EUROGRAPHICS '98 / N. Ferreira and M. Göbel
(Guest Editors)

Volume 17, (1999), Number 3

An Enhanced Spring Model for Information Visualization

Holger Theisel and Matthias Kreuseler

University of Rostock, Computer Science Department, PostBox 999, 18051 Rostock, Germany
{theisel/mkreusel}@informatik.uni-rostock.de

Abstract

In this paper we present a new technique for visualizing multidimensional information. We describe objects of a higher dimensional information space as small closed free-form-surfaces in the visualization. The location, size and shape of these surfaces describe the original objects in information space uniquely. The underlying enhanced spring model is introduced. The technique is applied to two test data sets.

Keywords: *information visualization, multidimensional information space, spring model, free-form-surface.*

1. Introduction

In recent years the visualization of large multidimensional information has become one of the "hot topics" in scientific visualization. New techniques to handle this problems have been developed, such as cone trees ([9]) and fisheye views ([10]). Several techniques try to map correlations of the objects in higher dimensional information space to spatial correlations in the visualization space (2D or 3D): objects with similar properties are visualized spatially close to each other. The underlying physical model is often a spring model.

In [7] the objects are documents, and the dimensions are key words. Using a spring model, the objects are placed between the dimension points in 2D. [1] extends this to 3D and introduces more facilities for browsing and interacting with the document space. A similar spring model is applied in [6] to visualize DNA sequences. In [3], an extended mass-spring model is introduced. The objects are initially placed on the inner of two concentric spheres; the dimension points are placed on the outer sphere. Between object points and dimension points springs are attached. [4] places the objects on spheres of interest in order to preserve the spatial relations in information space. [5] introduces a self-organizing system where certain repelling and attracting forces are between the objects. Then a state of balance is computed. [8] exploits a similar approach where spring forces are used to place graphical representations of information objects and objects' attributes in 3D according to certain object similarities. [2] uses a combined spring and gravitational model for visualizing large graphs.

The data we consider in this paper consist of objects in the n-dimensional information space. Visualizing these data means finding an appropriate map from information space to the (two or three dimensional) visualization space.

Every dimension of the information space is related to a point $\mathbf{d}_i \in \mathbb{R}^2(\mathbb{R}^3), (i = 1,...,n)$. An object O in the information space is a n-tuple $(c_1,...,c_n) \in \mathbb{R}^n$ $(c_1,...,c_n > 0)$. It can be considered as the coordinates of the object in information space. (As an example, consider the objects as text documents and the dimensions as certain key words. Then the coordinates $(c_1,...,c_n)$ of an object are the frequencies of appearance of the key words in the document.)

Using the classical spring model ([1], [6], [7]), an object $O = (c_1,...,c_n)$ is related to a point $\mathbf{p}$ in visualization space. We consider n springs - from each dimension point $\mathbf{d}_i$ to $\mathbf{p}$. The stiffness of the springs are set to the values $c_1,...,c_n$. Then the location $\mathbf{p}$ is searched where the spring model is in balance. For fixed $\mathbf{d}_i$ we can compute this location explicitely:

$$\mathbf{p} = \frac{\sum_{i=1}^{n} c_i \cdot \mathbf{d}_i}{\sum_{i=1}^{n} c_i}. \qquad (1)$$

Figure 1 gives an illustration for $n = 4$.

Using the classical spring model for placing objects in the visualization space has the following advantages, so that the technique is useful and often applied:
– The location of $\mathbf{p}$ gives spatially intuitive information about the object O: the more O is related to the i-th dimension (i.e., the bigger c_i is), the closer moves $\mathbf{p}$ towards $\mathbf{d}_i$.
– Since every object is visualized only as a point, a large number of objects can be visualized. The result is a point cloud.

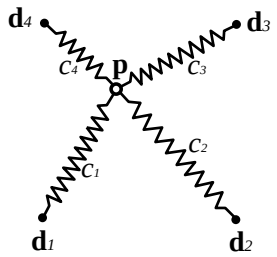

Figure 1: *Visualizing the object $O = (c_1, ..., c_4)$ of a four dimensional visualization space. The dimension points $\mathbf{d}_1, ..., \mathbf{d}_4$ are fixed, $\mathbf{p}$ moves to the position of balance of the spring system.*

– Objects with similar properties are spatially close in the visualization. We can search for clusters in the visualized point cloud.

There are limitations to the classical spring model as well:
a): ambiguity: objects with different coordinates $c_1, ..., c_n$ may collapse to one point in the visualization. This means similar objects are spatially close in the visualization, but we do not know whether or not objects spatially close in the visualization are similar.
b): insensitivity to coordinate scalings: the objects $(c_1, ..., c_n)$ and $(k \cdot c_1, ..., k \cdot c_n)$ with $k > 0$ cannot be distinguished in the visualization because they are mapped to the same point.

Figure 2 illustrades these limitations.

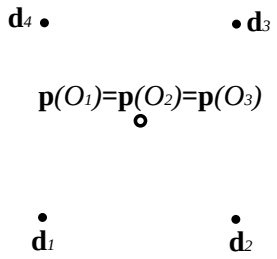

Figure 2: *Limitations of the classical spring model: the objects $O_1 = (1,2,1,2)$ and $O_2 = (2,1,2,1)$ collapse to one point (ambiguity). So do the objects O_1 and $O_3 = (2,4,2,4)$ (insensitivity to coordinate scalings).*

To decrease the impact of the drawbacks a) and b), research has been done on finding appropriate locations for the dimension points $\mathbf{d}_i$. In [1] and [7] the points $\mathbf{d}_i$ can be moved interactively. [3] solves a mass-spring system numerically for finding suitable locations of the $\mathbf{d}_i$. All these approaches can limit the drawbacks a) and b) but do not solve them.

