowded regions. The very slight movements of the relaxation do not change the topology of the generated map, but only locally adjust the point positions for a better display.

4 Relevance maps for Web searching

The technique described above has been tested in the context of a Web search. The RMAP (Relevance Map) prototype system was built to experiment and demonstrate the effectiveness of relevance maps as a tool for visualizing the result of a WWW query.

To retrieve information from the Internet dealing with a specific topic, several index servers, search engines, and query languages were developed (for several examples see [17, 8, 16, 22, 25]). The answer of a search engine to a query is often a large set of candidate documents. Rather than presenting the answer as a long linear list, the RMAP system displays them in relevance maps that reflect the relevance of the documents to various features that are likely to be of interest to the user. The system enables the user to query standard index servers, view the results as a map, interact with the map, adjust the layout (if needed) using coarser or finer feature granularity, and access or browse through the relevant data.

The implementation of the prototype is based on the architecture seen in Figure 8. The first component performs a Web search according to the user request. In the prototype we used the AltaVista index server as a query engine. The returned set of (pointers to) documents is referred to below as the *result population*. Next, the documents of the result population are loaded and analyzed, and an initial set of candidate features is selected. The relevance of each document to each feature is then determined in a process called *classification*. The feature selection and the document classification are based on an analysis of the document's text. The use of keywords and a feature database, as well as user interaction, is beyond the scope of this paper, and is described in [3].

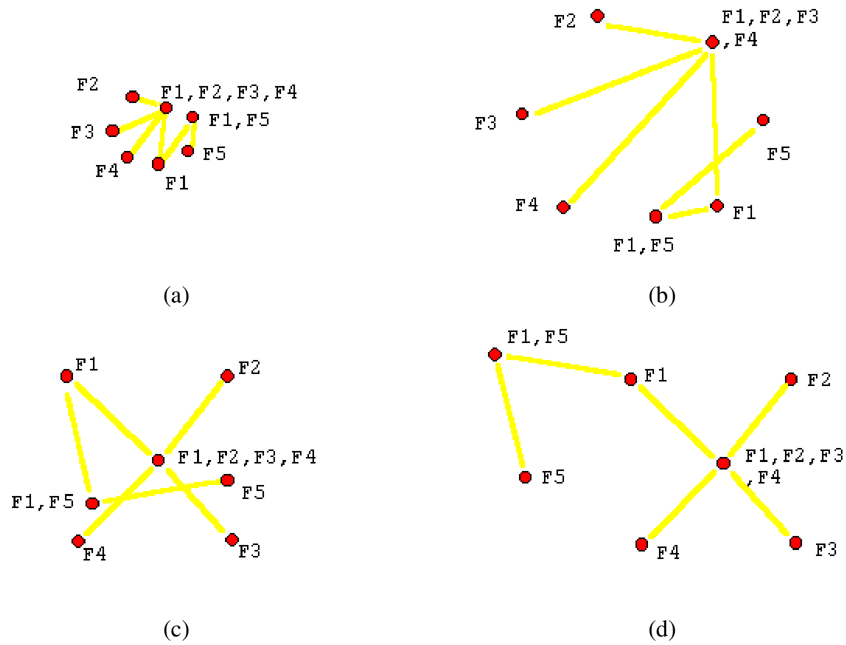


Figure 6: *The effect of the four rules on a simple case.*

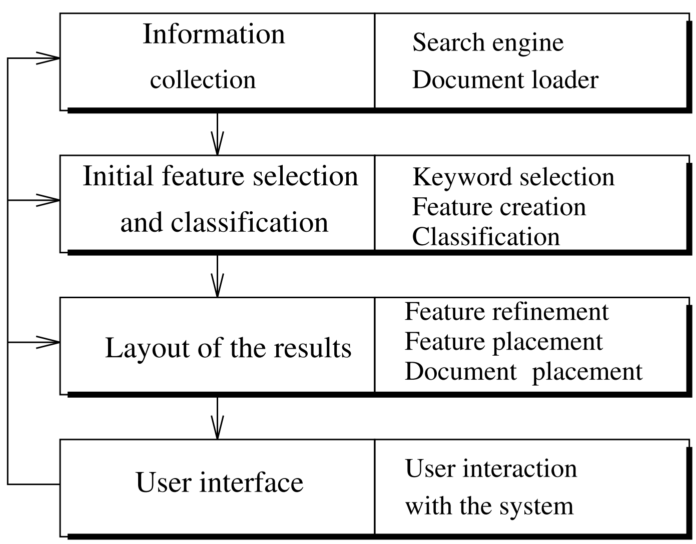


Figure 8: The prototype building blocks.

