# Visual Interfaces for Semantic Information

Paul Janecek
Human Computer Interaction Group
Database Laboratory
Swiss Federal Institute of Technology (EPFL)
CH-1015 Ecublens, Switzerland
paul.janecek@epfl.ch

Pearl Pu
Human Computer Interaction Group
Database Laboratory
Swiss Federal Institute of Technology (EPFL)
CH-1015 Ecublens, Switzerland
pearl.pu@epfl.ch

## Abstract

By capitalizing on human's powerful visual processing capabilities, visual interfaces play an increasingly important role in how we access, analyze, and understand information. At the same time, as information becomes more semantically annotated, interfaces that can explicate semantic relationships of data enable us to verify the encoded information. In this paper we present semantic fisheye views (SFEV) and discuss its properties in integrating semantic relationships into information browsing, and seeking tasks. We describe the formalisms and visual mapping functions of SFEVs, and then present two applications: one for visualizing the structure and relational properties of information in constrained displays, and another for learning and exploring semantic information in image databases.

## Keywords

Visual interfaces, data semantics, semantic fisheye views, information visualization, intelligent interface to information retrieval, WordNet.

## INTRODUCTION

Humans rely increasingly on visual interfaces in information exploration, seeking and access tasks. Furthermore, the interactivity of such interfaces is capable of following the changing context in which users interact with information, an important feature of today's information-driven society. At the same time, exploration of semantic relationships visually can also uncover data relationships unseen by the machines, which have motivated many researchers combining visual interfaces with text or data mining techniques [6].

Two scenarios motivated the research described in this paper:

1. a user booked his flight on his desktop PC, but is now accessing his itinerary via a hand-held device for the returning trip. He would like to review his itinerary in more or less the same format (order of appearance and syntax of words) as displayed on his desktop PCs. With limited screen size though, he is unable to see the itinerary in its entirety. However, if some parts of the itinerary (especially concerning return portion) can be omitted as long as the parts about the departure are there. He has some recollection of the position of each item, so that if he wants, he can move to an omitted item and ask for details.
2. a user is exploring a collection of images. Initially her query is very vague: horses. Since images have usually few words describing them, the matching based on keywords can only return images with the word "horse" or some variation of that word in the annotations. By navigating through a set of words obtained from this initial set, she has discovered that a new query term 'horse riding' can be used to more accurately retrieve images of people riding horses. Moments later, she decided to explore the 'knight horse' sense as defined by WordNet. It is clear from the images that 'knight horse' refers to pieces used in chess. To focus her search, she will now enter the term 'chess' and 'knight horse' at the same time. She hopes to find a focused set of images.

In the first scenario, as the user is temporally and location-wise more concerned with departure, the data elements semantically related to the departure airport and time are more important to him. However, nothing prevents him from checking the arrival airport and time. Thus a context + focus display technique will be ideal, allowing seeing relevant (to user task) information in emphasis, but nevertheless permitting the shifting of interest to items. The second scenario, on the other hand, calls for a visual interface capable of following users query terms as they learn new vocabularies and domain knowledge. We summarize the requirements on the overall properties of such an interface tool for semantic information as follows:

<u>Context + focus:</u> Information browsing and seeking is highly opportunistic [bates]. Users must be able to perceive other possible search goals while exploring the current one. This means that the interface should include what surrounds the focus of user's current interest (context) as well as the focus itself. The visual emphasis of displayed items must be determined based on their semantic distance to the focus, thus providing a semantic context for information perception and exploration. Furthermore, overview and distortion techniques can achieve an combination of general structure, while maintaining detailed information on the focus.

<u>Dynamic selection of focus and contexts:</u> The interface must be dynamic, because users information needs shift when their task context changes, and thus their interest shifts. Displayed items will follow this shift by recalculating their respective semantic distance to the new focus, and their relative visual emphases in relation to current interest.

<u>Multiple foci and multiple contexts:</u> Information seeking is multi-directional with several search goals. One important user activity in exploring multi-directional search is to compare the results and detect patterns and evidences (visual inferencing). Therefore, the interface must be able to support several foci and

their respective contexts. Users should be able to select any item or a group of them as a focus and any number of foci. The interface should display the corresponding contexts so to allow distinction and comparison.

Flexible modeling of semantics: to explore several semantic relationships among data, a flexible mechanism must be used. That is, users must be able to define any type of semantic relationships they want, but guided by the systems.

In this paper, we introduce semantic fisheye view, a general display method attempting to satisfy the above requirements. We first describe a set of techniques giving rise to the ideas behind semantic fisheye views (SFEV); the formalisms and various algorithms comprising a SFEV; the types of semantics relationships that can be modeled; and two applications of SFEV for information browsing, exploration and seeking tasks, followed by a section on related works that compares SFEV to other visual interfaces for information retrieval.

## BACKGROUND

As the size of data collections grow, it is increasingly difficult to represent all of the information in the limited space of a display, and to navigate within the representation at different levels of detail. These problems are even more challenging for small displays, such as in handheld devices.

