International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 18, No. 5 (2004) 867–890
© World Scientific Publishing Company

World Scientific
www.worldscientific.com

A STUDY ON COMBINING IMAGE REPRESENTATIONS FOR IMAGE CLASSIFICATION AND RETRIEVAL*

CARMEN LAI*,‡, DAVID M. J. TAX†,§, ROBERT P. W. DUIN*,¶,
ELŻBIETA PĘKALSKA*,‖ and PAVEL PACLÍK*,**

*Pattern Recognition Group, TU Delft, Lorentzweg 1,
Delft, 2628 CJ, The Netherlands
†Fraunhofer FIRST.IDA, Kekuléstr.7, Berlin, D-12489, Germany
‡C.Lai@ewi.tudelft.nl
§D.M.J.Tax@ewi.tudelft.nl
¶R.P.W.Duin@ewi.tudelft.nl
‖E.Pekalska@ewi.tudelft.nl
**P.Paclik@ewi.tudelft.nl

A flexible description of images is offered by a cloud of points in a feature space. In the context of image retrieval such clouds can be represented in a number of ways. Two approaches are here considered. The first approach is based on the assumption of a normal distribution, hence homogeneous clouds, while the second one focuses on the boundary description, which is more suitable for multimodal clouds. The images are then compared either by using the Mahalanobis distance or by the support vector data description (SVDD), respectively.

The paper investigates some possibilities of combining the image clouds based on the idea that responses of several cloud descriptions may convey a pattern, specific for semantically similar images. A ranking of image dissimilarities is used as a comparison for two image databases targeting image classification and retrieval problems. We show that combining of the SVDD descriptions improves the retrieval performance with respect to ranking, on the contrary to the Mahalanobis case. Surprisingly, it turns out that the ranking of the Mahalanobis distances works well also for inhomogeneous images.

Keywords: Data representation; image classification; image retrieval; one-class classification; dissimilarity; classifier fusion.

1. Introduction

In the problem of image 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 requires advanced matching methods. In this study, we describe a matching approach based on combining of

*The final work on this paper was done at the Information and Communication Theory Group, TU Delft, P.O. Box 5031, 2600 GA, Delft, Netherlands.

multiple image representations. We investigate, if combining multiple representations improves the retrieval performance with respect to a single representation i.e. ranking.

Building an image representation is a first step in designing of a retrieval system.[1,3,4,7,10,14] Usually, an image or an image region is encoded by a single feature vector containing information on image features like texture, shape or color. Such characteristic features are extracted from images in the database and stored. In a retrieval process, a feature vector for the query image is first extracted and then used for finding images which are the most similar to the query. Such low-level image features are often too restrictive for a description of images on a conceptual or semantic level.[3]

In this study, we represent images by sets or clouds of feature vectors. As we have described in Ref. 6, a cloud of points representation may account for heterogeneous substructures in images. Two clearly distinct objects in an image (for instance, a sculpture and a background) will be represented by two separate clusters in the feature space. In the single feature vector representation, the information on both objects will be mixed.

Image representations, based on sets of feature vectors, are used, for example, by Maron and Ratan,[8] who proposed to construct a semantic concept by learning it in a supervised way from a set of positive and negative image examples. Once trained, images from a database may be ranked according to their similarity to the concept. Our aim is, on the contrary, to measure the similarity between images without specifying the concepts or labeling the images.

A complication of the cloud representation is a possible high overlap (due to imperfect image features) between clouds of points obtained from semantically different images. A cloud, representing one image, may be significantly covered by another cloud coming from a different class. In such a case, both clouds become virtually indistinguishable. Our aim is to investigate, if this problem could be overcome by combining individual cloud descriptions.

To enhance the performance of content-based retrieval systems, the retrieval problem may be transformed into the image classification,[3] where images in the database are grouped into semantically meaningful classes. In this way, the semantic gap is reduced, since the image features can be selected such that particular characteristics of the classes are captured. In this paper, we will first focus on image classification problem, since different retrieval approaches can be easily evaluated and compared when classes are available. Next, we will discuss the image retrieval problem.

The paper is organized as follows. Section 2 discusses the representation of images by clouds of points. In Sec. 3, the image retrieval problem is presented. Then, the retrieval based on ranking of images is formalized and finally, the main contribution of this paper, the combination of individual cloud representations, is introduced. The experiments on two image databases are described in Sec. 4. The first database serves as the illustration of image classification problem, while the second refers to image retrieval. The results are further discussed in Sec. 5. In this study, we focus on merits of a specific combination strategy for image retrieval, not on a design of a high-performance image retrieval system. However, for the sake of completeness, we also discuss computational complexity of the proposed retrieval strategy and actual time demands of our experimental implementation. The final section summarizes our conclusions.

2. Representation of Images

To represent an image as a set of feature vectors, simple characteristics, like average filter responses in small image patches around individual pixels will be used. Each image patch is encoded as a feature vector, storing information on e.g. color and texture.

2.1. *Image features*

For the sake of image retrieval, 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 texture detectors. Another possibility is to capture color characteristics, like energy or entropy in the particular color channels. Besides such filters, a number of other techniques can be used to extract interesting information. The image features may refer to the number of corners, sharpness of the edges, bending energy of curves, change in orientation, etc. Some specific detectors can be used to describe particular patterns in the classes, if available, e.g. in order to detect a face, a building or human skin. These detectors will define features which can discriminate better between a number of patterns present in images.

