Database retrieval: the use of combined dissimilarities

Carmen Lai[*], D.M.J. Tax[‡], R.P.W. Duin[†], Elżbieta Pękalska [†], Pavel Paclík [†]

[*]Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

[†]Pattern Recognition Group, Department of Applied Physics, Faculty of Applied Sciences, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands

[‡]Fraunhofer Institute FIRST.IDA
Kekuléstr.7, D-12489 Berlin, Germany

email:{c.lai}@its.tudelft.nl

Keywords: data representation, image retrieval, one-class classification, dissimilarity, classifier fusion

Abstract

In image retrieval systems, the key point is the description of the set of images. In this paper we show that a representation using a cloud of points offers a flexible description but suffers from class overlap. We propose a novel approach for describing clouds of points based on the support vector data description (SVDD). We show that combining image descriptions using dissimilarities improves the retrieval precision. Further we propose a method to select an efficient and robust subset of classifiers. We investigate the performance of the proposed retrieval technique on a database of 368 images.

1 Introduction

In the problem of image database retrieval, we look for a particular image in a large collection of images. If an example, or a query image is available, we would like to find images which are similar to this query, according to our (human) perception. The construction of an automated system for such a search would, therefore, require advanced matching methods in order to achieve this goal. A crucial point is an appropriate image representation.

A number of data representations has been investigated for image database retrieval [8, 2, 5, 1]. Usually, an image or an image region is encoded by a single feature vector containing color-, texture-, or shape-based information. In Lai et al.[4] we compared this approach to a novel one proposed by us. Instead of a single feature vector, we encode an image as a set of feature vectors. Images are thereby represented by clouds of points. We showed that this approach is more powerful and flexible than a single feature representation.

To represent an image as a set of feature vectors, we use simple features, like average intensities in small image patches around individual pixels. Each image patch is again encoded by a feature vector, storing information about color and texture. The complete image is then represented by a set or a cloud of such vectors. This cloud may be described in different ways. For example, a one-class classifier may be trained on the cloud, dividing the feature space into two regions. In the first one, it is assumed that the feature vectors are similar to the cloud and are, therefore, accepted by the classifier. In the second region, the vectors lie outside the cloud description and are rejected. Here, we will make use of a support vector data description (SVDD)[9, 10], a classifier inspired by the support vector classifier[11]. An alternative way to represent clouds is based on Mahalanobis distance. In the paper, we investigate the implications of combining cloud representations using SVDD and Mahalanobis distance.
Although in this cloud representation the storage and computational costs are much higher than for the single vector representation, it is simpler to detect substructures in the original images. Two clearly distinct objects in the image (for instance, a sculpture and a background) will result in two separate clouds in the feature space. In the single feature vector representation, the information of both objects will be mixed.

A complication of the cloud representation is a possible overlap between clouds obtained from different images. It might happen, that one of the clouds is completely covered by another cloud. Although all the pixels lie within the description of the query image, their distribution is completely different. We show that by combining the information of several clouds descriptions this problem can be solved. It will improve the retrieval process and allow new approaches for image retrieval systems.

This work is a squeeze version of a paper that we submitted for a journal publication. In section 2, we present the image retrieval problem and the use of the combination of individual cloud representations. Several approaches are presented in order to select an efficient subset of SVDDs. The experiments on an image dataset are described in section 3. Their results are further discussed in section 3.4. Finally, conclusions are summarized in section 4.

2 Image database retrieval

Let us denote by I_D an image database with N images I_i, $i = 1, ..., N$. The image retrieval problem is formulated as a selection of a subset of images similar to a given query image I_Q. In our application, images in the database can be assigned to classes which describe images coming from the same origin, e.g. grain textures, sky images, images with flowers etc. Therefore, whenever we speak about a class, we mean a group of similar images. By this, an image retrieval strategy can be tested in a more objective way. Such a retrieval strategy is defined in two steps: image representation and a similarity measure between the query image and images stored in the database.

