EFFICIENT BROWSING IN IMAGE DATABASES USING A HIERARCHY OF KERNEL PCA SUBSPACES

Marcel Spehr[1], Frank Herrlich[1], Stefan Hesse[2] and Stefan Gumhold[1]
[1]*Technical University of Dresden - Computer Graphics and Visualization Lab - TU Dresden, Fakultät Informatik, Nöthnitzer Straße 46, D-01187 Dresden*
[2]*SAP AG - Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany*

ABSTRACT

We present a novel approach for designing the search functionality in large unlabeled image databases. It combines *Relevance Feedback*, *Hierarchical Browsing* and *Kernel PCA*, uses a *Mixture-of-Gaussian* to model feature space distributions and different visualization techniques of high dimensional feature spaces. Given an image database, finding a specific single or set of pictures is achieved by assisting the user to find an as-short-as-possible browsing path through the database. Our system relies on describing each picture with an appropriate feature vector that results from applying *Kernel PCA* to image and textual based similarity matrices. We solve the page-zero-problem by presenting the centroids of a hierarchical clustering in feature space as initial suggestions. The user can then steer the search by selecting positive and negative examples which define a *Mixture-of-Gaussian* density in the parameter space. New suggestions are drawn according to this density and the user is thus directed to the desired image category. A user study proved our system to be practical and beneficial for *category search* tasks.

KEYWORDS

CBIR, image similarity, relevance feedback, Kernel PCA, image features, semantic gap

1. INTRODUCTION

With the establishment of consumer digital photography private photo collections grow larger each year. Collections exceeding 10.000 images are common. This leads to the demand for new organization and navigation schemes to access the content of these image database management systems (*IDBMS*).

Programmers implementing an *IDBMS* face different problems. How to measure the similarity of images? Given limited screen size, which visualization techniques are most adequate to support searching and presentation of search results? How can user interaction be incorporated in subsequent search steps? Here we propose a combination of well known and established techniques to build an integrated system for efficiently browsing large databases of images.

We suggest a component based approach to give the system's developer as much flexibility as possible for implementing extensions. We also keep in mind that the final assembly of parts should hide the additional complexity from the user, who is just interested in the browsing capabilities. The components are responsible for (1) measuring similarity between images, (2) visualization of feature spaces, and (3) supporting the user's interaction. Figure 1(a) outlines (1) and (3) of our approach.

Component (1) derives appropriate feature descriptors for the images. Our system shall be as flexible as possible to be applicable in diverse application scenarios. We enable the user to supply a set of arbitrary image similarity measures as suit his needs. The resulting similarity matrices are henceforth processed in a *Kernel PCA* to achieve metric feature spaces that contain image descriptors. This procedure decouples the special case of a particular image database and use case from the rest of the search pipeline.

Visualization of feature space properties (2) is central to our needs. In our application scenario (see "Case Study" in section 4), the user is confronted with a large database of paintings and some more or less understandable abstract descriptors of them. The visualization component is responsible for conveying their meaning to the user in an understandable way. We implemented three data views: a) ordered regular grid, b) star charts, c) parallel coordinates.

There are two common ways of formulating a query to an *IDBMS* - either by text or a/several query image/s. In our scenario we want to refrain from using textual or label information to achieve a multipurpose system. Therefore we work with queries by example. The system presents suggestions from which the user chooses interesting and uninteresting images to approach his goal as R*elevance Feedback* (3).

The remainder of the paper starts with an overview of earlier work in the fields of *Relevance Feedback* and a discussion of ways to model high dimensional feature spaces as well as visualization techniques to navigate them. Then we start to describe our approach by first stating the system requirements that our solution is based on. The *Kernel PCA* algorithm that produces the image descriptors and our hierarchical browsing scheme is shortly introduced. Section 3.4 presents the way we map the user's intention to a parametrizable probability distribution in feature space using a *Mixture-of-Gaussian* model. We conclude the discussion of our work by introducing the visualization components that support the search task and shortly discuss the additional usefulness of the user's interaction data for supervised classification tasks. The final part of this paper presents a user study in which a database of 20.000 paintings and a set of 10 similarity measures is used in a *category search* scenario.

Our work's contributions are as follows. We designed a search system that distinguishes itself by being: I. Easily extensible by consistently treating arbitrary symmetric similarity measures. II. Scalable with respect to several of its components. And III. Flexibly applicable with the help of different views. To our knowledge the used techniques were never employed in this combination to solve the image retrieval task.

2. RELATED WORK

Since its early ancestor, the QBIC system (Flickner et al. 1995), content based image retrieval systems have come a long way. Due to the huge number of works in the area of navigation in high dimensional spaces for image retrieval we will only summarize the key ideas and works of those fields that inspired our work. Recent advances in content based image retrieval are thoroughly recaptured by (Datta et al. 2008).

