A Framework for Physically-Based Information Visualization

T. C. Sprenger, M. H. Gross, A. Eggenberger, M. Kaufmann*
Computer Science Department
Swiss Federal Institute of Technology
ETH Zürich, Switzerland
*Swiss Bank Corporation, IT-Camp Basel, Switzerland

Abstract The following paper describes a framework for the visualization and analysis of economic data. It can be employed in the context of risk analysis, stock prediction and other tasks being important in the context of banking. The system bases on a quantification of the similarity of related objects, which governs the parameters of a mass-spring system, organized as two concentric spheres. More specifically, we initialize all information units onto the surface of the inner sphere and attach them with springs to the outer sphere. Since the spring stiffnesses correspond to the computed similarity measures, the system converges into an energy minimum, which reveals multidimensional relations and adjacencies in terms of spatial neighborhoods. In order to simplify complex setups we propose an additional clustering algorithm for postprocessing. Furthermore, depending on the application scenario we support different topologic arrangements of related objects. In addition, we implemented various interaction techniques allowing semantic analysis of the underlying data sets. The versatility of our approach is illustrated by two examples, namely a comparison of agricultural productivity and an analysis of the relation between interest rates and other economic data.

1 Introduction

The visualization of complex, multidimensional, non-numeric information and of their relationships is an emerging subfield of increasing importance in scientific visualization. Nowadays, global computer networks and distributed data bases, such as the world wide web (WWW), provide platforms for new dimensions of retrieval systems for information units. As a consequence, the scientific visualization and computer graphics communities have been challenged to develop advanced tools for understanding, navigating and interactively analyzing the associated information spaces. However, as opposed to most of the classical data sets in scientific visualization, information spaces carry over entirely new qualities of problems. The most important ones can be summarized as follows:

- *Multidimensional relationships:* information units are generally related to many other units. The resulting topological organization corresponds to a multidimensional graph. Thus, adjacencies cannot be visualized straightforwardly and have to be mapped into subspaces. Here, we can carry over some interesting methods being already used in graph layout.
- *Measuring similarity:* In contrast to many scientific data sets, information space is an abstract entity and there is no specific reason to employ Euclidean metrics to

project similarities into a 3 dimensional subspace. Moreover, up to now there is no mathematical framework or paradigm, on how to map scores and similarities provided by retrieval systems onto a model in a three-dimensional world.

- *Clustering and hierarchies:* The huge amount of information forces the use of a multiresolution setup. Hence, appropriate methods for the clustering of objects and for interactive level of detail control are needed.

Due to the importance of information visualization for many applications, various interesting approaches can be found in literature and excellent surveys are available [20]. [5], [14] for instance, visualized text documents and clusters as galaxies and themescapes, whereas [4] proposed cone trees which specifically address hierarchical organization. Another promising method is [12] or [16], who essentially used self-organizing schemes and neural networks to arrange information objects of the WWW. In a more general understanding, multidimensional visualization problems have been stressed in [6] or [1]. Here, mathematical projection algorithms were introduced to map data into subspaces, while preserving their most important features. Interestingly, many current methods use physically based paradigms, such as [18] or [13], where information units are taken as nodes of some generalized mass spring system revealing the structure of relations upon relaxation. These types of multidimensional visualization methods have been studied extensively in graph theory, and efficient algorithms had been introduced for fast graph relaxation, such as [11] and [12].

The work reported in our paper was inspired by the research summarized above. However, unlike existing methods, our approach was mostly application driven, where the context was focussed on visualization problems arising in financial service providing. Therefore, we define a propriety mathematical framework for quantifying relationships in information space. In addition, we propose a visualization paradigm that considers all information units as initially located on the inner part of two concentric spheres and as attached with springs to the outer one. The strengths of relations of different objects are correlated to the stiffnesses of springs between them. After initialization, the system converges into an equilibrium stage by solving the underlying differential equations using popular strategies [2], [15]. Thus, the energy minimum represents spatial adjacencies of objects which are similar to each other in information space. Moreover, since the similarity matrix also encodes the topology of the underlying graph, standard algorithms from graph theory can be employed to discover the indirect links of objects and to find minimal paths between them. In order to simplify the geometric complexity of large scale data sets, we propose an additional clustering by computing an ellipsoidal hull around individual objects. The ellipsoid is parametrized by the principal components of the underlying cluster.

The organization of the paper is as follows: First we introduce the mathematical definition of similarity and explain our metric. Section 3 discusses the paradigm of two concentric spheres and describes the strategies for initial positioning and clustering of information units. In section 4 interactive analysis algorithms are elucidated. Finally, we illustrate the performance of our method by two examples: a comparison of agricultural productivity and an analysis of the relation between interest rates and other economic parameters.

