Visualization by Examples: Mapping Data to Visual Representations using Few Correspondences

Marc Alexa and Wolfgang Müller

Darmstadt University of Technology, Department of Computer Science, Interactive Graphics System Group, Rundeturmstr. 6, 64283 Darmstadt, Germany
{alexa,mueller}@gris.informatik.tu-darmstadt.de,
WWW home page: http://www.igd.fhg.de/~{alexa,mueller}

Abstract. In this paper we propose a new approach for the generation of visual scales for the visualization of scalar and multivariate data. Based on the specification of only a few correspondences between the data set and elements of a space of visual representations complex visualization mappings are produced. The foundation of this approach is the introduction of a multidimensional space of visual representations. The mapping between these spaces can be defined by approximating or satisfying the user defined relations between data values and visual atributes.

1 Introduction

The visualization of scalar and multivariate quantitative data involves the mapping of data onto a visual scale. The principles of such a mapping of scalar data to scales of visual attributes are well-known for a number of basic scales. In color mapping, data values are mapped to appropriate hue of lightness values. Scatter plots are based on the principle of mapping data values to positions, or more exactly, to distances from an axis. Other variables often applied for such purposes are scale, form, and texture. Bertin [4] describes a general methodology of how to select an appropriate mapping to these visual variables and how to combine them. Cleveland gives a ranking of their effectiveness [9]. For multivariate data with two upto five dimensions more complex color and texture scales can provide solutions in some cases. For the visualization of local variations higher dimensions data glyphs have been proposed and been applied successfully. Chernoff faces [8] and stick figures [13] are well-known examples for this visualization approach. Tufte gives a good overview of relevant visualization techniques and discusses their effectiveness in certain application areas [17]. Nevertheless, a generally accepted set of visualization rules does not exist.
All the techniques mentioned above represent fundamental approaches to the visualization of scalar and multivariate data. The general understanding is that no single of these visualizations is effective in all possible situations. One visualization may – and should of course – produce new insights and, by this, new questions which again produce the need of different views to the data. A good

number of applications have proven that a user can gain knowledge about some unknown data more effectively when provided with highly interactive techniques [15]. Techniques such as Focusing and Linking [7] extend this interactivity even further by connecting different views to the data using interactive feedback.
However, effective visualization still is very, very difficult. There are a number of reasons for this. First, a mapping of application data to these fundamental variables involves an abstraction. If the user is familiar with the idea of this mapping, he may understand the generated visualization pretty good. However, often the application context is lost and the visualization which was applied to simplify the analysis of the data involves an analysis step or experience by its own.
Second, it is not easy to visualize a number of parameters using these fundamental mappings only. Usually, the visualization of multiparameter data involves the generation of application specific models and solutions. General solutions and general scales do not exist for this purpose. Consequently, the user needs methods to define a visual scale for such applications very quickly and easily. Such methods do hardly exist to date.
Last, and may be even most important, it is still difficult to change visualization parameters and to produce a new, appropriate visualization intuitively. For example, to highlight a specific data value one has usually to modify the data filtering or to completely redefine the mapping of the data to visual attributes. A direct and interactive modification of local data mappings is hardly provided by any visualization technique today.
In this paper, we introduce a new approach for the visualization of scalar and multivariate data, which addresses the problems presented above. This approach is based on the direct and interactive specification of local data mappings.

2 Visualization by Examples

The general paradigm of our visualization approach is to enable the user to visualize some data by specifying the mapping of a small number of selected data values. We call this Visualization by Example.
This approach can be explained best with an example. A simple mapping of some scalar quantitative data to color can be defined by linking two arbitrary data values with appropriate color values. This results in a linear mapping. Further links can be supplied to adjust the mapping locally, resulting in a more sophisticated mapping function. The visualization of any local feature and its neighborhood is directly and intuitively controlled by the user and may be easily changed when provided with appropriate visual attributes or objects on which the data may be mapped. Note that this strategy is fundamentally different from the selction of a color scale and applying this scale to all data.
While the direct and interactive linking of data values with visual representations is not new, it has not been combined with appropriate methods for the approximation of data mappings based on the supplied correspondences.
In the example presented above this approach seems simple and easy to follow.