In general, feature values may be incomparable to each other. To avoid the dominance of one feature with a large variance, the data is preprocessed by weighting individual features on the basis of a dataset mean and standard deviation. A scaling is used to emphasize differences between individual images in the database.

Assume that 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 in an element-wise way as follows:

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

where F_k is the kth feature in the dataset F. All features of all images are rescaled according to this weight vector. This weighting strategy, inspired by text retrieval, was proposed by Rui *et al.*[12] for image retrieval.

2.2. *Cloud representation*

The complete image is described by a set of vectors in a feature space. This set may often be inhomogeneous and consist of several clusters. In this paper, we call this set of vectors *a cloud of points*. Such a cloud can be represented in different ways. In this paper, two approaches are considered.

The first approach relies on a boundary one-class classifier built by using a support vector data description (SVDD)[15,16] on a cloud of points. Feature vectors inside a boundary are considered to be similar and, therefore, accepted by the classifier. The vectors lying outside the boundary are rejected. The retrieval system uses such one-class classifiers trained on images from the database.

An alternative way to represent a set of points is based on the assumption of relatively homogeneous image clouds. Such clouds can be modeled by Gaussian distributions. This naturally leads to image comparison by the Mahalanobis distance.

We have chosen these two types of cloud descriptions because of their different properties. Although the assumption of homogeneous clouds leads to a simple comparison method, it often does not hold in practice. It means that the information in multimodal clouds will be influenced by the area of high densities and then averaged out in the direction of large covariances. On the other hand, the SVDD is a very flexible classifier able to detect boundaries of separate clusters. Therefore, it is an attractive option for the description of multimodal clouds.

2.2.1. *Support vector data description*

For the completeness of the paper, we will give a brief description of the SVDD; see Refs. 15 and 16 for details. Let a dataset, called a target set, be represented by M vectors in a K-dimensional feature space, i.e. $\{\mathbf{x}_i \in \mathcal{R}^K, i = 1, \ldots, M\}$. To describe the domain of the target set, we enclose the data by a hypersphere of minimal volume. Let the hypersphere be described by the center $\mathbf{a} \in \mathcal{R}^K$ and the radius R. A graphical representation for a 2D case is shown in Fig. 1.

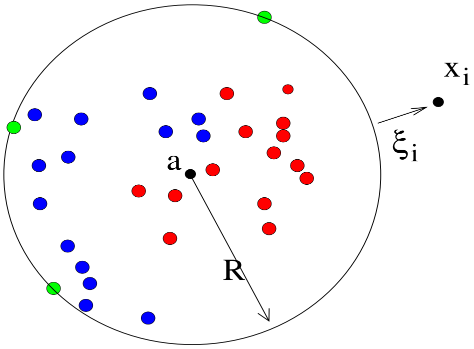


Fig. 1. Graphical representation of the (hyper)sphere around some target data in a 2D space. One vector $\mathbf{x}_i$ is rejected by the description (i.e. an error).

To permit some outliers in the training set, the distance from $\mathbf{x}_i$ to the center $\mathbf{a}$ must not be strictly smaller than R^2, but larger distances should be penalized. Therefore, slack variables ξ_i are introduced to measure the distance to the boundary. An extra parameter C has to be introduced for the trade-off between the volume of the hypersphere and the number of outliers. Now, we minimize L — both the radius of the hypersphere (and indirectly the volume) and the distance from the outliers to the boundary, requiring that (almost) all the data is inside the hypersphere:

$$\text{Min.} \quad L(R, \mathbf{a}, \boldsymbol{\gamma}) = R^2 + C \sum_i \xi_i \tag{2}$$

$$\text{s.t.} \quad \|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \xi_i\,, \quad i = 1, \ldots, M\,. \tag{3}$$

The constrains (3) can be incorporated into L (2) by applying Lagrange multipliers $\boldsymbol{\alpha}$ and optimizing the Lagrangian.[2] Then, the center $\mathbf{a}$ can be expressed in terms of the $\boldsymbol{\alpha}$ and the data vectors $\mathbf{x}_i$ as[15] $\mathbf{a} = \sum_i \alpha_i \mathbf{x}_i$ with $0 \leq \alpha_i \leq C$, $\forall\, i$ and $\sum_i \alpha_i = 1$. In practice, it appears that many α_i become zero. The vectors $\mathbf{x}_i$ corresponding to the positive α_i are then called *support vectors*, since they appear to lie on the boundary (in Fig. 1 these are marked by three light gray circles). Since $\mathbf{a}$ depends just on a few support vectors, the remaining vectors can be disregarded. The radius R is determined by calculating the distance from the center $\mathbf{a}$ to any support vector $\mathbf{x}_i$ on the boundary. Then, a vector $\mathbf{z}$ is accepted by the SVDD if:

$$\|\mathbf{z} - \mathbf{a}\|^2 = (\mathbf{z} \cdot \mathbf{z}) - 2 \sum_i \alpha_i (\mathbf{z} \cdot \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \leq R^2\,. \tag{4}$$