2 Mathematical Foundations

This section introduces the mathematical foundations required to understand the approach. First, our metric for similarity in information space is elaborated, since it represents a major prerequisite for visualization. Furthermore, we briefly review the principles of the dynamics of mass spring and particle systems.

2.1 Measuring Similarity in Information Space

One of the very challenging problems of information visualization is the definition of a mathematical framework for the quantification of similarity of entities in information space. We decided to found our framework on vector spaces, but it also supports probabilistic, Boolean and Euclidean approaches for information retrieval. Specifically, we assume the metric as being computed in a preprocessing step, and providing the input parameters of a physically-based system.

Recalling some foundations of data base research we formalize the information retrieval process as follows: Let $\mathbf{A}=\{A_1, .., A_k\}$ be a set of attributes (key words) and let‘s consider n objects $\mathbf{O}=\{O_1, .., O_n\}$, where each object O_i is assigned to a score vector $\mathbf{s}_i$ of dimension k. The component s_{il} represents the relative importance of document O_i with respect to key word A_l and is assumed to be bounded by [0, 1].

We now define the so-called *similarity* c_{ij} of two objects O_i and O_j by the dot product of the associated scores normalized with respect to the dimension k:

$$c_{ij} = c_{ji} = \sqrt{\frac{\mathbf{s}_i \bullet \mathbf{s}_j}{k}} = \sqrt{\frac{1}{k}\sum_{l=1}^{k} s_{il}s_{jl}} \quad i, j = 1\ldots n \tag{1}$$

The resulting elements $0 \le c_{ij} \le 1$ form a symmetric, positive definite $n \times n$ *similarity matrix* $\mathbf{C}$:

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & \cdots & \cdots & c_{nn} \end{bmatrix} \tag{2}$$

where $c_{ij} = c_{ji}$, $1 \le i, j \le n$.

The *self-similarity* c_{ii} of an object O_i is provided by the normalized length of the score vector

$$c_{ii} = \frac{|s_i|}{\sqrt{k}}, i = 1\ldots n \tag{3}$$

Note, that $\mathbf{C}$ also reflects whether or not two objects are related. Hence, it can be considered as some sort of generalization of the so-called adjacency matrix, well-known from graph theory [8]. The similarity defined by (1) can be interpreted as the projection of one score vector onto another. Needless to say that the quality of the selected key words and retrieval algorithms is crucial for the quality of our similarity matrix.

The matrix from above quantifies the strengths of relations of information units in an abstract space. At this point in time, we have to analyze the associated topology by which adjacent objects and their relations are described. We end up in a *network type model* or graph, such as illustrated in Figure 1a, which is obviously one of the fundamental topologies for visualizing related information units.

However, if we target at visualizing how particular objects match a predefined string of key words, the upper model is apparently not well shaped. Moreover, a new *optimal* object O_{n+1} has to be added, whose scores are all set to: $s_{(n+1),l} = 1$, $l = 1,..,k$. The similarity matrix is constructed by computing the dot products between all objects and the one newly inserted O_{n+1}. That is, we relate all objects to a theoretically optimal score vector. The structure of the associated $(n+1)x(n+1)$ matrix reveals only elements on the diagonal and in the last column and row, respectively.

$$\mathbf{C}_{star} = \begin{bmatrix} c_{11} & 0 & \dots & c_{1(n+1)} \\ 0 & \dots & \dots & \dots \\ \dots & 0 & \dots & c_{k(n+1)} \\ c_{(n+1)1} & \dots & \dots & c_{(n+1)(n+1)} \end{bmatrix} \qquad (4)$$

$c_{ij} = 0, (i \neq j) \wedge 1 \leq i, j \leq n$ and $0 \leq c_{i(n+1)} = c_{(n+1)i} \leq 1, 1 \leq i \leq n$.

Based on the metric defined by (1) the resulting elements compute to

$$c_{(n+1)i} = c_{i(n+1)} = \sqrt{\frac{1}{k}\sum_{l=1}^{k} s_{il}} \qquad (5)$$

As a consequence, the underlying topology has changed and converts into a *star type* arrangement, such as depicted in Figure 1b. The central object is supposed to be the theoretical optimum and taken as a fix point. It is clear that a combination of both *network type* and *star type* topology is accomplished straightforwardly. Formally, we can also construct a third arrangement like the one of Figure 1c. It enables us to visualize both relations of objects between each other and with respect to a predefined score. A detailed elaboration of the associated visualization paradigm will be given in the next section.