Browsing techniques are presented in the work of (Matkovic et al. 2009) on visual analysis techniques in feature space. They provide standard tools like parallel coordinates to brush coordinate subspaces but have limited navigational capabilities. (Bartolini et al. 2007) focus on *personalization actions* as facilities to adapt the local browsing structure and thus enhancing the personal experience a user has while using the application. (Moghaddam et al. 2004) place images according to their pairwise similarity in a 2D context. They allow user adaption of the similarity values. (Ding et al. 2008) exemplify how a hierarchical browsing algorithm can speed up image retrieval significantly.

Many attempts on improving image search with more sophisticated visualization techniques can be found in the literature. (Hedman et al. 2005) compare a fish-eye view on the image space with different standard views off the data. (Combs & Bederson 1999) analyze if zooming improves image search. (Moghaddam et al. 2004) work on the integration of user models for improving visualization systems for personal photo libraries. (Brivio et al. 2010) finally is a recent approach to embed thumbnail images in a 2D map based on weighted Voronoi diagrams. For a more elaborate overview of different display, summarization and exploration techniques we refer the reader to (Camargo & González 2009).

(Rui et al. 1998) introduced *Relevance Feedback* as *Rocchio's algorithm* to the field of image retrieval. After that it became an excessively employed technique in the CBIR context. (Meilhac & Nastar 2002), (Su et al. 2003) and (D. Liu et al. 2006) worked on adapting the idea to different application scenarios and improving it. For a detailed overview of the current state of the art please refer to (Thomee & Lew 2007).

Incorporation and learning from user interaction data can be done in multiple ways. I. e. (Cox et al. 2002) demonstrated the usage of a Bayesian framework with PicHunter. (Fogarty et al. 2008) let the user re-rank search results and thereby influence future retrieval orders. (Campbell 2000) on the other hand explicitly model the uncertainty a user experiences when judging the relevance of examples images. We followed the approach of (Qian et al. 2002) and modeled the distributions in feature space by using a *Mixture-of-Gaussian* in combination with *Relevance Feedback*. We combined their scheme with the approach of (Daoudi et al. 2008) who utilize a configurable *Kernel PCA*. We further enhanced our system by supplying an appropriate visualization and interaction engine for the image search.

The basic and most difficult problem of all feature based image retrieval systems is that they must bridge the *semantic gap* between abstract features and semantic meaning. (Yang et al. 2006) can serve as exemplary

for a key idea in this area - the propagation of already semantically labeled data to unlabeled data by automatic algorithms. Section 3 details our approach for its solution. For an overview of the current state of the art (Y. Liu et al. 2007) provide a good start.

3. OUR APPROACH

3.1 Design Requirements

In professional contexts like medical or engineering sciences large bodies of image data pose no exception any more. Since hardware costs continued to decrease for a long time storing large sets of images became also feasible for the average computer user. Given these constraints scalability is the first major issue a modern semi-automatic image retrieval system must address. For a holistic system we must tackle it at different stages. We do so by on the one hand employing a hierarchical browsing procedure which allows the user to first restrict his search to a certain subspace of the whole feature space. On the other hand the dimensionality of the feature space is adjustable. This is achieved by choosing different parameter values for the *Kernel PCA*.

Easy extensibility of the set of similarity measures that is used to define the final feature vector for each image is also indispensable. Here we argue that the simple inclusion of additional fixed metric feature vectors is not flexible enough for describing arbitrary distances. The usage of a *Kernel PCA* solves this constraint. Since the system simply expects a similarity matrix as input, it is very easy from a developers point of view to use all information he sees fit for a problem specific set of similarity measures between images. Hence, because of its consistent interface the employment of our technique for diverse image domains is straightforward. Online dimension reduction to a variable degree is in principle possible but not yet implemented for our system.

Finally the system's benefit is directly linked to its intuitive usability. We try to keep the interface to the system as simple as possible. In the standard settings the user only faces a canvas of images. If he is more experienced, additional visual analysis techniques of our system can be used.

3.2 From Arbitrary Similarity Measures to Image Descriptors in $\Re^n$

As mentioned before an image retrieval system's power lies in the integration of different similarity measures $s^i(I_k, I_l)$, $i \in 1...m$ between images I_k and I_l. In the remainder of this section the superscript i denotes the i th similarity measure. Here we only demand that $s^i(I_k, I_l) = s^i(I_l, I_k)$ and$s^i(I_l, I_l) \geq s^i(I_k, I_l) \forall k \neq l$. These m measures can be defined on the image signal directly, be feature based, take textual annotations into account or be based on studies in which human subjects were asked to rate perceptual distances between images.

Using these similarity functions $s^i(I_k, I_l)$ pairwise between all N images in the database produces m $N \times N$ dimensional quadratic similarity matrices S^i. Like (Bishop 2006) we interpret these values as dot products in an arbitrary feature space. An eigenvalue decomposition of each S^i delivers m sets of eigenvectors V^i (here each V^i denotes a matrix whose rows are the eigenvectors) and vectors of eigenvalues Λ^i. Compression can now be achieved by choosing just the r^i rows of V^i with the largest corresponding eigenvalue in Λ^i.