Fisheye (also called "focus + context") views are interactive visualization techniques that address these problems by directly relating the visual emphasis of information to a measure of the user's current interest. These techniques reveal hidden relationships in a representation by visually emphasizing the most relevant objects and de-emphasizing less relevant objects. These techniques also create compact displays of information by showing only the most relevant objects. For example, graphical distortion techniques may be used to increase the detail of objects near the focus and progressively reduce the detail of more distant objects. In a general sense, fisheye views are constructed by pairing a function to measure interest with one or more emphasis techniques.

Fisheye views were originally developed as a way of balancing local detail and global context in the interface, based on how humans conceptually structure and manage large collections of information [6]. In his seminal research paper on the subject, Furnas remarked that people tend to recall information with respect to its semantic importance, in greater detail in the conceptual "neighborhood" of the current focus, and only "landmark" information at greater conceptual distances. This general structure is found in many semantic contexts, from the cognitive maps people use to navigate through the physical environment, to their knowledge of organizational structures. The term "fisheye" is an analogy to a wide-angle lens that shows an area around the focus in greater detail, and radically distorts more distant objects to fit in the periphery.

## FORMALISMS OF SEMANTIC FISHEYE VIEW

A SFEV gives a score, called the Degree Of Interest (*DOI*), to each element of a given set of data. This score is defined as the difference between the *a priori interest* (*API*) of that element and the distance, *D,* between the element and the current focus, *fp* [14].

$$DOI(x \mid fp = y) = API(x) - D(x, y)$$

API and D (distance) are themselves functions, which are determined by the underlying semantic structure. In general, DOI is an algorithm for determining the most relevant set (high scored data elements) among all data depending on the current user task. Simple DOI functions without using API and D also exist, such as those determined by user selection and database queries. These DOIs give only binary or nominal information (e.g., "selected", or "not selected") as scores. More complex functions, such as the "relevance score" of an information retrieval engine, give an ordered distribution of interest that can be interactively refined by the user. The results of these ordinal or quantitative interest functions can often be further analyzed, for example through statistical methods or clustering algorithms, to reveal the structure and distribution of relevant results [3]. Interest (or semantic relevance) of each object is calculated with respect to a particular focus.

Furnas originally applied this equation to abstract data structures, such as hierarchies, structured text, and calendars, but suggested that it could be applied in any domain where *API* and *distance* functions could be defined.

The *API* is the importance of an object independent of any focus. Objects with a high *API* are considered "landmarks" in the information space. The *API* may be either specified (e.g., measured from user studies), or derived algorithmically from properties of the information collection (e.g., calculated from structural metrics). For example, Mukherjea calculated the *API* of nodes in the WWW as a combination of connectedness, in/out degree, access frequency, and depth [12]. *API* may also change over time to reflect user interaction [1,10,15]

The *distance* function is a measure of the conceptual distance between the focus and a data element in the collection. This is the component of *DOI* that responds to changes in the user's focus. We are investigating how to integrate multiple types of distance metrics based on information content, structure, user tasks, and interaction.

In order to design more sophisticated SFEVs, capable of modeling hybrid API and Distance functions, we have proposed the following general *DOI* function [8]:

$$DOI_{context} = f(API, w_i, dist_i)$$

Where *dist* and *w* are one or more distance metrics and their associated weight. We foresee the weight vector, ***w***, as a method for designers, and potentially users, to specify the relative importance of distance functions in different contexts. This contextual weighting factor is the most significant difference between this work and the general *DOI* proposed by Rüger, et al. [15].

## DISPLAYING SFEV

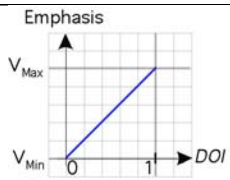
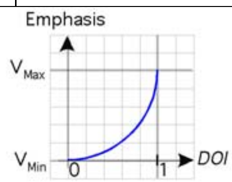
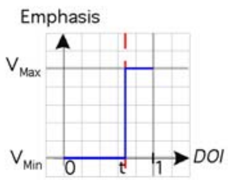
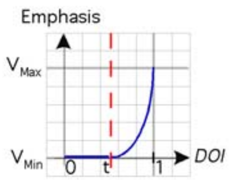
A fundamental goal of visually representing information is to replace cognitive tasks by perceptual tasks, thus enabling users to "*see*" rather than "*read*" the relationships between information. The second step in implementing a semantic fisheye view is the visual rendering of *DOI* in the display, calculated by emphasis algorithms. An *emphasis algorithm* consists of a transform function (what information and how their relative importance will be displayed), and a visual scale (what symbols will be used to display them).

## Visual Transformation Functions – Distortion Techniques

A visual transformation function defines the mapping between the calculated *DOI* (x axis) and the final visual representation (y axis) using methods such as the size and position of objects in a display.

Table 1 shows a taxonomy of transformation functions for different types of fisheye views (adapted from Leung and Apperley [9]).