Note that the model of a hypersphere will not be appropriate for a general case. Analogously to the method of Vapnik,[17] all the inner products in the form of $(\mathbf{x} \cdot \mathbf{y})$ can be replaced by the kernel functions $K(\mathbf{x}, \mathbf{y})$. Especially, the Gaussian kernel:

$$(\mathbf{x} \cdot \mathbf{y}) \to K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / s^2) \tag{5}$$

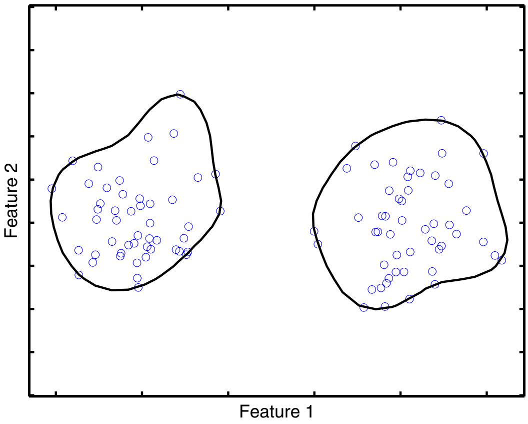


Fig. 2. An inhomogeneous cloud of points with the boundary determined by the SVDD with a Gaussian kernel.

provides a good data transformation.[15] This yields a much more flexible description. Please note, that a single nonlinear kernel, such as the Gaussian kernel, is able to detect several clusters in the data as illustrated in Fig. 2.

This Gaussian kernel (5) contains an extra free parameter, the width s, which influences both the complexity and tightness of the boundary. For a small s, the SVDD resembles a Parzen density estimator, while for a large s, the original hypersphere is obtained.[16] As shown in Ref. 16, s can be set with the maximally allowed rejection rate p_{rej} on the target set.

For the trade-off parameter C, a new variable $\nu = \frac{1}{MC}$ is defined, which describes an upper bound for the fraction of target vectors outside the description.[13] When the user specifies beforehand the rejection rate p_{rej}, just either s or ν can be determined. Therefore, here, we set ν to a fixed value of 1%. The value of s is then optimized such that the user-specified fraction p_{rej} of the target data is rejected.

3. Proximity in the Context of Image Retrieval

Let us denote by I_D an image database with N images I_i, $i = 1, \ldots, N$. The image retrieval problem is formulated as a selection of images, which are the most similar to the given query image I_Q. Such a retrieval strategy is roughly defined by two ingredients: (1) an image representation and (2) a proximity measure between the query image and the images stored in the database. The notion of a proximity of two images plays then a key role. On this basis, the images are judged similar to the query and, therefore, retrieved by the system. In the following sections, we will first describe the proximity criterion that we choose. Then, we will describe the retrieval process based on a direct ranking of the image proximities. Finally, we will introduce our combining strategy used to perform the retrieval search.

3.1. *Proximity criterion*

Assume that a cloud C_i, consisting of M_i feature vectors, represents the image I_i. A cloud of points can be then described by the SVDD. Let B_i^{SVDD} be a one-class classifier constructed for the image I_i. For a vector $\boldsymbol{x}$, coming from the cloud C_i, $\boldsymbol{x} \in C_i$, it is defined as:

$$B_i^{\mathsf{SVDD}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ is accepted by the SVDD}\,, \\ 0 & \text{if } \boldsymbol{x} \text{ is rejected by the SVDD}\,. \end{cases} \tag{6}$$

In order to train the SVDD classifier, see Sec. 2.2, the user has to specify the fraction of target vectors p_{rej} that will lie on the boundary, i.e.:

$$\mathrm{Prob}(B_i^{\mathsf{SVDD}}(\boldsymbol{x}) = 0 \;\&\; \boldsymbol{x} \text{ is on the boundary} \,|\, \boldsymbol{x} \in C_i) = p_{\mathrm{rej}}\,. \tag{7}$$

This means that the boundary vectors are here considered as outliers.

The directed dissimilarity between the image I_i and the image I_j is defined as the fraction of points from the cloud C_i rejected by B_j^{SVDD} as follows:

$$d(I_i, B_j^{\mathsf{SVDD}}) = \frac{1}{M_i} \sum_{\boldsymbol{x} \in C_i} (1 - B_j^{\mathsf{SVDD}}(\boldsymbol{x}))\,. \tag{8}$$

The smaller the fraction of outliers $d(I_i, B_j^{\mathsf{SVDD}})$, the more similar the images I_i and I_j.

Alternatively, the clouds of points can be compared by using the Mahalanobis distance. The symmetric Mahalanobis distance between two images I_i and I_j is defined on their clouds as:

$$d_M(I_i, I_j) = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \Sigma_{i,j}^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)\,, \tag{9}$$

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

3.2. *Direct ranking*

First, we will describe how to perform image classification (or retrieval) using the SVDD representation of images. Later we will show how the same criteria are applied to the Gaussian representation of the image clouds.

For a given database of N images, the cloud representations C_i as well as the corresponding SVDD classifiers B_i^{SVDD} are available. Our reasoning starts from the $N \times N$ matrix $D = (d_{ij})$; see Fig. 3. The rows of D point to the image clouds of points, while the columns refer to their SVDD classifiers. Therefore, the generic element $d_{ij} = d(I_i, B_j^{\mathsf{SVDD}})$, computed by Eq. (8), stores the fraction of points from the cloud C_i rejected by the classifier B_j^{SVDD}.