However, the generalization of this approach calls for a mathematical model describing the data objects, flexible visual scales, and the mapping between this values based on a small number of parameters and features.
In addition, the user will have some additional knowledge about the data in many cases, which can be exploited with our approach. Moreover, the user might want to highlight several data values by mapping them to special representations.
There exist approaches to describe data spaces based on mathematical models [6]. Mathematical models have currently been proposed to describe the spaces of visual scales and representations in combination with the use of morphing to construct the corresponding graphical objects for this purpose [2] [12]. This approach allows to define rich sets of useful visual representations from only a few graphical base objects. As such, this method provides the appropriate foundation for a Visualization by Example.
In the following section we will discuss the fundamentals and characteristics in more detail.

3 Space of Visual Representations

For our visualization technique, the visual representations have to be structured as a multidimensional space. That is, a visual object has to be element of an n-dimensional space and represented by a vector $r \in \mathbb{R}^n$.
For many visual scales such a representation is quite natural:

- Color can be represented by real values in $[0, 1]^3$, color scales might be represented by real numbers in $[0, 1]$.
- Size or position is naturally a real number.
- For textures one defines a number of real valued parameters, which control their appearance.
- In general: If the visual representations are used to depict quantitative data there has to be a reasonable understanding in terms of real valued vector spaces.

In [2] and [12] we have proposed a more flexible way to define spaces of visual representations: Given a number of graphical objects of any class (images, polyhedra, etc.) we construct a space by morphing among these objects. Morphing is usually applied to generate animations and, as such, exploited only for blending between two objects. In this approach, a morph among multiple objects by performing several morphing operations between two objects subsequently defines a multidimensional space of objects, whereby each of the morphing operations adds another dimension to the space. Barycentric coordinates can be used to represent elements of that space. Elements are constructed by morphing between two objects recursively. In that case, a space constructed from n graphical base objects is an affine space with dimension $n - 1$ (a more rigorous treatment of the mathematical properties of such spaces can be found in [1]).
Since morphing is applicable to produce scales such as color, position, size, etc. we understand this to be a generalization of these techniques to define spaces of

visual representations. For the mapping technique explained in the next section, however, the actual construction of the space is not relevant. All we need are visual representations that can be referred to by n-vectors.
The strength of using morphing techniques to generate visual representations of data becomes evident when applied to multivariate data. As mentioned before, it has been proven difficult to find intuitive visual representations for multivariate data and multidimensional objects. By morphing among multiple objects one could visualize multivariate data as elements of a space of graphical objects.

4 Mapping Data to Coordinates

The main idea of our approach is to let the user define several relations between data values and graphical representations. These correspondences are used to construct a mapping from data to visual representations. We want to allow the user to define any number of correspondences, usually beginning with only a small number of correspondences. Depending on the application the user might decide to generate an affine mapping in any case, or, if no simple affine mapping can be found, to accept a non-linear function to depict the mapping.
To define a single correspondence, the user first chooses a data value and then searches the space of graphical representations for a suitable element. That is, the user gives an example for the intended relation between data and visualization. We denote the space of data values as V^d, i.e. each vector consists of d variates. Assume a number e of data vectors $\boldsymbol{v_0}, \boldsymbol{v_1}, \ldots, \boldsymbol{v_{e-1}}$ should be mapped to a coordinate $\boldsymbol{r_0}, \boldsymbol{r_1}, \ldots, \boldsymbol{r_{e-1}}$ in the space of visual representations. If $e \leq d$ then an affine mapping $a : V^d \rightarrow \mathbb{R}^n$ exists that satisfies $a(\boldsymbol{v_i}) = \boldsymbol{r_i}$ for all $i \in 0, \ldots, e-1$. However, if $e > d$ that mapping does not necessarily exist.
We suggest to use a linear mapping whenever possible, i.e. in case $e \leq d$. The reason for this is that visual scales are still meaningful, properties of the data variates are preserved in the visualization, and the order of the data values is not changed.
If e is greater than d we offer three choices:

1. An affine mapping that fits the given correspondences as far as possible.
2. A non-linear mapping that satisfies all relations by locally changing a linear approximation.
3. A non-linear mapping that satisfies all relations by globally interpolating the relations.