The approach to be presented here is able to solve the drawbacks a) and b). The main idea is to assign an object not only with a point but with a small closed free-form curve (surface). The location of the curve (surface) gives spatial information similar to the location of the object point in the classical spring model. The additional size and shape information of the curve (surface) gives more intuitive knowledge about the object and solves the drawbacks a) and b).

The reason for using small free-form curves (surfaces) lies in the fact that humans react rather sensitively to small changes in the shape of curves (surfaces). Thus curves (surfaces) provide a way of presenting a lot of information in a small area - a precondition for visualizing a higher number of objects.

Section 2 introduces the underlying enhanced spring model. Section 3 describes how to get the curves (surfaces) out of the model. Section 4 discusses the visualization aspects of the approach. Section 5 applies the approach to two example data sets and discusses the results.

2. The enhanced spring model

As in the classical spring model, we place a fixed point $\mathbf{d}_i \in \mathbb{R}^2(\mathbb{R}^3)$ for every dimension of the information space. For a given object $O = (c_1, ..., c_n)$, we consider a point $\mathbf{p}$ in $\mathbb{R}^2(\mathbb{R}^3)$. We attach n springs with the constant stiffness $c > 0$ to $\mathbf{p}$. The other ends of the springs are named $\mathbf{p}_1, ..., \mathbf{p}_n$. Now we consider n more springs - from $\mathbf{p}_i$ to $\mathbf{d}_i$ with the stiffness c_i for $i = 1, ..., n$. The points $\mathbf{p}, \mathbf{p}_1, ..., \mathbf{p}_n$ are free movable; the points $\mathbf{d}_1, ..., \mathbf{d}_n$ are fixed. Then we search for the state of balance of this spring system. Figure 3 gives an illustration.

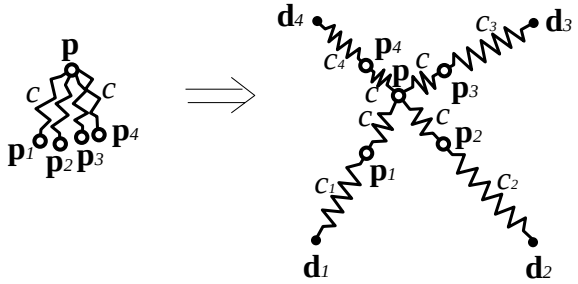

Figure 3: *The spring model for an object $O = (c_1, ..., c_4)$ of four dimensional information space. The object is described by the points $\mathbf{p}, \mathbf{p}_1, ..., \mathbf{p}_4$. The constant c is considered to be bigger than $c_1, ..., c_4$.*

Applying this spring model, an object $O = (c_1, ..., c_n)$ is described by the $n+1$ points $\mathbf{p}, \mathbf{p}_1, ..., \mathbf{p}_n$. The constant c is

considered to be at least one scale bigger than $c_1,...,c_n$. Its influence will be discussed later on.

For given dimension points $\mathbf{d}_1,...,\mathbf{d}_n$ and constant c, the location of the points $\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$ describing the object $O=(c_1,...,c_n)$ can be computed explicitly:

$$\mathbf{p}=\frac{\sum_{i=1}^n w_i\cdot\mathbf{d}_i}{\sum_{i=1}^n w_i} \tag{2}$$

with

$$w_i=\frac{c_i}{c+c_i}\quad\text{for } i=1,...,n. \tag{3}$$

Then we obtain for $\mathbf{p}_1,...,\mathbf{p}_n$:

$$\mathbf{p}_i=\frac{c\cdot\mathbf{p}+c_i\cdot\mathbf{d}_i}{c+c_i}\quad\text{for } i=1,...,n. \tag{4}$$

Obviously, the locations of $\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$ depend on the value of the constant c. We study the special cases of converging c to infinity and zero:

$$\lim_{c\to\infty}\mathbf{p}=\lim_{c\to\infty}\mathbf{p}_1=...=\lim_{c\to\infty}\mathbf{p}_n=\frac{\sum_{i=1}^n c_i\cdot\mathbf{d}_i}{\sum_{i=1}^n c_i}. \tag{5}$$

This means that for $c\to\infty$ the points $\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$ collapse to the point which describes the object in the classical spring model. Therefore, the classical spring model is contained as a special case in the enhanced spring model. For $c\to 0$ we obtain

$$\begin{aligned}\lim_{c\to 0}\mathbf{p} &= \frac{\sum_{i=1}^n\mathbf{d}_i}{n}\\ \lim_{c\to 0}\mathbf{p}_i &= \mathbf{d}_i\quad\text{for } i=1,...,n.\end{aligned} \tag{6}$$

For $c\to 0$, the locations of $\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$ do not depend on the object anymore. Therefore, c should be chosen rather large.

The points $\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$ describe an object $O=(c_1,...,c_n)$ uniquely. This means that from given $\mathbf{d}_1,...,\mathbf{d}_n,\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$ and c we can compute all c_i $(i=1,...,n)$. To do so we only have to solve the linear system of the equations (2)-(4) and the unknowns $c_1,...,c_n$. This gives an unique solution if no two of the dimension points coincide and the points $\mathbf{p},\mathbf{p}_i,\mathbf{d}_i$ are collinear for $i=1,...,n$.

3. Obtaining closed curves (surfaces)

Even if the points $\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$ describe an object uniquely - for the visualization of an object $n+1$ separate points are not suitable. We therefore define a closed curve (surface) which gives an intuitive imagination of the location of $\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$. In the following we only describe the surface case for a 3D visualization. The case of a closed curve in a 2D visualization can be obtained as a special case of the surface.