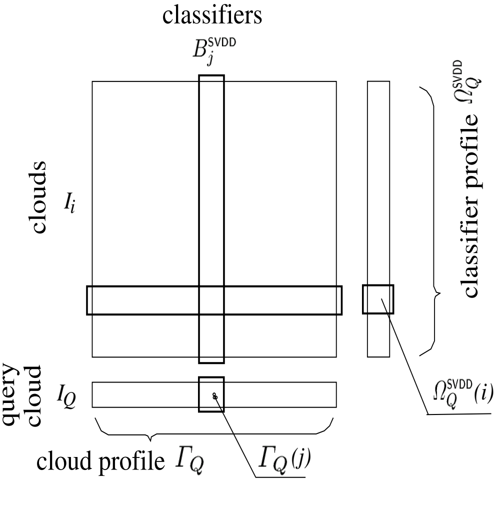


Fig. 3. Combination scheme for image database retrieval.

The relations between an image and the remainder of the database can be evaluated in two ways. The first one relies on the row information of the dissimilarity matrix D, which we call a *cloud profile*. The profile Γ_i shows how well the cloud C_i fits to the boundary of all one-class classifiers.

The cloud profile for the query image I_Q is defined as follows; see also Fig. 3:

$$\Gamma_Q = [d(I_Q, B_1^{\mathsf{SVDD}}), d(I_Q, B_2^{\mathsf{SVDD}}), \dots, d(I_Q, B_N^{\mathsf{SVDD}})]\,, \quad i = 1, \dots, N\,. \tag{10}$$

This vector evaluates the responses of the query cloud to all the SVDDs in the database. By ranking of the query cloud profile Γ_Q, the classifiers with the lowest fraction of outliers (the smallest dissimilarities) are identified and the corresponding images are returned as the most resembling the query.

The second viewpoint uses a *classifier profile* Ω_j^{SVDD} based on the jth column in the matrix D. It expresses the dissimilarities of all image clouds to the classifier B_j^{SVDD}, i.e. the fraction of the image clouds rejected by B_j^{SVDD}. As Fig. 3 shows, a classifier profile Ω_Q^{SVDD} of the query image is defined as:

$$\Omega_Q^{\mathsf{SVDD}} = [d(I_1, B_Q^{\mathsf{SVDD}}), d(I_2, B_Q^{\mathsf{SVDD}}), \dots, d(I_N, B_Q^{\mathsf{SVDD}})]\,, \quad i = 1, \dots, N\,. \tag{11}$$

This vector presents the responses of the query classifier to all the clouds in the database. By ranking of this profile, we find out which clouds are better accepted by the query classifier, and therefore, judged similar to the query image. Note that since D is asymmetric, the classifier profile Ω_Q^{SVDD} and the cloud profile Γ_Q differ.

Please note that eventually, the ranking procedure uses only a column or a row vector of the complete matrix D.

For defining the image proximities based on the Mahalanobis distances, the same strategies as described above, can also be used. The difference is that the matrix D describes now the Mahalanobis distances $d_M(I_i, I_j)$ instead of $d(I_i, B_j^{\mathsf{SVDD}})$. Note, that such matrix D is symmetric and, consequently, the cloud and classifier profiles are alike.

3.3. *Combining strategy*

In such a retrieval context, a high overlap of clouds representing semantically different images (or from different classes) may be problematic. For instance, in the SVDD approach, it may happen that one SVDD boundary completely contains another one, originating from a different image class. If a new query cloud is applied to both boundaries and is surrounded by the smaller one, it will also be accepted by the larger boundary. Therefore, the two images will be both considered equally similar to the query image even if they come from different classes. This, of course, lowers the performance of image retrieval based on direct ranking. To overcome this problem, we propose to combine the information given by classifiers in the profile. The query cloud profile is now compared to the cloud profiles of other images in the database. This means that a proximity of two images is now defined in a new way, as a similarity between their cloud profiles. For this purpose, different dissimilarity measures can be used, for instance, the Euclidean distance:

$$D_E(I_Q, I_i) = \|\Gamma_Q - \Gamma_i\|\,, \quad i = 1, \ldots, N\,, \tag{12}$$

or the cosine distance, based on the inner product between the cloud profiles:

$$D_{\cos}(I_Q, I_i) = \frac{1}{2}\left(1 - \frac{(\Gamma_Q)^T\,\Gamma_i}{\|\Gamma_Q\|\,\|\Gamma_i\|}\right), \quad i = 1, \ldots, N\,. \tag{13}$$

In this way, the responses of individual SVDDs are combined to express the dissimilarity between the query and the images in the database. The images, most similar to the query, are retrieved by ranking of the dissimilarities $D_E(I_Q, I_i)$ (or $D_{\cos}(I_Q, I_i)$).

This approach is similar to the decision process based on multiple classifiers, proposed by Kuncheva *et al.*,[5] where the decision templates are created by averaging over all the training objects in the class. In our experiments, individual classifiers are constructed for all single images in the database.