The second and third mapping are constructed using the same approach as discussed in section 4.2. Both need an approximation of the affine mapping similar to the one used in the first approach. In the upcoming subsection we explain how to calculate that affine mapping, independent of the number of given relations.

4.1 Finding an affine mapping

We want the mapping a to be represented by a matrix multiplication. Thus, we are searching for a $n \times d$ matrix A that maps from V^d to coordinates in the space of visual representations. In case $e \leq d$, A has to satisfy the simultaneous equations

$$\begin{aligned} A\boldsymbol{v_0} &= \boldsymbol{r_0} \\ A\boldsymbol{v_1} &= \boldsymbol{r_1} \\ \vdots &= \vdots \\ A\boldsymbol{v_{e-1}} &= \boldsymbol{r_{e-1}}, \end{aligned} \tag{1}$$

in case $e > d$ we like to minimize the residual

$$(\|A\boldsymbol{v_0} - \boldsymbol{r_0}\|, \|A\boldsymbol{v_1} - \boldsymbol{r_1}\|, \ldots, \|A\boldsymbol{v_{e-1}} - \boldsymbol{r_{e-1}}\|) \ .$$

We now first solve the first case. The techniques we employ here will automatically produce a solution to the second case.
If we look at the i-th row $\boldsymbol{a_i}$ of A we get the simultaneous equations

$$\begin{aligned} \boldsymbol{a_i v_0} &= r_{0_i} \\ \boldsymbol{a_i v_1} &= r_{1_i} \\ \vdots &= \vdots \\ \boldsymbol{a_i v_{e-1}} &= r_{e-1_i}. \end{aligned} \tag{2}$$

We define the $d \times e$ matrix

$$B = \begin{pmatrix} - & \boldsymbol{v_0} & - \\ - & \boldsymbol{v_1} & - \\ & \vdots & \\ - & \boldsymbol{v_{e-1}} & - \end{pmatrix} \tag{3}$$

to rewrite the simultaneous equations in (2) as a matrix equation:

$$Ba_i^T = \left(r_{0_i}, r_{1_i}, \ddots, r_{e-1_{i,}} \right)^T \tag{4}$$

The solutions of these n systems of linear equations yield the rows of A. In order to solve one of these systems we use the Singular Value Decomposition (SVD, [14]). The SVD is a decomposition of a matrix into a product of an orthogonal matrix, a diagonal matrix and again an orthogonal matrix. This decomposition is always possible [11]. For most applications the values of the diagonal matrix (singular values) are of particular interest. If any of the singular values is zero, the above matrix equation has no solution. If we replace any zero singular value by infinity, we can invert the diagonal matrix (the orthogonal matrices are invertible anyway). By multiplying both sides of (4) with the three inverted matrices we always find a solution for a_i, no matter what the condition of B is. The SVD has several nice properties that are interesting for our problem:

1. It gives a stable solution in the quadratic case, even in the presence of degeneracies in the matrix.

2. It solves the underspecified case in a reasonable way, i.e. out of the space of solutions it returns the one closest to the origin.
3. It solves the overspecified case by minimizing the quadratic error measure of the residual.

Using the SVD we can compute all rows of A and thus have found the affine mapping we were searching for.

4.2 Non-linear mappings

We want to find a mapping a the satisfies all equations $a(\boldsymbol{v_i}) = \boldsymbol{r_i}$. This could be seen as a scattered data interpolation problem where we try to find a smooth interpolation between the values $\boldsymbol{r_i}$ given at locations $\boldsymbol{v_i}$. Contrary to some other application domains of scattered data interpolation, we deal with different and high dimensions of the vectors and typically the number of relations is close to the dimension of the input data.