Defining a surface out of the control points $\mathbf{p},\mathbf{p}_1,...,\mathbf{p}_n$ using usual approaches like Bezier- or B-Spline surfaces fails because these approaches depend on the order of the control point sequence. For our problem, the points $\mathbf{p}_1,...,\mathbf{p}_n$ should not be in a particular order.

We define a closed surface as a deformation of a small sphere around $\mathbf{p}$:

$$\mathbf{x}(\lambda,\phi)=\mathbf{p}+f(\lambda,\phi)\cdot\begin{pmatrix}\cos\phi\cdot\cos\lambda\\ \cos\phi\cdot\sin\lambda\\ \sin\phi\end{pmatrix} \tag{7}$$
$$(0\le\lambda<2\pi;-\frac{\pi}{2}\le\phi\le\frac{\pi}{2}.$$

In (7), $\mathbf{p}$ is a 3D point and $f(\lambda,\phi)$ is a certain bivariate scalar function. For $f(\lambda,\phi)=\text{const}$, (7) describes a sphere around $\mathbf{p}$ in spherical coordinates. This sphere will be deformed by the function $f(\lambda,\phi)$. The special case of a 2D visualization is contained in (7) by setting $\phi=0$.

To define $f(\lambda,\phi)$ we introduce the auxiliary functions

$$f_i(\lambda,\phi)=\begin{cases}\dfrac{(\mathbf{p}_i-\mathbf{p})\cdot\begin{pmatrix}\cos\phi\cdot\cos\lambda\\ \cos\phi\cdot\sin\lambda\\ \sin\phi\end{pmatrix}}{\|\mathbf{p}_i-\mathbf{p}\|} & \text{if } (\mathbf{p}_i-\mathbf{p})\cdot\begin{pmatrix}\cos\phi\cdot\cos\lambda\\ \cos\phi\cdot\sin\lambda\\ \sin\phi\end{pmatrix}>0\\ 0 & \text{else}\end{cases} \tag{8}$$

for $i=1,...,n$. Figure 4 illustrates (8).

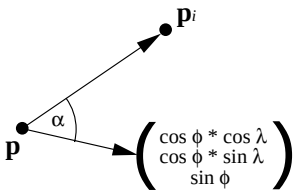

Figure 4: *The auxiliary function* $f_i(\lambda,\phi)$*:* $f_i(\lambda,\phi)=\cos\alpha$ *if* $\cos\alpha>0$, *else* 0.

Now we can define the deforming function $f(\lambda,\phi)$ by

$$f(\lambda,\phi)=f_0+\sum_{i=1}^n\|\mathbf{p}_i-\mathbf{p}\|\cdot(f_i(\lambda,\phi))^{sh}. \tag{9}$$

Generally, the surface $\mathbf{x}(\lambda,\phi)$ shows deformations in all directions $\mathbf{p}_i-\mathbf{p}$. The bigger $\|\mathbf{p}_i-\mathbf{p}\|$ is, the stronger the deformations are. The constant *sh* determines how strong the deformation is performed in the directions close to $\mathbf{p}_i-\mathbf{p}$. It has similarities to the shiny exponent in Phong's illumination model: the bigger *sh* is, the sharper is the deformation (highlight in Phong's model). The impact of *sh* is illustrated in figure 7. The constant f_0 is for preserving continuity and parametrization regularity of the surface. In fact, for $f_0>0$ and *sh* a natural number, the surface defined by (7)-(9) is C^{sh-1}-continuous. Thus we should choose $sh>2$ in order to obtain a C^1-continuous closed surface.

Another property is the fact that the points $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$ are inside the closed surface defined by (7)-(9).

We consider a first example of a 2D visualization.

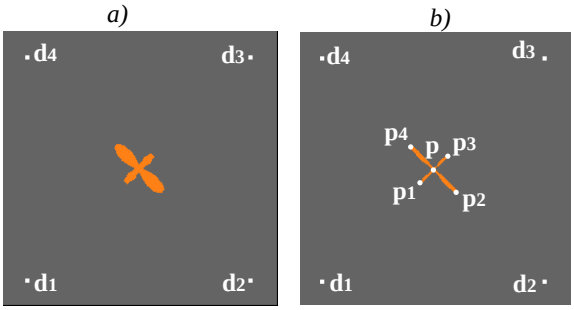

Figure 5: *a): Visualizing the object $O = (1,2,1,2)$ with $sh = 10$. b): The same object visualized with a high sh ($sh = 100$). We obtain $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$ directly from the curve.*

Figure 5a shows the visualization of the object $O = (1,2,1,2)$. The closed curve was visualized as the boundary of a filled area. The curve gives us three kinds of information: location, size and shape. The location of the whole curve gives initial information about the object. The size gives information about how big the components of an object are generally. An object with generally high components c_i $(i = 1, \ldots, n)$ produces a bigger closed curve. The deformations of the curve show which dimensions the object is more related to. In our example the object is more related to $\mathbf{d}_2$ and $\mathbf{d}_4$ than to $\mathbf{d}_1$ and $\mathbf{d}_3$.

A surface defined by (7)-(9) describes an object uniquely. This means that from the location, size and shape of such a surface we can uniquely infer all components $c_1, \ldots, c_n$ of the original object. For showing this, suppose sh is high. Then the surface converges to the line segments $(\mathbf{p}, \mathbf{p}_1), \ldots, (\mathbf{p}, \mathbf{p}_n)$ (figure 5b illustrates). This means that from the surface we obtain the location of $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$ directly. As shown in section 3, these points give the object uniquely. Figure 6 shows the visualization of the objects from figure 2.