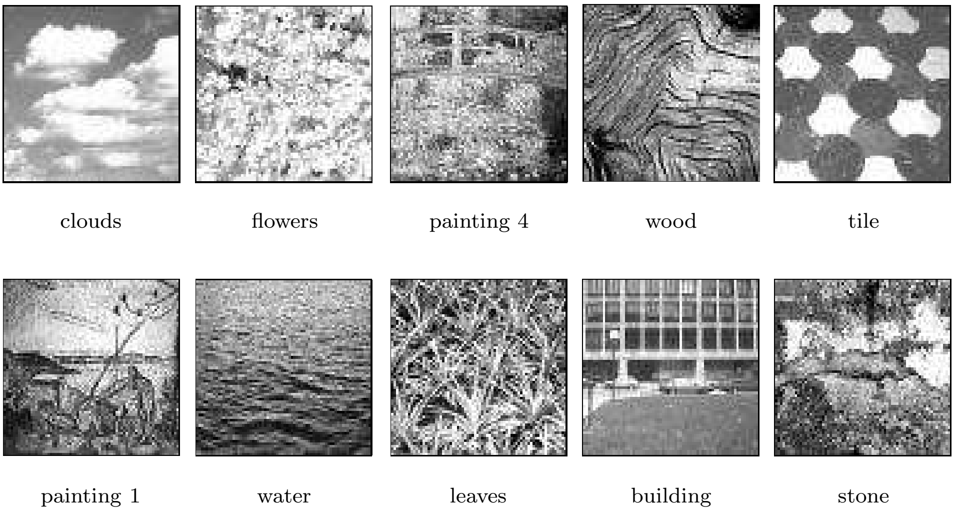
Similarly to the method of cloud profiles, the entire classifier profile, consisting of the responses of a chosen SVDD to all the image clouds, can be combined. The images are then compared by evaluating the dissimilarities between the classifier profiles. These are again based on the Euclidean or cosine distances, as previously defined in Eqs. (12) and (13), where Γ_Q and Γ_i are now replaced by Ω_Q^{SVDD} and Ω_i^{SVDD}, respectively.

In the introduced combining schema, the number of classifiers or image clouds in a profile is as large as the number of images in the database. This is not essential, since a profile with a smaller set of classifiers or clouds may be used as well, by using a concept of representation sets.[11] In this way, the computational complexity can be significantly reduced. This aspect is investigated further in our experiments; see Sec. 4.

In order to investigate the effects of combined dissimilarities for the Mahalanobis set up, the above strategy will be applied to the distance matrix $D = (d_M(I_i, I_j))$. Note, however, that due to the symmetry of D, we can restrict it to the cloud profiles only.

4. Experiments

In this section, we will describe a set of experiments for the problems of image classification and retrieval. In our first application, Sec. 4.1, images in the database are 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 semantically similar images. In the context of image classification, the retrieval strategy can be tested in a more objective way, which is our goal here. However, in Sec. 4.2, we also describe an experiment referring to the image retrieval problem. There, only some classes are specified leaving most images unlabeled.



clouds flowers painting 4 wood tile

painting 1 water leaves building stone

Fig. 4. Examples of images from the MIT database.

In our experiments, the Matlab toolbox `dd_tools`[a] has been used for the computation of the SVDD classifiers.

4.1. *Image classification problem*

In this section, we describe a set of experiments performed on a database of images organized into classes. First, we compare several strategies for computing and combining similarities between image representations with respect to the retrieval performance. Later, we investigate possible ways of reducing the computational complexity of the image classification problem and finally, we discuss the results.

4.1.1. *Experimental set-up*

Our experiments are based on 23 512 × 512 mostly homogeneous images obtained from MIT Media Lab.[b] Each original image is cut into 16 128 × 128 nonoverlapping pieces representing a single class (see Fig. 4). Therefore, we use a database of 368 images organized in 23 classes.

The image features used are the absolute values of the responses of 10 different Gabor filters. These 10 features were chosen by a backward feature selection from a larger set of 48 Gabor filters with different smoothing, frequency and direction parameters. A cloud is composed of 500 vectors randomly taken from the image, each vector being an average of 9 × 9 pixel neighborhood. The choice of 500 is a compromise between a higher noise sensitivity and the computational complexity.

[a] `http://www-ict.ewi.tudelft.nl/∼davidt/dd_tools.html`

[b] `ftp://whitechapel.media.mit.edu/pub/VisTex/`

The images of the database are, one by one, considered as the queries. The retrieval precision is computed using all 368 images. For each query image, 16 most similar images are found. The retrieval precision is then defined as the average fraction of images originating from the same class as the query, i.e.:

$$P = \frac{1}{368} \sum_{I \in I_D} \frac{\#\,\text{images of the same class as } I \text{ in the first 16 retrieved}}{16} \cdot 100\% \,. \tag{14}$$

4.1.2. *Evaluation of the classification system*

In this set of experiments, we evaluate the performance of the basic classification system. The results are summarized in Table 1. First, we investigate the behavior of the SVDD. We build the SVDD for the cloud of points, setting 20% of the points to the boundary, i.e. $p_{\text{rej}} = 0.2$; see Eq. (7).

In the approach of cloud profiles, the responses of the classifier to the query cloud form a cloud profile, as described in Sec. 3.3. It can be seen from Table 1 that the direct ranking gives a low performance of 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 precision of 81.4% and 81.6% using the Euclidean and the cosine distance, respectively.

In the second approach, a single SVDD trained on the query cloud is used and applied to other image clouds. Ranking in the classifier profile yields the precision of 72.0%. Combining the classifier profiles leads to a precision of 80.4% and 81.0%, again for the Euclidean and the cosine distance, respectively.