**Table 1. Visual transformation functions for SFEVs**

| | Linear | Non Linear |
|---|---|---|
| Continuous |   |   |
| Non-continuous |   |   |
| | | |

Geometric distortion techniques apply these transformation functions to spatial scales, such as position and size. For example, the graphical fisheye views of Sarkar & Brown [15] can be defined by a continuous non-linear transformation, and Rao's Table Lens [13] can be defined by a non-continuous linear transformation.

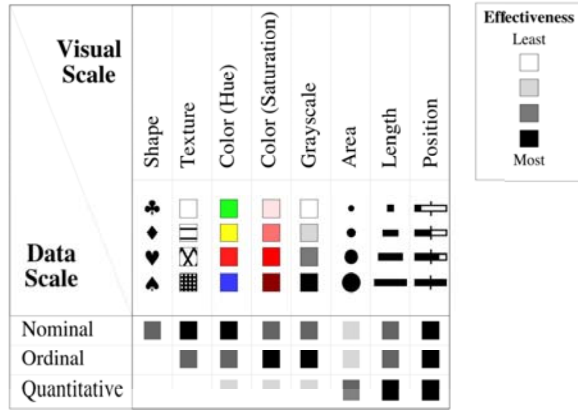
These transformation functions can also be applied to other visual scales. For example, in the flight information browser (described later), we created a hybrid fisheye view by applying a continuous linear transformation to grayscale and size, and a non-continuous linear transformation to level of detail.

### Visual scales

Ideally, the visual scale will maximize the use of pre-attentive perception for revealing the relationships in information. The effectiveness of the visual scale in representing *DOI* is the scale of information most appropriate for the encoding the data (i.e., nominal, ordinal, and quantitative). Position is the most effective attribute for encoding all scales of information, hence the particular effectiveness of charts for comparing two variables and maps for representing spatial relationships. Table 2 is a summary of visual scales based on Bertin's classical work modified to include more recent research [8].

In summary, a semantic fisheye view can be developed using this framework based on two general components: a *DOI* function and one or more emphasis algorithms. The *DOI* function consists of an *API* function and one or more weighted distance functions, and an *emphasis algorithm* consists of a transform function and a visual scale.

**Table 2. The relative effectiveness of visual scales for representing nominal, ordinal, and quantitative data.**




## TYPES OF SEMANTICS

Interest functions and emphasis techniques may be based on different semantics of the data represented in the interface, such as content, structure, history of interaction, and relevance to a specific user task.

Content-based metrics are derived from the values of information attributes (e.g., similarity measures in information retrieval). Structure-based metrics are derived from the structure within and between information (e.g., path distance in graphs and hierarchies). Herman, et al., describe both of these types of metrics in much greater detail [7]. User task-based metrics are derived from an analysis of the relevance of information for completing specific tasks (e.g., the relative order of steps in a process). Interaction-based metrics are derived from the recency and frequency of a user's interaction with information (e.g., the history of a user's navigation in hypertext).

To illustrate SFEV's ability to model a wide range of semantic relationship in different user context and its various functions in action, we describe two applications. The first one applies SFEV to display a small set of semantically relevant information for the purpose of allowing users to browse a big set of data in a constrained display, while the second is for exploring word semantics in image retrieval when users' goals are ill-defined in the beginning (berry picking).

## APPLICATION 1: SFEV FOR DISPLAY ON MOBILE DEVICES

Here we explore the location-based semantic information based on data structure, content, while employing emphasis techniques such as font size, gray scales, and filtering.

The data from Figure 1 is a simplified itinerary showing a flight departing from Geneva, stopping for several days in Denver, continuing to Oakland, back to Geneva from San Francisco International airport a few days later. The first two columns are the numbers of the flight legs and flight segments, and are not usually shown in an itinerary. Displaying all of this information on a mobile device will require zooming and scrolling. It is not possible to have an overview of the whole itinerary while having the detailed information on some parts of the itinerary at the same time. Using SFEV, however, we able to track user's current interest depending on the data item that he has selected. From this focus, we can calculate DOIs for the other data items.

Then emphasis techniques will map these DOI scores to a combination of font sizes, levels of detail (words are entirely spelled vs. abbreviated), gray scales, or a combination of all of them.

| # | Seg # | Date | Airport | Dep/Arr | Time | Airline | Fligh... | Duration |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 17.1 | GVA | Dep | 12:15 | SA | 138 | 8h45 |
| 1 | 1 | 17.1 | JFK | Arr | 15:00 | | | 2h50 |
| 1 | 2 | 17.1 | JFK | Dep | 17:50 | SA | 8178 | 5h23 |
| 1 | 2 | 17.1 | DEN | Arr | 20:27 | | | 16h58 |
| 2 | 1 | 20.1 | DEN | Dep | 09:15 | UA | 389 | 1h40 |
| 2 | 1 | 20.1 | OAK | Arr | 10:55 | | | 1h40 |
| 3 | 1 | 23.1 | SFO | Dep | 21:20 | SA | 109 | 11h00 |
| 3 | 1 | 24.1 | ZUR | Arr | 17:20 | | | 0h50 |
| 3 | 2 | 24.1 | ZUR | Dep | 18:10 | SA | 940 | 0h45 |
| 3 | 2 | 24.1 | GVA | Arr | 18:55 | | | 12h35 |

