Visualization of Temporal Distances

Gerik Scheuermann Thomas Wischgoll

University of Kaiserslautern
Department of Computer Science, Computer Graphics & CAGD
P.O. Box 3049, D-67653 Kaiserslautern
Germany
E-mail: [scheuer|wischgol]@informatik.uni-kl.de

Abstract

In order to visualize *temporal distances*, i.e. the time for traveling from one place to another, we arrange some selected cities according to these distances. In this way, the new positions reflect the connectivity of these cities with respect to time. Unlike existing approaches using tables, our method facilitates a global examination of the connectivity of a whole country. For the database, any connectivity information can be used as long as it is ensured that it is unambiguous. Therefore, any transport system can be considered and even a mixture of such systems could be visualized.

Keywords: information visualization, physically based modeling, transport system, deformation

1 Introduction

Till now, tables are used to visualize the durations of traveling from one city to another which we call temporal distances. Such a description can hardly be evaluated globally. It is only useful for answering local questions about the quality of connections. For an examination of the global interconnectivity, a global view is needed. Such a global method is presented in this paper which arranges the cities approximately proportional to the temporal distance. This results in a contraction within regions that have fast connections and in an enlargement in regions with slower connections. The global effect of a change of a schedule can be analyzed in this manner. The cities have to be repositioned according to the old and new schedule. Then one can compare the results: When the latter is compared with the former a shorter distance between two cities means that the connection is faster now and vice versa. In the same way, one might check if building new railway tracks improves connectivity. The advantage is the consideration of the surrounding regions instead of only focusing on the connected cities.

The first problem we encounter, is the acquisition of valuable data for the temporal distances. In this paper, we use the durations of railway connections because these times are constant under the assumption that the trains arrive on time. Traveling by car is not very predictable, so we cannot get fixed values for the traveling time. Therefore one has to eliminate interferences, such as traffic congestions, otherwise it is not possible to define unique temporal distances. But an interesting investigation in this context might be to compare the visualization of an ideal journey by car with respect to time with a more lasting one because of traffic congestions. This would show the delays produced by traffic congestions at a global point of view and would allow a global comparison to other different transportation systems.

It is also possible to use a mixture of transport systems, for instance a combination of normal trains and the German TransRapid, if the relevant durations are provided in a unique manner. With this information one can visualize improvements in the connectivity by new transport systems.

The organization of the paper is as follows: Section 2 describes the main ideas of the presented method while section 3 shows the results in the application area. Finally, we give a conclusion in section 4 and some ideas for future work in section 5.

2 Description of the Method

First, we have to give an exact definition for temporal distances.

Definition 2.1 *Let A and B be two points and C_{AB} a set of connections between A and B with respect to any transport system. Additionally, let $T : C_{AB} \to \mathbb{R}$ be a map which describes the duration of the given connection. Then the temporal distance between A and B is defined as:*

$$t = min(\{T(x)|x \in C_{AB}\}) \tag{1}$$

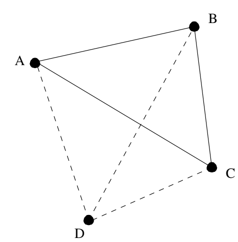

Figure 1: Arrangement of four points

As already mentioned, the cities should be placed in such a way that their distances are approximately proportional to the respective temporal distance. The main problem is that one can place at most three points, named for instance A, B and C, in a plane with predefined distances. When placing the fourth point D as shown in figure 1 with the exact distances to the other three points, it will generally not lie in the same plane. Consequently, we have to approximate the distances to facilitate that all points are part of the same plane:

Criterion 2.2 *Let $S \subset \mathbb{R}^2$ be a finite set of points with $S = s_1, \ldots, s_n$ and $d_{ij} = \|s_i - s_j\|$. Further, let t_{ij} be the temporal distance between the points s_i and s_j in accordance with definition 2.1. Then we are looking for an arrangement of the points $s_1, \ldots, s_n$ that fulfil the following equation:*

$$\sum_{i=1}^{n}\sum_{j=0}^{n}|d_{ij} - t_{ij}| =$$

$$min\left(\left\{\sum_{i=1}^{n}\sum_{j=0}^{n}\left|\,\|s'_i - s'_j\| - t_{ij}\right| \,\middle|\, s'_1, \ldots, s'_n \in \mathbb{R}^2\right\}\right) \tag{2}$$

This results in a problem we already know from graph drawing as described in [1], [2] and [3]. In order to solve it we use a simulation of mass particles connected with springs to each other as described in [4] to obtain an approximation to the solution of formula 2. The main idea is to place a particle with unit mass at the geographic position of every city. These particles are connected by springs with restlengths proportional to the temporal distance of the connections. Generally, this results in a mass-spring-system under tension. Allowing the particles to move, this mass-spring-system starts to oscillate. A damping of the springs slows down the oscillation of the system. Finally, a equilibrium is reached where the deviation of the restlengths from the actual lengths of the springs is at a minimum. This equilibrium gives the desired new coordinates of the cities. The existence of an equilibrium is guaranteed by the damping factor. This solution can easily be computed with an ODE solver described in [5] but for a faster algorithm one can use [6].