Since the latency time for loading a document may be high, they are loaded in parallel. Also, the user can limit the number of documents to be considered, in which case only the first n documents returned by the index server are loaded. The documents are stored locally for possible further usage (e.g. if the user wants to see the data displayed with respect to a different set of features). The last component interacts with the user. The relevance map is used as a working tool for the user. It thus has to provide convenient means for accessing the documents on the map, adjusting the layout and using coarser or finer feature granularities, adding documents to the display, storing the data, and submitting additional queries.

The relevance map is connected to a standard Netscape browser. The connection between the browser and the map is bi-directional. By selecting an information item (document), the browser loads the associated page, and the document icon on the map is highlighted. In the other direction, whenever the user loads new documents through the browser, the documents are automatically classified and placed into the relevance map according to their grades. The user can also directly modify the relevance map, by changing or deleting the positions of the feature nodes and of the document points. The system is built in a modular way, so that each of the four main components can be easily replaced by other possible implementations. This architecture enables one to experiment with different search engines, feature selections and classifications, layouts, and user interface techniques.

Example

We demonstrate the use of the system with an example. Assume the user is interested in finding information about the Beatles band and its members. He submits the query “Beatles”. The system issues the query, analyzes the returning documents, and extracts about 30 candidate features for to focus, including the band member names. The features are then offered to the user who selects the names of the four band members, and bounds the number of the documents to be displayed to 50. *

We first used a high threshold for the creation of composed nodes. The resulting map is shown in Figure 9. In the map, we can see five feature nodes. Four of them correspond to the actual band members (the nodes labeled Paul, George, John, Ringo). The fifth node is a composed node that was automatically created to highlight the fact that most of the documents in the center mention all four members.

Lowering the threshold for the creation of composed nodes, generated the relevance map shown in Figure 10. This map includes an additional node – the ”George and John and Paul” node. It reflects the fact that among the documents in the center there is another

* A similar display is obtained without limiting the number of results, except that in that case the screen is more loaded, and zooming on the various areas is thus needed to see the full titles of documents.

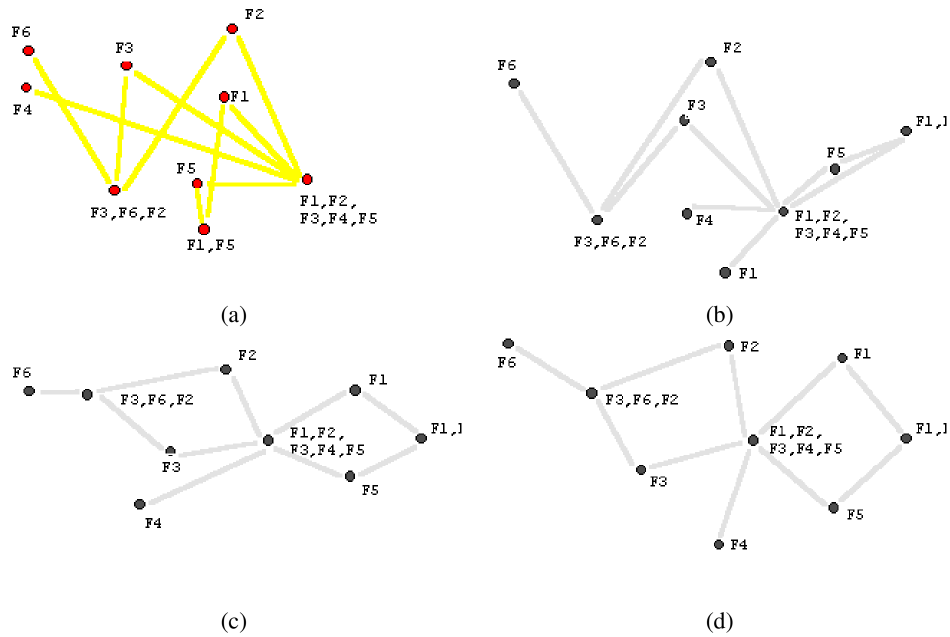


Figure 7: *Four simulated annealing iterations over a graph.*

significant group of documents, documents mentioning the three members but not Ringo. The simulated annealing algorithm placed the feature nodes so that the document location is most intuitive according to the information in hand.

