Sifting through Visual Arts Collections

H. Pflüger, T. Ertl[1]

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

Abstract

We introduce a visualization system for large image sets which combines a distance function, a clustering and a projection method. The distance function, the clustering and the projection methods run so fast that they can calculate new results during the interaction with the user and can therefore be adapted dynamically to the context of the investigation and the requests made by the user at any given moment. The system aims to facilitate investigations which take similarity between images in terms of human perception into account. Similarity in terms of human perception is highly context and task dependent and cannot be described by a metric in the mathematical sense. Functions reflecting similarity in terms of human perception have to be adapted dynamically to the context of the investigation as well as to the tasks assigned at any given time. Our system thus shows that these requirements can be met in principle, and we propose it as a basis for developing specific applications and suitable surfaces in collaboration with experts for whom such tools are useful, as for instance experts of art theory.

1. Introduction

The number of digital images people can access is enormous and continually increasing. Specialized Web portals such as Google or Flickr are offering billions of digital images. While limiting ourselves to works of visual arts reduces these myriads of images significantly, they still number in the range of several million. The image archive Prometheus (www.prometheus-bildarchiv.de), for instance, has more than one million digital images on display. It is obvious that on this kind of scale, no one human being is able to sift manually through such a number of images. These bodies of images either contain great redundancies, or one has to accept that one can only survey small selections of these bodies. In either case, one is left with the question of how to get an overview of the entire body of images and how to make that body tangible.

The theory of art provides ways of structuring quantities of images. For example, different styles help to classify bodies of images, both in terms of appearance and content. Additionally, image details such as the names of the artists, the images' titles, their years of origin, painting techniques, and sizes, help in the process of categorizing large bodies of images. Such annotations can help making structures and correlations accessible to users and may enable them to find samples suitable for their purposes. Formal metadata of the images (artists' names, titles of images, years of origin, painting techniques, and sizes) usually accompany the works' digital copies, and assignations of the styles and characteristics of the images are available for the most important and best known works of art. However, assigning works of art and choosing representative artworks require ex-

[1]Authors addresses: H. Pflüger, and T. Ertl, Institute for Visualization and Interactive Systems (VIS), University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany; emails: {Hermann.Pflueger, Thomas.Ertl}@vis.uni-stuttgart.de.

pert analysis and are therefore costly; and they fall short when one is dealing with the ever increasing total number of works of art. In addition, such assignations are most often ambiguous and depend on the issue at hand as well as on the context.

A system to analyze and explore large sets of images of visual arts should have the following properties.

- Similarity between images in terms of human perception obviously constitutes an important relationship between images, and we believe that it is essential in the study of works of visual art to consider this property. The computation of visual similarity is usually calculated by means of complex functions (section 3). Therefore, the image distances for clustering or visualization have to be calculated via a distance function and not via a metric of a feature space (section 2); in the process, metadata or features retrieved from image annotations or the image itself can be integrated into the distance function.

- There is no way a human being can examine all the individual elements in a large set of images. For this reason, we do not use individual images as objects of an investigation, but rather groups of images. The optimal number of images per group depends on the images which are clustered and on the specific conditions during the investigation. Large clusters may be taken if there are many similar images in the set, or if the task is to get an overview of the total amount. On the other hand, it is necessary to form small groups or even take single images when individual subsets are to be examined in more detail. Therefore, the clustering should be done dynamically.

- Similarity between images depends on the given context (e.g., paintings, drawings, photos, color or black and white images, figurative or abstract images), the task (e.g., getting an overview, exploring structures or relationships, or classifying tasks), and the current state of an investigation. The distance function and the visualization in general should therefore be adaptable during the investigation for any requirements that may arise at any given time of an investigation.

- The system should be a human-in-the-loop system and therefore follow the intuition, tasks, requirements, and precognition of the user, ensuring that that person be given incremental insight into a collection.

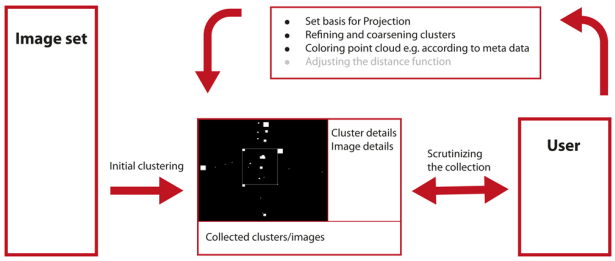

Figure 1: Schematic workflow of our visualization system.

The validity of distance functions simulating human perception is weak. Their dependence on the given context and tasks is too high to transfer them into a metric in the mathematical sense. In our opinion, tools for structuring works of art need to be supported by experienced users; the structuring process can only be done interactively with a human being involved in the system, which leads to the above listed conditions. To our knowledge, the usual methods are too sophisticated and costly for such tools. In our work, we present methods that meet the conditions described above: a distance function similar to David Lowe's algorithm SIFT [17], optimized for visual art (section 3); a simplified k-means++ clustering (section 4); and a 2D visualization reflecting similarity between images/clusters (section 5). In section 6 we show an interface that takes advantage of these methods and enables users to sift through large image sets and visualize structures given by metadata (Figure 1). We show that the methods are fast enough for the intended applications, that their results are precise enough for the intended applications (section 7 and 8), and we show in a user study that useful applications can be derived with the help of these methods (section 9).

We applied our system to a collection of about

5,000 images while staying fully interactive without any significant delay. The collection includes portraits, individuals or groups of people, architecture, landscapes, and abstract paintings, done in different painting and drawing techniques. We also conducted performance tests with up to one million images (frames of cell phone videos). For some of our evaluation test we used a set of 200,000 images (frames of cell phone videos); other evaluation tests run with much lower number of images because of a large computation time (see section 7 and 8). The user study was done with 676 images because for this study it was not feasible to conduct the manual clustering of more images without support that had to be done for comparison purposes.

We thus demonstrate that the proposed system is feasible in principle and that useful applications can be generated with it. Of course, this work represents only a first step and is likely to be regarded as a feasibility study. Specific applications and suitable surface designs must be developed in collaboration with experts from the field of art theory.

2. Related work

The analysis of large quantities of images usually consists of the following steps: Defining features that characterize the images; mapping the images in a feature space in which every feature defines a dimension (Figure 2); and analyzing structures of the data in the feature space.

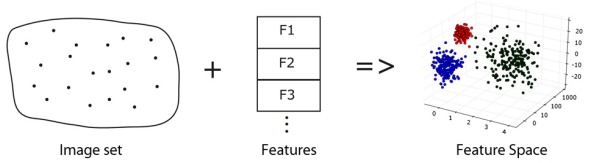

Figure 2: Images represented by features.