Obviously this procedure does not scale well. The size of the S^i grows quadratically with the number of images in the dataset. We address this issue by using a subset of q^i data points. Let $S'^i \in \Re^{q^i \times q^i}$ be the similarity matrices of this subset, V'^i their eigenvectors and Λ'^i their eigenvalues. Let further be $D^i \in \Re^{N \times q^i}$ the submatrix of S^i that contains in each row the similarity values from all images to the subset's images.

The new image features now result from

$$\underbrace{\begin{pmatrix} D^i_{11} & \cdots & D^i_{1q^i} \\ \vdots & \ddots & \vdots \\ D^i_{N1} & \cdots & D^i_{Nq^i} \end{pmatrix}}_{D^i} \times \underbrace{\begin{pmatrix} V'^i_{11} & \cdots & V'^i_{1r^i} \\ \vdots & \ddots & \vdots \\ V'^i_{q^i1} & \cdots & V'^i_{q^ir^i} \end{pmatrix}}_{V^i} = \underbrace{\begin{pmatrix} f^i_{11} & \cdots & f^i_{1r^i} \\ \vdots & \ddots & \vdots \\ f^i_{N1} & \cdots & f^i_{Nr^i} \end{pmatrix}}_{F^i} \quad (1)$$

This illustrates how we can adapt our procedure to the number of data values by choosing appropriate values for the r^is and q^is according to the capabilities of the machine that runs our application. The value of q^i only affects the feasibility of S'^is eigenvalue decomposition. For our case study we used a subset of $q^i = 500$ of 20.000 data points. Choosing r^i (the number of rows of V^i) depends on the storage capabilities of the system and the requirements to the quality of the final conserved data variance in feature space. A full discussion of the choice of r^i is out of scope for this paper. We refer the interested reader to the terms elbow criterion that analyzes the percentage of variance explained by the reduced data respectively the *Akaike information criterion* that weighs model complexity to model accurateness. In our implementation we suffer an additional constraint because we are holding all F^i in main memory. Though by using an out-of-core data structure that would not be necessary.

We achieve equal data ranges in each of the r^i new feature dimensions by normalizing with the inverse squares of the corresponding eigenvalues in Λ'^i. This corresponds to a data whitening process which facilitates further image browsing and feature evaluation methods.

3.3 Hierarchical Browsing and the Page-Zero-Problem

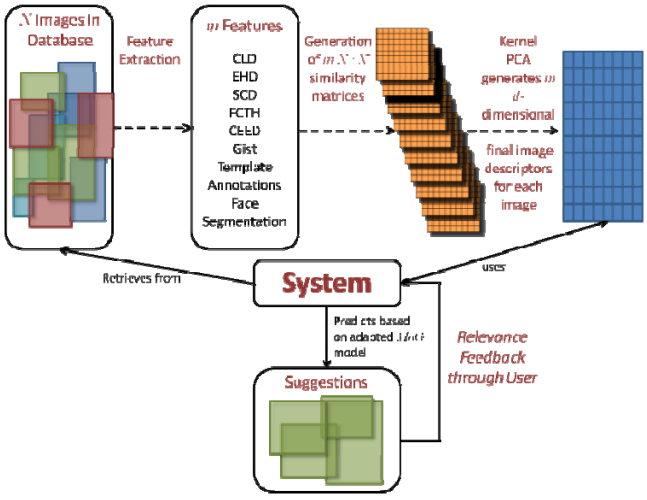
At this stage of the discussion we suppose that each image I is affiliated with a set of feature vectors $f^i(I) \in \Re^{r^i}$ in metric feature spaces. The last paragraph showed how we adapt the dimension of the data points to our needs. Yet, the mere number N of images poses problems. A hierarchical procedure suggests itself. We assume the user's search pattern in feature space for our application to be from coarse (rough similarities to target category) to fine (fine tuning of result within target category). We support this browsing pattern by offering different, increasingly large browsable subsets of the image set. These subsets are constituted by the centroids of a *KMeans* clustering with an increasing number of k for each hierarchy level. At the very bottom all images are reachable. This also elegantly solves the page-zero-problem. The initial suggestions of our system are the centroids at the highest hierarchy level. This clustering has to be done only once. Adding additional images to the database can partly be compensated by making them accessible at the lowest level.

Having scalability in mind as a main system requirement, usage of a global clustering algorithm like *KMeans* usually forbids itself. We overcome this issue by running the clustering algorithm on a reduced representation of the data points. We do this by first running a standard PCA on the covariance matrix $C = F^T F$ given by the f^i. Let

$$F(I) = \begin{pmatrix} F^1 & \cdots & F^m \end{pmatrix} \in \Re^{N \times \sum_{i=1}^{m} r^i} \quad (2)$$

We preserve only the 3 dimensions with highest variance and thus achieve sufficient data reduction. Furthermore we partition this data in an axis parallel fashion in each dimension by creating an octree. Each octant is afterwards clustered independently. The level of the octree can be adapted to N. For our use cases a level of 3 was sufficient.

