SIGRAD 2015
L. Kjelldahl and C. Peters (Editors)

Analysis of Visual Arts Collections

H.Pflüger & T.Ertl

Institute for Visualization and Interactive Systems (VIS), University of Stuttgart

Abstract

In this paper, we introduce a projection technique that aims to place points representing individual images in a two-dimensional visualization space so that proximity in this space reflects some sort of similarity between the images. This visualization technique enables users to employ their visual ability to evaluate the significance of metadata as well as the characteristics of classification methods and distance functions. It can also be used to recognize and analyze patterns in large sets of images, and to get an overview of the entire body of pictures from a given set. The projection technique only uses a similarity function for calculating a suitable distribution of the points in the visualization space and has a linear time complexity.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Systems]: Information Search and Retrieval—all subtopics Picture/Image Generation [I.3.3]: Viewing algorithms—Clustering [I.5.3]: *–classification*

1. Introduction

Our aim is the development of techniques which facilitate the search, exploration, and general structuring of large collections of works of visual arts. To this end we are searching for methods which can calculate and visualize relationships between pictures. The similarity between images in terms of human perception is obviously an important relationship between pictures, and there are projection techniques that place points representing individual data instances in a two-dimensional visualization space so that proximity in this space reflects some sort of similarity. These techniques support many analysis tasks for data sets. We use an algorithm that is well suited to calculate the similarity of pictures of visual arts in terms of human perception. However, this algorithm generates no vector representation reflecting similarity, which is the basis of most of the common projection techniques. Common methods are based on a given vector representation of the data or their time complexity is at least quadratic. Our proposed method has linear time complexity and is thus applicable to very large sets of images. Furthermore, it only requires a distance function for calculating the distribution of the points.

2. Similarity of Images

We use the algorithm given in [PHRE15] to calculate similarity between pictures. The algorithm is based on the assumption that fixations during the perception of visual arts, along with their surroundings, constitute important image information both for recognizing and for comparing pictures (more information in [HS95], [Saa93], [JC80], and [HNA*10] Chapter 3.2). The algorithm calculates a sequence of 100 fixations per image. The positions and the image information within a radius of 32 pixels of the simulated fixation points of two pictures are the basis for calculating 18 lokal comparison features (more information in [PHRE15] and [Aly11]).

The assumption underlying the method is that all features have an impact on how people perceive similarity, but that it is not known how strong this influence is. The method deals with the issue of choosing appropriate weighting factors by implicitly performing a weighting: To calculate the similarity between a picture P and a number of other pictures, the method first calculates all comparison features $x_{i,j}$ (i: feature number; j: picture id) between the picture P and the rest of all available pictures. Next, the mean value AM_i and the standard deviation σ_i is calculated for each of the comparison features. Now, the similarity between Picture P and another picture is regarded as the sum of normalized comparison factors $\tilde{x}_i$: $\tilde{x}_i = (x_i - AM_i)/\sigma_i$. This approach takes the range of the similarity features into account, and normalizes the variance of the distances between all objects and a single object to 1.

The evaluation of the similarity function in [PHRE15] revealed that the method is capable of identifying those pictures in a set that are similar to a given picture in terms of human perception; nevertheless, not every picture that the calculations determined to be similar to a given picture was perceived as being similar by human viewers. As a distance function, the method has some drawbacks, because for every picture in a set of pictures the comparison factors are calculated/weighted individually: The function is not symmetric; the function does not obey the triangle inequality; in order to calculate the similarity of two pictures, it is necessary to compare one of the two pictures with all the pictures in the set. However, the projection technique presented in this paper is designed to handle these problems with linear time complexity.

3. The Projection Technique

For the intended projection technique there is only a distance function available and the technique should have linear time complexity. The methods commonly used are not suitable for this case: One class of similarity-based projection techniques are the methods of force-directed placement (see e.g. [Ead84], [Cha96], and [TMN03]). The fastest of these methods has a time complexity of $O(n^{5/4})$ [MC04] and would be fast enough for our purposes, but all methods with a time complexity faster than $O(n^2)$ work on the basis of a given distribution of the points in a multidimensional space, which is not available in our case. Another similarity-based projection technique is t-SNE [VdMH08], a variation of Stochastic Neighbor Embedding [HR02]. The technique keeps the low-dimensional representations of very similar data points closely together, which is advantageous for our purposes. Optimized versions have linearithmic time complexity [vdM14], but again, all methods with a time complexity faster than $O(n^2)$ work on the basis of a given distribution of the points in a multidimensional space. Another class of methods is focused on preserving cluster structures (e.g. in [CBP09] and [CLRP13]), which would be desirable for our purposes. However, they need either a given distribution of the points in a multidimensional space, or the distance of all pairs of points, which results in quadratic time complexity. Another drawback all procedures described above have in common is that these algorithms are iterative; thus, the results vary strongly with changing starting positions or if new objects are added. Therefore, the results are not stable.