Images with identical feature values are considered identical, images with similar feature values are considered similar. The distance/similarity between two images can be easily calculated by using a metric of the feature space, as for example the Euclidean distance. This makes it easy to search for identical or similar images, or to find images with specific characteristics. Additionally, there are very efficient methods which use such distance metrics to cluster images, analyze relationships between features, and visualize image sets as point clouds to show structures and distances between the images (see, e.g., [1], [2], [3], [4]).

However, we do not consider the procedure we just outlined suitable for our purposes. The properties listed in the last section require high-speed clustering and fast similarity-based visualization on a 2D display. Time complexity has to be at least quasi-linear, so that interactivity is also possible for large image sets. There are many methods for clustering (see, e.g., [5]) and for distance-preserving 2D projections (see, e.g., [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]). However, all methods which are based on distance functions and have a time complexity faster than $O(n^2)$ work on the basis of a given distance matrix (which requires n^2 image comparisons) or a distance function that obeys the triangle inequality. In [16], Faloutsos shows a method for embedding data in a Euclidian pseudo space, which uses only a distance function and has only a linear time complexity, but the distance function used in this case has to obey the triangle inequality. Unfortunately, the result of distance functions calculating similarity in terms of human perception constitutes no metric in the mathematical sense. In particular, no exact statement can be made about the similarity between image A and image C if only the similarity between A and B and the similarity between B and C are known. We assume that if A is calculated as similar to B, and C as similar to B, then there are aspects that make A similar to C, but even this is not certain. It is possible to construct examples that disprove this. A monochrome red painting, for example, is similar to a portrait painted in red in terms of color. At the same time, this portrait is similar to a portrait drawn in charcoal because both images are portraits, even as the red monochrome image has nothing in common with the charcoal portrait. Large values in the calculation of image distances mean only that the corresponding image pairs are dissimilar and cannot be meaningfully distinguished.

The cluster and projection methods which employ

distance functions that are expected to obey the triangle inequality can possibly also be used when the underlying distance function does not satisfy the triangle inequality. It is, however, difficult to understand the outcome of this case; additionally, the highly efficient methods with a linear time complexity which means that the number of image comparisons is less than cn (n: number of objects; c: a constant factor) have factors c which are significantly larger than 1 (see, e.g., [6]), which is why these methods are still too slow for the intended use at hand.

3. Similarity between images

It is necessary to define some kind of relationship between images to find structures in sets of images, to build classes or groups of related images, and to find representatives for these classes and groups. Visual similarity between images is obviously an important category of relationship between images. Metadata or features retrieved from image annotations or the images themselves can be combined with these visual similarity features. We experiment with the use of the style and content of images cited in their annotations as well as with the limitation of the similarity comparison to relevant areas detected by means of object recognition; however, these are aspects that go beyond the scope of this work.

3.1. Similarity between images in terms of human perception

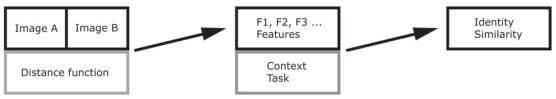

Figure 3: Schematic workflow for calculating similarity.

Methods for image recognition are usually based on David Lowe's algorithm SIFT [17] (Figure 3). These methods are used to determine characteristic areas, as for example, salient regions, and descriptors for these areas, such as local histograms. If two images have matching areas and descriptors, the images are considered to be equal (see e.g. [18], [19]). The scale of accordance can also be taken as a measure for similarity. Central to the comparison of images by means of this method is the determination of appropriate areas in the images. Here, mathematical methods aimed at identifying points with high salience, such as edge detectors, are typically used; the surroundings of these points are taken as characteristic areas of the respective image. A disadvantage of these methods is that they require about 1,000 characteristic areas per image to find an image in a large collection of images with a sizeable measure of reliability (see [18]), which in turn results in large memory requirements and high computation times. Another disadvantage is that the characteristic regions usually take only visual conspicuousness into account, as for example luminance or color contrasts, and do not consider what people actually perceive with their retinas and what is important for human perception (see e.g. [20]). Thus, these methods compare only image details and calculate similarity more theoretically than in terms of human perception.

3.2. The distance function

We use the distance function we presented in [21] for our calculation of similarity between images. 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 images (more information in [22], [23], [24], [25]). 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 images are the basis for calculating 18 features for a given pair of images. In addition, we now calculate 8 global features for pairs of images; these features are concerned with global characteristics of images, like color scheme, granularity, and complexity. In this way we get 26 similarity features for every pair of images where each feature takes different aspects of similarity into account. 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 performing a normalization which is implic-

itly a weighting: To calculate the similarity between a image P and a number of other images, the method first calculates all comparison features $x_{i,j}$ (i: feature number; j: image id) between image P and the rest of all available images. Next, the mean value AM_i and the standard deviation σ_i are calculated for each of the comparison features. Now, the similarity between image P and another image is regarded as the sum of normalized comparison factors $\tilde{x}_i = (x_i - AM_i)/\sigma_i$. 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. One could also use a weighted sum with learned weightings for a specific destination. However, this has not yet been applied.

Although our distance function does not explicitly use object recognition, high-level objects (e.g., faces, eyes) and mid-level objects (e.g., horizon line, simple geometric objects) are considered when comparing images. Subjects focus on high-level objects and mid-level objects more often than on simply salient regions that are not connected to objects involving cognition (see, e.g., [26]). For example, subjects often fixate on particular regions in a face displayed in an image (e.g., eyes, mouth, and ears). This is also the case with the simulated fixations. High-level and mid-level objects are therefore characterized by individual patterns (see, e.g., [27]), a fact which comes into play when comparing images with our distance function. When images are compared for similarity, one has to consider the context. In Figure 4a, color plays an important role. Therefore the group of colored images will be perceived as similar, as will the group of black and white images. In Figure 4b, color plays no role. Therefore, the distribution of the fixations is important. In this case, the faces that look to the right will be perceived as similar, as will the faces that look left. Our method for calculating similarity takes such aspects into account by considering the images in the current image set for weighting the similarity features, which is especially important for the comparison of images of visual art.

Using the proposed method for calculating similarity, we have to consider the following properties:

- In order to calculate the similarity of two images, it is necessary to compare one of the two images with all images that are relevant for the comparison, which generally is the whole set of images. If one then does compare one of the pair of images to the set, all distances of the chosen image to all images of the set are known. Therefore, the comparison of two images requires the same number of image comparisons as the comparison of one image with all images of the set, namely n comparisons, with n as the number of the images in the whole set.

- The result of the method is context and image dependent. Clustering and the projections should be done every time the context is significantly changed.