To avoid an overall expansion or contraction of the mass-spring-system, one has to scale the restlengths because the temporal and the real distance normally have nothing in common. We determine a scaling factor using the following formula where d_{ij} and t_{ij} are defined in the same way as in criterion 2.2:

$$s = \frac{\sum\limits_{i \in S, j \in S} d_{ij}}{\sum\limits_{i \in S, j \in S} t_{ij}} \tag{3}$$

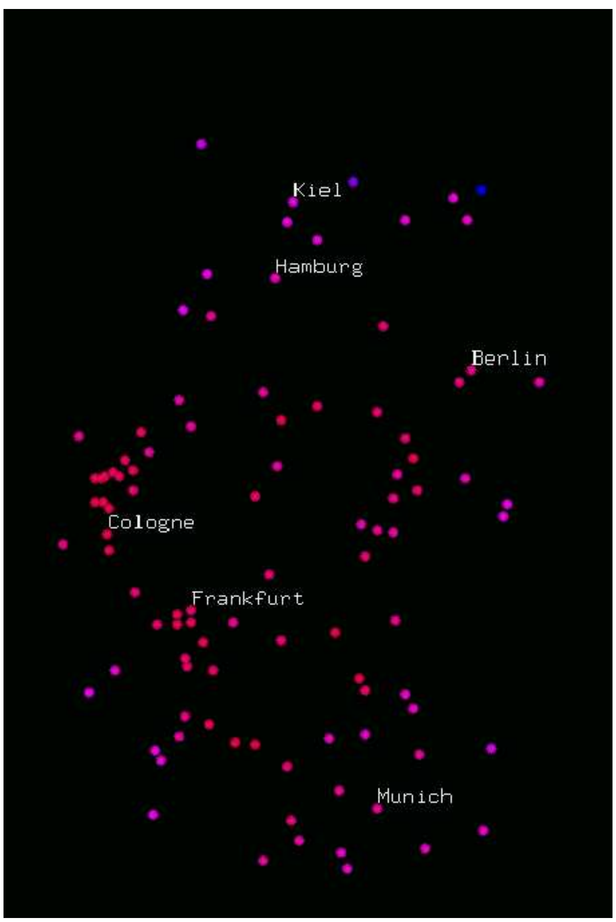

Figure 2: Initial positions of the cities

This scaling factor facilitates that the positions of the cities are only influenced by the temporal distances.

A similar approach is used by [7] where information is placed on a sphere. Therefore mass particles are placed on the inner sphere of two concentric spheres. Every mass particle is connected to each other with a spring where the stiffness of the springs is proportional to the similarity of the corresponding information. Additionally, a connection between the mass particles and the outer sphere is introduced, so that the mass particles stay on the sphere.