The idea of our method consists of several techniques: assuming that the objects in question are represented by points in some unknown multi-dimensional space in which the Euclidian distance between the objects is proportional to the similarity of the objects; using principal component analysis (see e.g. [CC08]) to get the first two principal directions; building a plane space with these directions; and projecting the points into this space. The principal component analysis is a commonly used approach to reduce a multi-dimensional problem to a two-dimensional problem. As there is only a distance function available, and we want to avoid quadratic time complexity, we have to create a suitable approximation solution.

- As an approximation for the first principal direction we construct a line AB through the points A and B with the greatest distance. To calculate the points with the greatest distance, n^2 distance calculations are necessary (n is the number of objects), so we calculate these points with an algorithm that, again, is only an approximation but has linear time complexity: We start with an arbitrary object and calculate the object with the greatest distance to this point. Then we once again calculate the point that is farthest from the object just calculated. We take the last two points as points A and B, which have approximate maximum distance to each other (see [OK89]).
- In order to project all objects O_i orthogonally to line AB, we calculate the similarity distances a_i, b_i, and c (see Figure 1). We take the line AB as our first dimension for the visualization space with A as the origin, and define $x_B = 1$. The coordinate value x_i can be calculated as follows: $x_i = a'_i/c$ where $a'_i = \sqrt{(a_i{}^2 + c^2 - b_i^2)/2c}$.
- As an approximation for the second principal direction we consider the line RS through the two points with the greatest distance, measured orthogonally to line AB. R is the point with the greatest orthogonal distance $h_i = \sqrt{a_i^2 - a_i'^2}$ to line AB (see Figure 1). The point with the greatest distance $r'_i = \sqrt{r_i^2 - (a'_r - a'_i)^2}$ to R (see Figure 1) is taken as S.
- We take the line RS as our second dimension with R as the origin, and define $y_S = 1$. The coordinate value y_i for the second dimension is calculated analogously to the first dimension.

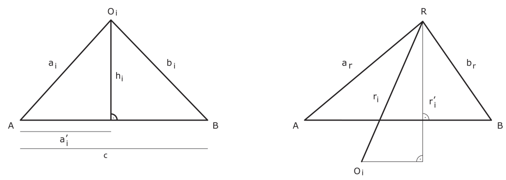

Figure 1: *Geometric relationships.*

The distance function does not obey the triangle inequality. It is possible that $c > a_i + b_i$, $a_i > c + b_i$, or $b_i > c + a_i$, the last two cases only in the second direction. In these cases the coordinate values cannot be calculated in the manner described. In order to get a suitable value for x_i in this cases, we increase or decrease the distances a_i and b_i with the factor k so that equation $c = ka_i + kb_i$ is satisfied, and can now calculate x_i with the values ka_i and kb_i. Having y_i we proceed analogously. The entire method has linear time complexity. In particular, the weights of the comparison factors must be determined only for the 5 objects A, B, R, S and the start object of the iteration that calculates A and B. The points A, B, R, S can be retained, if only a small number (compared to the total number) of images are added or removed. In this case, the distribution of the points is maintained.

In order to evaluate our method we used a corpus of visual arts containing about 4,000 pictures. Large values in the calculation of image distances mean only that the corresponding image pairs are dissimilar, and distances with values above 0.5 (on a scale from 0 to 1) cannot be meaningfully distinguished. Therefore, we attach great importance to good correlation only for small distances. Thus, we calculated the sample correlation coefficient not only for all object pairs but also for pairs within a certain distance range. The sample correlation coefficient for all object pairs was 0.42; for pairs with a distance of less than 0.5 the coefficient was 0.54; and for pairs with a distance of less than 0.3 the coefficient was 0.63. The relative deviation of the Euclidean distance from the value of the distance function was 0.4 for all pairs, 0.33 for pairs with a distance of less than 0.5, and 0.3 for pairs with a distance of less than 0.3.