The difference in precision of the ranked SVDD with respect to the cloud and classifier profiles is caused by the fact that usually $d(I_i, B_Q^{\text{SVDD}}) \neq d(I_Q, B_i^{\text{SVDD}})$. Moreover, note that $d(I_i, B_i^{\text{SVDD}}) = 0.2$ (by our setup), so it may happen that $d(I_j, B_i^{\text{SVDD}}) < 0.2$ for some other image $I_j \neq I_i$, especially if the cloud of the I_j is (mostly) inside the cloud I_i. Apparently, in a number of cases, the query cloud

Table 1. Experimental results: precision of different retrieval methods on the MIT database.

Image Representation	Method	Precision [%]
SVDD/Cloud profile	single cloud	58.9
	combined (Euclidean)	81.4
	combined (cosine)	81.6
SVDD/Classifier profile	single classifier	72.0
	combined (Euclidean)	80.4
	combined (cosine)	81.0
Mahalanobis distance	single cloud	78.9
	combined cloud profile (Euclidean)	61.6
	combined cloud profile (cosine)	70.0

highly overlaps with other image clouds, hence the ranking of cloud profiles gives a worse performance than the ranking of classifier profiles.

It appears that the combination of the SVDD classifiers, both cloud and classifier profiles, yields a significant improvement with respect to a single SVDD. We think this large benefit occurs due to combining weak SVDD classifiers with a high variance.

As described in Sec. 3, the image clouds can be compared by the Mahalanobis distances. A direct ranking in these cloud profiles gives a precision of 78.9%. This good result can be explained by the homogeneous character of most images, resulting in almost normally distributed clouds. By applying the proposed combinations of cloud profiles, we obtained the precision of 61.6% and 70.0% for the Euclidean and cosine distance, respectively. Contrary to the SVDD classifier, combining Gaussian models lowers the retrieval performance with respect to ranking.

4.1.3. *Classifier selection in combining cloud profiles*

In the previous set of experiments, the cloud profiles were built using all available SVDDs. We suppose that the information given by all classifiers is redundant, e.g. due to homogeneity within the classes. Therefore, only a subset of the images ("prototype" images) may be chosen to be first described by the SVDDs and then used to build a profile. In this section, we evaluate three selection criteria generating a robust subset of SVDD classifiers. Following selection procedures use the class information and are, therefore, suitable only for image classification problems. Random selection method, appropriate for image retrieval, is discussed in the next section.

In order to have a general testing procedure we need independent training and test sets. We build a test set with 23 images, each coming from a different class and the training set with remaining 345 images. The precision formula in Eq. (14) is updated accordingly.

The first approach is a *systematic* search for relevant classifiers. One by one, the one-class classifiers are removed and the performance of remaining SVDDs is computed. The classifier with the highest score is 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 length of the profile itself. We choose to evaluate the performance starting with the total amount of 345 classifiers and continue until two. This is essentially a backward selection, considering responses of individual SVDDs as features. The algorithm even allows the removal of all SVDDs of an entire class.

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 23 SVDDs, one for each class, is removed at once. Different combinations are tested to find the least relevant set, which, once removed, leads to the highest performance.

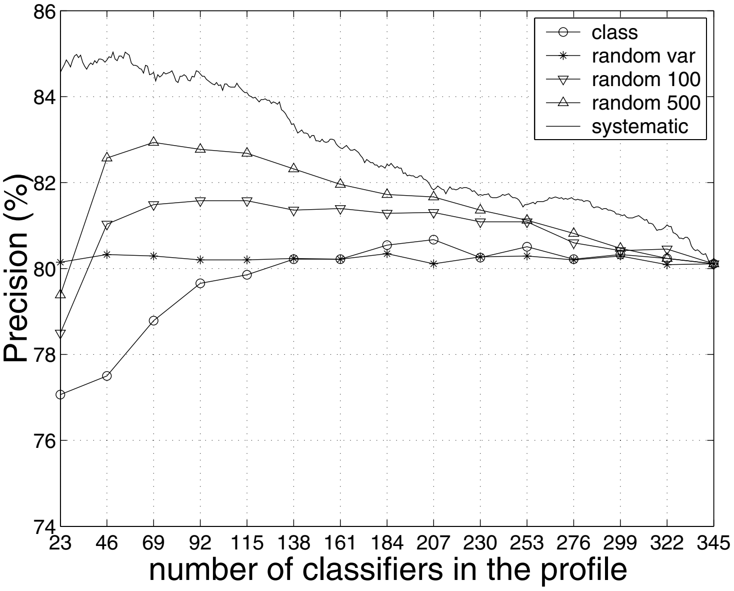


Fig. 5. The performance estimates of three selection criteria as a function of the number of SVDDs in the profile.

In order to obtain profiles of a small size, this process is iterated 16 times, starting from a complete profile with 345 SVDDs until the set of 23 SVDDs is reached.

In order to figure out whether an equal representation of classes is desirable, we implemented a *random* approach. A randomly chosen subset of classifiers with the same size as in the previous approach is removed at once. Also in this case, the process is iterated 16 times. The random selection is constructed such that it is comparable to the class approach.

Figure 5 presents the performance as a function of the number of SVDDs in the cloud profile. It clearly shows that by reducing the number of classifiers in the profile, the performance increases. The systematic approach reaches the optimal performance for about 50 classifiers. A smaller profile is also more efficient with respect to the computational complexity and memory requirements.