As explained before, it seems desirable to have an affine mapping from the data values to the space of visual representations. Therefore, we always start with a linear approximation of the mapping (as calculated in the previous section) and then fit the relations in the mapping by tiny adjustments. For these adjustments we use radial sums. The idea of combining an affine mapping with radial sums for scattered data interpolation is considered in e.g. [3] and [16] (for two-dimensional vectors, only).

Hence, we define a by

$$a(\boldsymbol{v}) = A\boldsymbol{r} + \sum_j \boldsymbol{w_j} f(|\boldsymbol{v} - \boldsymbol{v_j}|), \boldsymbol{v} \in V^d, \boldsymbol{r} \in \mathbb{R}^n \tag{5}$$

where $\boldsymbol{w_i} \in \mathbb{R}^n$ are vector weights for a radial function $f : \mathbb{R} \rightarrow \mathbb{R}$. We consider only two choices for f:

1. The gaussian $f(x) = e^{-x^2/c^2}$, which is intended for locally fitting the map to the given relations.
2. The shifted log $f(x) = \log \sqrt{(x^2 + c^2)}$, which is a solution to the spline energy minimization problem and, as such, results in more global solutions.

We compute A beforehand as explained in the previous section. Thus, the only unknown in (5) is a pure radial sum, which is solved by constituting the known relations

$$\boldsymbol{r_i} - A\boldsymbol{r_i} = \sum_j \boldsymbol{w_j} f(|\boldsymbol{v_i} - \boldsymbol{v_j}|) \tag{6}$$

This can be written in matrix form by defining

$$F = \begin{pmatrix} f(0) & f(|\boldsymbol{v_0} - \boldsymbol{v_1}|) & \ldots f(|\boldsymbol{v_0} - \boldsymbol{v_{e-1}}|) \\ f(|\boldsymbol{v_1} - \boldsymbol{v_0}|) & f(0) & \ldots f(|\boldsymbol{v_1} - \boldsymbol{v_{e-1}}|) \\ \vdots & \vdots & \ddots \quad \vdots \\ f(|\boldsymbol{v_{e-1}} - \boldsymbol{v_0}|) & f(|\boldsymbol{v_{e-1}} - \boldsymbol{v_1}|) & \ldots \quad f(0) \end{pmatrix}$$

and separating (6) according to the n dimensions of $\boldsymbol{r_i}$ and $\boldsymbol{w_i}$:

$$F \begin{pmatrix} w_{0_i} \\ w_{1_i} \\ \vdots \\ w_{e-1_i} \end{pmatrix} = \begin{pmatrix} r_{0_i} - \boldsymbol{a_i r_0} \\ r_{1_i} - \boldsymbol{a_i r_1} \\ \vdots \\ r_{e-1_i} - \boldsymbol{a_i r_{e-1}} \end{pmatrix}, i \in 0, \ldots, n-1 \quad (7)$$

Again, we solve these n equations by calculating the SVD of F. This time we are sure that an exact solution exist, because the solvability for the above radial functions f can be proven [10].

5 Results

We will demostrate the techniques at two examples. These examples show two principally different application scenarios:

- The first example shows the mapping from multivariate data onto low-dimensional visual representation. That is, the dimension of the data is much higher than the dimension of the representations.
- The second example shows a mapping from scalar data onto either basic or more complex, multiparameter representations. Here, specific aspects of the scalar data set are mapped to a specific channel of the visual attribute enhancing the expressiveness of the visualization.

5.1 Visualizing city rankings

In this example we visualize an overall (scalar) ranking of cities in the USA. Suppose we want a visual aid for a decision which of the major cities would be nice to live in. In order to quantify the different amenities and drawbacks of these cities we use data from "The places rated almanac" [5]. This data contains values for nine different categories. That means, we need to project nine-valued vectors onto scalar values.

To visualize the ranking of the cities we use a Chernoff-like approach. The faces are generated by morphing among a standard set of facial expressions (see [2]). In this example we make use of only a smile and a grumble, defining a one-dimensional visual scale. Thus, the degree of smiling represents the living quality determined by a combination of the nine data attributes from [5].