2.1 Image preprocessing

For the sake of image discrimination, images should be represented in a feature space such that the class differences are emphasized. A convenient way to extract good features is to apply a bank of filters to each image in a database. These filters may be, for example, wavelets, Gabor filters or other detectors. In many cases, the filters will give response values which are incomparable to each other. To avoid that one filter with large variance will dominate, the data is preprocessed by weighting individual features on the basis of a dataset mean and standard deviation. We use a scaling that emphasizes differences between individual images in the database.

Assume we have constructed a dataset F containing N K-dimensional feature vectors, representing all images in the database. The weight vector $\mathbf{w}$ is computed element-wise in the following way (see also[7]):

$$w_k = \frac{1}{\text{mean}(F_k)} \log_2 \left[\text{std} \left(\frac{F_k}{\text{mean}(F_k)} \right) + 2 \right], \quad (1)$$

where F_k is the k-th feature in the dataset F. All features of all images are rescaled according to this weight vector.

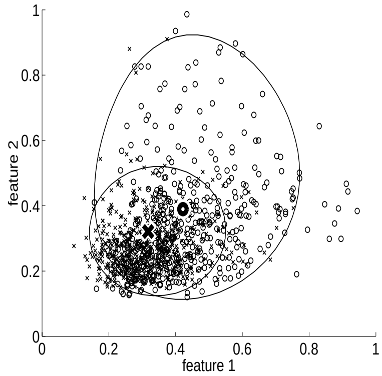

Figure 1: Two clouds of points and corresponding SVDD boundaries.

2.2 Cloud representation

Different image regions may contain different information, especially when images are inhomogeneous. Descriptions, preserving this heterogeneity, are desirable. Our proposal is to represent images by clouds of points in a feature space.

Assume we have a cloud C_i representing the image I_i. The cloud consists of M_i feature vectors, storing the information from a number of image patches. A patch is a pixel neighborhood used for computing features. We suppose that compact clouds are easier to separate from each other. We propose to describe a cloud by a one-class classifier based on the SVDD.

For an accurate description of the SVDD see[9]. Basically we enclose the cloud by an hypersphere. We minimize the volume of this hypersphere by fitting the boundary on the points of the cloud. This is a one-class classifier because it defines only the points that are inside the SVDD boundary, which are called *target*. It doesn't make any distinction between the points that are outside the boundary, which are all called *outliers*. If the original image contains multimodal information, the corresponding representation may results in several separate clouds in the feature space. The svdd used to describe such an image is able to detect and enclose this clouds. Therefore the corresponding description will be composed by several closed boundaries.

In order to minimize the boundary of the SVDD classifier, the user has to define the percentage of target objects (points) that will lie on the boundary f_i. Let B_i^{SVDD} be a one-class classifier constructed for image I_i. For a vector $\boldsymbol{x}$, coming from the cloud C_i, i.e. $\boldsymbol{x} \in C_i$, it is defined as:

$$B_i^{\text{SVDD}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ is accepted by the SVDD} \\ 0 & \text{if } \boldsymbol{x} \text{ is rejected by the SVDD} \end{cases} \quad (2)$$

Our classifiers are trained such that a fraction $f_i = 0.2$

of target vectors lie on the boundary, i.e.:

$$\text{Prob}\left(B_i^{\text{SVDD}}(\boldsymbol{x}) = 0 \,\&\, \boldsymbol{x} \text{is on the boundary} \mid \boldsymbol{x} \in C_i\right) = f_i, \quad (3)$$

which means that the boundary vectors are here considered as outliers. An example of two clouds and the corresponding SVDDs in 2D feature space is given in Figure 1. Each original image is homogeneous, therefore the svdd boundary is a single blob around the corresponding cloud.

The dissimilarity between images I_i and I_j is defined as the percentage of points from cloud C_i, rejected by the one-class classifier B_j^{SVDD}, as follows:

$$s(I_i, I_j) = \frac{1}{M_i} \sum_{\boldsymbol{x} \in C_i} (1 - B_j^{\text{SVDD}}(\boldsymbol{x})), \quad (4)$$

The smaller the percentage of outliers $s(I_i, I_j)$, the more similar the images I_i and I_j.