- The result of the method constitutes no metric in the mathematical sense. In particular, no exact statement can be made about the similarity between image A and image C if only the similarity between A and B and the similarity between B and C are known. However, we assume that if A is calculated as similar to B, and C as similar to B, then there are aspects that make A

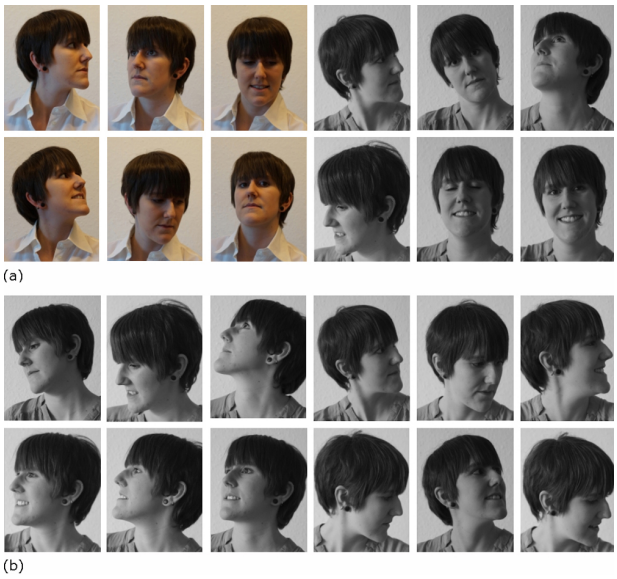

Figure 4: Context dependency in similarity reflections.

similar to C, but it is possible to construct examples that disprove this assumption. Large values in the calculation of image distances mean only that the corresponding image pairs are dissimilar and cannot be meaningfully distinguished.

Although the distance function can be used with any digital image, it is especially suited for images of visual art for the following reason: Our distance function analyzes salient regions in images, and common place images often contain elements that are not essential to the image content. If these elements are visually conspicuous, they affect our distance function negatively. Artists, however, paint major contents usually in a distinctive and salient way, and therefore, regions with important content usually coincide with visually salient regions. In this way our distance function especially considers substantive elements.

4. Clustering

There is no way for a human being to examine every single image of a large set of images. Therefore, we do not consider individual images as objects for our investigation, but rather groups of images. The optimal number of images per group depends on the images which are clustered, and on the circumstances of the investigation. Large clusters may be used if there are many similar images in the set or if the task is to get an overview of the total image set. However, forming small groups is required when individual areas are to be examined in more detail. Therefore, we start with coarse clusters and refine them according to the requirements of the users (Figure 1).

4.1. Building the initial set of clusters

To get initial clusters we start in a way similar to the k-means++ method by spreading k initial clusters:

- Randomly choose c as the first cluster representative. For each image x, compute $D(x)$, which is the distance between c and x. Set $C(x) = c$ for all x; $C(x)$ describes the recent cluster representative to which x is assigned.
- Choose the image with the maximum* value $D(x)$ for the next cluster representative c. For each image x compute $D'(x)$, which is the distance between c and x. For all x: if $D'(x) < D(x)$, set $D(x) = D'(x)$ and $C(x) = c$;
- Repeat the last step until k clusters have been chosen.

* This is different to the original method, and takes the characteristics of our distance function into account.

Now k images were selected as representatives for k clusters, and for each image x, $C(x)$ is the cluster representative to which x is assigned. The selection of clusters and the distribution of the images among them is already as good as the similarity calculation rendered by our distance function (see section 7). If the distance function can be improved - as, for example, by means of learning appropriate weighting factors of the comparison features of the distance function or using appropriate metadata - the clustering can be improved by applying a step of the k-mean iteration:

- For all clusters: calculate the distances between all cluster members with the normalization factors that are determined on the basis of the cluster members and choose as the new representative the image for which the sum of the distances to all cluster members is minimal.
- For all images: calculate the distance to all new representatives with the new normalization factors. Choose as new cluster the cluster to whose representative the distance is minimal.

Our initial clustering and the optimization step take into account that our distance function provides comparable values only for small values. Choosing which image belongs to a cluster requires only small function values. Large function values will only be used to ensure that the cluster representatives are not too close together. The initial clustering requires kn and the optimization step $n^2/k + kn$ image comparisons, where n is the number of images, and k the number of clusters. One image comparison takes

about 0.03 milliseconds computation time (we used an Intel i7-2600 processor with 3.4 GHz and 16 GB computer memory); therefore, the initial clustering for large image sets cannot be done while a user is in the process of investigating. The refining and coarsening described in the following section, however, can run during the investigation.

4.2. Refining and coarsening clusters

Every element in a set of images determines a scale on which every image within the set has a value calculated by our distance function. To divide a cluster we calculate the scale of A, which is the representative image of the cluster, and the scale of B, which is the image with the largest distance to A. Now we separate the set into two groups. One group contains the images that are closer to A, the other group those closer to B. To do this procedure, $2n$ image comparisons (n = the number of images) are necessary for the calculations of the distances to images A and B. If the algorithm were to divide the cluster into two equal parts - a feat that could be carried out without further (costly) image comparisons - this algorithm would be appropriate to carry out a clustering with $2n\ log_2 k$ image comparisons (n is the number of images in the set, and k the number of clusters). In the first step the initial set would have n images; in the second step there would be two sets, each with $n/2$ images; in the third step there would be four sets, each with $n/4$ images; and so on. Doing this, however, is unsuitable for clustering image sets because of the characteristics of our distance function. The scale spanned by the images A and B just reflects similarity between the images close to A or B. Clustering this way would be based on large values of the distance function and allocate images that are similar neither to A nor to B, which would lead to wrong decisions, because images cannot be meaningfully distinguished based on large distance values. However, if the set consists only of two groups of images, or of similar images, the algorithm is very well suited to refine the set. Therefore, we use this method to refine clusters.

Starting with the initial set of clusters, the user can refine the clusters stepwise with the method just described. This way, clusters are divided into two clusters until every cluster consists of only one image. Every step requires $2n$ image comparisons as described above, where n is the number of the images represented by the clusters. For large image collections this is too large a number to deal with during the investigation. Therefore, we divide only those clusters where the resulting clusters lie in the visible range of the visualization space. The representatives to be expected after dividing a cluster are already known beforehand; one is the representative image of the cluster that is to be divided; the other is the image with the largest distance to it, which has already been calculated while building the cluster that is to be divided. It is possible to construct cases where this approach may still lead to long waiting times during the examination of very large collections; however, when only a manageable number of clusters in the visualization space is selected, then no appreciable delay is caused by the refinement. To coarsen the set of clusters we start with the initial set of clusters and refine the set up to the level which is one below the current level. This leads to stable clusters if the similarity function is not changed, and even when the similarity function is changed during the investigation, the clusters of the lowest level remain stable. An advantage of this method for refining and coarsening the clusters is that it opens up the possibility of changing the similarity function during the investigation, a possibility we are planning to apply in the future.