One way to find this mapping might be to inspect the nine different categories and try to find some weights for the values. This requires not only to define nine values, also the correlation to the outcome of this mapping does not take into account the user's knowledge about the cities.

A more intuitive approach is to allow the user to supply a ranking based on personal experience. Remark that a ranking of a subset of all cities is sufficient. In figure 1 only three examples were given to generate an affine mapping. Namely, Chicago was thought to be nice to live in and was mapped to a smiling

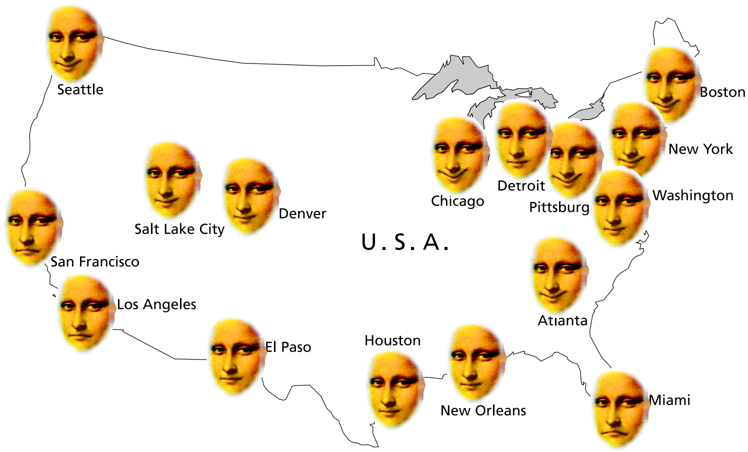

Fig. 1. Cities of the united states represented by mona lisa faces. The representation is generated from 9-valued ranking vectors. The mapping was defined by mapping Chicago to a smile, Washington to neutral face, and Miami to a grumble.

face, whereas Miami was unacceptable and mapped to a grumble. Additionally, Washington appeared nice but expensive and, therefore, mapped to a neutral face.

5.2 CT scan data

In this example, we inspect CT scan data from the "Visible Human"-project. The data is given as 16-bit data values on a 512 by 512 grid. A standard linear mapping of the relevant CT data to gray values is depicted in figure 2. Note, that this image could be produced by picking the two boundary values to define an affine map.
In figure 2 the soft tissue is display relatively bright. We can adjust this for a better distinction of bones and soft tissue by simply seleting one of the data values from the soft tissue and assigning a dark gray to it. This time an affine mapping is a bad choice, because the three correspondences cannot be satisfied. Instead, we fit the mapping globally to the data value - gray value pairs by using radial basis sums with the shifted logarithm as the radial function. The resulted is depicted in figure 3 and clearly shows the advantage in comparison with figure 2.
If we take a closer look at figure 3 we find a brighter substructure in the stomach. We would like to bring this region of data values to better attention in the visual representation. We do this by mapping a data value of this region to a red color. That is, instead of using gray values in the visualization we now use RGB color. Note, that it is not necessary to use specific two-dimensional color scales: We simply specify which data value maps to which RGB triple. The gray value

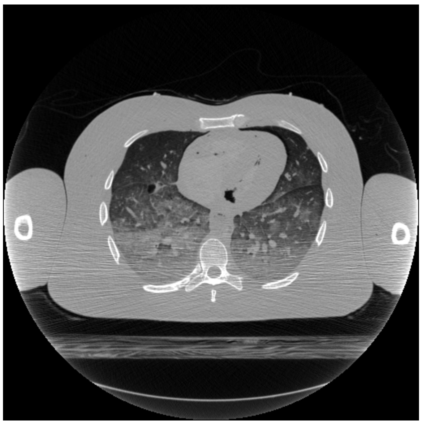

Fig. 2. The CT-scan of the chest of a man. This image is generated from the raw CT-data by linearly mapping the range of usefull CT-data values to a greyscale