4. Visualizing the closed curves (surfaces)

For visualizing a curve (surface) related to an object O, we have to determine the following parameters: c, sh, f_0 and the locations of $\mathbf{d}_1, \ldots, \mathbf{d}_n$. All these parameters are chosen globally, i.e. they are the same for every visualized object.

The location of the $\mathbf{d}_i$ influences the intuitivity of the visualization but not the uniqueness of the object description. We placed the $\mathbf{d}_i$ equally distributed on a unit sphere.

The parameter c has influence on the size of the surface. A higher c leads to a smaller surface. For $c \to \infty$, the surfaces collapse to points.

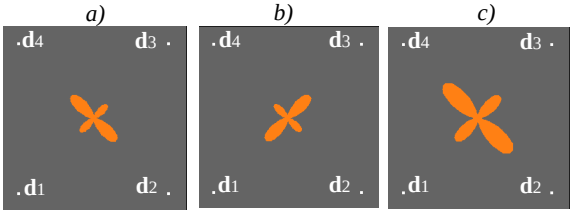

Figure 6: *Visualizing the objects a): (1,2,1,2); b): (2,1,2,1); c): (2,4,2,4). In contrary to the classical spring model shown in figure 2, these objects can be well distinguished. For all objects in this figure we have chosen $sh = 10$.*

The parameter sh influences how sharp the deformations are around the directions $\mathbf{p}_i - \mathbf{p}$. The extreme case $sh \to \infty$ is shown in figure 5b.

The parameter $f_0 > 0$ is necessary for continuity preserving of the surface. It should be chosen rather small.

The influence of the parameters c and sh is shown in figure 7 for the 2D case.

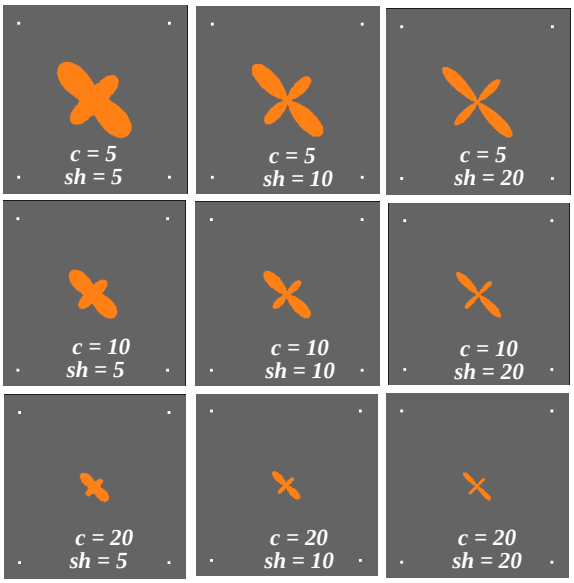

Figure 7: *Visualizing the object (1,2,1,2) with different parameters sh and c.*

For visualizing a higher number of objects we used the following visualization scenario:
a): Get a global impression by visualizing all objects with a high parameter c. The objects are almost points, the visualization is the same as from the classical spring model. We try to detect clusters in the visualization. Objects which are

not spatially close (for instance objects in different clusters) are not similar to each other. For objects in a cluster we have to go on analyzing:
b): Zoom into a cluster and increase c. The points will appear as closed surfaces. If two surfaces have different shape, the objects are not similar to each other. If two surfaces have a similar shape but different sizes, one object is the scaled image of the other one. If the surfaces are similar in shape and size, the objects are similar to each other.

The only problem here is how to distinguish between objects that are in the same location. In this case the surfaces may intersect and hide each other in part or completely. A solution to this problem is to offer the option of visualizing the surfaces transparently or in wire frame representation. In addition, the surfaces are visualized in different colors. In this way we can distinguish between surfaces in the same location as long as they are not exactly identical. For the test data sets described in the next section, this problem did not occur.

5. Applications and results

We applied the visualization technique to two test data sets which are publicly available on WWW. Both data sets have been initially explored in [11].

The first data set † contains information about 38 automobiles including miles per gallon, weight, drive ratio, horsepower, displacement and number of cylinders. This means we have 38 objects in a 6-dimensional information space. The dimension points $\mathbf{d}_1, \ldots, \mathbf{d}_6$ were placed in an equidistant way on the unit sphere. Figure 8 gives an overview over the data set in a 3D visualization. We used the parameters $c = 15, sh = 10, f_0 = 0.2$. Each of the closed surfaces was approximated by 1944 triangles. The visualization on a Silicon Graphics Indigo 2 Workstation under IRIS Explorer was computed in approx. 13 seconds. The navigation through the scene is possible in an interactive time.

In figure 8 we recognize a cluster of surfaces in the upper middle part of the picture. For the cars lying in this cluster we can make the assumption that they have similar properties; but we have to check it. For doing this, we zoom into this area and visualize with $c = 30$. Figure 9 shows the result. As we can see here, the surfaces for Buick Century Special, Dodge Aspen and AMC Concord D/L are similar in location, size and shape. These cars have, therefore, similar properties. The other 8 surfaces in figure 9 are also similar to each other in location, size and shape. Thus the properties of these 8 cars are similar to each other as well.

The second data set ‡ measures various quality of living parameters of US cities. It is a 9 dimensional data set with 329 objects. The values for all dimensions are normalized to $[0,1]$ in such a way that the higher the number the better the city.

The 9 dimensional points were placed equally distributed on a sphere. Figure 10 gives an overview over the data set. Most of the objects are in one big cluster, there are only a few outliers. Figure 11 shows the cluster in more detail. Figure 12 shows the magnification of the five objects lower right of figure 11. As we can see here, the surfaces for Springfield, St. Louis, Baltimore and Hartford have similar shape and size. These cities therefore have similar living conditions. The shape of the surface related to Miami-Hiale differs from the 4 other cities. The living conditions in Miami-Hiale are therefore different in comparison to the four other cities, even if the five closed surfaces are spatially close to each other.