(a) Flow diagram of feature definition and user interaction



(b) *Mixture-of-Gaussians* distribution $p(f^i)$ in dimension i with 3 single *Gaussian* summands and corresponding mean values μ_j^i fitted to the dataset in dimension f^i. Squares represent images I placed according to their feature value $f^i(I)$ along the axis f^i. Red: Labeled negative. Blue: Positive. Green: To be suggested in the next round. For an uncluttered plot all I that are not currently shown or will appear in the next round are not displayed

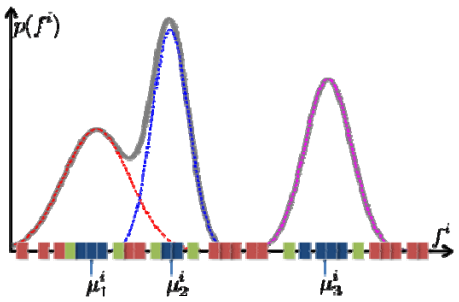


Figure 1. Models of retrieval system (a) and parameterization of feature distribution (b)

3.4 Guided Browsing using Relevance Feedback and a Mixture of Gaussian Density

Usually the specific semantic context that the user has in mind for his search defines manifolds $T^i \subset \Re^{r^i}$. The goal of the concept search is to map out and deliver all images I with $f^i(I)$ within these manifolds. How can this search be visually guided by the system and reachability of all images guaranteed? We assume, that these manifolds can be approximated with *Mixture of Gaussian* probability distributions p^i. We estimate the parameters of these distributions based on the user feedback. This *Relevance feedback* enables the system to guess the user's intentions and thus leading him to the desired image - respectively image category - by sampling image suggestions from feature space according to the p^is.

The problem we are trying to solve can also be stated as defining a set of gradients in the direction where the desired image subset lies in the feature space. Therefore abstract feature dimensions originating from the dimension reduction must be made available to the user. As already mentioned our user interface mainly consists of a canvas that presents images suggested by the system. The user can decide which ones correspond to his search target and label them as positive. He can also label every image as negative that would lead him away from his target. This input is used to "learn" the T^is. Since our assumption states that at least some of the features must correspond to meaningful object categories that are of interest to the user the probability to find a desired image in some of the $\Re^{r^i}$ must not be uniform. Otherwise our system would obviously fail. We decided to describe the current estimate of the target distribution with a *Mixture of Gaussian* model (*MoG*, see figure 1(b)). Once defined it is easy to analyze.

There are multiple ways how such *MoG*s can be defined in the feature spaces $\Re^{r^i}$. Each positive sample could define one p_j^i. This would lead to a very fine granular model which is most possibly not the desired outcome. Finding the appropriate number of modes of the distribution for an aggregate approach usually involves probing for different candidate numbers of modes k^i by running first the *K-Means*-, then an *Expectation Maximization*-algorithm and finally deciding with Akaike or Bayesian information criterion which value of k^i defines the model complexity best to explain the data.

We use an alternative technique that integrates the negative samples. (Zhou & T. S. Huang 2003) state that most systems ignore them. The reason being, that in high dimensional feature spaces positive examples might well describe one single class, but negative samples usually stem from many different classes. From that it follows that the user can never label enough negatives to describe all of them.

In contrast we use the negative examples as separators between the p_j^is. We assure that each p_j^i is defined by positive samples, that are not separated by negative samples in *any* dimension. This results in different k^i for each iteration but our experiments proved this procedure to be quite effective.

Let a_j^i be the number of positive samples that define p_j^i divided by the number of all positively labeled images. Let further be μ_j^i their r^i-dimensional vector of mean values and Σ_j^i their $r^i \times r^i$-dimensional covariance matrix.

$$p^i(f^i) = \sum_{j=0}^{k^i-1} a_j^i \cdot p_j^i(f^i, \mu_j^i, \Sigma_j^i) \tag{3}$$

with $\sum_{j=0}^{k^i-1} a_j^i = 1$. The p_j^i are given by

$$p_j^i\left(f^i, \mu_j^i, \Sigma_j^i\right) = \frac{1}{(2\pi)^{\frac{r^i}{2}} |\Sigma_j^i|^{\frac{1}{2}}} \quad \exp\left(-\frac{1}{2}\left(f^i - \mu_j^i\right)^T {\Sigma_j^i}^{-1} \left(f^i - \mu_j^i\right)\right) \tag{4}$$

In each iteration of user feedback this distribution is adapted. The number of selected positive examples by the user is usually smaller or about the same size as the generally high feature dimension. According to (Hoffbeck & Landgrebe 1996) this generally renders the covariance matrices Σ_j^i meaningless. We deal with this problem by assuming independence between feature dimensions which results in diagonal matrices Σ_j^i. This would correspond to the assumption of independence between the features within the image class and we can model

$$p_j^i\left(f^i, \mu_j^i, \Sigma_j^i\right) = \prod_{h=1}^{r^i} \frac{1}{\sqrt{2\pi}\sigma_h^i} \exp\left(-\frac{\left(x - \mu_h^i\right)^2}{2\sigma_h^{i2}}\right) \tag{5}$$

Since it is very easy to fit this simple product of one dimensional *Gaussians* the performance of our system benefits.