The class approach yields, on the other hand, the worst performance, because not all the classes are equally representative, as discussed in Sec. 5. Therefore, it is not useful to enforce a regular class organization in the selection process.

The performance of a random approach is in between the results of the class and the systematic methods. The line denoted *random var* in the graph, corresponds to the random selection, which is comparable to the class approach. 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.

Fig. 6. Examples of images in the image retrieval experiment.

Nevertheless, the computational complexity is much lower (*systematic*: $345 \cdot 344/2 = 59,340$, *random 500*: $16 \cdot 500 = 8,000$ criteria evaluations).

4.2. *Image retrieval problem*

In this section, we study further the combination of cloud profiles in the context of image retrieval. Our experiments are based on the Surrey image database,[9,10] which originally contains 3,483 various images. Here, we selected a smaller database created by the first 500 images. These images describe various scenes from the television news like people in the city, buildings, trains, mountains, sea shores, etc.; see also Fig. 6 for some examples.

For each image in the database, 12 color and 21 texture features are computed[9,10] such as: energy, entropy, mean and variance in each of the R, G and B channels and the discrete cosine, Gabor and wavelet transforms. We follow the same type of preprocessing as in the previous experiment by forming a cloud of 500 randomly selected points, now in the 33-dimensional space. To avoid the scale dependency, all the data is normalized in the same way as described in Sec. 2.1.

In the experiments, described below, we perform image retrieval on all 500 images. In order to evaluate the retrieval performance, we identified three classes within this database. The classes are: `news:` news readers presenting the news (24 images), `beach:` sandy beaches (12 images) and `sea:` (8 images). These are still semantically well-defined classes for a human observer, however, broad in variability of scenes. The `sea` class is the most homogeneous, since the images present very similar scenes: the sea, waves and the rocky shore. The `news` class shows a moderate variability due to different closeups and the number of people present. The `beach` class is the most heterogeneous, since it describes a large variability of activities on the beach, like people sun-bathing or playing volleyball (see the rightmost image in the bottom row in Fig. 6 for an example).

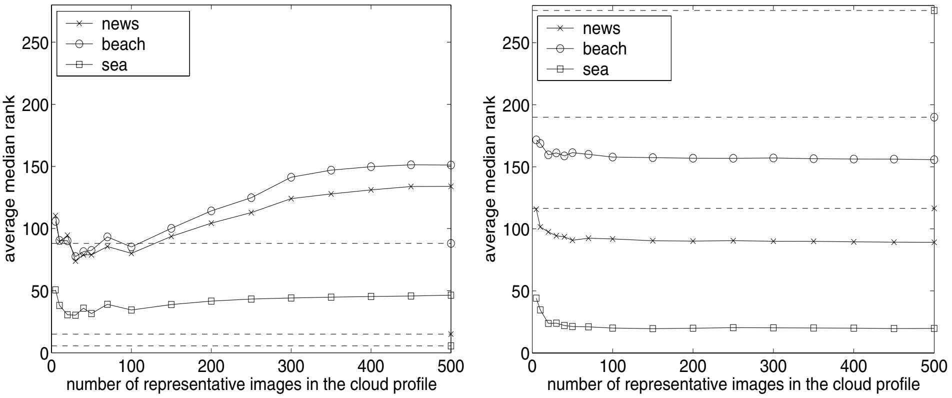
Note that studying the retrieval of images for this database is significantly different from the classification problem considered before. Even though three classes are created, they are only agreed upon for this evaluation task. Moreover, many other images in the database exist that contain similar type of information e.g. semantic (like a single person or large object) or in color (like a blue background) as the images assigned by us to classes. Hence, the whole setup points in the direction of image retrieval.

4.2.1. *Evaluation of the retrieval system*

All the experiments in this section are performed using the cloud profiles for both the SVDD responses and the Mahalanobis distances. The SVDD is built by setting 20% of the points to the boundary, i.e. $p_{\mathrm{rej}} = 0.2$; see Eq. (7).

The images of the three classes are, one by one, considered as queries. For each query, we rank the positions of all retrieved images and store the median rank of the images from the same class. Then, we compute the average over the median ranks, denoted as $\bar{r}_{\mathrm{med}}$ for each class. Note that in a perfect case, the averaged median ranks are: 13 for the `news` class (23 images to rank for each query in this class), 6 for the `beach` class and 4 for the `sea` class.

Our retrieval experiments are based on the cloud profiles. As before, we use two approaches. In the first one, we rank the entire profiles directly and in the second one, we combine the profiles by the Euclidean distance and then rank the Euclidean distances between the query and other images in the database. Since we claim that only some images are sufficient to build the cloud profiles, we perform our study on images randomly selected from the database to be used in the cloud profiles. Because the class labels are not available for all the images in the database, a systematic selection is impossible. In order to study the behavior of different methods we



(a) Gaussian model

(b) SVDD model

Fig. 7. The averaged median rank $\bar{r}_{\mathrm{med}}$ for the three classes: `news`, `beach` and `sea` versus n_{repr}, the number of representative images in the cloud profile. Dashed lines refer to the ranking procedures, while the solid lines correspond to the combined cloud profiles. The results are averaged over 30 repetitions.

analyze the averaged median rank $\bar{r}_{\text{med}}$ for each class as a function of the number of images randomly selected to represent the cloud profile n_{repr}. The results are averaged over 30 repetitions. This is presented in Fig. 7.