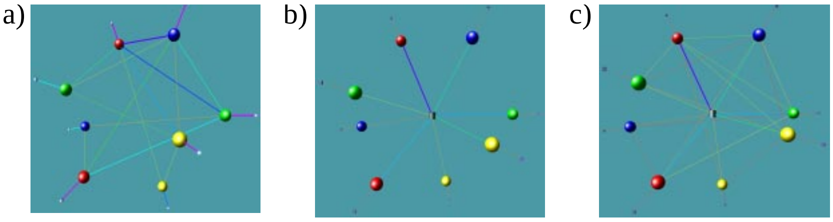

Fig 1. Different type of visualization models for multidimensional relations in information space: a) Network b) Star c) Combined Setting

2.2 Mass-Spring Systems

Mass-spring systems are linear finite elements and have been used widely and successfully in computer graphics [2], since the underlying physics is based on linear differential equations and is straightforward to implement. The principal equation that governs the attractive force $\mathbf{F}_{ij}$ between two attached masses m_i and m_j at spatial locations $\mathbf{r}_i$

and $\mathbf{r}_j$ and velocities $\mathbf{v}_i$ and $\mathbf{v}_j$ is given by

$$\mathbf{F}_{ij} = c_{ij}(\mathbf{r}_j - \mathbf{r}_i)\left(1 - \frac{l_0}{|\mathbf{r}_j - \mathbf{r}_i|}\right) + f(\mathbf{v}_j - \mathbf{v}_i) \qquad (6)$$

where c_{ij} stands for the spring stiffness, l_0 represents a bias length and f is the friction.

Applying Newton's law converts (6) into a second order linear differential equation. Note, however, that a straightforward discretization and integration results in algorithms of quadratic complexity. Therefore, it is recommended to invoke more sophisticated techniques, such as the ones proposed in [14] or [12].

3 Arranging Information on a Sphere

In the following section we describe the visualization paradigm which combines the metric introduced earlier with the physically based mass-spring approach. More precisely, we assign the computed similarities to the individual stiffnesses of springs linking information units to each other. Hence, we end up with an intermediate mapping of similarities onto parameters of a physically-based system. This approach has already proved to promise good results [18], [13] and [14]. In this context we address two major novelties of our framework: topology and arrangement of objects at initialization.

3.1 The Paradigms

Given a set of documents $O_1,..,O_n$ the visualization method employs two concentric spheres for the initial positioning of information units as illustrated in Figure 2a. The objects are placed on the surface of the inner sphere and are attached with springs to the *virtual* outer sphere and to each other. The major advantage of this arrangement is the degree of symmetry inherent to the geometry of a sphere of radius R. Consequently, the model handles our three fundamental topologies from Figure 1.

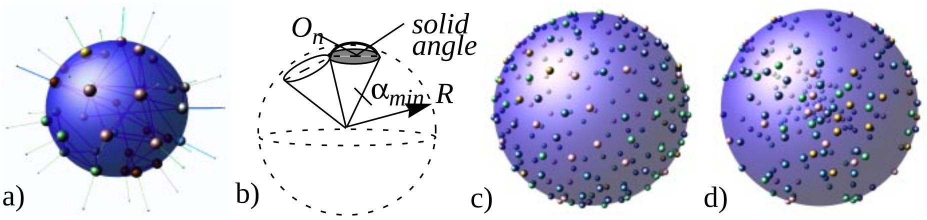

Fig 2. a) Initialization of information objects on a virtual sphere b) Poisson disc sampling for initial positioning c) Poisson disc distribution for: $f_d = 1.0$, d) $f_d = 0.3$

3.2 Initialization Procedures

Positioning of Objects

First, for a given set of *n* objects, the initial positions on the spherical surface have to be found. A straightforward positioning could be at random. However, due to the problems arising with small numbers of objects, we recommend a Poisson disc sampling procedure of the surface [9]. As depicted in Figure 2b each object is assigned to a solid angle element and the distance of two objects must not exceed a distance threshold d_{min}, computed by the following relation:

$$d_{min} = 2Rf_d\sin\left(\frac{\alpha_{min}}{2}\right) = 4R\sqrt[4]{1-\frac{f_d}{n}}\sqrt{\left(1-\sqrt{1-\frac{f_d}{n}}\right)} \qquad (7)$$

The influence of the factor f_d on the uniformity of the distribution of the initial positions is illustrated in Figure 2c and Figure 2d.

We have to be aware that our model corresponds to a multidimensional graph visualization problem [11] and is prone to all problems associated with subspace procedures [1]. Consequently, in order to prevent the system from convergence to a local minimum, tightly related objects should be positioned as close as possible to each other upon initialization. It is clear, that we won‘t succeed in the general case, but the traversal strategy reported below avoids most initial ill-conditioning.