Once the p^is are estimated one could start sampling from the set of all N images and use rejection sampling based on the probability of these images according to p^i to find new suggestions. This procedure will invariably be slow. We greatly enhance its speed by using *the Approximate Nearest* Neighbor library of (Mount & Arya 1997). It creates *a kd-tree* from the data points and allows very efficient retrieval of neighbours in a metric vector space. We make use of it by retrieving only the g nearest neighbours to the currently positively images in all m feature dimensions. Each feature dimension can make g initial suggestions. From these initial suggestion set which is usually smaller than $g \times m$ due to related similarity measures we sample our new suggestions. g varies over time and can be enlarged when there happens to be too few neighbours to present enough suggestions through the GUI. Usually we chose g=30.

This procedure also helps us in assigning importance values to the m features. These are useful when one has to decide which feature dimension is most valuable to explore further. We assign the importance values according to the frequency with which a suggested image by feature i was subsequently labeled as positive by the user.

3.5 Visualization Techniques

Visualization is one of our most essential system components. Dimension reduction techniques like the introduced *Kernel PCA* are based on the assumption that the directions in feature space that contain the majority of variance of the data values are most important because they are caused by hidden variables. In many areas this assumption totally holds true (e. g. measured 2D human height/weight data corresponds to age and sex). However, for very high dimensional image descriptors that are massively reduced to a few dimensions their inherent meaning is lost or at least hardly nameable. The visualization component's task is to convey the meaning of this abstract directions to the user and thus bridge the *semantic gap*.

Visualizing images in their feature space is a much researched topic. Our implementation offers different views on the dataset. Starcharts, parallel coordinates and grid based views are available and prove their assets and drawbacks in different scenarios according to their abilities.

The star char view mode distinguishes itself by highlighting the feature values of a selected image on the corresponding axis in red (see figure 2(a)). The user can thereby analyse which images pair according to different distances in different feature dimensions.

To avoid cluttered scenes with overlapping images we implemented an iterative procedure that first positions the images according to their respective feature value in the star chart in increasing order of their probability p^i. Then we let the more probable, hence more important images, push their overlapping counterparts away along the difference vector between the two of them. We let this procedure run until all overlaps are resolved. This ensures that the positions of the most important images are maintained as best as possible.

The second expert mode is inspired by the classical parallel coordinate view of high dimensional data (see figure 2(c)). Unlike the classical depiction the data points are not connected by lines across the axes. In fact, the images act as their own visualization of the data point. As can be seen from the screenshot, hovering the mouse cursor over one image highlights it on all the other axes. This representation has the clear advantage of separating feature dimensions. Figure 2(c) exemplifies its benefit. It shows the final context of a successful search during the case study in section 4. Even though the task is not yet introduced, one can clearly see the different distributions along the axes. It appears that feature 7 (from above, Template) is the most suited for describing the target class. Feature 10 has not been used.

The different views on the data are useful for different user groups. Expert users have the possibility of browsing each feature dimension individually. This means they can explore all $\sum_{i=1}^{m} r^i$ single dimensions separately using either the star chart view or the parallel coordinate view by simply clicking on the axes or choosing it from a drop-down menu.

Complementing the expert mode we provide an uncluttered more common presentation for the technically uneducated users. Here we show a grid layout in which the ordering of images occurs according to the probability value $p^i(f^i(I))$ (see figure 2(b)). This can be viewed as a *rank-ordered top-k returns* classifier.

3.6 Browsing Strategies

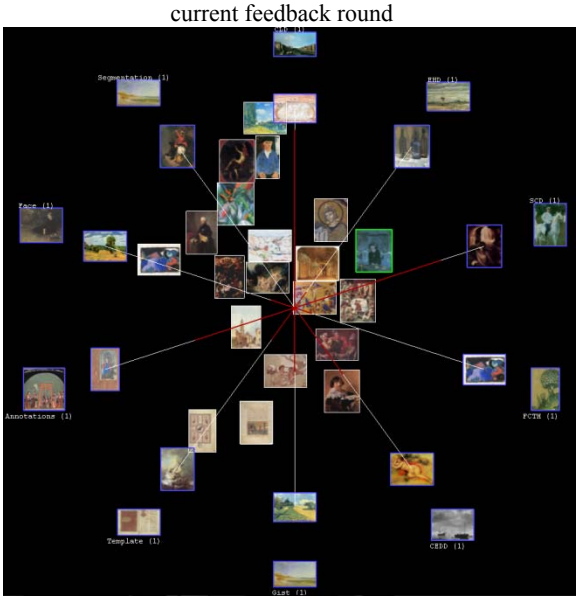
Having explained the visual interface to our system we finish its discussion by describing how the different feature dimensions can be navigated to the user's benefit. We have m r^i dimensional feature spaces. Clearly it is not possible to show them all at once. We provide different facets to navigate them. Our initial view makes the images browsable in a space spanned by the dimensions f_0^i. Star chart and parallel coordinates view display the respective feature values along the axis. The user can either explore the combination of these most prominent directions of each feature or alternatively *jump into* one dimension i to utilize its r^i feature dimensions.