† available at http://lib.stat.cmu.edu/DASL/Stories/ClusteringCars.html

‡ available at http://lib.stat.cmu.edu/datasets/places.dat

6. Acknowledgements

The authors would like to thank Prof. Heidrun Schumann from Rostock University for her constant encouragement and support over the research presented in this paper. M. Kreuseler thanks Siemens AG Munich for their support.

References

1. S. Benford, D. Snowdon, C. Greenhalgh, R. Ingram, I. Knox and C. Brown. VR-VIBE: A Virtual Environment or Co-operative Information Retrieval. Proceedings Eurographics '95, pp. C349 - C360, Blackwell Publishers, 1995.
2. R.F. Cohen an M.L. Huang. Online Information Visualization of Very Large Data. Proceedings Visualization'97 Conference, Phoenix, IEEE, 1997.
3. M.H. Gross, T.C. Sprenger and J. Finger. Visualizing Information on a Sphere. Proceedings of Information Visualization '97 Symposium, Phoenix, pp. 11-16, IEEE, 1997.
4. M. Hemmje, C. Kunkel and A. Willet. Lyber World - A User Interface Supporting Fulltext Retrievel. Proc. of SIGIR '94, Dublin, 1994.
5. R.J. Hendley, N.S. Drew, A.M. Wood and R. Beale. Narcissus: Visualizing Information. Proceedings of Information Visualization '95 Symposium Atlanta, pp. 90-96, IEEE, 1995.
6. P. Hoffman, G. Grinstein, K. Marx, I. Grosse and E. Stanley. DNA Visual And Analytic Data Mining. Proceedings of Visualization '97 Conference, Phoenix, pp. 437-441, IEEE, 1997.
7. K.A. Olsen, R.R. Korfhage, K.M.Sochats, M.B.Spring and J.G. Williams. Visualization of a Document Collection: The VIBE System. Information Processing and Management, Vol 29, No 1 pp. 69-81, Pergamon Press Ltd. 1993.

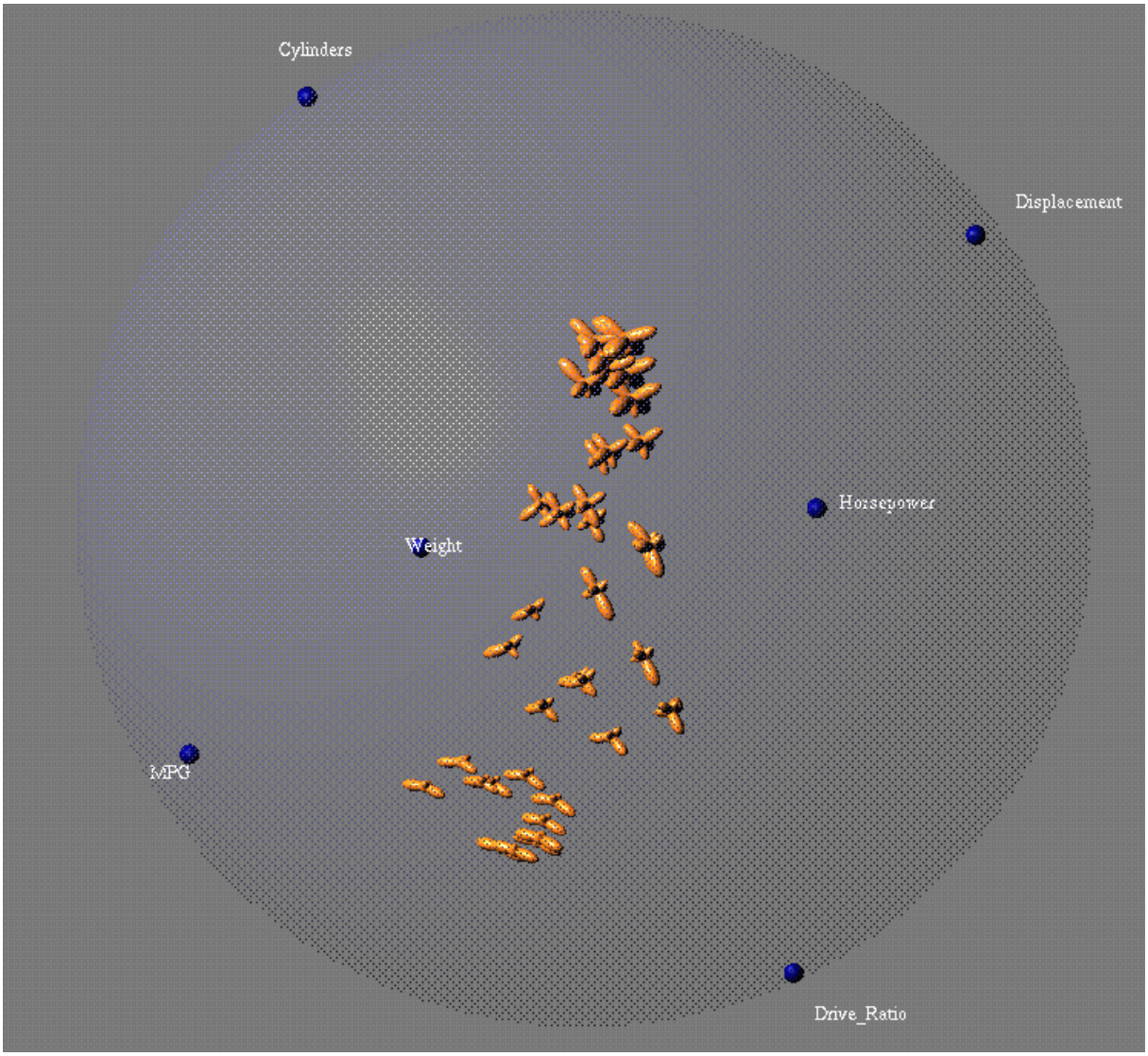

Figure 8: *Visualization of the car data set (38 objects, 6 dimensions) - overview.*

8. J. Panyr, U. Preiser and T. Führing. Kontextuelle Visualisierung von Informationen. Proceedings "19. Oberhofer Kolloquium über Information und Dokumentation", Oberhof, pp. 217-228, 1996 (in German).