Alternatively, the clouds of points can be compared by using the Mahalanobis distance. This distance assumes that the clouds of points posses a homogeneous structure, i.e. in a type of unimodal hyperellipsoids, and, thereby, they can be described by normal distributions. The Mahalanobis distance between two images I_i and I_j is defined as:

$$d_M(I_i, I_j) = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \Sigma_{i,j}^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j), \quad (5)$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ are estimated mean vectors of the corresponding clouds and $\boldsymbol{\Sigma}_{i,j}$ becomes an estimated common covariance matrix.

2.3 Image similarity

A retrieval system should evaluate similarities between a query image and the images of a database, in order to retrieve the more similar ones. In the following, we describe two strategies to perform database retrieval on the basis of SVDD representation of images (see Figure 2). For a given database of images I_D, the cloud representations C_i as well as corresponding SVDD classifiers B_i^{SVDD} are available. The matrix S is $N \times N$, where N is the number of images in the database I_D. The rows indicate the clouds of points, while the columns refer to their SVDD classifiers. Therefore, the generic element s_{ij}, computed by Eq. (4), stores the percentage of objects from the cloud C_i rejected by the classifier B_j^{SVDD}.

The relations between an image and the remainder of the database can be evaluated from two points of view. The first one is based on a row of the dissimilarity matrix S which we call a *cloud profile*. The profile Γ_i shows how the cloud C_i fits to the boundary of all one-class classifiers. The second viewpoint uses a *classifier profile* Ω_j^{SVDD} based on a column j in the matrix S. It shows the dissimilarities of all image clouds to the classifier B_j^{SVDD}.

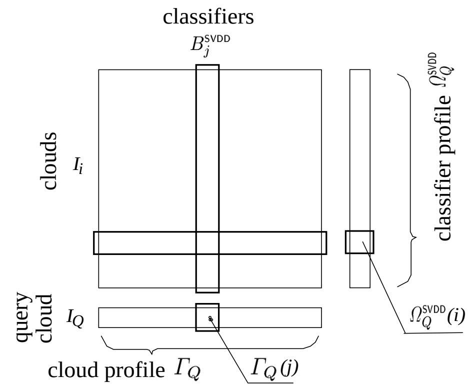

Figure 2: Combination scheme for image database retrieval.

The cloud profile of a given query image I_Q is defined as follows:

$$\Gamma_Q = [s(I_Q, I_1), s(I_Q, I_2), \ldots, s(I_Q, I_N)],\ i = 1, \ldots, N \quad (6)$$

see Figure 2. This vector evaluates the responses of the query cloud to all the SVDDs in the database. By ranking the query cloud profile Γ_Q, the classifiers with the lower percentage of outliers are identified. In this way, just a single classifier is used to find the most resembling images to the query.

A serious problem of this retrieval setup is caused by overlapping clouds, representing images from different classes. For instance, it may happen that one SVDD boundary completely contains another one, originating from a different image class. Consequently, the query cloud surrounded by the boundary of a smaller cloud will also be accepted by the larger boundary. Therefore, the two images will be both considered similar to the query image even if they come from different classes. This, of course, lowers the performance of the whole image retrieval system. To prevent such inconvenient situations, we propose to combine all the information given by the classifiers, using the entire cloud profile. Of course, this requires that all the SVDDs are trained in advance. We compare the query cloud profile with the cloud profiles of the other images in the database. For this purpose, different dissimilarity measures can be used, for instance the Euclidean distance:

$$D_E(I_Q, I_i) = ||\Gamma_Q - \Gamma_i||,\ \ i = 1, \ldots, N \quad (7)$$

Another possibility is the cosine distance, based on the inner product between cloud profiles:

$$D_{cos} = \frac{1}{2}\left(1 - \frac{(\Gamma_Q)^T\, \Gamma_i}{||\Gamma_Q||\, ||\Gamma_i||}\right),\ i = 1, \ldots, N \quad (8)$$

In this way, the responses of the individual classifier are combined to express the dissimilarity between the query image and the images in the database. The images, most similar to the query, are then retrieved by ranking the dissimilarities $D_E(I_Q, I_i)$.