**Figure 1. A flight itinerary displayed in a tabular representation.**

## Degree of interest function used in flight display

The *DOI* used in this prototype for each cell of the tabular display is calculated using the following equation:

$$DOI(p) = API(p) - \sum_{i=1}^{n} w_i dist_i(p, fp)$$

The *API* is a scale representing the levels of importance of data items based on our analysis of user's requirement for an overview (**Error! Reference source not found.**). At the highest level is the initial departure and final arrival *DateTime* and *Airport* of each *FlightLeg*. At the next level is the *DateTime* and *Airport* of each *Layover*, followed by individual *Flight* information. At the lowest level are the *Segment.# and* duplicate information, such as duplicate dates. Notice that if we apply the filtering emphasis technique, the data items in light gray with small font sizes will not be displayed at all. In that case, the overview will be much more compact, while an overall context is still maintained.

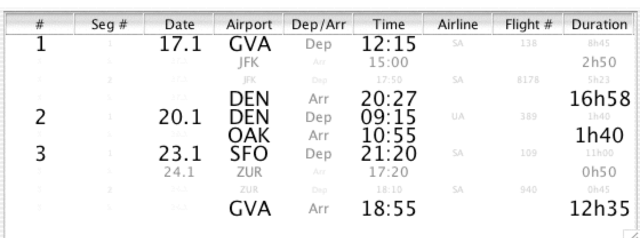
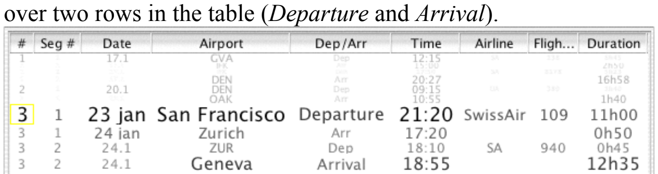


**Figure 2. An overview of the itinerary created by setting DOI=API using a font-size emphasis algorithm.**

We have implemented two general types of distance metrics in the prototype: structure-based and content-based. *Structure-based* metrics (Figure 3) calculate distance based on the actual or derived structure of the data using entity relations [8]. For instance, departure and arrival share the same structural semantic because they are at the same level in the ER model of a flight. The first structural metric used in the prototype calculates the path distance between *Flights*, which are represented over two rows in the table (*Departure* and *Arrival*).



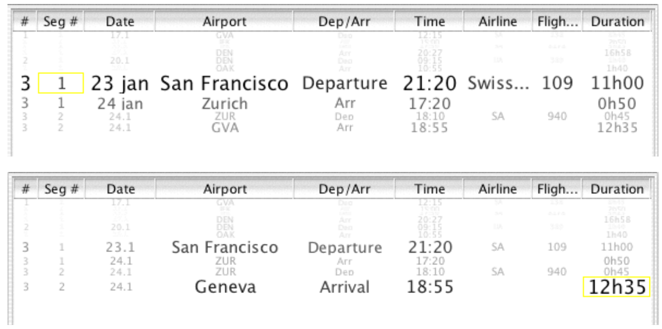


**Figure 3. Structure-based metrics. In the top two figures, *dist* is based on the hierarchical path distance between *FlightLegs* (first column) and *Flight Segments* (second column). In the bottom figure, *dist* is based on the *DateTimes* used to calculate *Duration* (last column).**

The second type of structural metric implemented in the prototype is based on the relationship between *Durations* and the *DateTime*s they are calculated from. For example, a *Flight.Duration* is calculated as the differenced between the *Arrival.DateTime* and the *Departure.DateTime*. Figure shows the change in *DOI* calculated by different structural distance metrics as the focus moves from *FlightLeg* 3 (top figure), to *Segment* 3.1 (middle figure), to the *Duration* of *FlightLeg* 3 (bottom figure).

*Content-based* metrics compare the similarity of the focus value to the values in other cells in the same column (Figure ). Each column has a different type of data (e.g., date, time, airport), and in this prototype the distance is simply a matching algorithm. More sophisticated strategies could use semantically richer distance models (e.g., the geographic distance between airports).

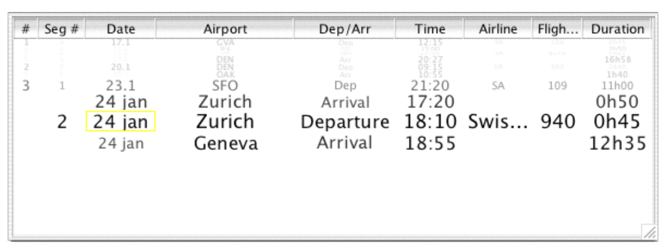


**Figure 4. Content-based metrics. When the focus is in the *Date* column, *dist* is calculated on the similarity between dates.**

## Emphasis algorithms used for flight display

We have implemented five different emphasis algorithms in the prototype: font size, grayscale, row height, semantic zoom, and filtering. Four of the transformation functions associated with these algorithms are shown in Figure 5 (please refer to [8] for details). In the default view, all data elements are displayed as they were. Using variations on gray scales, important information is displayed with dark fonts, while unimportant ones are in light gray. Using level of details, unimportant information has been abbreviated while no change has been made to others. A hybrid approach which combines gray scale, size of fonts, and level of details is shown in the last row. As mentioned earlier, filtering emphasis algorithms can be used to take out rows in light gray or abbreviated forms from the tabular display, achieving compact displays of itineraries.