At times the user may want to take a closer look at the neighbourhood of a specific image without generating suggestions. This usually happens when the target class is roughly found. We offer the user the possibility to retrieve the nearest neighbours of a selected image in the currently enabled feature space. This allows exploration of local feature space image population and supplements our example and suggestion based browsing interface.

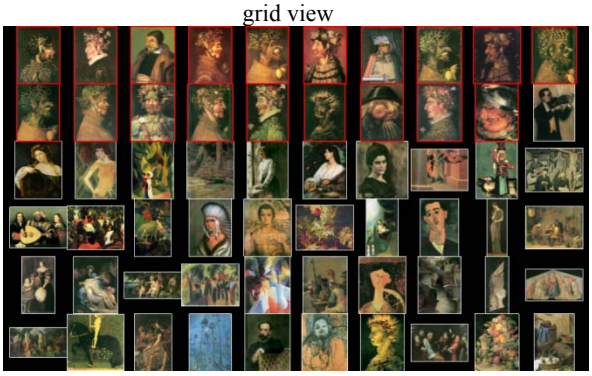
3.7 Analysis

The design of our system offers some additional benefits. As mentioned before it is difficult to bridge the *semantic gap* between the abstract features and the user's interpretation of the image's content. After the user is done with his search additional information can be inferred from the *MoG* model about his actual intentions. We automatically retrieve weighting factors for each feature dimension and thus infer importance values by looking at the density of feature value occurrences. If the feature values cluster along a dimension f^i in one or more groups, then f^i seems to be characteristic for the target category. If the values are uniformly distributed along feature i, it probably carries no meaning. This information could subsequently be used for fully automatic classification algorithms.

(a) Screenshot of our image browser with star chart representation of the *page-zero* images in feature space. Blue framed images represent extremes of the feature dimension. The green framed image is currently selected. Red coloration of axes shows the feature values of this image. The unframed images are suggestions for the current feedback round



(b) Images are ordered according to their relevance in our grid view



(c) Positively labeled images in parallel coordinate view. The green frames highlight one image on all coordinate axes.

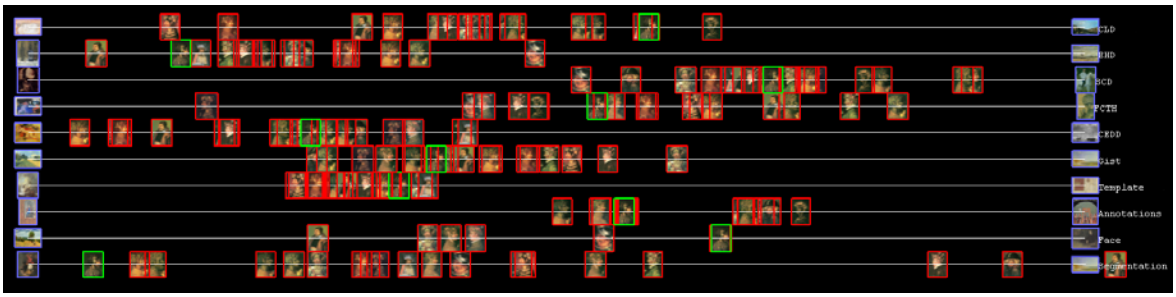


Figure 2. Views on image data that our system offers

4. CASE STUDY

Evaluating complex analytic tools for visual investigation scenarios in high dimensional feature spaces is a highly debated topic. I. e. the procedure (Owen 2007) proposes is unfortunately hard to do using a small user group.

There are some plausible arguments against *target image search* scenarios. They are somehow far-fetched because of the relative rareness of the need to find one specific image that the user has in mind. Additionally the insufficient short term memory of humans complicates the task. We can only approximately remember the picture we are looking for. A database of 20.000 images, as is used in our application scenario, inevitably contains many similar images. At some point of browsing the database it becomes virtually impossible to decide which images shall be labeled positive and negative next. Due to the mentioned problems we decided to measure the system's performance in a *target class search* scenario because here the success rate is easily measurable.

The following parts illustrate the evaluation of our search system. The evaluation's aim is to gather information about the user's effectiveness, efficiency and satisfaction concerning the quality of results provided by our system.