3 Results

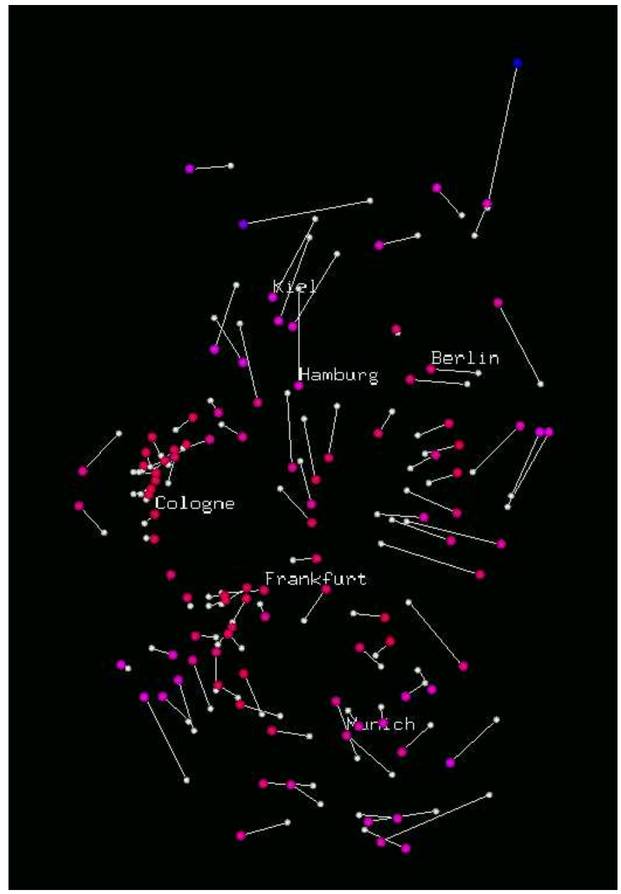

Figure 3: Cities arranged according to temporal distances

This section illustrates the visualization results. To get the data for the temporal distances, we chose all 96 cities in Germany with an InterCity railway station and asked for the 9216 connections between every pair of cities. For that purpose, we used the service provided by the *Deutsche Bahn AG* on their WorldWideWeb pages[1]. Then we defined the temporal distance between two cities by scaling the duration of the respective connection with an appropriate factor described in equation 3 to avoid an unnecessary expansion or contraction.

Figure 2 shows the initial positions of the cities while figure 3 displays the arrangement of the cities according to the temporal distances. To enhance the identification of the markers for the cities, some cities are labeled by their name. The label is placed with

[1]URL: http://bahn.hafas.de

Figure 4: Initial map with positions of the cities

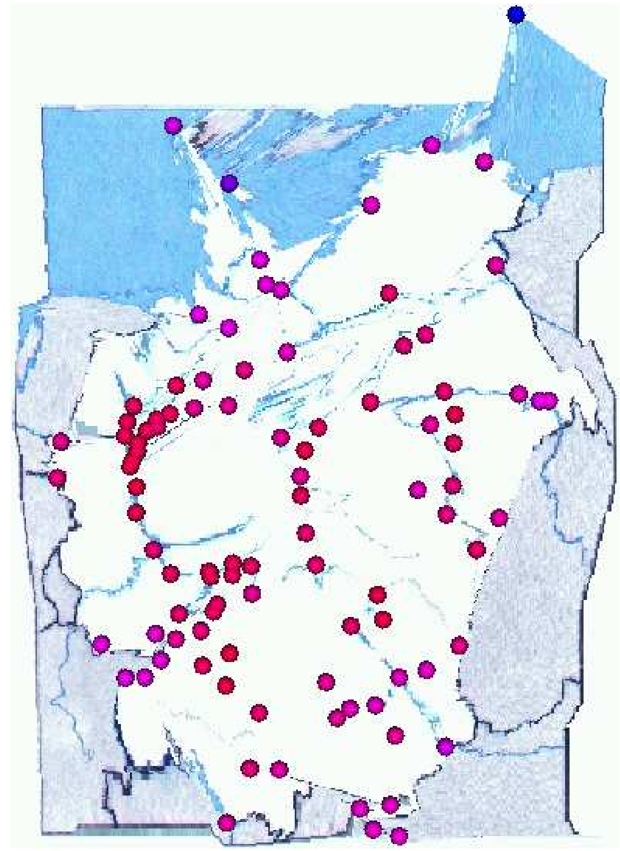

Figure 5: Deformed map with cities arranged according to temporal distances

its lower left corner at the position of the marker. Additionally, lines are drawn from the original coordinates of the cities, which are marked with a smaller white ball, to the displaced position of the marker.

To analyze the relaxation of the springs in the used mass-spring-system the markers are colored from red to blue. For the color scale, an analogy to visualization of temperature is used: Cities with relaxed springs are colored blue. Red markers represent cities where the difference between the length of the springs and the corresponding restlengths reaches the highest level; concerning the coloring this can be compared to a liquid under pressure which gets warm. A red marker shows only that the springs are not as relaxed as a spring connected to a city with a more blue marker, but it is not a sign for a failure of the algorithm. For example, the city in the upper right corner lying on an island has a slow connection to the other cities, so the distances to the other cities are enlarged by the algorithm. Because of its position near to the border the springs can relax until they reach their restlengths which is visualized by the blue marker.

In figure 3, one can see that the cities in Eastern Germany are pushed away from the center. This is due to the fact that the railway tracks are still slower there than in the western part, so the temporal distances are larger. Another effect is that the cities connected by the same railway track get arranged on a string. This can be seen in the left middle part of figure 3 where cities like *Cologne* and *Bonn* are placed.