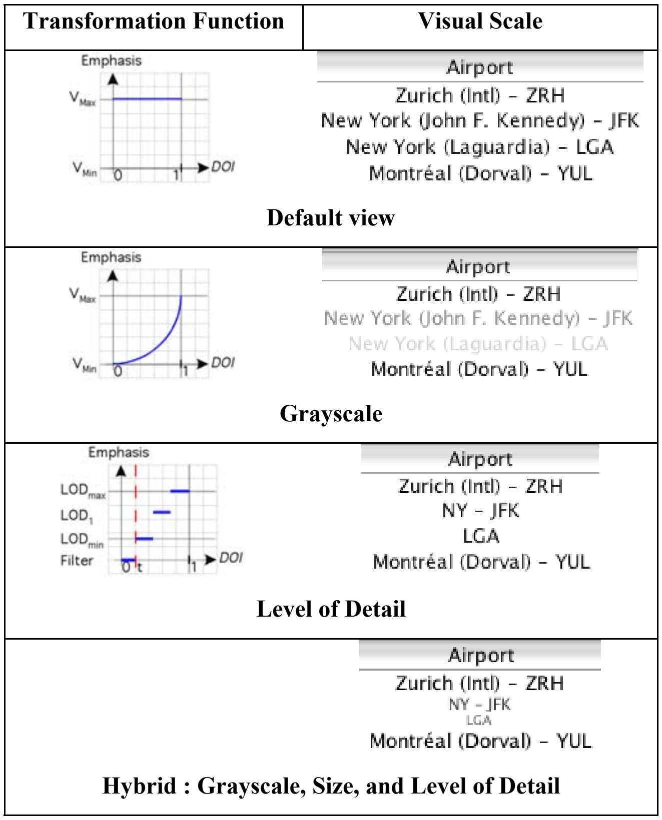



**Figure 5. Applying transformation functions to visual scales**

## APPLICATION 2: SFEV FOR INFORMATION SEEKING

We have developed a SFEV for information seeking in a large database of annotated images (56000 images). This prototype integrates two types of semantic information about the images. First, each image has a caption and a set of, on average, 25 keywords (a vocabulary of 28000 unique keywords over the entire collection). Second, we use WordNet, an extensive network of semantic and lexical relationships between words, to disambiguate and extend the image keywords.

To create a semantic fisheye view integrating these two models, we have defined a *DOI* function that combines the cosine similarity measure of the vector space model with the path distance between senses in WordNet. We linearly scale image size to reflect *DOI*.

In the interface, the user is given access to the keywords, their senses, and an overview of the collection of images. When a user selects a keyword (the "focus"), the related senses for the keyword are shown (the semantic "context" of the word), and all the visible images in the collection are dynamically resized to reflect their *DOI* (the similarity "context" for the query). This visual feedback allows the user to immediately evaluate the effectiveness of their query, and rapidly change their focus to other keywords, senses, or images.

### Types of user tasks

Our prototype supports these user information seeking tasks:

1. Search by key words
2. Query expansion by adding terms learned from keyword list appearing in retrieved images
3. Query expansion by adding terms learned from related words found in WordNet
4. Sense disambiguation from consulting WordNet
5. Narrow down result set by adding more keywords that are semantically related to the query, thus obtaining a more focus result set

### Examples

After a user has typed the keyword 'horse,' the query term is used to get all words containing that term, such as Arabian horse, horse riding, and etc. Figure 6 shows the set of images retrieved by SFEV using DOI using only the vector space model. As expected, the results are very non-homogenous, containing all kinds of images. Suppose the user selects 'horse shoe.' SFEV will now display a much more focused collection of images containing 'horse shoe.' Noticing that there is another term 'horseshoe,' she now selects both terms. This will strengthen the focus of word semantics for 'horse shoe', thus enabling SFEV to display better results (Figure 7).

In a different scenario, the user consults WordNet on 'horse,' for sense disambiguation. Realizing that 'knight horse' is a piece used in chess, she added the word 'chess' in the keyword list. SFEV now shifts its focus to chess related horses and displays images accordingly (Figure 8). When 'knight' and 'chess' are finally selected simultaneously as the foci of SFEV, the images displayed are almost purely those pertaining to the concept of knight horses used in chess (Figure 9).

## FUTURE WORK

We have implemented SFEV to satisfy several key requirements important to information browsing, exploring and seeking. Multiple foci is supported to strengthen SFEV's semantic emphasis by allowing related words to be used simultaneously for a query. We are currently exploring multiple foci in terms of discovering semantic relationships between two or more concepts to support opportunistic search. Furthermore, we plan to perform several user studies in order to validate our hypotheses regarding SFEV's role for providing a flexible and wide range of modeling of semantic information.

