

# Projective Mapping: A non-iterative method for the layout of multidimensional data

Karina Vashta Assiter  
Computer Science and Systems, Wentworth Institute of Technology  
Boston, Massachusetts 02115, USA

## ABSTRACT

In this paper, we present a fast, non-iterative method for the layout of large-scale, high-dimensional data. In this method (*Projective Mapping*) samples in a dataset  $S$  are mapped into a representation set  $R$  based on the centroid of their geometric relationships to a set  $T$  of pre-positioned references. The time complexity of this *Projective Mapping* method is  $O(kh)$  (where  $k$  is the number of user placed samples in  $R$  that are used for the mapping and  $h$  is the number of samples). We define *faithfulness* as preserving distances precisely, and show how Projective Mapping maps samples from  $S$  into  $R$  faithfully when the references in  $T$  are mapped faithfully for an  $x$ -dimensional subspace of an  $n$ -dimensional space mapped into an  $x$ -dimensional reference space. We demonstrate that the faithfulness property of Projective Mapping can be used to test the dimensionality of a dataset embedded in an  $n$ -dimensional  $S$ . Additionally, we show that any linear transformation applied to the references as they are mapped from  $S$  to  $R$  is also applied to the samples.

**Keywords:** Computing Techniques, Data Mining, Layout, Faithfulness, Computational Geometry.

## 1. INTRODUCTION

One of the challenges for computer scientists and information scientists is how to layout large-scale, multidimensional datasets on a computer screen in a manner that demonstrates the underlying data relationships. For example, a plant in a catalogue might have 20 different attributes describing its characteristics. Any program that displays the plant dataset should limit distortion of inter-plant distances, reduce run-time complexity, and provide both a view of the plant characteristics and an overview of inter-plant relationships.

### Objective of layout

The objective of *layout* is to create a mapping  $\Phi$  from  $h$  samples in a sample set  $S$  to representative points in a representation set  $R$ , so that *similar* samples in  $S$  are mapped *close together* in  $R$ . As an illustration (Figure 1),  $P_i$  and  $P_j$  are samples in  $S$  that are mapped by  $\Phi$  into  $Q_i$  and  $Q_j$  in  $R$ , respectively. A *sample*  $P$  in  $S$  is a vector of either numbers or non-numerical attributes, while  $R$  is a two- or three- dimensional 'map' of the samples.

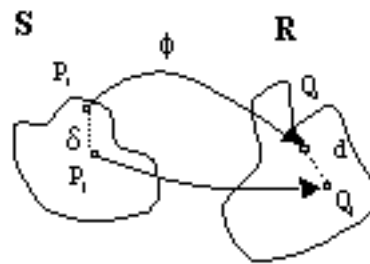


Figure 1 - Illustration of Mapping

### Proximity

*Similarity* between samples in  $S$  is a metric function that is within the group of *proximity* measures. Proximity is defined as distance in space, time or some other way and is important for layout because it establishes the relative relationships between samples in  $S$  that need to be preserved in  $R$ . Proximity functions return a real value based on how far apart the samples lie in the measurement space (in an abstract sense). There are two kinds of proximity measures: similarity (function value is greater when the samples are closer together), and dissimilarity (function value is greater when the samples are further apart). In our illustration of mapping (Figure 1) we use the dissimilarity function  $\delta$ , which is often just Euclidean distance.

### Methods of mapping

Traditional layout methods move points around in  $R$  in order to minimize an *error function*  $E^1$  that describes how accurately proximity values in  $S$  are related to distances in  $R$  (for all pairs). The error function's influence on minimization is to preserve either the actual proximity values between samples, as in Metric MDS (Sammon, 1969) or the rank order<sup>2</sup> of the proximity values, as in Non-Metric MDS (Kruskal, 1964). Unfortunately, traditional layout methods, generally, have run a time complexity of  $O(h^2)$ , where  $h$  is the number of samples.

### Faithfulness in mapping

A mapping  $\Phi: S \rightarrow R$  of an  $x$ -dimensional subspace of an  $n$ -dimensional space into an  $m$ -dimensional reference space is *faithful* if the mapping preserves proximities precisely so that  $\delta(P_i, P_j) = d(\Phi(P_i), \Phi(P_j))$ , when  $P_i$  and  $P_j$  are in  $S$ . Most often with  $n$ -dimensional data,  $x=n$ . In the case where  $x>m$  the data will be distorted; no method of layout can prevent that. We

<sup>1</sup> Error functions for layout are often called the stress of the layout configuration.