This approach is similar to the decision based on multiple classifiers, proposed by Kuncheva *et al.* [3], where the decision templates are created by averaging over all training objects in a class. In our experiments, individual classifiers are constructed for all single images in the database.

The second way of computing similarities between a query I_Q and the other images is based on the classifier profiles. As Figure 2 shows, the classifier profile Ω_Q^{SVDD} of the query image is defined as:

$$\Omega_Q^{\text{SVDD}} = [s(I_1, I_Q), s(I_2, I_Q), \ldots, s(I_N, I_Q)],\ i = 1, \ldots, N \tag{9}$$

This vector presents the responses of the query classifier to all the clouds from the database. By ranking of this profile, we can find out which clouds are better accepted by the query classifier. This method employs again just a single cloud and it is, therefore, sensitive to the cloud overlap. As discussed in the approach based on cloud profiles, we can combine the entire classifier profile, i.e. the responses of clouds to a particular SVDD. The images are compared by evaluating the dissimilarities between the classifier profiles. These are again based on Euclidean and cosine distances, as previously defined in Eq. (7) and (8), where Γ_Q and Γ_i are now replaced by Ω_Q^{SVDD} and Ω_i^{SVDD}, respectively.

Above, the number of classifiers or clouds in the profile is as large as the number of images in the database. This is not essential, because a profile with a smaller set of classifiers or clouds may be used as well, using a concept of representation sets [6]. In this way, the computational complexity can be significantly reduced. This aspect is investigated further in the next section. Due to the duality of the problem, we will focus on the use of the cloud profiles.

The same strategies for image retrieval, as discussed above, may be also used for Mahalanobis distances. The matrix S now contains Mahalanobis distances $d_M(I_i, I_j)$ instead of $s(I_i, I_j)$. Note, that S is symmetric and there is, therefore, no difference between cloud and classifier profiles.

2.4 Selection of classifiers in the cloud profile

In the previous section, we introduced the idea of profile as a tool for combining image representations. In the following, we focus on the cloud profile storing responses of all individual SVDDs. We suppose that the information given by the classifiers is redundant. Images of the same class are similar, and so are their cloud representations. Therefore, only one or few classifiers of the same class may be selected to form the profile. We investigate different approaches to select such a subset of classifiers.

The first approach is a *systematic* search for relevant classifiers. One by one, the one-class classifiers are removed and the performance of the remaining SVDDs is computed. The classifier with the highest score may be deleted as superfluous. In order to further decrease the number of SVDDs, this process is iterated. The stopping criteria may be a threshold on the retrieval performance or on the size of the profile itself. This is essentially a backward selection considering individual SVDDs as features. The algorithm does not set additional constrains on deleting less relevant classifiers. It even allows the removal of all SVDDs of one class.

In order to take into account the class organization of the database, we proposed another method, which we call a *class* approach. Instead of a single SVDD, a subset of classifiers, one for each class, is removed at once. Different combinations are tested to find the less relevant set, i.e. the one that, once removed, gives the higher performance. In order to obtain a smaller size of profiles, this process may also be iterated, using the same stopping criteria as previously proposed. As a consequence, the profiles are build up with a fixed number of classifiers for each class. Therefore, all classes are equally represented.

In order to figure out whether equal representation of classes is desirable, we implemented the third, *random* approach. A subset of classifiers, with the same size as in the previous approach, is removed at once. But, instead of taking out one classifier from each class, this subset is chosen randomly from the complete set of classifiers available. By iterating the procedure we reduce the number of classifiers, i.e. decrease the size of the profiles.

The goal of these approaches is to investigate the relation between the size of a cloud profile and the retrieval performance.

3 Experiments

In this section we describe a set of experiments performed on a dataset of images. We evaluate the retrieval performance when images are represented by clouds of points. First, we compare several strategies for computing and combining similarities between images. Later, we investigate possible ways how to reduce the computational complexity of the image retrieval problem and finally, we discuss the results.

3.1 Experimental set-up

Our experiments are based on 23 512×512 mostly homogeneous images obtained from MIT Media Lab (see ftp://whitechapel.media.mit.edu/pub/VisTex/). Each original image is cut into 16 128×128 non-overlapping pieces representing a single class. Therefore, we use a database with 23 classes and 368 images. Figure 3 shows few examples of the database images.