These values show that the presented point placement method presented here largely preserves the similarity between the objects. Together with a computation time of about one second for 4,000 pictures (we use an Intel i7-2600 processor with 3.4 GHz and 16 GB computer memory) and linear time complexity, the presented method is therefore well suited for the task at hand.

4. The Display

The display is divided into three linked views (Figure 2). The size of the views depends on the size of the top left visualization space, which is variable in size. The top right view shows those pictures which are in the light gray area of the visualization space. The picture displayed in the top left corner of this view, however, is the one next to the mouse cursor in the visualization space. The bottom view shows collected pictures, which can for example be stored or used to build up clusters. Detailed information about each image is available: A double-click on an image opens a window that displays information such as metadata, linked texts, or detail shots.

5. Visualization of Patterns

Metadata, such as the names of the artists, the pictures' titles, their years of origin, painting techniques, and sizes, help to structure large amounts of images. The points in the visualization space can be colored according to user selected metadata thus showing their distributions (Figure 3). That way, the significance of the selected metadata can be easily understood. In contrast, when the significance of metadata is known, the characteristics of the similarity function that was used can be tested in this way. When a classification algorithm is used for the coloring of points, the visualization space shows the characteristics of the classification algorithm.

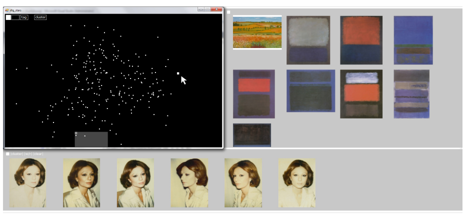

Figure 2: *The main Display.*

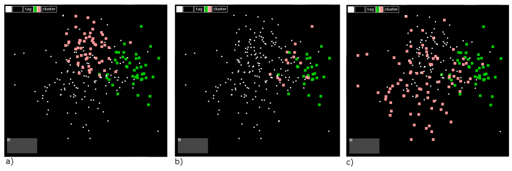

Figure 3: *(a) Landscapes (green dots); portraits (red dots). (b) Pictures painted by Hockney before 2008 (green dots); painted after 2007 (red dots). (c) Pictures painted by Hockney (green dots); pictures painted by Rothko (red dots).*

6. Reducing complexity

In large image collections, pictures are often featured several times with only small variations. Therefore, it seems only natural to visualize similar images only through one representative, which is why we included an agglomerative cluster method [KR08] to find those similar images:

- For each object pair, we apply the distance function in both directions and calculate the Euclidian distance of the images in the visualization space. Of the three distance values, we take the largest. The Euclidian distance must be adapted to the values of the distance function. We do this by multiplying it by a factor. This factor must be determined experimentally, which in our case was not problematic. We used the same factor for all sets of pictures. In contrast to the distance values of the distance function the Euclidean distance also takes the distance to all nearby objects into consideration, and, in all calculations, the results were better when the Euclidean distance was taken into account.
- We sort the object pairs according to their distance values.
- Starting with each object as the potential core of a cluster, we merge the clusters. To do so, we set the maximum distance value and progress through the ordered list of pairs of objects starting with the pair with the lowest distance value. We merge two clusters if the distance values of all pairs of the cluster members are less than the maximum value.
- We begin with a maximum distance of 0.01 and repeat the previous step while always increasing the maximum distance value by 0.01. We end the procedure once the maximum distance value has reached a specified limit.

We tested the cluster method on the whole corpus of 4,000 pictures as well as on a subset of 232 images. Applied to the small subset with the maximum distance set to 0.3, 44 clusters were created and they were chosen in a way that no subsequent manual improvement was necessary (Figure 4). If the maximum distance was set to 0.5, the number of clusters was cut in half, but in some clusters images were grouped together that did not fit well in terms of human perception. Applied to the set with 4,000 pictures with a maximum distance of 0.3, the results were still good, but not as good as in the previous case. The number of clusters was in this case about one third of the number of images, and could not be significantly reduced with manual rework.

The problem with the method presented here is that its time complexity is quadratic. The sorting has linear time complexity, because we do not have to sort precisely, but only have to assign the pairs to 100 distance classes (0.0 - 0.01 to 0.99 - 1.0). But we have to calculate the distance for every pair of pictures. The computation time for clustering the corpus of 4,000 images is about two minutes. Therefore, the method is only suitable for image sets with a size of several thousand pictures.