We start with an initial weighting of all objects, where the weight w_i is defined by the sum of all adjacent similarities.

$$w_i = \sum_{j=1}^{u_i} c_{ij} \quad ; \quad u_i\text{: number of objects adjacent to } O_i \qquad (8)$$

This weighting emphasizes the importance of units with strong relations. From there, a list **O**={O_1,..,O_n} of all objects is built which is sorted with respect to the weights w_i. The assignment of information objects to the computed surface positions is figured out by a *breadth-first* strategy. The algorithm starts from the most important object and assigns first positions to all directly linked objects. These positions are ranked according to their distance from the initial one in 3D space. From there, the procedure traverses the list recursively until all objects are assigned. A pseudocode fragment for the method is given by:

```
O={$O_1$,..,$O_n$}        // initial object list //
P={$P_1$,..,$P_n$}        // initial positions //
T={}                      // list of sublists //
while (O not empty) do
   fetch object $O_i$ from O | $w_i$ = $w_{max}$
   assign random position $P_k$ to $O_i$: $O_i$ ->$P_k$
   remove $P_k$ from P
   generate list $L_i$ of all directly
   linked objects $O_l \in$ O sorted with respect to $c_{il}$
   keep $w_i$ with $L_i$
   add $L_i$ into T
   while (T not empty) do
        fetch sublist $L_m$ from T | $w_m$ = $w_{max}$
        remove all objects $\in$ $L_m$ from O
        while ($L_m$ not empty) do
             fetch next object $O_j$ from $L_m$
             assign free position $P_k$ to $O_j$ |
             dist ($P_k$,$P_m$) = min: $O_j$ -> $P_k$
             remove $P_k$ from P
             generate list $L_i$ of all directly
             linked objects $O_l \in$ O sorted with
             respect to $c_{lj}$
             keep $w_j$ with $L_j$
             add $L_j$ into T
```

od

od

od

It should be stated, that a *depth-first* traversal had been implemented as well but experimental results perform similarly.

Spring Stiffness Assignment

Another important aspect is the computation of the stiffnesses of the anchor springs attaching the objects to the outer sphere. It is clear that we have to take into account the individual connectivity of a single information unit. In order to understand the computation, one should be aware that the anchor spring influences the degree to which closely related objects converge to each other during energy minimization. Therefore, we propose the following method, where, for convenience, all relations hold for $t = 0$:

Let l_0 be the initial length of the anchor spring i and let $\boldsymbol{F}_a^i$ be its attractive force. The resulting force $\boldsymbol{F}_h^i = \sum_{j=1}^{u_i} \mathbf{F}_{ij}$ is computed from all springs linking object $O_j = m_j$ to others according to the adjacency matrix. This is shown in Figure 3a.

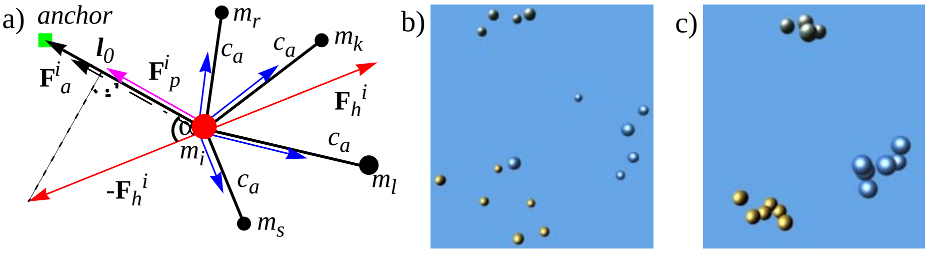

Fig 3. a) Spatial arrangement of the resulting force vector for initial spring stiffness computation b) Clustering of information units for different cluster-factors: $f_c = 0.9$ c) $f_c = 0.2$

The computation of the anchor spring stiffness is figured out by projecting $-\mathbf{F}^i{}_h$ onto the vector $\mathbf{l}_0$, which defines the direction of the anchor spring. Introducing an additional factor f_c, we set

$$|\mathbf{F}^i{}_a| = |\mathbf{l}_o|c_j = f_c|\mathbf{F}^i{}_p| = f_c\cos\alpha|\mathbf{F}^i| \tag{9}$$

and obtain the relation for the required anchor spring stiffness c_i:

$$c_i = \frac{f_c\mathbf{l}_0 \bullet (-\mathbf{F}_h{}^i)}{l_0{}^2} \tag{10}$$

In order to compute $\mathbf{F}^h{}_i$ all springs are supposed to have a uniform stiffness given by averaging:

$$c_a = \frac{1}{u_i}\sum_{j=1}^{u_i} c_j \tag{11}$$

Experiments have shown that this approach is superior to the immediate usage of the individual spring stiffnesses, since it balances differences of individual spring lengths. Note, that the same relations hold for the star type topology.