The visual load

For a Web search application the representation of the information credentials in the map is vitally important to the success of the visualization map. The object title should be contained in a minimal display bounding box. In some cases creating a thumbprint of the object, or placing one of its embedded pictures might be used as an identified icon. Since the display size is limited, and the amount of information is considerable large, the visualization problem, known as the visual load, is encountered [6]. Overloaded displays are less intuitive and their usefulness is lost. The visual load of complex maps can be alleviated by hierarchies and/or hyperbolic zoom. cascon95-multisurf Fisheye

An hierarchy of features creates several levels of containment, and lowers the number of objects displayed in each level. Hierarchies are useful in visually loaded regions where too many information items are closely placed. This solution is familiar from road maps where overloaded regions are referenced to a separate zoomed-in map. Hyperbolic display provides zoom-in regions of interest while zooming-out the background. The effect is like that of an electronic magnifying glass which uses hyperbolic deformation, known as fish eye view.

Our layout method is independent of the above techniques. However, we have implemented a zoom-in mechanism with which the user can scale-up the region of interest. Figure 11 shows a zoom on the John Lennon region of the map. The hyperlinks to documents are displayed with icons taken from the documents themselves. These provide more information about the correlation between the map presentation and the documents.

5 Related work

The field of information visualization has been explored by many researches in the last few years. Works in Neural networks[21], information retrieval[13], scientific and information visualization[24, 2, 18] provide a wide range of solutions, designed to alleviate problems of displaying high number of dimensions.

Neural network based systems, designed to learn the characteristics of n-dimensional space and to display a map of relations between the objects in the space, was introduced by Kohonen [21]. Later work with this system introduces updates and evaluation of the method, showing some capabilities of associative thought, but still does not represent the space in a comprehensible and intuitive way[12].

Scientific generic visualizations such as parallel coordinates [18] and pixel representations [24], used in the information retrieval field, also provide a visualization picture, which is optimal to data mining. It allows the researcher to locate data behavior, but again, looses the space/objects intuitive relations [13].

The intuitive approach described in this paper is a variation of Venn diagrams. These were used in several previous systems such as VIBE[23], Lyberworlds[14], Cougar [13], Infocrystal [30], Idm [32], and others. We briefly describe below the systems and explain how our work differs from them.

VIBE is a tool for displaying high dimensionality of documents representation, where query terms are applying an attraction to the relevant documents. As mentioned in [14], this creates a inconsistency problem, in case of more than 3 terms query. To minimize the problem the system allows the user to change the location of the terms, thus regenerating the result space projection.

Lyberworld [14], extends the VIBE method, by transforming the space into a relevance sphere allowing a higher degree of freedom for term placement, thus lowering possibility for inconsistency, in positions of items. Another extension of VIBE, is VR-VIBE [5], which reduces VIBE view inconsistencies by allowing the user to place the terms in 3D, thus creating a 3D space.

Cougar is a computer human interface based upon Venn diagrams, developed for enabling users to asses documents similarity. The interface is is constrained only to three circles view (8 topology different regions), and higher dimensions are being expressed with using hierarchy set of diagrams. In any case, this interface is not used to display the data itselft but rather the collection sets of the data.

The Infocrystal system [30, 13] is a sophisticated interface allowing more than 3 dimensions of documents space to be demonstrated based upon placement scheme similar to Venn diagrams, but also extends this method by using shapes indicating the influences on each of the entities displayed.

Later work, within the SPIRE project included two technologies, called Galaxies, and Themescapes [31]. The first, displaying documents in their information space as stars, located according to their relevance to a set of terms, while the 2nd , displaying the documents in a 2.5 D (spatial terrain like) map build using the certain themes as attracting anchors in Themespaces.

Our solution differs from the above, by expanding the Venn diagram scheme to accommodate n dimensional data, exploring the data to be visualized, and providing an optimal solution for that space. Furthermore, placement of the terms, using the simulated annealing instead of asking for Human intervention allows to create a more optimal placement. This solution provides a satisfying solution in 2D map, generating an intuitive map.

6 Conclusion

In this paper we introduced *relevance maps*, a visualization paradigm to assist in analyzing the relevance of the objects to various features that are likely to be of interest to the user. A key observation is that for the type of information retrieval applications we are dealing with, the *exact* relevance measures are not significant, and approximation suffices. Thus accuracy can be traded for a clearer display. This is used to improve the display by refining the set of features and introducing the notion of composed features. An optimized layout for the map is then obtained using simulated annealing.