Dataset and Participants The evaluation was done using public domain paintings from the YORK Project DVD "10.000 Meisterwerke der Malerei" (York project 2013). Due to a number of close-ups the dataset contains nearly 20.000 unique images from various epochs. For this study we used 14 voluntary participants from the faculty of computer science. All participants had experience in using computers and common input devices. The participants were divided into two groups. The first group used our system, the second group used *Google Picasa* to accomplish the task described in the following section. Google Picasa has been chosen because of its easy to use interface and his wide capabilities for fluent browsing and handling large image sets. We also considered the usage of *imgSeek* (Cabral 2010) and *imagesorter* (Barthel 2008) but they either crashed due to memory constraints or computed too much on the fly.

Features for artwork search The quality of an image retrieval system directly depends on the features that are used to describe the images. Research from object and scene recognition respectively image and text retrieval applications resulted in a vast number of possible image descriptors and similarity measures. Since they generally originate from very different domains it is often difficult to decide how a meaningful combined similarity measure can be achieved. Concatenation is a straightforward way for doing so. However, it is generally not trivial to decide how to weigh each single feature dimension. As explained before we approach this in our work through *Relevance Feedback*.

To illustrate the system's flexibility we deliberately chose properties from very diverse feature domains. Low level features were taken from the MPEG-7 standard (Manjunath et al. 2002) or extensions based on it (Chatzichristofis & Boutalis 2011). Gist (Aude Oliva & A. B. Torralba 2001) is a mid-level feature that was shown to correlate with semantic image categories for natural images. To bridge some of the *semantic gap* contextual information were integrated by using a feature describing the distribution of faces recognized by a face detector.

User supplied semantic information can generally considered to be rare. Yet feature combinations of textual and image based information are quite promising for many different areas. Here we show how we can naturally achieve this combination of information sources with our approach. A customized text kernel using the textual annotations, which were provided with our image database, accompanies our feature set as a proof of concept.

The template feature stands for a low resolution version of the very same image it describes. Through the dimension reduction steps this feature actually reproduces the *IPC* base functions for image signals from (A. Torralba & A Oliva 2003).

The final feature we employ is based on a primitive segmentation algorithm based on the *KMeans* algorithm. It delivers a segmentation in 3 segments, is translational variant and color specific.

Table 1. Image features and similarity measures as *Kernel PCA* input for the artwork database

Image Property	Similarity Measure
Face distribution descriptor (Bradski 2010)	exp(Weighted L_1 distance)
CEDD (Chatzichristofis & Boutalis 2011)	Tanimoto Coefficient
FCTH (Chatzichristofis & Boutalis 2011)	Tanimoto Coefficient
CLD (Manjunath et al. 2002)	According to MPEG-7 standard
SCD (Manjunath et al. 2002)	According to MPEG-7 standard
EHD (Manjunath et al. 2002)	According to MPEG-7 standard
Gist (Aude Oliva & A. B. Torralba 2001)	$exp(L_2$ distance$)$
Template	$exp(L_2$ distance$)$
Segmentation	Dot Product
Annotations	Normalized String co-occurrence

Task for the Participants For the evaluation we created a scenario around receiving a big amount of unsorted pictures on DVD. The dataset of 20.000 pictures represents this unsorted collection. In our scenario an uninvolved participant demands for a collection of at least eight portraits from the painter Guiseppe Arcimboldo (1526-1593). The style of this painter is unique and his portraits are composed for example with vegetables, fruits or animals (see figure 2(b)). The participants were shown six sample images of the dataset to explain the unique style of these paintings. For completing the task the participants were requested to locate pictures within the dataset and collect them to a selection.

Procedure The evaluation was divided into two parts. The first part consists of introducing the task with image examples, the setting and the program to the participants. After that, the participants had to fulfil the task. The participants were not directed or influenced during the procedure, but their interaction was observed. For the second part the participants had to complete a questionnaire with a seven point Likert scale about their experiences and the satisfaction with the programs and the fulfillment of the task.

Results and Discussion All participants completed their questionnaires and were included in the analysis. The results were twofold. On the one hand the responses showed that the task has been easier fulfilled and was less mentally challenging with our combination of techniques as through simple browsing a large dataset. This covers the rating in which our system helps to save time to find the images in comparison to Googles *Picasa*. Scrolling in large datasets with Google Picasa has been experienced as strenuous. On the other hand the analysis showed that the learning curve for using our application is steeper than with Google Picasa. The selective picking of positive and false samples in our approach has been marked as difficult to learn. This can be found in the rating of necessary comprehension for using the programs. Google Picasa needs less comprehension than our approach. We observed that the initial view (*page zero*) of our application causes some irritations. The participants were confused by images without common visible similarities to the example images. The users seemed to be unsure, which pictures had to be marked positive or negative. We could fix this problem by offering different random subsets of the centroids (see section 3.3).

Efficiency and personal satisfaction with the result has been observed higher with our approach than by using the simple browsing of Google Picasa. During the evaluation with Google Picasa, the attention of the participant dropped when scrolling and the participant overlooked some clear result images. Besides of searching for the target images, the participants had more joy of use by finding interesting similar images from other painters which had not been part of the task. Finally we asked the participants, if they would use our application for their private image collection. The users would split the use. Because of its simple usability, they would use Google Picasa for small image data sets, but they could imagine to use our approach for medium or large sets.