(a) Clouds (b) Brick (c) Flower (d) Monet

Figure 3: Examples of images of the database.

The absolute values of responses of 10 different Gabor filters are used as features. These 10 features were chosen by a backward feature selection from the larger set of 48 Gabor filters with different smoothing, frequency and direction parameters. The outcome of each filter is again an image, with size 128×128. Instead of a single pixel, the average intensity in 9×9 pixel neighborhood is used. Using this preprocessed data, we build cloud representation of images. Each cloud consists of 500 patches randomly selected from the image. The choice of 500 is a compromise between a higher standard deviation (noise sensitive) for small number of patches, and a computational complexity.

The images of the database are, one by one, considered as queries. The retrieval precision is computed using all 368 images. The presence of the query image in the training set leads to a slightly optimistic performance estimate. We decided for this approach because it allowed us to work with the complete dissimilarity matrix. For each query image, 16 most similar images are found. The retrieval precision for each query is then defined as the percentage of returned images, originating from the same class as the query. The total precision of the retrieval method is then the average precision over all 368 individual queries, i.e.:

$$P = \frac{1}{368} \sum_{I \in I_D} \frac{\text{\# relevant images}}{16} \cdot 100\% \qquad (10)$$

3.2 Experiment 1: Evaluation of the retrieval system

In this set of experiments we evaluate the performance of the basic retrieval system. First, we investigate the behavior of the proposed SVDD one-class classifier. We build the SVDD for the cloud of points, setting 20% of the points to the boundary; see (3). As described in section 2.3, the similarities between images can be computed from two different perspectives. In the classifier profile approach we use a single SVDD, trained on the query cloud and we apply it to other images, represented by clouds. It follows from Table 1 that, by ranking the single profile, the total precision is 72.0%. We choose to combine these clouds further. This leads to a decision based on the dissimilarities between classifier profiles of the query and the other images. Here, two different distance measures are considered: the Euclidean distance and the cosine distance for which the precision of 80.4% and 81.0% is reached, respectively.

In the second approach, the classifier responses to a query cloud are combined to form a cloud profile, as described in section 2.3. The images, most similar to the query, may be obtained by a direct ranking in the cloud profile. It can be seen from Table 1 that the ranking approach has a low performance (58.9%) because the results are based on single pairs of classifiers and clouds. By computing dissimilarities between cloud profiles, we effectively combine the classifiers. In this way, we gain the performance of 81.4% and 81.6%, respectively.

As described in the section 2.2, the cloud of points representation also allowed us to express similarity between images using the Mahalanobis distance, defined in Eq. (5). After ranking these distances, we obtain a precision of 78.9%. This good result can be explained by the homogeneous character of the images, resulting in almost normally distributed clouds. By applying the proposed combination of cloud profiles, we obtained the precision of 82.5% and 82.3% for Euclidean and cosine distance, respectively.

It follows from the results in Table 1 that in all cases, combining outperforms the retrieval based on the single image. We observe, that combination of the SVDD classifiers yields a significantly large improvement with respect to a single SVDD. We conclude that SVDD may easier reach good results by combining due to its high variance. The improvement is small for the Mahalanobis distances.

Image representation	Method	Precision [%]
SVDD/Classifier profile	single classifier	72.0
	combined (E)	80.4
	combined (cos)	81.0
SVDD/Cloud profile	single cloud	58.9
	combined (E)	81.4
	combined (cos)	81.6
Mahalanobis distance	single cloud	78.9
	combined row profile (E)	82.5
	combined row profile (cos)	82.3

Table 1: Experimental results: Precision of different retrieval methods.

3.3 Experiment 2: Selection of classifiers in the cloud profile

In the previous set of experiments, the cloud profiles were built based on all available SVDDs. Only a subset of database images ('prototype' images) may be described by SVDDs and used to build a profile. Similarities between images may be still measured, as they are based on profile patterns. In this section we evaluate several selection criteria to obtain an efficient and robust subset of classifiers.