<sup>2</sup> Rank order refers to the condition where the proximity values are arranged in either ascending or descending order.

have proven [1] that when the stress is 0, then the mapping is faithful, and the result of the layout in  $\mathbf{R}$  represents the structure of the data in  $\mathbf{S}$ , precisely. Thus, faithfulness is a sufficient test of the dimensionality of the data in  $\mathbf{S}$ . The traditional optimization methods, generally, do not map samples faithfully<sup>3</sup> in the degenerate case of a one or two-dimensional data set in  $\mathbf{S}$ .

### Subjective Layout

With a subjective layout method, a sample's placement in the layout  $\mathbf{R}$  depends upon its relationship with user defined (and positioned) anchors, such that inter-object relationships may or may not be preserved. In most subjective layout methods, anchors in  $\mathbf{T}$  are queries (keywords, mailing lists, etc.) instead of representatives of samples from  $\mathbf{S}$ . Like the SemNet system [3], we define a subjective layout algorithm from  $\mathbf{S}$  to  $\mathbf{R}$ , where each map is based upon choosing and placing a tuple of reference points  $\mathbf{T}$  in  $\mathbf{S}$  into layout positions  $\mathbf{Q}$  in  $\mathbf{R}$ , such that any sample  $P$  in  $\mathbf{S}$  is mapped to  $\mathbf{R}$  based on the average of its geometric relationships to valid tuples of references in  $\mathbf{T}$ .

In the rest of this paper, we introduce the Projective Mapping method, including notation, figures and a complete set of algorithmic steps. Then, we demonstrate the performance of the method on simple geometric data structures. This leads to a comparison between the results of Projective Mapping and the results of a traditional method of layout (Sammon mapping). Finally, we conclude with the benefits and weaknesses of the method, and discuss future work.

## 2. PROJECTIVE MAPPING

In this method, we place a point in  $\mathbf{R}$  based on the average geometric position calculated from valid pairs of references in  $\mathbf{T}$ . Each valid pair of references  $\{S_j, S_k\}$  determines a line  $L_1$  in  $\mathbf{S}$  (Figure 2). We determine where  $L_1$  intersects with the perpendicular line  $L_2$  projected from  $P$ . This gives us two parametric time values  $t_1$  and  $t_2$ ;  $t_1$  is how far the intersection point,  $J$ , is along  $L_1$ , and  $t_2$  is how far  $P$  is along  $L_2$ .  $t_1$  can be used to determine where a mapped intersection point  $J'$  falls along the line  $L_1'$  between the two mapped references  $R_j$  and  $R_k$  in  $\mathbf{R}$ .

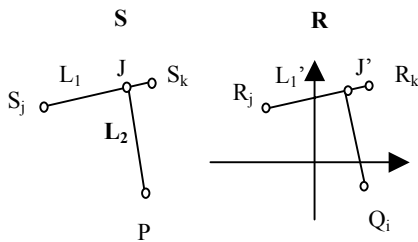


Figure 2 - Projective Mapping Illustration

If we find the unit vector perpendicular to  $L_1'$  in  $\mathbf{R}$ , then  $Q_i$  will fall in the direction of that vector, at a distance determined by the second time parameter, and a scale factor. The scale factor is the ratio between  $|S_j - S_k|$  in  $\mathbf{S}$  and  $|R_j - R_k|$  in  $\mathbf{R}$ .

<sup>3</sup> A test for faithfulness of layout is whether or not the calculated stress of the layout configuration is 0.

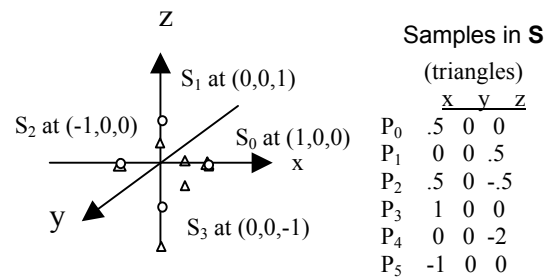
The direction of the perpendicular toward  $Q_i$  is determined by projecting  $P$ ,  $S_j$ , and  $S_k$  onto a projection plane,  $P$ ; if projected  $Q_i$  falls to the right of the projection of  $L_1$  then  $Q_i$  is placed to the right of  $L_1'$  in  $\mathbf{R}$ .

The complete algorithm for mapping consists of applying the projection method to each pair of references, and then averaging [1].