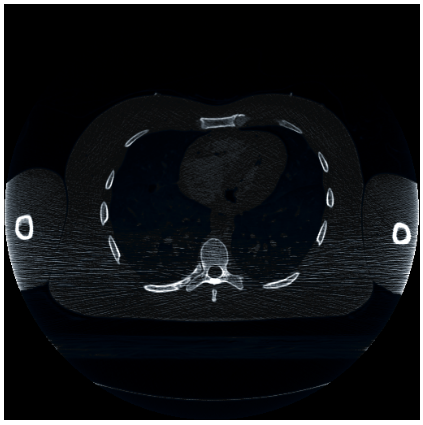

Fig. 3. Here, the CT scan was generated by a mapping defined from three correspondences. The background was mapped to black, the bones were mapped to white, and the soft tissues surrounding the lung were mapped to dark grey.

representations of the three correspondences defined earlier are mapped onto corresponding RGB values. The resulting mapping is shown in figure 4 in the color section. Note, how the empty structures are colored in the complementary color of red. This gives a nice distinction of empty spaces and tissues.

6 Conclusion

In this paper we presented a new approach to the construction of visual scales for the visualization of scalar and multivariate data. Based on the specification of only a few correspondences between data values and visual representations, complex visualization mappings are produced, hereby introducing a Visualization by Examples.
This approach exploits the user's knowledge about the data in a more intuitive way. Moreover, the user is enabled to adapt the visualization interactively and easily. The technique of Visualization by Examples can be used in combination with any visual representation.

References

1. Marc Alexa and Wolfgang Müller. The Morphing Space. Proceedings of the WSCG '99, Plzen, Chzech Republic, 1999
2. Marc Alexa and Wolfgang Müller. Visualization by Metamorphosis. IEEE Visualization '98 Late Breaking Hot Topics Proceedings, 33-36, 1998

3. Nur Arad and Daniel Reisfeld. Image Warping Using few Anchor Points and Radial Basis Functions. Computer Graphics Forum, 14, 1, 23-29, 1995
4. J. Bertin. Semiology of Graphics, The University of Wisconsin Press, 1983
5. Richard Boyer and David Savageau. Places Rated Almanac. Rand McNally, 1985, also available from http://www.stat.cmu.edu/datasets/places.data
6. Ken Brodlie. A classification scheme for scientific visualization. in: A. Earnshaw and D. Watson: Animation and Scientific Visualization - Tools and Applications. Academic Press Ltd., London, 1993, pp. 125-140
7. Andreas Buja, John Alan McDonald, John Michalak, and Werner Stuetzle. Interactive Data Visualization using Focusing and Linking. Proc. IEEE Visualization '91, Boston, 1991, pp. 156-163
8. H. Chernoff. The use of faces to represent points in k-dimensional space graphically. Journal of American Statistical Association, 63, 361-368, 1973
9. William S. Cleveland. The Elements of Graphing Data, Wadsworth Advanced Book Program, Pacific Grove, 1985
10. Nira Dyn. Interpolation and approximation by radial and related functions. Approximation Theory VI, Vol. 1, Academic Press, 1989, pp. 211-234
11. Gene H. Golub and Charles F. van Loan. Matrix Computations. The Johns Hopkins University Press, Baltimore, 1983
12. Wolfgang Müller and Marc Alexa. Using Morphing for Information Visualization. ACM Workshop on New Paradigms in Information Visualization and Manipulation, Bethesda, 1998
13. Ronald M. Pickett and Georges G. Grinstein. Iconographics display for visualizing multidimensional data. Proceedings IEEE Conference on Systems, Man, and Cybernetics, 514-519, 1988
14. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical Recipes. Cambridge University Press, Cambridge, 1992
15. Penny Rheingans. Color, Change, and Control for Quantitative Data Display. Proc. IEEE Visualization '92, Boston, 1992, pp. 252-259
16. Detlef Ruprecht and Heinrich Müller. Image Warping with Scattered Data Interpolation. IEEE Computer Graphics and Applications, 15, 2, 37-43, 1995
17. Edward R. Tufte. The Visual Display of Quantitative Information. Graphics Press, Cheshire, Connecticut, 1983