For further examination it is useful to have a map corresponding to the cities so that one can identify them easier. In the case of the repositioned cities this map has to be deformed appropriately. Figure 4 shows the initial map of Germany with all considered cities. To get a distorted map with respect to the deformed positions of the cities we use an approach similar to those mentioned in [8], [9], [10] or [11]. Therefore we start with a triangulation of the cities at their initial positions. Then the vertices are moved towards the new positions of the cities computed as described in section 2 as long as possible without destroying the triangulation. To prevent too long and small triangles the algorithm guarantees that the vertices do not get to close to the edges. Then one can use the initial triangulation and the actual one to define a deformation by computing a linear map for each triangle which maps the initial into the actual one. After that, the vertices are triangulated again and the same procedure starts all over again until every vertex has reached its desired position or no more movement is possible. In the latter case the edge which prevents a further movement must be bent around the desired position of that vertex. This step guarantees that every position can be reached.

The resulting map is displayed in figure 5 in which one can clearly recognize that the city at the upper right corner is moved upwards and the surrounding island is following this displacement. The same is valid for the earlier stated facts: The east part of Germany gets enlarged because of the longer temporal distances in that region. Also the middle part of the river *Rhein* near the *Ruhrgebiet* is not deformed very much because of the fact that the cities like *Cologne* or *Bonn* lying directly at that river are connected by the same railway track and consequently get arranged on a string.

4 Conclusions

We presented a useful tool for examining the connectivity of every kind of transport system with respect to time. The main advantage is that the results can be analyzed directly at a global point of view without complex studies of the produced material. As already stated in the introduction, there are several different applications. When comparing different situations with respect to connectivity it is useful to provide a fix scaling factor in spite of determining it by the algorithm. Otherwise the results may get disturbed by a different increase or decrease in size.

Because of its low requirements on the given database – only the durations of the connections have to be provided in a file with a simple structure – it can be extended to every kind of transport system and even a mixture of these.

5 Future Work

As the interested reader may already have recognized, the shown deformation is only C^0-continuous but not even C^1. Therefore some regions can show nasty displacements which is mainly evoked by the process of bending edges. Consequently, future research has to improve this deformation and might replace it by a C^1-continuous mapping. Another problem can also be discerned: Regions, where the number of cities and therefore the number of mass particles in the simulation process of section 2 is small, shrink unproportionally compared to regions with a higher density of cities. This has to be avoided to ensure a more precise picture of temporal distances.

Because of the fact that we only use one traveling direction between two cities, we like to consider both directions in our future research, perhaps by deforming the underlying plane according to the difference between the two temporal distances.

Another interesting application would be to use international airports as cities and its air traffic to determine the temporal distances. Therefore the cities have to be placed on a globe which allows the use of the radial direction as an additional degree of freedom which can be used to visualize the direction with greater temporal distance in analogy to the flat case.

6 Acknowledgement

The authors thank the *Deutsche Bahn AG* for enduring the massive requests on their WorldWideWeb server to get the durations of the railway connections created by our shell scripts.

References

[1] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software Practice and Experience*, 2(11), 1991.

[2] A. Frick, A. Ludwig, and H. Mehldau. *A Fast Adaptive Layout Algorithm for Undirected Graphs*, pages 388–403. Proc. Graph Drawing. Springer-Verlag LNCS 894, 1995.

[3] J. Cohen. Drawing graphs to convey proximity: an incremental arrangement method. *ACM Trans. CHI*, 4(11):197–229, 1997.

[4] Andrew Witkin, editor. *An Introduction to Physically Based Modeling*. Siggraph 95, August 1995.

[5] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[6] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. *IEEE Visualization*, pages 127–132, 1996.

[7] M. H. Gross, T. C. Sprenger, and J. Finger. Visualizing information on a sphere. *IEEE Information Visualization*, pages 11–16, 1997.

[8] W. R. Tobler. *Map Transformations of Geographic Space*. PhD thesis, University of Washington, 1961.

[9] W. R. Tobler. Pseudo-cartograms. *The American Cartographer*, 13(1):43–50, 1986.

[10] Herbert Edelsbrunner and Roman Waupotitsch. A combinatoral approach to cartograms. *Computational Geometry*, 7:343–360, 1997.

[11] Donald H. House and Christopher J. Kocmoud. Continuous cartogram construction. *IEEE Visualization*, pages 197–204, 1998.

[12] Will Schroeder, Ken Martin, and Bill Lorenson. *The Visualization Toolkit*. Prentice Hall, 1997.