The factor f_c can be considered as a clustering factor, which takes immediate influence on the degree to which related objects converge to each other during simulation. Figure 3b and Figure 3c depict results of an experimental setup for different values of f_c.

4 Exploration of Information Space

Further important issues to be addressed concern the graph topology of related documents in information space. This formalism enables us to carry over some of the fundamental algorithms [8] to support interactive analysis and clustering, where in addition a PCA based clustering mechanism allows to simplify the structure of complex subregions. Minimal path procedures allow discovery and quantification of indirect links between objects.

4.1 Discovering Relations using Minimal Paths

For some applications it is useful to explore indirect relationships rather than immediate ones. This holds also for probabilistic settings where the links and adjacencies are marked by probabilities p<1. In order to provide tools for interactive exploration we take advantage of the graph structure of the visualization problem. Here, the indirect links which optimize a specific cost function are interesting for analysis.

Let $c_{ij} = p_{ij}$ be the probability that ranks the strength of two related objects O_i and O_j. Assuming statistical independence, we compute the probability p_{acc} of two objects O_S and O_E indirectly linked through a specific path by:

$$p_{acc} = \prod_{i,\,j \in path(S,E)} p_{ij} \tag{12}$$

Formally, paths from one node to another are derived from the *transitive closure* in graph theory, a Boolean matrix, whose elements are either 1, if two nodes are connected or 0 if not. Since these types of algorithms belong to the standard repertoire of computer science textbooks, we refer to those [10].

4.2 Clustering and Level-of-detail (LOD)

In order to simplify the geometry and topology of complex object arrangements it is necessary to provide an efficient level-of-detail strategy. Initial work for information visualization is reported in [7] who accomplished simple clustering by wrapping hyperspheres around groups of objects. The transparency of the hyperspheres was controlled as a function of the distance to the viewer. Unlike this approach we propose a K-means and PCA based clustering mechanism [19] which will be explained in the upcoming section.

The basic idea is to wrap ellipsoids around each cluster whose shape is controlled by the principal components of the respective cluster. The method is designed as a two pass procedure, where in a first step all objects in the scene are divided into a set **K** of disjoined subsets. The algorithm passes though all objects and assigns a new cluster if the distance of the current object to all existing clusters exceeds a threshold *delta*. This variable finally governs the granularity of the generated clusters. A pseudocode fragment is given below:

```
O={$O_1$,..,$O_n$}  // initial object list; $r_i$ = position of $O_i$ //
K={}         // set of clusters; $m_j$ = centroid of cluster $K_j$ //
while (O not empty) do
    fetch object $O_i$ from O | minima = delta
          iterate over all clusters $K_j \in$ K do
              if (|$m_j$ - $r_i$| < minima) then
                $K_{min}$ = $K_j$
                minima = |$m_j$ - $r_i$|
              fi
          od

          if (minima < delta) then
              add $O_i$ into $K_{min}$ | update $m_{min}$
          else
              create new cluster $K_{new}$
              add $O_i$ into $K_{new}$ | $m_{new}$ = $r_i$
              add $K_{new}$ into K
          fi
    od
end
```

The second pass comprises the parametrization of an affine map which transforms the initial 3D ellipsoidal shape appropriately into the scene. For cluster K_j this transform is defined by a translation vector $\mathbf{m}_j$, a scaling matrix $\mathbf{S}_j$ and a rotation matrix $\mathbf{R}_j$. Let n_j be the number of objects in cluster K_j we obtain the required translation vector immediately as the centroid of the cluster:

$$\mathbf{m}_j = \frac{1}{n_j} \cdot \sum_{i=1}^{n_j} \mathbf{r}_i \quad ; \quad \mathbf{r}_i\text{: object positions in cluster } K_j \tag{13}$$

In addition, the 3x3 Covariance-matrix of the cluster is given by

$$\mathbf{M}_j = \frac{1}{n_j} \cdot \sum_{i=1}^{n_j} (\mathbf{r}_i - \mathbf{m}_j) \cdot (\mathbf{r}_i - \mathbf{m}_j)^T \tag{14}$$