In order to have a general testing procedure we need independent training and test sets. Each class of similarity is made by sixteen different images. If we select from the entire database one image from each class, we have 23 images that can be used to form an independent test set. Consequently, the training database consists of 345 images with 15 images per class. We updated the precision formula in Eq. (10) accordingly.

When a selection criterion is applied to the training set, one or several inferior classifiers are detected. By removing these classifiers the system attains higher performance than by removing all the other possible combinations of SVDDs. Therefore, we judge these classifiers as less useful, and remove them from the training and test profiles. The performance of the retrieval system based on this smaller profile description is evaluated on the test set. In order to avoid that the criterion is influenced by particular test set, we apply the same method to sixteen different training and test sets and average the results.

As described in section 2.4, three different approaches are proposed in order to find a small number of SVDDs to form an efficient cloud profile. These approaches are applied to each of the sixteen training sets. The systematic method removes one SVDD at a time. It is iterated 343 times. Therefore, we evaluate the performance starting with the total amount of 345 classifiers and continue until two. In the class approach, 23 SVDDs are removed simultaneously, with the constraint that they come from different classes. The procedure is iterated 16 times, starting from a complete profile with 345 SVDDs until the smallest set of 23 classifiers. In the random approach, also 23 SVDDs are removed together. The choice of which subset is deleted is based on the training performance for different random subsets. The procedure is iterated 16 times. We constructed the random selection to be comparable with the class approach.

Figure 4 shows the performance as a function of number of SVDDs in the cloud profile. It turns out that choosing the small number of classifiers using the systematic approach improves the retrieval performance. Apparently, the information stored in the complete profile is redundant. Smaller profile is more efficient regarding the computational complexity and memory requirements.

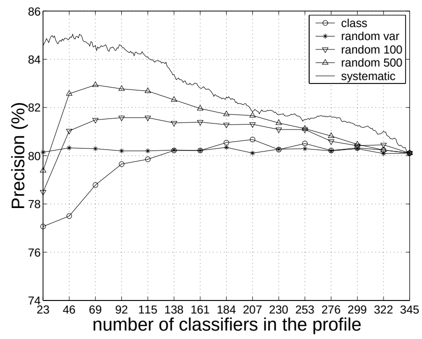

Figure 4: Performance estimates of three selection criteria as a function of the number of classifiers in the profile.

The class approach yields, on the other hand, the worst performance. We may argue that not all the classes are representative. Therefore, it is not useful to enforce a regular class organization in the selection process. This explanation is also supported by the results of the random selection. The line denoted *random var* in the graph, corresponds to the random selection, which is comparable to the class approach. It attains the same or better performance not using any class information.

We have also investigated two other settings of the random approach: *random 100* and *random 500*. In these cases, we select the right subset from 100 or 500 randomly generated subsets in each step, respectively. It follows from our results, that by evaluating more subsets per step, the performance gets closer to the systematic approach. Nevertheless, the computational complexity is much lower (*systematic:* $345 \cdot 344/2 = 59\,340$, *random 500:* $16 \cdot 500 = 8000$ criteria evaluations).

3.4 Experiment 2: Discussion

By using a cloud profile, we are building a new feature space, where responses to particular SVDDs form the features. The systematic selection of SVDDs in the profile shows that using a smaller number of classifiers is better than the entire available set. In fact, while the performance reached by using all the SVDDs is 81%, the maximum performance of 85% is achieved by using only 50 SVDDs. Figure 5 shows the detailed results of the systematic approach, when the number of classifiers in the profiles is decreasing one by one (starting from 60). The performance remains almost constant until the profile size reaches about 22, Then, it decreases in a pronounced way, together with the number of classifiers. It is interesting that the original precision of 81% is achieved again with only 17 classifiers. This suggests that only few SVDDs may be used to generate a feature space, suit-