5. Interactively adaptable projection

To help users to analyze and explore large sets of images we project image clusters into a 2D space. Our approach allows users to interactively adapt the projection to their analysis needs. This results in requirements for the projection method that are not met by current methods. We propose a new method that meets these requirements.

5.1. Requirements for projection

The projection technique should reflect the relationships among clusters which represent the images of a large set of images. Clusters that belong together are to be displayed closely together. However,

it cannot be expected that there is one optimal solution. The degree to which clusters belong together is defined by many features. The presentation of the clusters on a 2D display cannot exactly reflect their calculated distance even if the distances could be described by a metric in the mathematical sense. Additionally, the relationships among clusters are dependent on the context and the task at hand. Therefore, the distance function has to be adapted during the investigation. An obvious solution to this complication is to come up with different views. We thus have several requirements for the projection method in our case. The basis for calculating distances is a distance function that does not obey the triangle inequality. Therefore, the projection technique must be able to calculate the position of the clusters using only this distance function. Additionally, the calculations for the projection have to be fast enough to run during the investigation of the image set, and should have linear time complexity to remain fast even if a large number of clusters is on hand. In order not to confuse the user, the mapping function should provide stable solutions when clusters are being refined, and therefore, new objects have to be included.

The commonly used methods do not meet the required conditions: One class of similarity-based projection techniques are the methods of force-directed placement (see, e.g., [7], [8], [10]). The fastest of these methods has a time complexity of $O(n^{5/4})$ [10] 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 [11], a variation of Stochastic Neighbor Embedding [12]. The technique keeps the low-dimensional representations of very similar data points closely together, which is advantageous to our purposes. Optimized versions have linearithmic time complexity [13], 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. [14], [15]); preserving clusters would be desirable for our purposes. However, they require 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. In [16] Faloutsos shows a method for embedding data in a Euclidian pseudo space, which uses only a distance function, has only a linear time complexity, and would allow using sophisticated algorithms based on this Euclidian space. However, the distance function used in this case has to obey the triangle inequality, which is something our distance function does not do. The same problem exists when using multilevel algorithms (e.g. [6]). In the course of stepping from a coarse level to a finer one, it is expected that the distance function obey the triangle inequality or that it be able to calculate meaningful results for larger distances. Another drawback all these iterative procedures have in common is that the results are not stable if new objects are added.

5.2. The projection method

The following describes our method, which maps clusters as points in a two-dimensional space. The clusters are always represented by a image, and the distance between clusters is calculated by means of those images.

- We start with the randomly chosen cluster A. Cluster A determines a scale on which every cluster of the set has a value calculated by our distance function. As already mentioned, it is impossible to extrapolate a significant relationship between clusters on the basis of knowing only that they have the same, albeit large, distance to a specific cluster. In particular, it is wrong to conclude that two clusters with the same, though large, distance to a third cluster are similar. Thus we calculate cluster B with the greatest distance to A. We now have a one-dimensional scale with cluster A on one side and cluster B on the other.

- To account for clusters that are very different from both clusters A and B, we determine two other clusters, namely, C and D. To this end, we take from all images the distance d_{AB}, which is the smallest of the two distances to A and B; for C we choose the image with the greatest distance

d_{AB}. For cluster D we take the cluster with the greatest distance to C.

- We now form a square with the points A, B, C, and D as vertices. Clusters A and B form one diagonal, clusters C and D the other. When the clusters of the set are projected into this square according to their distances to clusters A, B, C, and D, all clusters in the center of the square have great distances to the clusters in the vertices. However, the significance of the similarity value decreases with the magnitude of the distance. We therefore determine one more cluster, E, which we place in the middle of the square. To this end, we take from all images the distance d_{ABCD}, which is the smallest of the four distances to A, B, C, and D; for E we choose the image with the greatest distance d_{ABCD}.

Into the square formed by clusters A, B, C, D, and E (Figure 5), we project the whole set of clusters with the following mapping function:

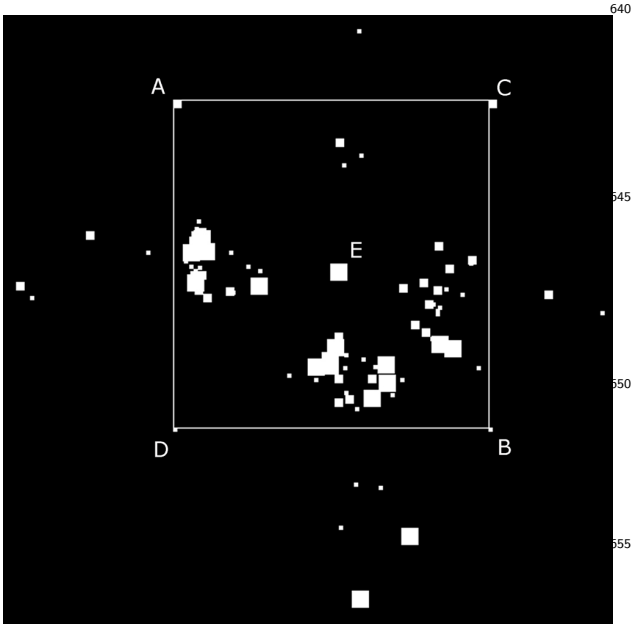

Figure 5: Point cloud in the visualization space with fixed clusters A, B, C, D, and E.

- We distinguish four cases. In case I, the cluster that has to be projected is closer to cluster A than to cluster B, and closer to cluster C than to D. In case II, the cluster to be projected is closer to cluster A than to cluster B, and closer to cluster D than to C. In case III, the cluster to be projected is closer to cluster B than to cluster A, and closer to cluster C than to D; and in case IV the cluster is closer to cluster B than to cluster A, and closer to cluster D than to C.

- For every cluster P, cluster E and the two clusters significant in each case determine the location of the cluster in the square ACDB. In case I, these significant clusters are the cluster A and C. The position of cluster P is calculated relative to the position of the significant clusters and cluster E on the barycentric coordinates which in case I are a, c, and e:

  $a = 1 - d_{PA})/(3 - d_{PA} - d_{PC} - d_{PE}$

  $c = (1 - d_{PC})/(3 - d_{PA} - d_{PC} - d_{PE})$

  $e = (1 - d_{PE})/(3 - d_{PA} - d_{PC} - d_{PE})$

  d_{PA}, d_{PC}, and d_{PE} are the distances of cluster P to clusters A, C, and E calculated by our distance function.