9. G.G. Robertson, J.D. Mackinlay and S.K. Card. Cone Trees: Animated 3D Visualizations of Hierarchical Information. Proceedings of ACM Conf. on Human Factors in Computing Systems, pp. 189-193, 1991.

10. M. Sakar and M.H. Brown. Graphical Fisheye Views. Commun. ACM, 37(12), pp. 73-84, 1994.

11. P.C. Wong and R.D. Bergeron. Multivariate Visualization Using Metric Scaling. Proceedings Visualization '97 Conference, Phoenix, pp. 111-118, IEEE, 1997.

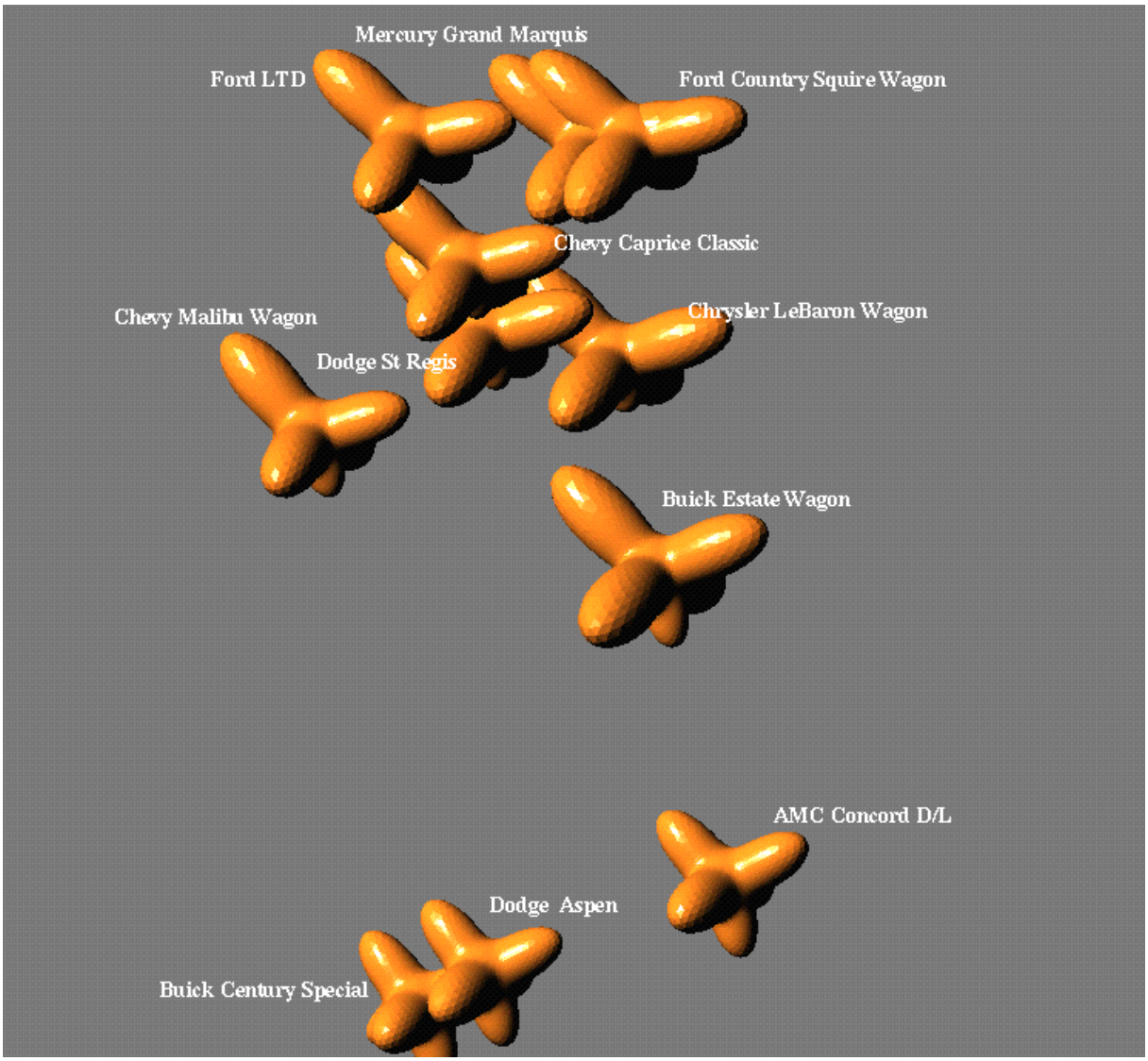

Figure 9: *Visualization of the car data set - zoom into the cluster.*