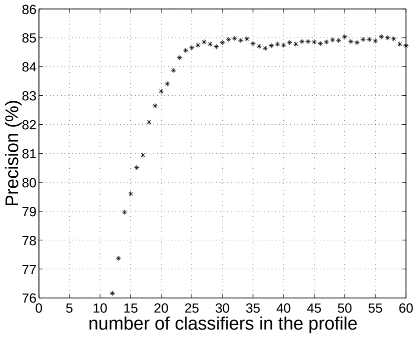

Figure 5: Performance of the systematic classifier selection

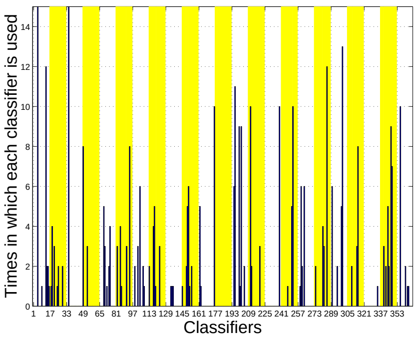

Figure 6: Classifiers held in the profiles as the best 23 by the sixteen test sets.

able for an image retrieval.

Figure 6 helps us to evaluate whether a particular classifier describing one image is more representative than the others. On the x-axis all possible SVDDs in the database are listed. The gray color denotes the classes of similarities. The bar for each classifier shows, how many times it was included into the best 23 SVDDs. Because there are 16 test sets, the same classifier can be requested a maximum of 15 times. It is evident from the plot that only a few SVDDs are used, often just a few per class. Moreover, not all classes are relevant. For example, the SVDDs referring to the class *painting1* (with indices between 177 and 192) are never used, and only the SVDD number 334 from the class *water* is selected once.

To better understand this behavior, we can visualize the classifier profile. Examples are given in figures 8 and 7, where the x-axis represents all the clouds, grouped according to the class of similarity. The y-axis shows the percentage of outliers when the given SVDD is applied. Large classifiers contain pixels of a number of different images, even those that belong to different classes. An example is shown in Figure 7(a). The profile of SVDD 135 of class *leaves* is very large and unspecific, therefore clouds from different classes are considered similar to the query.

Small classifiers suffer for the opposite problem.

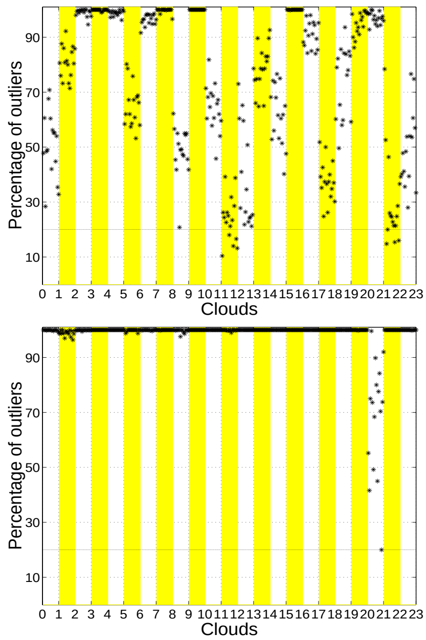

Figure 7: Percentage of outliers of two ”bad” classifiers applied to all images. A classifier from the class *leaves* in the upper and a classifier from the class *water* in the lower graph.

They reject as outliers the pixels of all images in the database, with the exclusion of those that belong to the same class. An example is shown in Figure 7(b). The profile of SVDD 334 of class *water* clearly identifies the clouds of the same class, but all the others are completely outside. This classifiers is thereby not very informative.

On the contrary, Figure 8 shows two examples of frequently selected SVDDs: numbers 35 (class *building*) and 285 (class *stone*). Both classifiers are sufficiently small to clearly identify images belonging to their class. At the same time, the clouds of the other classes show different numbers of outliers. Some are completely outside, others are partially accepted. We can infer that a good classifier is a compromise between small and big sizes.

It turns out that a suitable feature space may be generated by a limited number of classifiers. For new images, just a dissimilarity to this small subset is computed. The actual retrieval is eventually performed by combining the classifier responses. Proposed solution is flexible and computationally easier than the use of complete profiles.