  In the other three cases, we proceed analogously. The positions of clusters A, B, C, D, and E are set; the positions of the remaining clusters are derived from their calculated distances to these fixed clusters.

- If a cluster is outside square ACDB, the barycentric coordinates cannot be uniquely deduced from the distance values of our distance function. In addition, for clusters outside square ACDB, the distance values are quite large, so that their significance is relatively low. We check whether a cluster is outside square ACDB, and calculate the coordinates in this case as follows below, which leads to a clear positioning of these clusters. If, for example in case I, $d_{PA} + d_{PC} + d_{PE} < 2$, we assume that cluster P is inside triangle AEC. Otherwise we assume that P is outside the triangle. The coordinates

in the visualization space are then calculated as follows:

$$\tilde{e}_P = ((d_{PA} + d_{PC} + d_{PE}) \ / \ 2)$$

$$w_1 = x_E + (x_A - x_E)\ \tilde{e}_P$$

$$w_2 = x_E + (x_C - x_E)\ \tilde{e}_P$$

$$w_3 = (y_A + y_C) \ / \ 2$$

$$x_P = w_1 + (w_2 - w_1)\ d_{PA} \ / \ (d_{PA} + d_{PC})$$

$$y_P = y_E + (w_3 - y_E)\ \tilde{e}_P$$

x_A, x_C, x_E, x_P, y_A, y_C, y_E, and y_P are the Euclidean coordinates of the clusters A, C, E, and P in the visualization space.

In order to adapt the projection of the clusters to context and task, we use suitable subsets of clusters to calculate the fixed points. Users can choose among three options:

- The user can select all initial clusters. This allows him or her to get a stable outline. This projection defines a stable position of all active clusters in the visualization space.
- The user can select all clusters within the viewable range of the visualization space. In this case the projection reflects the relationship among the chosen clusters better than in the preceding case, and it brings together clusters that were not previously mapped together but belong together in the specific context. For example, a portrait painted in red is thus mapped both in the category of portraits and in the category of red images.
- The user selects user-defined clusters. In this way, he or she determines the context for the similarity consideration by means of a sample quantity, with the same advantages as in the case before.

The user can change the subsets and thus the fixed points at any time during the investigation and can thereby control the projection. The projection error increases considering all relations if only a subset is taken as the basis. However, the projection error decreases in this case if only the relations related to the selected subset are considered. In Figure 6 the projection is based on the chosen images in the bottom view. In this projection all images (the image set taken in section 9) dissimilar to the chosen images were projected at a large distance. Thus they were outside the chosen view.

The distance function could also be adapted to requirements arising during the investigation, although this possibility is not currently supported.

Figure 6: Projection based on the subset of images in the bottom view.

6. The display

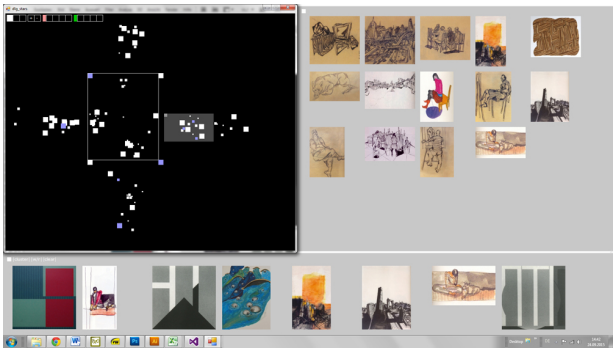

Figure 7: The main display. The collected images (bottom view) are colored blue in the point cloud.

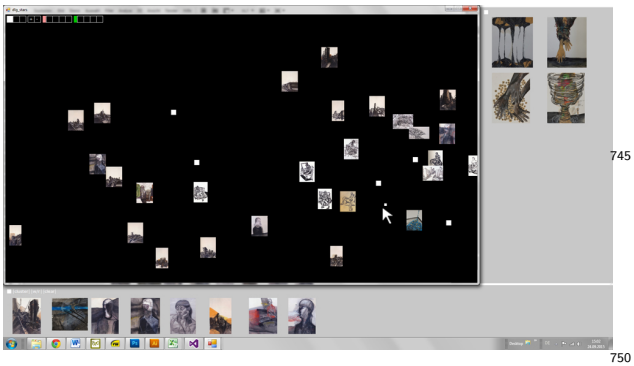

Figure 8: The main display after refining the clusters three times and zooming. The mapping function was calculated by means of the collected images (bottom view). The top right view shows the images of the cluster nearest the mouse cursor.

The display is divided into three linked views (Figures 7 and 8). The size of the views depends on the size of the top left visualization space, which is variable. The visualization space shows clusters as point clouds. The size of the points depends on the number of images. The smaller the number of images that are in a cluster, the greater is the point. If the cluster has only one image, this is shown as a small thumbnail. The idea behind this view is the concept that the smaller the number of images in a cluster is, the closer the viewer is to the clusters and the larger they therefore appear to him or her. The scale is adjusted so that no more than 2% of the area is covered with dots. The visualization space contains a gray area. Holding down the left mouse button, a user can drag this area across the screen. The representatives of all clusters within the area are shown in the top right view. With the right mouse button, the gray area can be changed. When the gray area is not selected, all images of the cluster closest to the mouse cursor are shown in the top right view. A double-click on an image in the top right view opens a window that displays information such as metadata, linked texts, or detail shots. The bottom view shows images collected by the user, which can be stored, used to build user-defined clusters, or used as the basis for calculating a new projection. The top of the visualization space features buttons the user can employ to control various actions:

- There are three buttons to calculate a new projection. Either all initial clusters are used as a base that is the starting situation; or all clusters that are in the visualization space are used; or all clusters collected in the bottom view are used.
- With two other buttons, the clusters may be refined or coarsened.
- With another button, the points in the visualization space can be colored according to user-selected metadata, showing their distribution in the visualization space (Figure 9). 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 manner. When a classification algorithm is used for the coloring of points, the visualization space shows the characteristics of the classification algorithm. During the selection process, it is possible to specify if only one image of the cluster, the representative image, all images, or the majority of the images must comply with a request.

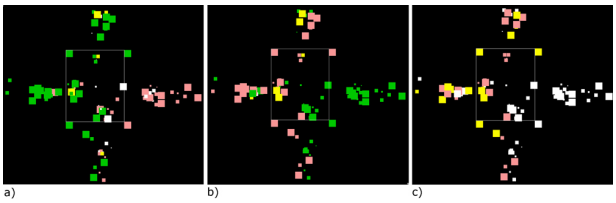

Figure 9: For these examples, the selection criterion majority was used. A point that was assigned both red and green was rendered yellow. a) Abstract Paintings (green dots); portraits (red dots). b) Images painted by Weiran Wang (green dots); Images painted by Frieder Kühner (red dots). c) Images painted by Frieder Kühner before 2012 (green dots) and after 2011 (red dots).