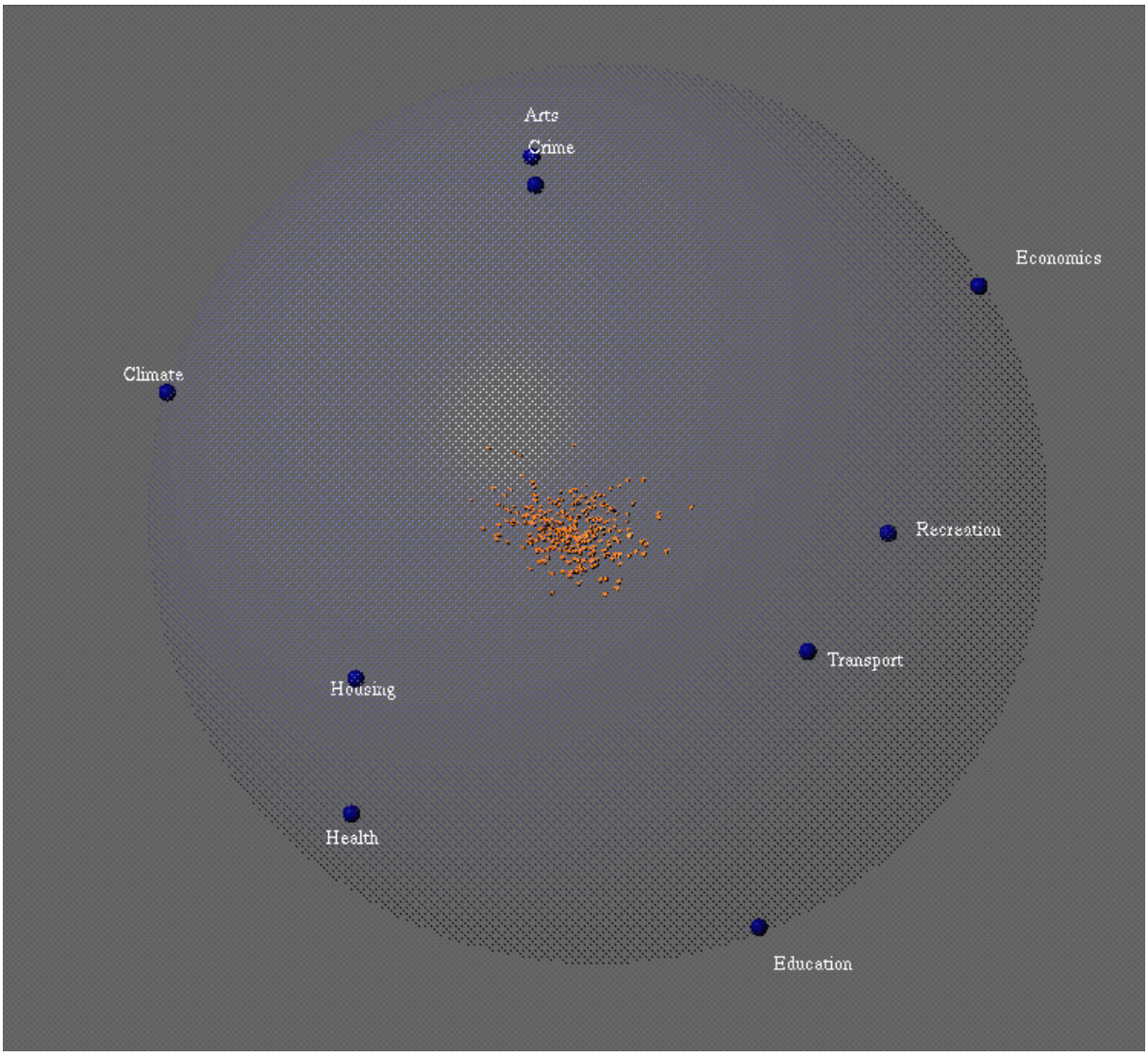

Figure 10: *City data set - overview.*

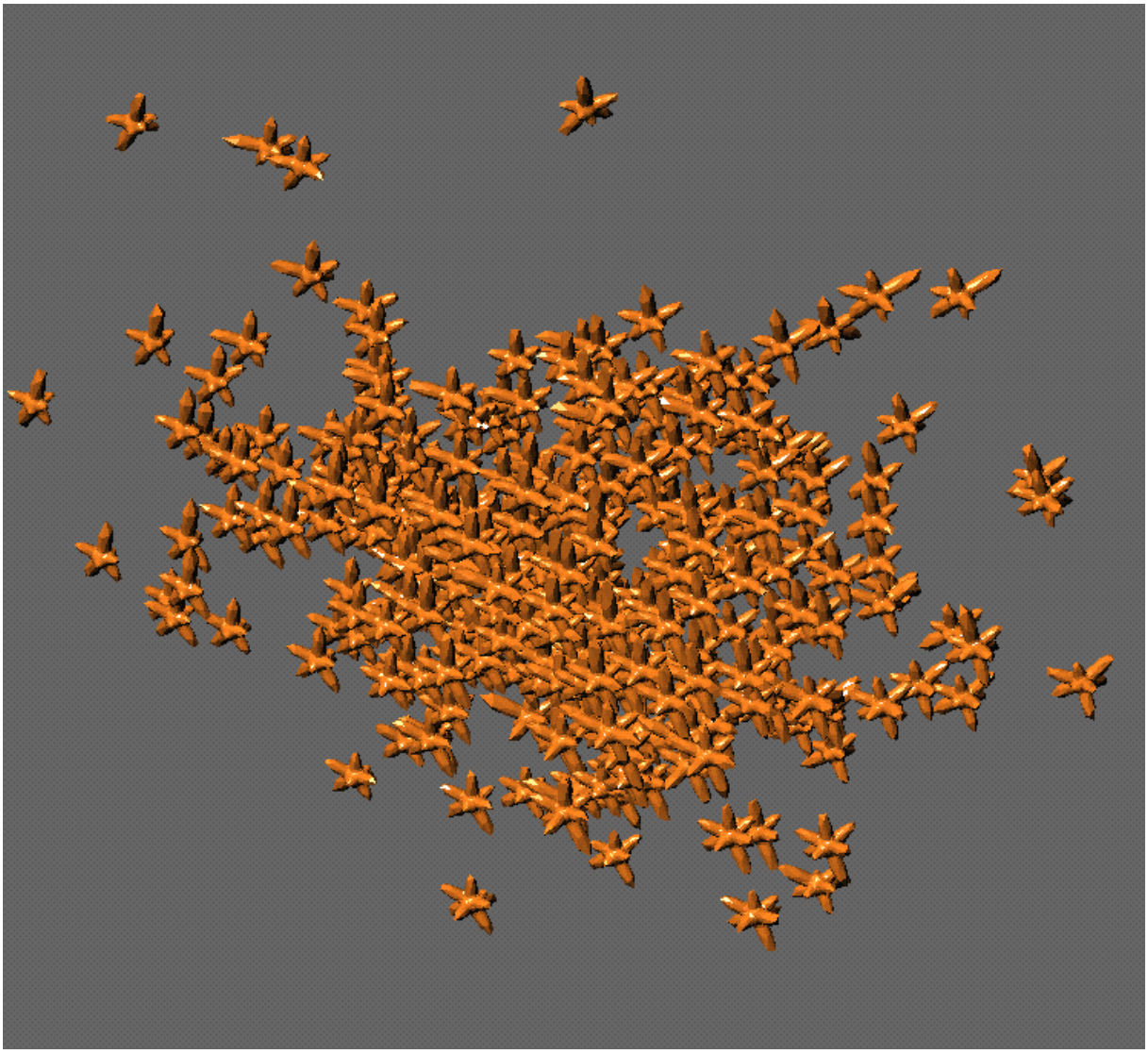

Figure 11: *City data set - the cluster.*

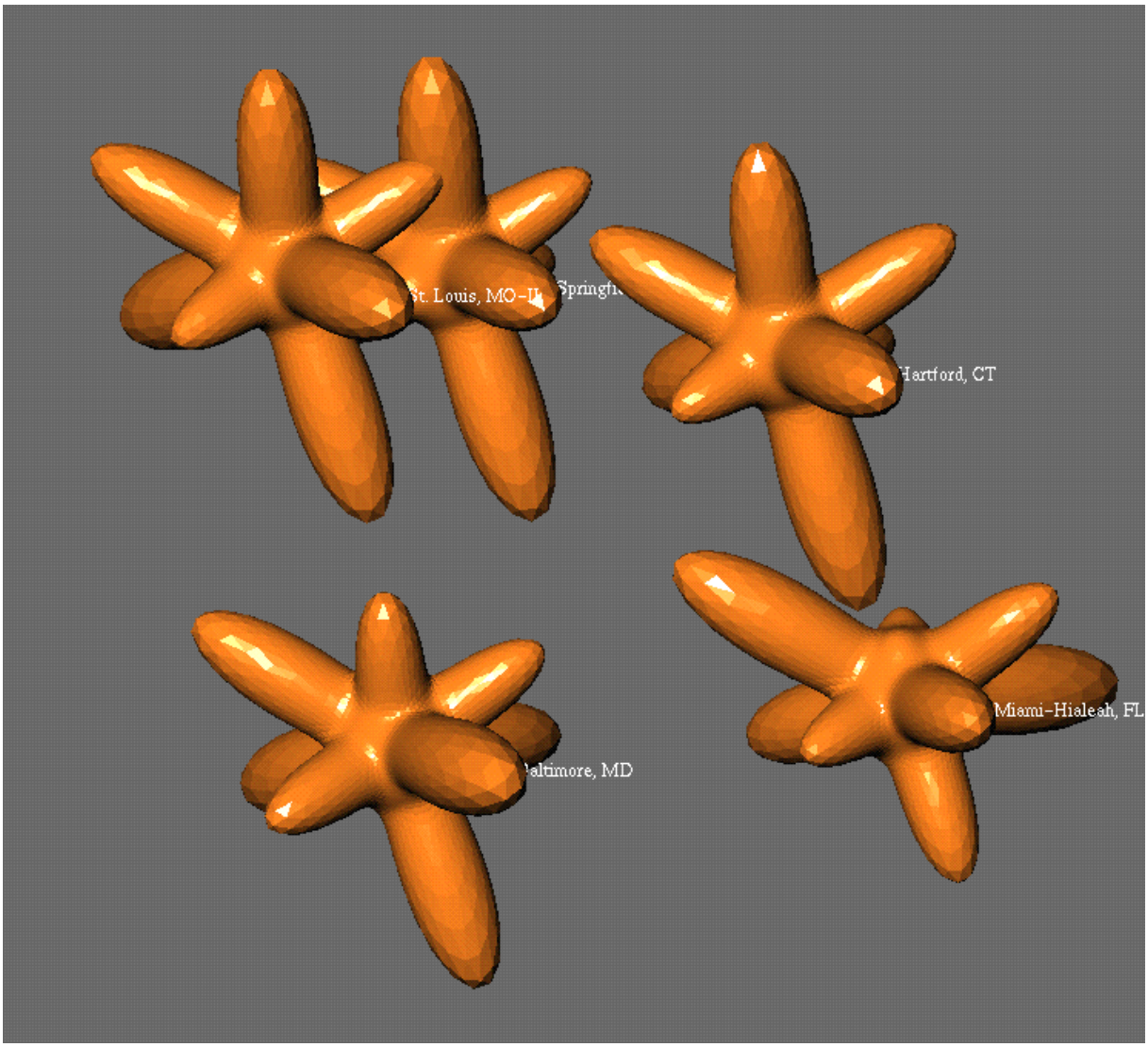

Figure 12: *City data set - zoom of the 5 objects lower right in figure 11.*