We noticed several characteristic outcomes of the search process. Once a target related image appeared at the beginning of the browsing process it is quite easy to mark positive examples. One tends to label many images as positive, that share a common target related attribute. As the search tends towards the end all suggested images become equally similar to the wanted image in regard to its properties first fixed. Then it becomes easier to define negative examples. At this point it is challenging to preserve the high density of the *MoG* function in the previously fixed target related feature range.

5. CONCLUSION

We presented a novel semi-automatic approach to solve the image retrieval task in presence of abstract features and unspecific target descriptions. It features an extremely simple interface which yet is powerful enough to convey meaning about the underlying distribution in feature space of the images and thus helps to bridge the *semantic gap*.

We described a modular pipeline that is easily adaptable to diverse image retrieval tasks. In this pipeline we combined well known standard techniques to solve the individual tasks. It distinguishes itself by allowing the definition of almost arbitrary similarity measures from which a metric feature space can be deduced. A multi-resolution hierarchy on the data points is employed to achieve scalability and interactivity for the browsing procedure. A new way of finding the parameters for the *MoG* model that describes the estimated distribution of the target class based on the current user interaction data was introduced. We use this model within our *Relevance Feedback* step to generate new suggestions for the user. We evaluated our work in a case study in which works of Guiseppe Arcimboldo had to be discovered among 20.000 paintings of other artists. This proved to be feasible.

Each image retrieval system faces a similar problem. What are the most potent image similarity measures that facilitate the search for a special image or an image category given a certain task and image domain? We plan to run test searches using our system for a given task with different feature subsets. Analysis of the estimated p_j^is according to their designated σ_j^is could prove beneficial. The smaller the respective variance entry for a feature dimension the better it is applicable for the investigated search task.

ACKNOWLEDGEMENT

This work was supported by the DFG Priority Program 1335: Scalable Visual Analytics.

REFERENCES

Book

Bishop, C.M., 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Chatzichristofis, S.A. & Boutalis, Y.S., 2011. *Compact Composite Descriptors for Content Based Image Retrieval: Basics, Concepts, Tools*, VDM Verlag.

Manjunath, B.S., Salembier, P. & Sikora, T., 2002. *Introduction to MPEG-7: multimedia content description interface*, John Wiley & Sons Inc.

Journal

Brivio, P., Tarini, M. & Cignoni, P., 2010. Browsing large image datasets through Voronoi diagrams. *IEEE transactions on visualization and computer graphics*, 16(6), pp.1261–70.

Campbell, I., 2000. Interactive Evaluation of the Ostensive Model Using a New Test Collection of Images with Multiple Relevance Assessments. *Information Retrieval*, 2(1), pp.89–114.

Daoudi, I., Idrissi, K. & Ouatik, S., 2008. Kernel Based Approach for High Dimensional Heterogeneous Image Features Management in CBIR Context. In *Advanced Concepts for Intelligent Vision Systems*. pp. 860–871.

Cox, I.J. et al., 2002. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. *Image Processing, IEEE Transactions on*, 9(1), pp.20–37.

Datta, R. et al., 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2), pp.1–60.

Flickner, M. et al., 1995. Query by Image and Video Content: The QBIC System. *Computer*, pp.23–32.

Hoffbeck, J.P. & Landgrebe, D., 1996. Covariance matrix estimation and classification with limited training data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7), pp.763–767.

Liu, Y. et al., 2007. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1), pp.262–282.

Moghaddam, B. et al., 2004. Visualization and user-modeling for browsing personal photo libraries. *International Journal of Computer Vision*, 56(1), pp.109–130.

Oliva, Aude & Torralba, A.B., 2001. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3), pp.145–175.

Owen, C.L., 2007. Evaluation of complex systems. *Design Studies*, 28(1), pp.73–101.

Rui, Y. et al., 1998. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 8(5).

Su, Z. et al., 2003. Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. *Image Processing, IEEE Transactions on*, 12(8), pp.924–937.

Torralba, A. & Oliva, A, 2003. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3), pp.391–412.

Zhou, X.S. & Huang, T.S., 2003. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 8(6), pp.536–544.

Conference paper or contributed volume

Barthel, K.U., 2008. Improved Image Retrieval Using Automatic Image Sorting and Semi-automatic Generation of Image Semantics. In *Proceedings of the 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*. Washington, DC, USA: IEEE Computer Society, pp. 227–230

Bartolini, I., Ciaccia, P. & Patella, M., 2007. PIBE: Manage Your Images the Way You Want! In *2007 IEEE 23rd International Conference on Data Engineering*. pp. 1519–1520.

Bradski, G., 2010. Face Detection using OpenCV.

Cabral, R.N., 2010. What is imgSeek? Available at: http://www.imgseek.net.

Camargo, J. & González, F., 2009. Visualization, Summarization and Exploration of Large Collections of Images: State Of The Art. In *Latin-American Conference On Networked and Electronic Media*.