By solving the Eigenproblem $\mathbf{M}_j \cdot \mathbf{u}_k = \sigma_k \cdot \mathbf{u}_k$ we compute the 3 Eigenvalues σ_{j1}, σ_{j2}, σ_{j3} and the associated Eigenvectors $\mathbf{u}_{j1}$, $\mathbf{u}_{j2}$, $\mathbf{u}_{j3}$ which define the required transformation matrices, where

$$\mathbf{S}_j = \begin{bmatrix} \sqrt{\sigma_{j1}} & 0 & 0 \\ 0 & \sqrt{\sigma_{j2}} & 0 \\ 0 & 0 & \sqrt{\sigma_{j3}} \end{bmatrix} , \mathbf{R}_j = \begin{bmatrix} u_{j1x} & u_{j2x} & u_{j3x} \\ u_{j1y} & u_{j2y} & u_{j3y} \\ u_{j1z} & u_{j2z} & u_{j3z} \end{bmatrix} \tag{15}$$

Note that the 3x3 Eigenproblem can be solved analytically.

Thus the transform is figured out by the following set of equations. We start from the implicit equation of the unit sphere with surface vector $\mathbf{x}_j$:

$$\mathbf{x}_j = (x, y, z) \mid x^2 + y^2 + z^2 = 1 \tag{16}$$

and perform a subsequent affine mapping by

$$\mathbf{x}_j^e = \mathbf{m}_j + \mathbf{R}_j \cdot \mathbf{S}_j \cdot \mathbf{x}_j^0 \quad ; \quad \mathbf{x}_j^e\text{: surface vector of the ellipsoid} \tag{17}$$

Due to the statistical properties of the principal components it is not guaranteed that all objects of a cluster are enclosed by the ellipse. Thus we carry out additional postprocessing and grow the hull until all objects are enclosed.

5 Examples

The following section illustrates the performance and versatility of our approach by using two different examples: the visualization of agricultural productivity of selected countries and the visualization and analysis of interest rates correlating with other important economic parameters. Note, however, that the 2D pictures of this section do not reveal the full 3D arrangement computed by our method [21].

5.1 Agricultural Productivity

A classical example for visualization and analysis of multidimensional relations is given by taking some items from the world market. We employed figures from the yearly production of different agricultural products for selected countries, some of which are listed in Table 1.

Table 1: Productions of different countries in 1994 (subset)

Product Country	milk	meat	wheat	rice	potatoes
USA	69250	32091	63133	8547	19050
Canada	7750	2989	23180	0	0
China	5610	41424	102005	175608	36160
India	32112	4117	57802	117600	16318
Brazil	15774	8082	2127	11166	0
France	24900	6179	29944	0	4903
Germany	28200	5772	16100	0	12260
Switzerland	3300	448	0	0	800
....	....	....	....	....	....
Total	460058	193809	534301	531341	283306

Set of products **A** ={*butter, milk, meet, wheat, rice, soja, potatoes, sugar, bananas, cacao, coffee, tea*}.
Set of countries **O** ={*USA, Canada, China, India, Pakistan, Vietnam, Sri Lanca, Russia, Romania, Brazil, Argentina, Cuba, Columbia, Costa Rica, Peru, Ecuador, France, Netherlands, Italy, Austria, Germany, Switzerland, Australia, Japan*}.

A can be considered as the set of key-words and **O** as the set of information units. The figures of Table 1 were taken as a basis to construct a similarity matrix **C** according to the metric of section 2.

Figure D and Figure E (see Appendix) show the results of the *network type* model approach by contrasting the initial setup to the energy minimum of the particle model. All objects are textured according to their regional location, the USA and Switzerland are presented by their flags. Line color and thickness reflect the strength of the connection. We observe that most objects are arranged around the USA which forms a cluster center due to its high productivity. Conversely, Switzerland, as a small country moved apart. Furthermore, the results of the clustering algorithm are presented, where the clusters are visualized as transparent hulls whose opacity is controlled by the distance to the camera position.

Further interactive analysis is depicted in Figure F (see Appendix) where all objects with a direct link to the USA are presented. Those countries also lay within a particular radius. Changing this threshold reveals gradually those countries competing immediately to the USA on the world market. The strengths of the competition is approximated by the elements c_{ij} of **C,** also indicated in the figure.

5.2 Long Term Interest Rates

In the second example we contrast our method with a traditional way of analyzing multidimensional relationships of economic indicators. The goal is to evaluate the influence of the indicators presented in the diagrams of Figure 4a on the long term interest rates of individual countries. Each of these indicators was computed relative to the USA as a reference. The state of the art approach, as depicted in Figure 4b, consists of producing bar charts showing the correlation with individual indicators for different countries. These charts form a basis for further interpretation performed by the financial analyst. In order to map the problem onto our visualization paradigm we start from a special instance of the *network type* model. By imposing displacement constraints we first generate a subset of objects which keep their position during relaxation. For visualization, we map our indicators on these object types. Conversely, we drop the anchor stiffnesses of all other objects to zero, that is we cut off their anchors. These freely movable objects represent the countries and are connected via links to all rigid objects from above. The spring stiffness of a link conforms to the correlation of the associated indicator to the long term interest rates of this country. Note that the movable objects are not interconnected.