4 Summary and Conclusion

The performance of an image retrieval system depends on an appropriate representation of image data. We propose to describe an image by a cloud of points. This is more robust to noise and, at the same time,

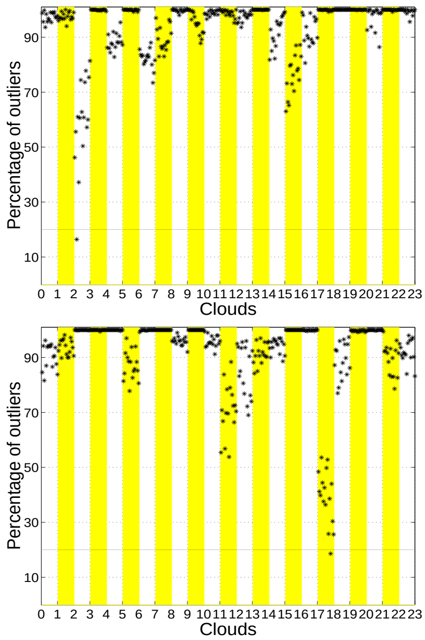

Figure 8: Percentage of outliers of two "good" classifiers applied to all images. A classifier from the class *Building* is used in the upper and a classifier from the class *Stone* in the lower graph.

sensitive to substructures in the data.

To apply this type of representation, a convenient way of measuring similarity between images must be defined. In the paper, we investigate two different representations. The first one, proposed by us, describes a cloud of points by the support vector data description (SVDD) method. In contrast to other methods based on the probabilistic approach, SVDD describes the data domain in a feature space. By this approach, images can be easily matched, based on the fraction of the points rejected by the description (the smaller, the better). We believe, that especially for inhomogeneous images, SVDD is a convenient image representation for database retrieval. The second representation, we investigated, is a Mahalanobis distance between clouds of points. It assumes normal distribution of data, which may not be met for inhomogeneous images.

Each image can be characterized either by a classifier or a cloud profile. Direct ranking of the query profiles gives a poor performance, due to large cloud overlaps. We found out that computing distances between complete profiles, i.e. combining corresponding image representations, is a better strategy. Combining full profiles is not convenient due to high computational complexity and information redundancy. Therefore, we have proposed several methods for selecting smaller profiles. Dissimilarity between images may still be computed due to unique profile patterns.

We have performed a set of experiments on a dataset of 368 images. We found out that if images are described by clouds of points, combining such a representation provides a powerful tool for image retrieval. By combining all the available information stored in the complete profile, we reach better results than by direct ranking of image dissimilarity. The systematic approach selects the small cloud profile with the best performance but is computationally intensive. We show, that reasonable performance increase is also reached in a faster way by random selection.

References

[1] S. Antani, R. Kasturi, and R. Jain. Pattern recognition methods in image and video databases: past, present and future. In *Advances in Pattern Recognition, Proceedings of SPR'98 and SSPR'98*, pages 31–53, Berlin, 1998. IAPR, Springer-Verlag.

[2] T. Huang, Y. Rui, and S.-F. Chang. Image retrieval: Past, present, and future. In *International Symposium on Multimedia Information Processing*, 1997.

[3] Ludmila I Kuncheva, James C Bezdek, and Robert P W Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.

[4] Carmen Lai, David M.J. Tax, Robert P.W. Duin, Elzbieta Pekalska, and Pavel Paclík. On combining one-class classifiers for image database retrieval. In *accepted for 3rd Int. Workshop on Multiple Classifier Systems, Cagliari, Italy, June 2002*, 2002.

[5] K. Messer and J. Kittler. A region-based image database system using colour and texture. *Pattern Recognition Letters*, 20:1323–1330, 1999.

[6] Elżbieta Pekalska and Robert P W Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8):943–956, 2002.

[7] Y. Rui, T. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS, 1997.

[8] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based Image Retrivial at the End of the Early Years. *IEEE Trans. Pattern Analysis and Machine Inteligence*, 22(12):1349–1380, 2000.

[9] D.M.J. Tax. *One-class classification*. PhD thesis, Delft University of Technology, http://www.ph.tn.tudelft.nl/˜davidt/thesis.pdf, June 2001.

[10] D.M.J. Tax and R.P.W Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199, December 1999.

[11] Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons., 1998.