Relevance maps have been implemented and tested, in the context of a Web search. The RMAP prototype system enables users to query standard index servers, interact with the map, adjust the layout (if needed) using finer or coarser feature granularity, and access the relevant data.

Better methods of extracting the features from the data and for refining the selection to get a better display, are of major importance for the success of the system. We plan to investigate and experiment with more alternatives and test their adequacy to the system. Furthermore, we intend exploring the advantages of using composed nodes, as well as examining and improving the placement algorithm, to accurately show complex feature relations.

The described prototype which joins existing members of the Internet visualization tools, are based on a multitude of techniques such as [7, 26, 10, 11, 1, 15, 28, 4, 6]. Combining these techniques, may produce better tools with better representation of the information space.

References

[1] K. Andrews. Visualising cyberspace: Information visualization in the Harmony Internet browser. In *Proceedings '95 Information Visualization*, pages 97–104, Atlanta, October 1995.

[2] M. Ankerst, D. Keim, and H. Kriegel. Circle segments': A technique for visually exploring large multidimensional data sets. In *Proc. of Visualization 96*, December 1996.

[3] J. Assa, D. Cohen-Or, and T. Milo. Relevance maps. *Technical Report, Computer Science Department, Tel-Aviv University*, 1997. *ftp.math.tau.ac.il/pub/daniel/assa.ps.gz*.

[4] E. Ayers and J. Stasko. Using graphic history in browsing the World Wide Web. In *Proc. of the Fourth International World Wide Web Conference*, Boston, December 1995.

[5] S. benford, D. Snowdon, C. Greenhalgh, R. Ingram, I. Knox, and C. Brown. Vr-vibe: A virtual environment for cooperative information retrieval. In *Proc. of Eurographics 95*, pages 349–360, Maastricht, The Netherlands, 1995.

[6] G. Collaud, J. Dill, C. Jones, and P. Tan. The continuously zoomed web - a graphical navigation aid for www. In *Proc. of Visualization 96*, December 1996.

[7] M. Consens, F. Eigler, M. Hasan, A. Mendelzon, E. Noik, A. Ryman, and D. Vista. Architecture and applications of the Hy+ visualization system. *IBM Systems Journal*, 33(3):458–476, 1994.

[8] Digital Equipment Corporation. Alta vista: Main page. 1996. *http://altavista.digital.com*.

[9] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM transactions on graphics*, 15(4):301–331, 1996.

[10] P. Doemel. WebMap – a graphical hypertext navigation tool. In *Proc. of the Second International World Wide Web Conference*, Chicago, October 1994.

[11] B. Gaines and M. Shaw. WebMap: Concept mapping on the Web. In *Proc. of the Fourth International World Wide Web Conference*, Boston, 1995.

[12] R. Orwig H. Chen, C. Schuffels. Internet categorization and search: A self organizing approach. *Journal of Visual Communication and Image Representation*, 7(1):88–102, 1995.

[13] M. Hearst. Using categories to provide context for full-text retrieval results. In *Proc of the RIAO 94*, New York, 1994.

[14] M. Hemmje. Lyberworld - a visualization user interface with full text retrieval. In *Proc. of SIGIR 94*, 1994.

[15] R. Hendley, N. Drew, A. Wood, and R. Beale. Narcissus: Visualizing information. In *Proceedings '95 Information Visualization*, pages 90–96, Atlanta, October 1995.

[16] Excite Inc. Excite: Main page. 1996. *http://www.excite.com*.

[17] Yahoo Inc. Yahoo: Main page. 1996. *http://www.yahoo.com*.

[18] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. of Visualization 90*, 1990.

[19] D. Keim and H. Kriegel. Visdb: A system for visualizing large databases. In *ACM SIGMOD International conference on Management of Data*, San jose, October 1995.

[20] D. Keim and H. Kriegel. Visualization technique for mining large database. *Journal of Computational and Graphical Statistics.*, 8(6):923–938, 1996.

[21] T. Kohonen. The self organizing maps. In *Proc of the IEEE symp. on Neural Networks95*, volume 78(9), pages 1464–1480, 1995.

[22] D. Konopnicki and O. Shmueli. W3QS: A query system for the World Wide Web. In *Proc. of VLDB'95*, pages 54–65, 1995.