## 3. EXPERIMENTAL RESULTS

### Two-dimensional subspace embedded in n-dimensional space maps faithfully to $\mathbf{R}$

If  $\mathbf{S}$  is n-dimensional, but actually represents a two-dimensional subspace, and samples in  $\mathbf{S}$  are mapped into a two-dimensional representation space  $\mathbf{R}$  via projective mapping, then the two-dimensional projection  $\mathbf{R}$  is faithful to the original two-dimensional subspace of  $\mathbf{S}$ . (see proof in [1]). In Figure 3, we see the result of a two-dimensional data set embedded in a three-dimensional space, mapped to a two-dimensional space.



a) 2D references in 3D sample space  $\mathbf{S}$



b) Layout in  $\mathbf{R}$

Figure 3 - Two-dimensional embedded in 3D

In order to verify this result, let us look at the reference mapping as coordinates in the following tables:

Table 1 - References mapped from  $\mathbf{S}$  to  $\mathbf{R}$

Index	In $\mathbf{S}$	In $\mathbf{R}$
0	(1,0,0)	(1,0)
1	(0,0,1)	(0,1)
2	(-1,0,0)	(-1,0)
3	(0,0,-1)	(0,-1)

We see that references are mapped faithfully in the table above. Now let us look at the samples:

Table 2 - Samples mapped from  $\mathbf{S}$  to  $\mathbf{R}$

Index	In $\mathbf{S}$	In $\mathbf{R}$
0	(.5,0,0)	(0.50, 0.00)
1	(0,0,.5)	(0.00, 0.50)
2	(.5,0,-.5)	(0.50, -0.50)

3	(1,0,0)	(1.00, 0.00 )
4	(0,0,-2)	(0.00, -2.00 )
5	(-1,0,0)	(-1.00, 0.00 )

As we see, samples are mapped faithfully so that they preserve distances.

### Linear Transformations

Linear transformations applied to references in **T** result in the same linear transformations applied to data samples. We prove this in [1] and demonstrate it in following example (Figure 4). In this example, lines in **S** are mapped to lines in **R**. **T** contains a regular grid of references. A single reference  $S_{12}$  in **T** is moved from its position between  $S_{11}$  and  $S_{13}$  in the grid, to  $R_{12}$  in the figure. All sample positions on a line in **S** are mapped to coordinates on a line in **R**.

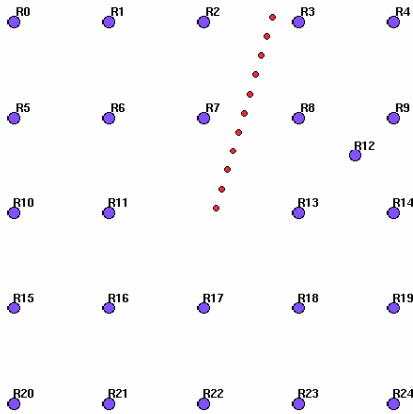


Figure 4 - Example of lines to lines

## 4. COMPARING METHODS

In this section, we compare Projective mapping with Sammon mapping [4] on two sets of data: a hypercube and the Iris dataset. We compare with Sammon mappings because its layouts are representative of the all-pairs layout methods.

### Hypercube tests

The hypercube samples were 00000-11111. For Projective mapping, the references in **S** were 00000-00011 and the associated references in **R** were 00-11. The plane for the projection was {00001,00010}. The result of Projective Mapping on the hypercube is seen in (Figure 5). The result looks like a flattened cube. The Sammon stress calculated on the final configuration was .235. The elapsed time was 18 seconds.



Figure 5 - Projective Mapping on the Hypercube

The result of Sammon mapping on the hypercube is seen in (Figure 6). The final stress converged to 0.118. The elapsed time was approximately 1 minute. The layout configuration does not look like a flattened cube; instead, it looks like a web where all samples are connected.

We did a set of tests on the hypercube [1], and in each case varied the angle of the projection plane. We found that Projective Mapping tends to preserve shape and structure, while Sammon Mapping tends to preserve proximities (and not shape and structure).



Figure 6 - Sammon Mapping on the Hypercube

### Iris dataset

The Iris dataset has 150 samples with three clusters of Iris flowers. When we ran Projective Mapping against the data set with three references, we get the result in Figure 7. The final stress was 1.190. The elapsed time was 1 minute and 30 seconds. The three clusters of the dataset do not have clear boundaries.

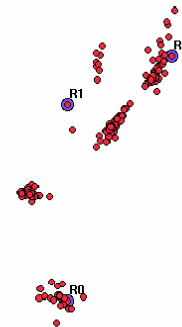


Figure 7 - Projective Mapping on the Iris dataset

Compare this to the Sammon mapping result in Figure 8. The initial configuration for the Sammon mapping run was a straight line of Samples along the principal axis of the data. The final stress converged to a local minimum of .0046. The elapsed time was 7 minutes. The Sammon mapping clearly shows the clusters of the dataset.