7. Evaluation clustering

First we evaluated the results using the distance function. For all clusters we calculated the average

distance d among all images of the set and the representative of a cluster as well as the average distance d_c of the cluster images to the representative of the cluster. The ratio of the average distance d to d_c is a measure of how well the images of the cluster are separated from the rest of the image set. Figure 10 shows the results of clustering our art corpus of about 5,000 images. The corpus included portraits, individual or groups of people, architecture, landscapes, and abstract paintings, done in different painting and drawing techniques. The images represented different styles. The range ran from photorealistic to distorted and from abstract to constructivist images. Figure 11 shows some typical clusters calculated by our cluster algorithm. Figure 12 shows the results of the clustering of 197,000 frames of 184 cell phone videos collected by Eva Paulitsch und Uta Weyrich (http://www.pw-video.com). Videos contain many highly similar images in succession; therefore, the factor d/dc is much better with video images. We also used the results of clustering the frames to evaluate the clusters in another fashion. Taking advantage of the usual high similarity between two successive images in a video, we counted the contiguous images of the longest sequence for each cluster. The percentage p of this count can be considered as a measure of the compactness of the clusters. Figure 13 shows the percentage p of our clustering, with p as a function of the number of clusters.

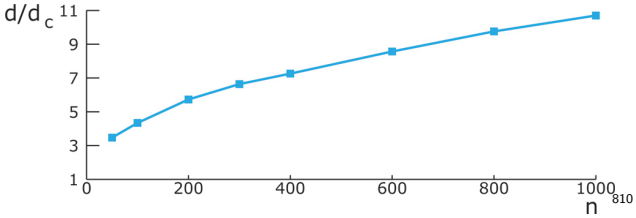

Figure 10: Average of factor d/d c as a function of the number of clusters (with images of our art corpus).

The evaluation shows that our clustering can form coherent groups of images. In the process, the smaller the selected groups are, the more compact and better defined they are. The evaluation by means of the video images shows that our clustering, together with our distance function, is actually capable of grouping

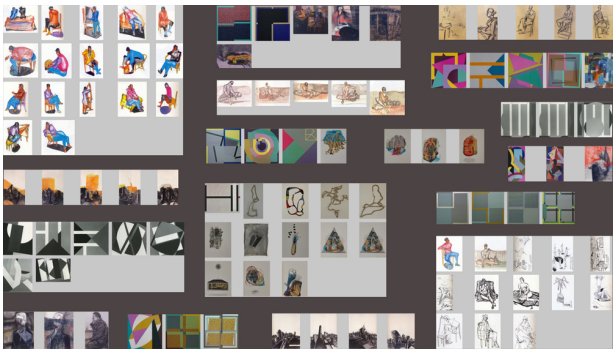

Figure 11: Some typical clusters calculated with our cluster method applied to a subset of our art corpus. The subset includes 634 images, and the number of clusters was 100. Clusters are separated by larger spaces.

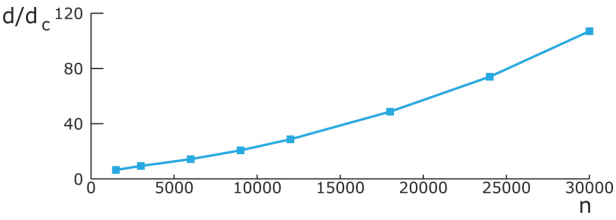

Figure 12: Average of factor d/d_c as a function of the number of clusters (with frames of several cell phone videos).

similar images. We assume that the limited nature of similarity functions makes more sophisticated clustering techniques unnecessary. The optimization step improved the results only marginally, and our user study (section 9) demonstrates that spectral clustering, together with calculating similarity with SIFT, did not lead to better results.

8. Evaluation - projection technique

The sample correlations coefficient between the calculated and the spatial distances served us as a measure of the quality of the projection. We based our distance between two clusters on the distances of their representatives calculated with our distance function. For the distance in the visualization space we took the Euclidean distance. Figure 14 shows the correlation factor as a function of the number of clusters. The results suggest that there is no direct re-

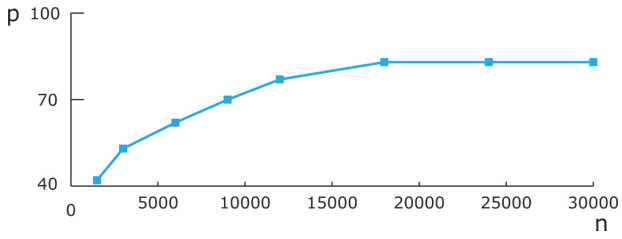

Figure 13: Average of p, which is the percentage of the number of images of the longest sequence in the cluster as a function of the number of clusters (with frames of several cell phone videos).

lationship between the number of clusters and the correlation factor.

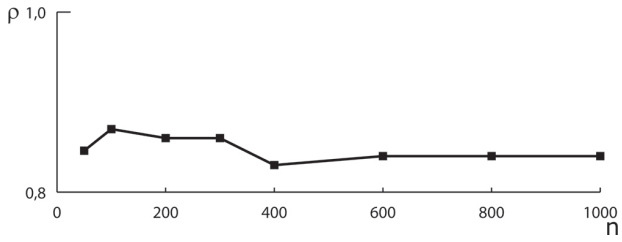

Figure 14: Sample correlations coefficient as a function of the number n of clusters.

As another way of studying the quality of our projection method we used a force-directed method to improve our results. After the projection of our method, we moved the positions of each cluster iteratively by a gradient descent algorithm which minimized the standard stress error $s(D,\Delta) = \sqrt{\sum_{i,j}(d_{i,j}-\delta_{i,j})^2 / \sum_{i,j}\delta_{i,j}^2}$, where $d_{i,j}$ are the Euclidean distances among the clusters, and $\delta_{i,j}$ the distances calculated by our distance function. The algorithm converged in all tests we conducted, which means that in all tests a local minimum had been reached. Figure 15 shows the sample correlation coefficient of the projection of 100 clusters, with the coefficient as a function of the number of iteration steps.