## RELATED WORKS

### VIRI: Visual Information Retrieval Interfaces

Other visual interfaces for information seeking tasks fall into two categories: query-level interfaces and search results analysis interfaces.

Query level (also known as search without query terms): organize documents into topical categories, select words representative of documents, display relationship among the categories. This interface increases users seeking experience by:

- better knowledge of what is contained in the collection
- from category names and relationships, users can more effectively choose a sub-collection to drill-down.

Examples are Khonen maps [11], and path finder networks.

Concerning search result analysis, the assumption is that the document collection is very large, and even when a query has been launched, a user is still confronted with a large set of documents. A method, such as the Scatter/Gather [4], clusters the retrieved documents into semantically related groups and label them with descriptive summaries. The goal is to provide a quicker way for user to read titles in clusters to determine whether he wants to pursue the cluster further. It's a way of providing relevance feedback to the system in a more efficient way: users only have to examine a few representative titles in each collection to determine relevant categories used to refine the search..

Two important differences exist between VIRI and SFEV. The fact that images cannot be compared using text retrieval methods on their semantics, it is not meaningful to cluster image collections according to keywords appear in them. On the other hand, to "view" an image is to determine its relevancy to the current information need is usually simpler then "reading" an article. Thus when results are displayed in SFEV, the actual image is displayed, rather than its label.

A more important difference is that SFEV tries to visually render semantic similarity and relevancy of the underlying content, rather than doing the clustering for a user. (Clustering in SFEV formalism correspond to calculate the DOI for each query, and display the results from each query terms in the corresponding clusters using some kind of filtering.) While VIRI stresses on information organization, SFEV emphasizes the exploration of semantic concepts in the content and leave the discovery of relationships to the users.

VIRI rely on visual, usually spatial, representations of collections to help users in understanding the relationships betweeen search results. SFEV, on the other hand, are interactive techniques that modify an existing view to make semantic relationships apparent. This dynamic aspect allows the user to continuously modify the view to explore new relationships.

## CONCLUSION

We have presented the framework of semantic fisheye views, which are a set of dynamic display techniques aiming at the uncovering of semantic relationships of data by exploiting users visual power for patterns. We described two applications of SFEV in order to show how it works in action: one used for browsing of data sets in constrained display areas, and another for exploring word semantics in finding images.

## REFERENCES


1. Bartram, L., Ho, A., Dill, J. and Henigman, F. (1995) The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces. In: Proceedings of the ACM Symposium on User Interface Software and Technology, pp. 207-215.
2. Bates, M. J. (1989) The design of browsing and berrypicking techniques for the online search interface. Online Review, 13(5), 1989, pp. 407--424.
3. Cleveland, W.S. (1993) Visualizing data, pp. 360, Hobart Press, Murray Hill, N.J.
4. Douglas R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. (1992) *Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections*", ACM SIGIR `92.
5. Kenneth C. Cox, Stephen G. Eick, Graham J. Wills, and Ronald J. Brachman. (1997) Visual Data Mining: Recongnizing Telephone Calling Fraud. Data Mining and Knowledge Discovery, 1(2):225--231.
6. Furnas, G.W. (1986) Generalized Fisheye Views. In: Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems, pp. 16-23.
7. Herman, I., Melancon, G. and Marshall, M.S. (2000) Graph Visualization and Navigation in Information Visualization: a Survey. IEEE Transactions on Visualization and Computer Graphics 6, 24-43.
8. Janecek, P. and Pu, P. (2002) A Framework for Designing Fisheye Views to Support Multiple Semantic Contexts. in International Conference on Advanced Visual Interfaces (AVI '02).
9. Leung, Y.K. and Apperley, M.D. (1994) A Review and Taxonomy of Distortion-Oriented Presentation Techniques. ACM Transactions on Computer-Human Interaction 1, 126-160.
10. Lokuge, I. and Ishizaki, S. (1995) GeoSpace: An Interactive Visualization System for Exploring Complex Information Spaces. In: Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, pp. 409-414, ACM.
11. Lin, X., Soergel. D. and Marchionini, G. (1991) A Self-Organizing Semantic Map for Information Retrieval. In: *The ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'91)*.
12. Mukherjea, S. and Hara, Y. (1997) Focus+Context Views of World-Wide Web Nodes. In: Proceedings of the Eighth ACM Conference on Hypertext, pp. 187-196.
13. Rao, R. and Card, S.K. (1994) The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In: Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems, pp. 222.
14. Ruger, M., Preim, B. and Ritter, A. (1998) Zoom Navigation. In: Computational Visualization: Graphics, Abstraction, and Interactivity (Strothotte, T., ed.), pp. 161-174, Springer-Verlag, Berlin.
15. Sarkar, M. and Brown, M.H. (1994) Graphical Fisheye Views. Communications of the ACM 37, 73-84.
16. Strothotte, T., ed. (1998) Computational Visualization: Graphics, Abstraction, and Interactivity, Springer-Verlag.