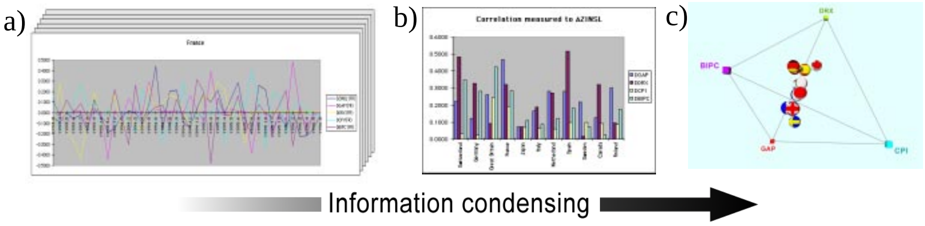

Fig 4. Condensing multivariante relationships: a) Stack of conventional diagrams b) Correlation tables c) Physically-based visualization paradigm

Figure G and Figure H (see Appendix) display two views on the relaxed model. The cubes at the vertices of the tetrahedral structure stand for the different indicators taken into account and the spheres representing the countries are textured with their flags. Although the definition of individual indicators is beyond the scope of this paper we observe that the interest rates of Canada correlate tightly with the index DRX, whereas Switzerland relates more closely to GAP and CPI. Conversely, Germany is located near the center of gravity of the plane spanned by DRX, BIPC and GAP and is hence equally influenced by those.

Conclusions and Future Work

We presented a new variant for physically-based information visualization and illustrated its versatility. The fundamental idea is to arrange all information units on the inner part of two concentric spheres and to attach them with springs to each other. Relaxation of the model figures out the structural relations in information space. Specifically, we are convinced that the physically-based approach fits nicely advanced I/O concepts with force and tactile feedback. Future research has to encompass a generalization of the physically-based approach including timeseries, limited lifetime of particles and advanced clustering algorithms using isosurfaces.

References

1. M. Gross, R. Koch: Visualization of Multidimensional Shape and Texture Features in Laser Range Data using Complex-Valued Gabor Wavelets, IEEE-Transactions on Visualization and Computer Graphics, Vol. 1, No. 1, pp.44-59, 1995
2. A. Witkin, D. Baraff, M. Kass: Physically-Based Modeling, SIGGRAPH tutorial course Notes No. 34, 1995
3. M. Harada, A. Witkin, D. Baraff: Interactive Physically-Based Manipulation of Discrete/Continuous Models, Proceedings Siggraph 95, pp. 199-208, 1995
4. J. Carriere, R. Katzman: Research Report - Interacting with Huge Hierarchies: Beyond Cone Trees, Proceedings of the IEEE Info. Vis. 95, pp. 74-81, 1995
5. J. Wise, et al.: Visualizing the Non-Visual: Spatial analysis and Interaction with Information from text Documents, Proceedings of the IEEE Info. Vis. 95, pp. 51-58, 1995
6. F. Young, P. Rheingans: Visualizing Strucutre in High-Dimensional Multivariate Data, IBM Journal of Research and Development, Vol. 35, No. 1/2, pp. 97-107, 1991
7. R. Hendley, et al.: Case Study - Narcissus: Visualizing Information, Proceedings of the IEEE Information Visualisation 95, pp. 90-96, 1995
8. J. Nievergelt, K. Hinrichs: Algorithms and Data Structures with Applications to Graphics and geometry. Englewood Cliffs: Prentice Hall, 1993
9. A. S. Glassner: Principles of Digital Image Synthesis, Morgan Kaufmann Publishers,San Francisco, 1995
10. T. H. Cormen, C. E. Leiserson, ans R. L. Rivest: Introduction to Algorithms, MIT Press, Cambridge, Massachusetts, 1994R. Koch, M. Gross, et al.: Simulating Facial Surgery Using Finite Element Models, Proceedings of SIGGRAPH 96, pp. 421-428, 1996

11. A. Frick, A. Ludwig and H. Mehldau: A fast adaptive layout algorithm for undirected graphs, Proceedings of Graph Drawing 94, LNCS 894, Springer Verlag 1995

12. I. Bruss, A. Frick: Fast Interactive 3-D Graph Visualization, Proceedings of Graph Drawing 95, Springer Verlag, LNCS 1027, p. 99-110