Figure 16 shows the standard stress error. The stress factor was calculated by using all cluster pairs in one instance, as illustrated in Graph I, and taking only the 25 % of the clusters pairs with the smallest

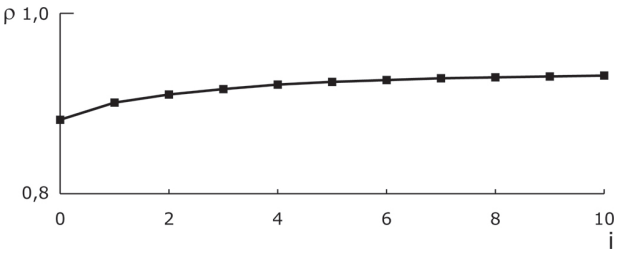

Figure 15: Sample correlations coefficient as a function of the number i of improvements. The number of clusters was 100.

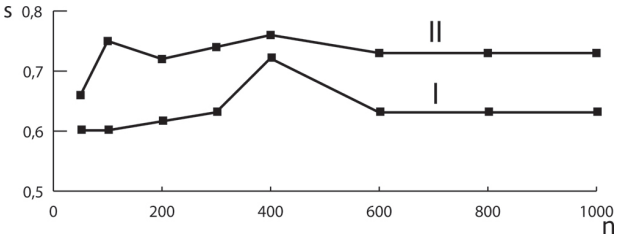

Figure 16: Standard stress error s as a function of the number n of clusters. In Graph I we used all cluster pairs; in Graph II we used only the 25 % of the clusters pairs with the smallest distance.

distance into account in the other instance, as illustrated in Graph II.

The stress error, which approximately describes the relative error between the calculated distance and the distance in the projection, is relatively high. Therefore, no exact match can be expected between the calculated distance and the Euclidean distance in the projection. However, the evaluation shows that there is a high correlation between the distance calculated with our distance function and the distance in the projection. That means that clusters that are similar according to our calculations are projected close to each other, and clusters that our calculations show to be dissimilar are projected with large distances between them. We can improve our results by using more sophisticated methods (see Figure 15); however, that requires a much higher computation effort. Because of their limited nature, the calculations of similarity functions can only approximate similarities in human terms. That is why we consider more sophisticated methods inappropriate.

9. User study

Art experts with whom we spoke confirmed that the classification of images based on their visual impact is an important task. However, it is also a very exasperating task to go through a large quantity of images and compare them individually in order to divide them into different classes, especially if there are no criteria to go by and if the images cover a wide range of image types. The user study described in this section shows that such a task can be supported with our similarity algorithm, and can even be automated to a certain degree. We used a collection of 676 images of visual arts. The collection included portraits, individual or groups of people, architecture, landscapes, and abstract paintings, done in different painting and drawing techniques. The images represented different styles. The range ran from photorealistic to distorted and from abstract to constructivist images. The task was to group the images into about 20 classes, so that the images of each class are similar to each other. The criteria for classification were not specified.

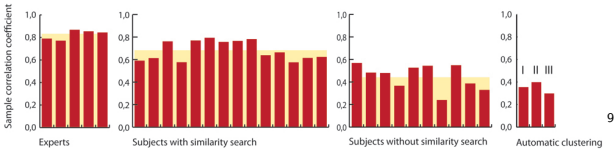

Figure 17: Sample correlation coefficients of the classifications made by the various participants and the automatically generated classifications. In the case of automatic classification: bar I depicts the value of our classification algorithm without optimization; bar II depicts the value of our classification algorithm with one step of optimization; bar III depicts the result of a clustering that calculated similarity with SIFT and clustered using spectral clustering.

In order to arrive at the classifications, we gave the participants the images in a uniform grid on a screen. The subjects could move the grid and zoom in and out. Via mouse click they could select images, choose a representative image, and create a group. After grouping the images, only the representative image was displayed. The remaining group elements were no longer shown; however, they could be reactivated and removed from the group, or new images could be added. Some of the participants were supported by a function that calculated a new order of the displayed images. They could choose an image, and subsequently the function placed all selected images next to the one chosen. The remaining images were displayed around the chosen and selected images according to their similarity to the chosen image; similarity was calculated by our distance function.

The study was carried out by three different groups. The first group consisted of five experts: four artists (Christine Gläser, Frieder Kühner, Angela Matthies, and Uta Weyrich) and an art historian (Stefan Borchardt). This group had no time limit and took 3-5 hours for the classification. They could make use of the similarity function to arrange the images by similarity and received individual support for handling the program interface. The other two groups were 18- and 19-year-old students of a class at a technically oriented high school. They were familiar with standard programs such as Photoshop, Illustrator or Flash. After 15 minutes of motivation and instructions on program handling, the students were divided into two groups (one group of 15 students and one group of 14 students) and then given 75 minutes to classify the images. The groups worked in separate rooms; they were permitted to help each other, but not to exchange data. One group could make use of the similarity function to arrange the images by similarity; the other group did not have this mode of assistance. There was one person to give both groups technical support. Four students of the group which could use the similarity function completed the task after 60 minutes. In the other group, 5 students resigned and gave up.

For purposes of evaluation we created a distance matrix. The values of the elements of this matrix were defined based on the results of the classifications made by the experts. The value of an element in this matrix was the number of experts who assigned the corresponding pair of images to the same class. We thus got a matrix (the average expert result) with the values 0, 1, 2, 3, 4, and 5. To evaluate both the classifications of the students and the automatically generated classification, we calculated the sample correlation coefficient between this distance matrix and the distance matrix we created from the

classification in question (containing the values 0 and 1). The basis on which to evaluate the classification of one of the experts was the distance matrix of the four other experts. Figure 17 shows the results of our evaluation. The sample correlation factors depicted in Figure 17 describe the correlations between the classifications of each participant and the average expert result. Considering the average expert result as the ideal depiction of togetherness, then these sample correlation factors reflect the quality of the respective classification. The experts had an average value of 0.82; the students that could use the assist function achieved an average value of 0.68; and the remainder of the students had an average value of only 0.44.

The experts all confirmed that the presentation of the images according to their similarity had greatly facilitated and accelerated the classification process, and that support of this type was necessary for the classification of very large numbers of images. Comparing the classifications made by the two groups of students shows that our distance function helps both to improve classifications significantly and to render them more homogeneous than they would be without this support. Additionally, it became apparent while watching the students at work that performing the classification with the help of the distance function was much easier than without. Without this help the students would have been overwhelmed if the number of images to be classified had been significantly higher.

In our opinion, the difference between the results of the group of students supported by the distance function and the expert group does not reflect any greater ability on part of experts to classify images. Rather, it shows that the experts used sophisticated and uniform criteria for classification, and that the students would have achieved similar results if the experts had explained the classification rules to the students before they performed the classification task.

The sample correlation coefficients for the automatic classifications are lower than the coefficients of the students. But experiments we carried out at the time have shown that the automatic clustering can be improved if it is trained for this task with appropriate training examples. If the automatic clustering is also supported – if, for example, a certain number of images that are difficult to classify and have to be identified by the clustering algorithm has been classified by experienced people – then even large numbers of images can be classified at reasonable cost.