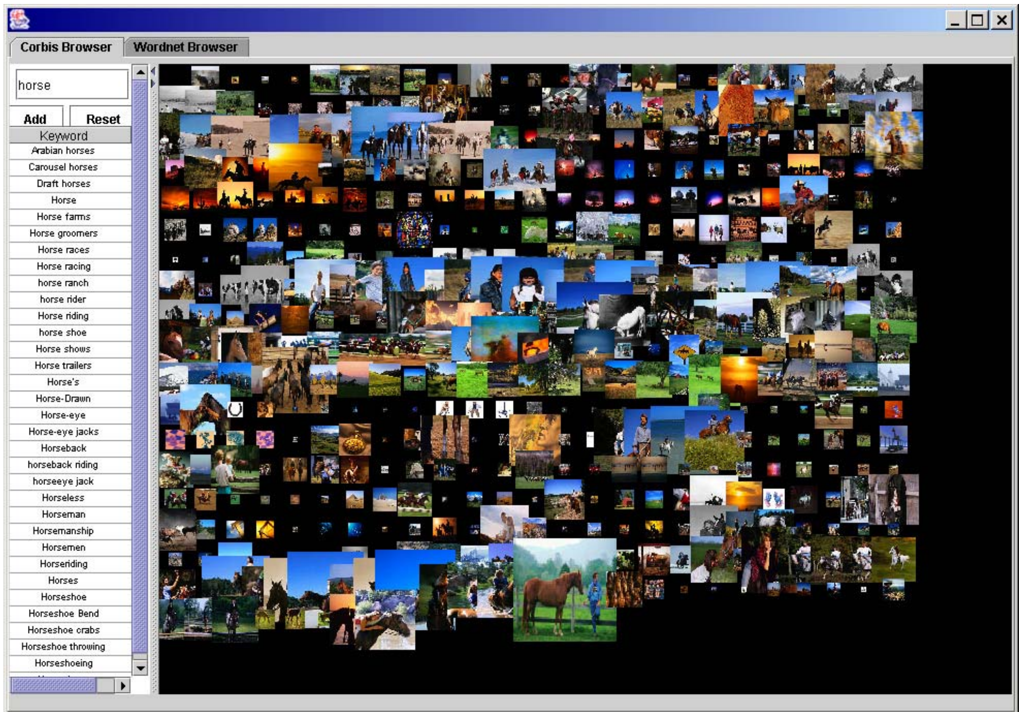



**Figure 6. Images retrieved by SFEV using keyword 'horse.' Only vector space model is used for calculating DOI.**

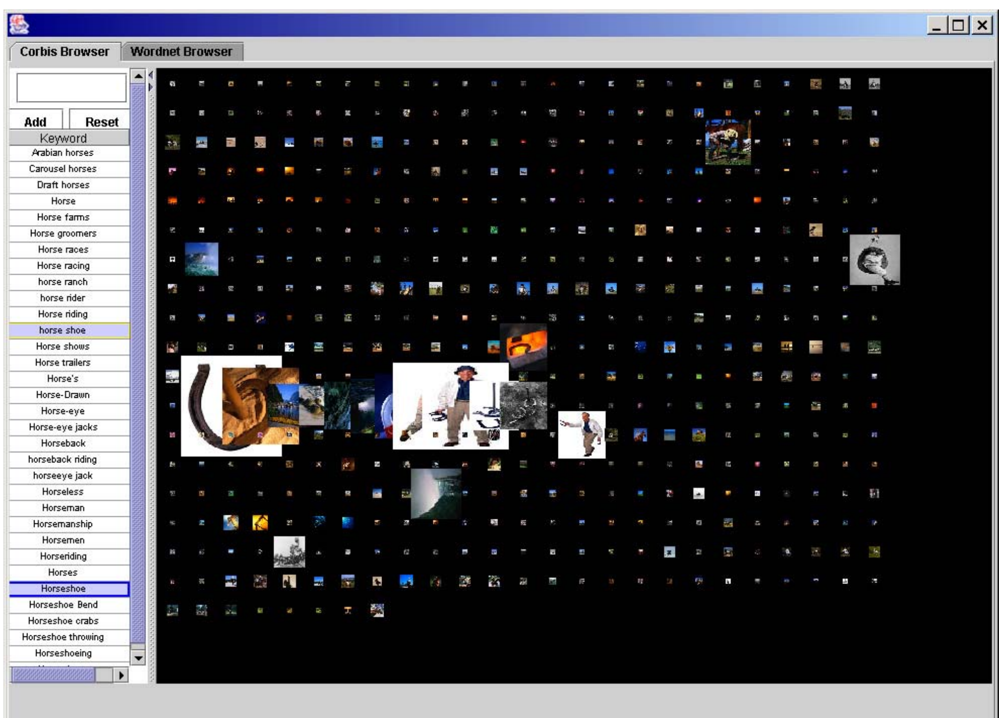



**Figure 7. 'Horse shoe' is selected as the focus in SFEV.**

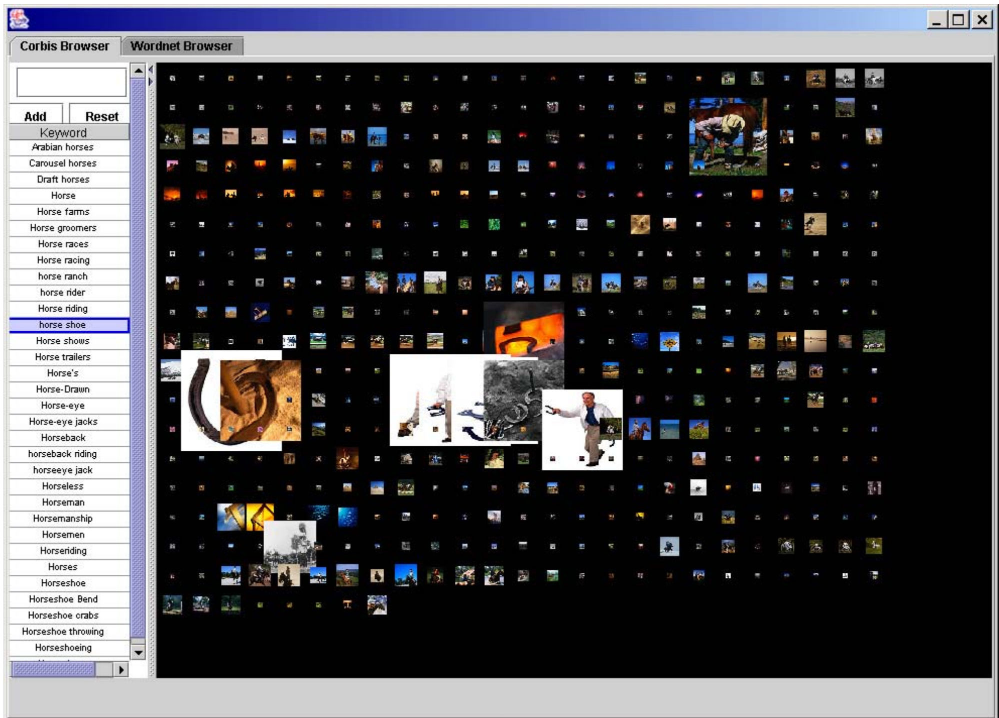