13. R. J. Hendley, N. S. Drew: Visualisation of complex systems, http://ww.cs.bham.ac.uk

14. M. Chalmers: A Linear Time Layout Algorithm for Visualizing High-Dimensional Data, Proceedings of the IEEE Information Visualization 96, pp. 127-132, 1996

15. A. Witkin: Particle System Dynamics, SIGGRAPH 96 Course Notes 34, pp C1-C12, 1996

16. A. Wood, et al.: HyperSpace: Web Browsing with Visualisation. Third International World-Wide Web Conference, Poster Proceedings, Darmstadt, Germany, pp. 21-25, 1995

17. T. R. Henry, S. E. Hudson: Interactive Graph Layout, Proceedings of the ACM SIGGRAPH Symposium, Proceedings ACM Siggraph Symposium on UI Software, 1991

18. C. L. Bentley: Animating Multidimensional Scaling to Visualize N-Dimensional Data Sets, Proceedings of the IEEE Information Visualisation 96, pp. 72-73, 1996

19. K. Fukunaga: Introduction to Statistical Pattern Recognition. 2nd Edition, New York: Academic Press, 1990

20. S. Card, S. G. Eick, N. Gershon: Information Visualization, SIGGRAPH 96 Course Notes 8, 1996

21. CGRG Homepage, ETH Zürich, http://www.inf.ethz.ch/department/IS/cg/html/research/infovis.htm

22. M. H. Gross, T. C. Sprenger, J. Finger: Visualizing Information on a Sphere, Technical Report, ETH Zürich, 1997

Appendix

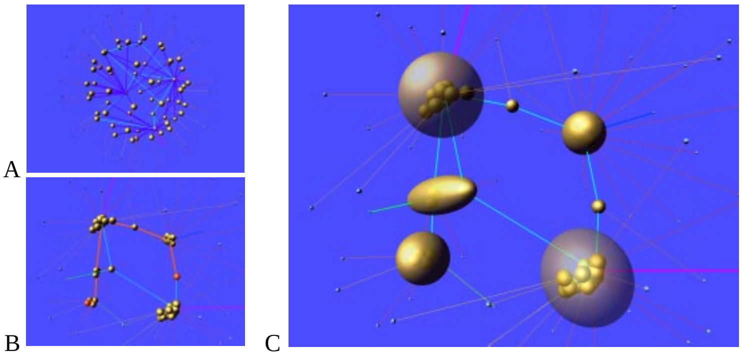

Fig E: Illustration of the clustering method: Initialization

Fig F: Model after relaxation and highlighted minimal path between two objects

Fig G: Disjoined clusters as transparent ellipsoids

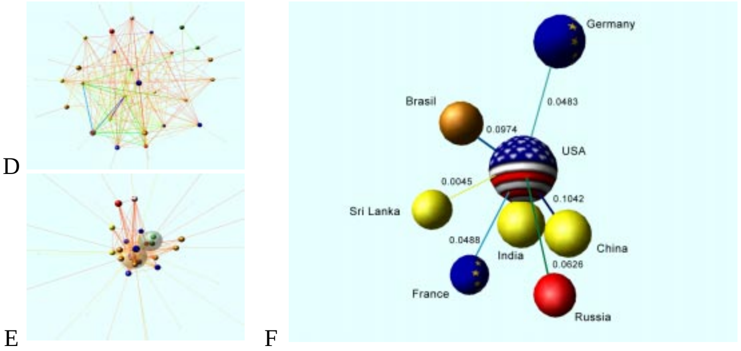

Fig D: Visualization of agricultural productivity: Initialization

Fig E: Clustered energy minimum (Parameters: $R = 600$, $l_0 = 100$, $f_c = 0.02$)

Fig F: Discovering competitors of the United States on the world market

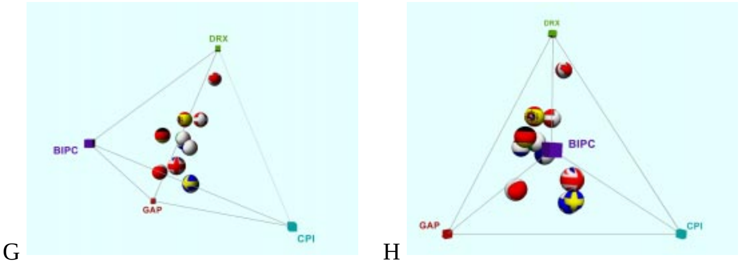

Fig G: Influence of economic indicators onto the long term interest rates of different countries

Fig H: Alternative view