10. Conclusion

It is obvious that people cannot capture and manage the enormous number of digital images they can access just by simply sifting and sorting them manually. The proposed visualization technique, however, shows that it is possible to make large numbers of images of visual arts tangible, analyzable, and manageable in their entirety with the aid of computer science and digital technology. Particularly in the field of art, coherences and structures in image volumes are strongly view- and task-dependent. Methods that depict images in a pre-defined feature space and then perform comparisons and create structures in this space are too rigid to effect this. Our system demonstrates that it is possible to realize an interactive user-controlled system that can be adapted to different contexts and tasks during work. The results of our clustering, the projection technique, and the calculation of similarity in general were, as far as we could observe in our tests, amazingly close to human perception and categorization. Standard methods for similarity calculation, clustering, and projection are not suitable for this task because they are either not suitable for the given conditions, or their performance is not sufficient for interactive work. The methods for similarity calculation, clustering, and projection presented here have been specifically developed for this task. Our evaluation has shown that these methods are fast enough for the intended applications, that their results are precise enough for the intended applications, and we have shown in a user study that useful applications can be derived with them. Thus we have demonstrated that the proposed system is feasible in principle and that useful applications can accordingly be generated with its help. Of course, this work represents only a first step, and is likely to be regarded as a feasibility study. Our future work aims to provide the user with more means for analyzing and structuring image corpora and to develop

specific applications and suitable surface designs in collaboration with art theory experts.

References

[1] D. Eler, M. Nakazaki, F. Paulovich, D. Santos, G. Andery, M. Oliveira, J. Batista Neto, R. Minghim, Visual analysis of image collections, The Visual Computer 25 (10) (2009) 923–937. doi:10.1007/s00371-009-0368-7. URL http://dx.doi.org/10.1007/s00371-009-0368-7

[2] L. Manovich, Data science and digital art history, International Journal for Digital Art History 0 (1). doi:10.11588/dah.2015.1.21631. URL https://journals.ub.uni-heidelberg.de/index.php/dah/article/view/21631

[3] G. P. Nguyen, M. Worring, Interactive access to large image collections using similarity-based visualization, J. Vis. Lang. Comput. 19 (2) (2008) 203–224. doi:10.1016/j.jvlc.2006.09.002. URL http://dx.doi.org/10.1016/j.jvlc.2006.09.002

[4] Y. Lacerda, H. de Figueiredo, C. de Souza Baptista, M. Sampaio, Photogeo: A self-organizing system for personal photo collections, in: Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on, 2008, pp. 258–265. doi:10.1109/ISM.2008.81.

[5] E. Pekalska, P. Paclik, R. P. W. Duin, A generalized kernel approach to dissimilarity-based classification, J. Mach. Learn. Res. 2 (2002) 175–211. URL http://dl.acm.org/citation.cfm?id=944790.944810

[6] S. Ingram, T. Munzner, M. Olano, Glimmer: Multilevel mds on the gpu, IEEE Transactions on Visualization and Computer Graphics 15 (2) (2009) 249–261. doi:10.1109/TVCG.2008.85. URL http://dx.doi.org/10.1109/TVCG.2008.85

[7] P. A. Eades, A heuristics for graph drawing, Congressus Numerantium 42 (1984) 146–160. URL http://ci.nii.ac.jp/naid/10000075358/en/

[8] M. Chalmers, A linear iteration time layout algorithm for visualising high-dimensional data, in: Visualization '96. Proceedings., 1996, pp. 127–131. doi:10.1109/VISUAL.1996.567787.

[9] E. Tejada, R. Minghim, L. G. Nonato, On improved projection techniques to support visual exploration of multidimensional data sets, Information Visualization 2 (4) (2003) 218–231. doi:10.1057/palgrave.ivs.9500054. URL http://dx.doi.org/10.1057/palgrave.ivs.9500054

[10] A. Morrison, M. Chalmers, A pivot-based routine for improved parent-fnding in hybrid mds, Information Visualization 3 (2004) 109–122.

[11] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (2579-2605) (2008) 85.

[12] G. E. Hinton, S. T. Roweis, Stochastic neighbor embedding, in: Advances in neural information processing systems, 2002, pp. 833–840.

[13] L. van der Maaten, Accelerating t-sne using tree-based algorithms, Journal of Machine Learning Research 15 (2014) 3221–3245. URL http://jmlr.org/papers/v15/vandermaaten14a.html

[14] J. Choo, S. Bohn, H. Park, Two-stage framework for visualization of clustered high dimensional data, in: Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on, IEEE, 2009, pp. 67–74.

[15] J. Choo, C. Lee, C. K. Reddy, H. Park, Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization, Visualization and Computer Graphics, IEEE Transactions on 19 (12) (2013) 1992–2001.

[16] C. Faloutsos, K.-I. Lin, Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets, SIGMOD Rec. 24 (2) (1995) 163–174. doi:10.1145/568271.223812.
URL http://doi.acm.org/10.1145/568271.223812

[17] D. G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99, IEEE Computer Society, Washington, DC, USA, 1999, pp. 1150–.
URL http://dl.acm.org/citation.cfm?id=850924.851523

[18] M. Aly, Searching large-scale image collections, Ph.D. thesis, California Institute of Technology (2011).

[19] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. van Gool, A comparison of affine region detectors, International Journal of Computer Vision 65 (2005) 1/2 43–72.

[20] R. J. P. W. Rodieck, The First Steps in Seeing, Sinauer Associates, Inc., 1998.

[21] H. Pflüger, B. Höferlin, M. Raschke, T. Ertl, Simulating fixations when looking at visual arts, ACM Trans. Appl. Percept. 12 (3) (2015) 9:1–9:20. doi:10.1145/2736286.
URL http://doi.acm.org/10.1145/2736286

[22] J. Hoffman, B. Subramaniam, The role of visual attention in saccadic eye movements, Perception & Psychophysics 57 (1995) 6: 787–795.

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

[24] M. A. Just, P. A. Carpenter, A theory of reading: From eye fixations to comprehension, Psychological Review 4 (1980) 329 – 354.

[25] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, J. van de Weijer, Eye tracking: A comprehensive guide to methods and measures, Oxford University Press, 2010.

[26] Y. Niu, R. M. Todd, M. Kyan, A. K. Anderson, Visual and emotional salience influence eye movements, ACM Trans. Appl. Percept. 9 (3) (2012) 13:1–13:18.

[27] P. Viola, M. J. Jones, Robust real-time face detection, Int. J. Comput. Vision 57 (2) (2004) 137–154.