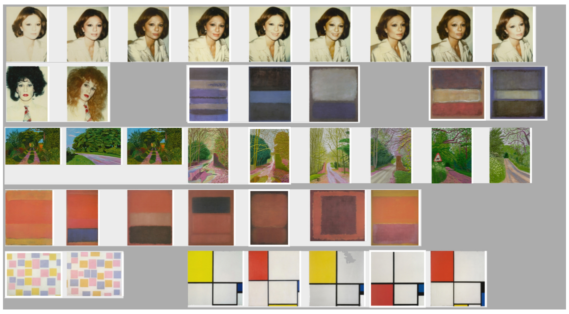

Figure 4: *Some typical clusters calculated with our cluster method applied to the set of 232 pictures. Clusters are separated by larger spaces.*

7. Conclusion

The evaluation shows that the presented projection technique preserves similarity to a high degree. Because it also has a linear time complexity, the technique is very suitable for the task at hand. The examples show that the similarity function used is well suited to identify similarities (e.g. the same artist, style, or subject matter). Nevertheless, we consider the provided method only as a first step towards an analysis tool for large image sets. The time complexity of the clustering method has to be improved without losing its current characteristics, so that the method can be applied to very large sets of images. However, our focus in the development of the tool is on the integration of the user into the analytical process. With visual analytics we want to include the user's knowledge and requirements in the design of the similarity function, the clustering process, and in the detection and visualization of structures.

References

[Aly11] ALY M.: *Searching Large-Scale Image Collections*. PhD thesis, California Institute of Technology, 2011. 1

[CBP09] CHOO J., BOHN S., PARK H.: Two-stage framework for visualization of clustered high dimensional data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on* (2009), IEEE, pp. 67–74. 2

[CC08] COX M. A. A., COX T. F.: *Handbook of Data Visualization*. Springer Berlin Heidelberg, 2008. 2

[Cha96] CHALMERS M.: A linear iteration time layout algorithm for visualising high-dimensional data. In *Visualization '96. Proceedings.* (Oct 1996), pp. 127–131. 2

[CLRP13] CHOO J., LEE C., REDDY C. K., PARK H.: Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *Visualization and Computer Graphics, IEEE Transactions on 19*, 12 (2013), 1992–2001. 2

[Ead84] EADES P. A.: A heuristics for graph drawing. *Congressus Numerantium 42* (1984), 146–160. 2

[HNA*10] HOLMQVIST K., NYSTRÖM M., ANDERSSON R., DEWHURST R., JARODZKA H., VAN DE WEIJER J.: *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2010. 1

[HR02] HINTON G. E., ROWEIS S. T.: Stochastic neighbor embedding. In *Advances in neural information processing systems* (2002), pp. 833–840. 2

[HS95] HOFFMAN J., SUBRAMANIAM B.: The role of visual attention in saccadic eye movements. *Perception & Psychophysics 57* (1995), 6: 787–795. 1

[JC80] JUST M. A., CARPENTER P. A.: A theory of reading: From eye fixations to comprehension. *Psychological Review 4* (1980), 329 – 354. 1

[KR08] KAUFMAN L., ROUSSEEUW P. J.: *Finding Groups in Data; Introduction*. John Wiley & Sons, Inc., 2008, pp. 1–67. 3

[MC04] MORRISON A., CHALMERS M.: A pivot-based routine for improved parent-fnding in hybrid mds. *Information Visualization 3* (2004), 109–122. 2

[OK89] ÖMER, KALANTARI B.: Approximating the diameter of a set of points in the euclidean space. *Inf. Process. Lett. 32*, 4 (Sept. 1989), 205–211. 2

[PHRE15] PFLÜGER H., HÖFERLIN B., RASCHKE M., ERTL T.: Simulating fixations when looking at visual arts. *ACM Transactions on Applied Perception 12*, 3 (May 2015). 1

[Saa93] SAARINEN J.: Shifts of visual attention at fixation and away from fixation. *Vision Research 33* (1993), 8: 1113–1117. 1

[TMN03] TEJADA E., MINGHIM R., NONATO L. G.: On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization 2*, 4 (Dec. 2003), 218–231. 2

[vdM14] VAN DER MAATEN L.: Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research 15* (2014), 3221–3245. 2

[VdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of Machine Learning Research 9*, 2579-2605 (2008), 85. 2