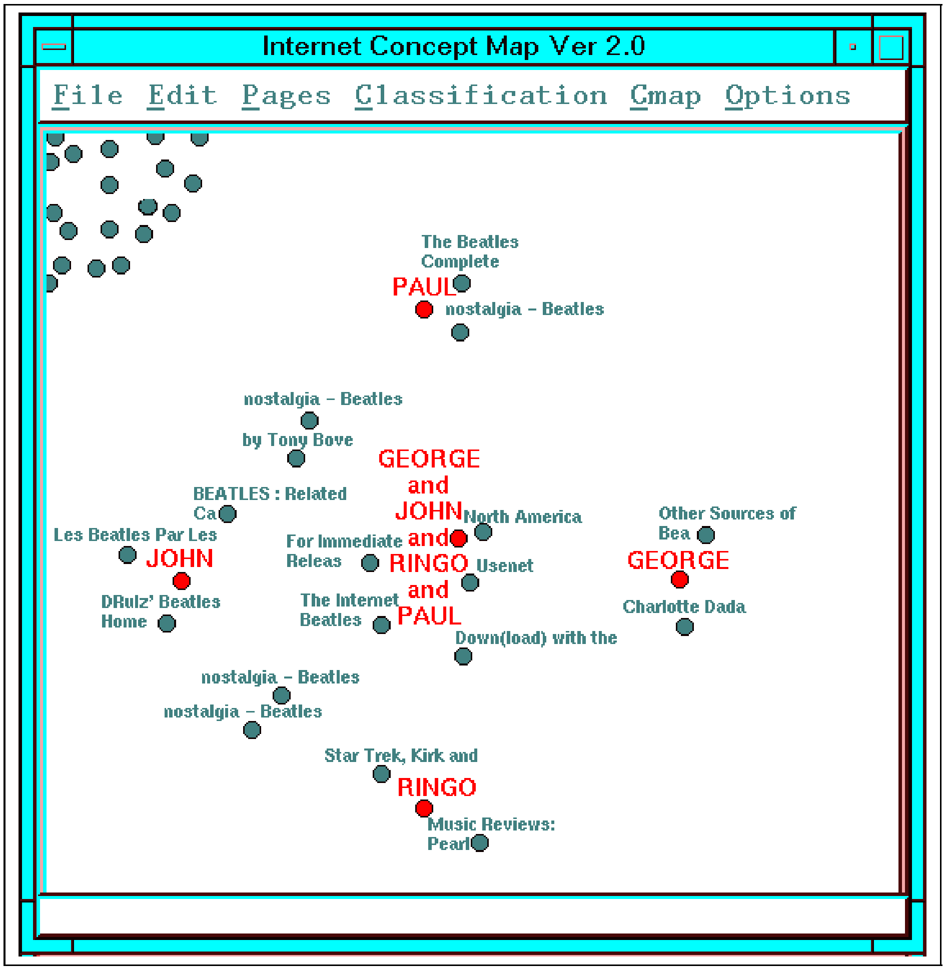


Figure 9: A relevance map generated for the Beatles query.

[23] R. Korfhage. To see or not to see - is that the query? In *Proc of the 14th Annual International ACM/SIGIR conference*, pages 134–141, Chicago, 1991.

[24] H. Kriegel and D. Keim. A system for visualizing large databases, system demonstration. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 1995.

[25] A. Mendelzon, G. Mihaila, and T. Milo. Querying the world wide. In *Proc. of PDIS'96*, 1996.

[26] S. Mukherjea and J. Foley. Visualizing the World Wide Web with the Navigational View Builder. *Computer Networks and ISDN Systems*, 27:1075–1087, 1995.

[27] F. Das Neves. The aleph: A tool to spatially represent user knowledge about the www docuverse. In *to appear in Proc. ACM Hypertext '97*, 1997.

[28] G. Robertson, S. Card, and C. Roberts. Information visualization using 3D interactive animation. *Comm. of the ACM*, 36(4):57–71, apr 1993.

[29] F. Ruskey. Venn diagrams. *The Electronic Journal of Combinatorics*, 1(3), 1996.

[30] A. Spoerry. Infocrystal: A visual tool for information retrieval & management. In *Proc. of information knowledge and management 93*, Washington D.C., 1993.

[31] J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proc. of information visualization 95*, Atlanta, Georgia, 1995.

[32] M. Zizi. Providing maps to support the early stage of design of hypermedia systems. In *Proc. of IWHD'95 International Workshop on Hypermedia Design*, December 1995.