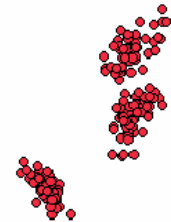


Figure 8 - Sammon mapping on the Iris dataset

## 5. CONCLUSIONS

The intent of this work was to find a fast subjective mapping method that would produce layouts in a representation space **R** that were faithful to a one- or two- dimensional sample space **S**. We also wanted the mapping to result in layout distances in **R** that preserved, as well as possible, the proximities between samples from an n-dimensional sample space **S**.

Projective Mapping is fast, with a run time of  $O(k^2 h)$ , and subjective, with references selected and placed by the user. It creates faithful layouts in one and two-dimensions. Additionally, any linear transformation that is applied to the references as they are mapped from  $\mathbf{S}$  to  $\mathbf{R}$  is also applied to the samples.

Projective Mapping can also map faithfully from an  $n$ -dimensional  $\mathbf{S}$  to an  $x$ -dimensional  $\mathbf{R}$ , when the dimension of the data subspace is less than or equal to the dimension of  $\mathbf{R}$ . Thus, the benefit of faithfulness is that it can test the dimensionality of a dataset embedded in an  $n$ -dimensional  $\mathbf{S}$ .

#### Preserving Proximities

A disadvantage of the method is it does not preserve proximities well enough (in some cases) when  $\mathbf{S}$  is  $n$ -dimensional. The spring model methods, such as Bead [2] and the gradient optimization methods, such as [4] and [5], result in lower stress values, where stress measures the goodness-of-fit of the distances in  $\mathbf{R}$  to the proximities in  $\mathbf{S}$ .

We would like to determine if the mapping could be improved by the use of multiple projective planes and associated views, e.g., along principle axes or even relative to an arbitrary orthonormal basis. One criterion might then be to find the low stress on average, rather than low stress for one view.

As we saw in the hypercube test, the method can give informative views of a dataset, providing a sense of the datasets' structure. We ran a series of tests [1] that varied the plane selection; as the plane rotated through space, each orientation of the plane provided a unique view.

#### Scale Factor

For each pair of references  $S_j$  and  $S_k$  the scale factor is based on the ratio between  $|R_j - R_k|$  and  $|S_j - S_k|$ . We found that a non-uniform scale factor can result in inaccurate placement of samples, with samples mapped into  $\mathbf{R}$  close to references that are far away from them in  $\mathbf{S}$ . Also, when mapping from  $N$ -dimensions to two-dimensions, the distances between mapped samples in  $\mathbf{R}$  may not need to be as large as the (scaled) proximities in  $\mathbf{S}$ .

In future work, we will use heuristics for making the distances that samples are mapped away from reference pair lines, uniform. This preserves the linear transformations (scaling, identity, rotation), but it can also prevent distortions.

#### Reference Set

When we tested our method on the Iris dataset [1], we found that randomly placed references resulted in the lowest stress values. In future work, we would like to run a variety of tests with both randomly and faithfully mapped references.

Another issue is how accurately we can map the references in  $\mathbf{T}$  when the number of references in  $\mathbf{T}$  grows large. One option would be to use the spring model on the  $k$  references in  $\mathbf{T}$ , and then apply the projective mapping method to all other samples. The run time for this option would be  $O(k^2) + O(k^2 h)$ , where  $h$  is the number of samples.

#### Volume of tests

We would like to do more tests varying plane selection, data dimensionality and reference set selection and size.

#### Generalized for mapping to $n$ -dimensions

Finally, once we understand the method better, we would like to see if Projective mapping could be generalized for an  $n$ -dimensional to  $n$ -dimensional mapping.

#### 6. REFERENCES

- [1] K.V. Assiter, Projective Mapping: A non-iterative method for the layout of multidimensional data. Dissertation. Tufts University, Medford, MA, 2001.
- [2] M. Chalmers and P. Chitson, "Bead: Explorations in Information Visualization", SIGIR '92: Proceedings of the Fifteenth annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Denmark: ACM Press, 1992.
- [3] K.M. Fairchild, S. E. Poltrock and G. W. Furnas, "SemNet: Three-Dimensional Graphic Representations of Large Knowledge Bases", Cognitive Science and its Applications for Human-Computer Interaction, R. Guindon, Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1988, pp. 201-233.
- [4] J.B. Kruskal, "Nonmetric multidimensional scaling: a numerical method", Psychometrika, Vol. 29, 1964, pp. 115-129.
- [5] J.W. Sammon, "A nonlinear mapping algorithm for data structure analysis", IEEE Trans. Computers, Vol. 18, No. 5, 1969, pp. 401-409.