**Figure 8. Image retrieval results have been strengthened by adding 'horseshoe' to 'horse shoe.'**

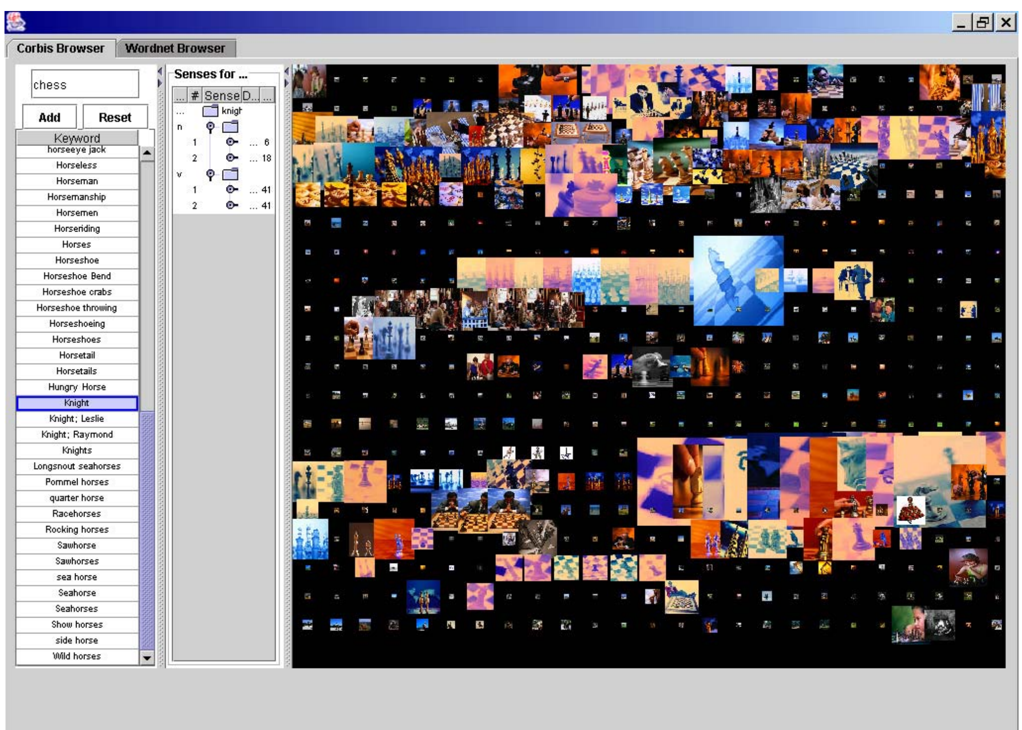



**Figure 9. Search results after WordNet expansion and selecting 'chess' with 'knight horse.'**