A number of observations can be made from our investigations. First of all, we see that in the Mahalanobis case the direct ranking gives very good results: $\bar{r}_{\text{med}} = 5.5$ for the `sea` class, $\bar{r}_{\text{med}} = 15$ for the `news` class and $\bar{r}_{\text{med}} = 88$ for the `beach` class. Then, we observe that the combination of the cloud profiles does not improve the results, except for the `beach` class and 30–50 representative images in the cloud profile. Moreover, with the growing size of the cloud profile n_{repr}, the combining yields increasingly worse performance. Concerning the SVDD classifiers, the direct ranking attains bad results, which can be significantly improved by the combined cloud profiles. Still, the best results in the SVDD case are worse than direct ranking in the Mahalanobis case.

5. Discussion

The cloud representation of images offers a good retrieval performance on both studied datasets. However, both investigated models behave differently. The Gaussian model is based on the density and, therefore, focuses on a global description. The SVDD concentrates on the boundary description, instead, and it is more sensitive to local characteristics of the cloud. This sensitivity is directly linked to the fraction of points set up to lie on the boundary p_{rej}. Higher values of this parameter make the boundary tighter and more complex, low values make it wider and more smooth.

The retrieval capabilities of both models may be better understood by looking at the distances between images inside and outside of a class. Table 2 summarizes these values for the Surrey database. For each labeled image we computed mean and standard deviation of distances to the remaining images of its class and to all other images. Numbers in the table are then averaged within each class.

It follows from Table 2 that intra-class Mahalanobis distances are very small compared to the distances to other images. This explains why the direct ranking performs well. High standard deviations of the distances from the query to other images suggest that the values in the profile are too scattered and do not provide a

Table 2. Means and variances of distances within the class of the query and to all other images in the Surrey database.

	Gaussian Model				SVDD Model			
	Inside Class		Outside Class		Inside Class		Outside Class	
	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.
`news`	50.47	26.36	285.06	1616.28	65.59	12.97	66.61	18.98
`beach`	150.80	235.26	502.01	2610.05	76.33	10.74	73.57	16.86
`sea`	80.52	38.25	684.52	3048.18	90.93	5.65	71.18	15.16

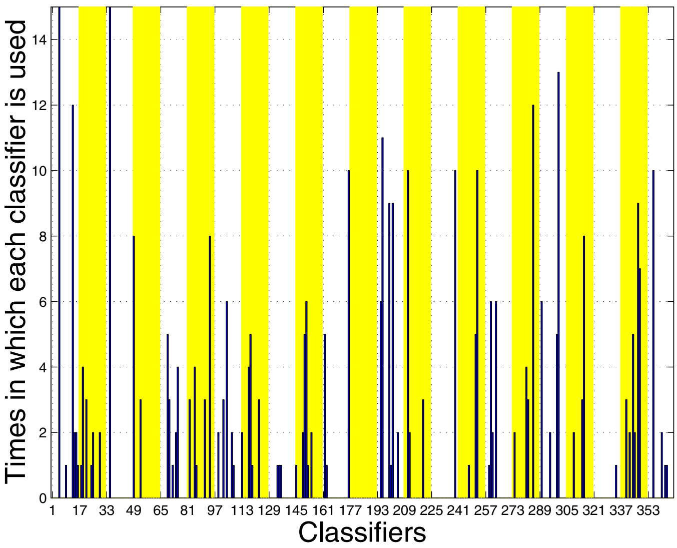


Fig. 8. Classifiers held in the profiles as the best 23 by the 16 test sets.

distinct class pattern. That is probably the reason for a low combining performance (see Fig. 7(a)). Figure 9, in the first two columns, presents images retrieved by the direct ranking and by the combined profiles in the case of the Gaussian model. The image of a news reader, enclosed in a thick frame, was used as a query. The first eight images, ranked as the most similar to the query are given. Images from the same class as query (`news`) are marked with a black square (■). While the direct ranking gives almost perfect results, combining the cloud profiles returns three unrelated images. We have also observed a low combining performance on the MIT database with more homogeneous images (see Table 1).

In case of the SVDD model, the situation is different. Although the direct ranking shows very poor results, the performance is significantly improved by combining. As it can be seen in Table 2, intra-class and extra-class distances are comparable. The information in the profile is spread more uniformly over a number of images. A single SVDD boundary is, therefore, a weak classifier which explains a low ranking performance (the larger p_{rej}, the weaker the classifier). Responses of several classifiers still convey a pattern, specific for a given class. Because of that, combining the SVDD responses is beneficial.

An example of the retrieval based on the SVDD is given in the third and fourth columns of Fig. 9. Please note that the image, most similar to the query, is not the query itself. Remember that the image self-dissimilarity $d(I_i, B_i^{\mathsf{SVDD}})$ is set by using p_{rej} to 0.2 (see Eq. (8)). A query cloud attains zero (or close to zero) dissimilarity to an image, if it is completely contained in the boundary of this image. That happens for the first returned image in Fig. 9. Ranking of the SVDD responses finds, apart from the query, only one image from the `news` class. By combining the cloud profiles,

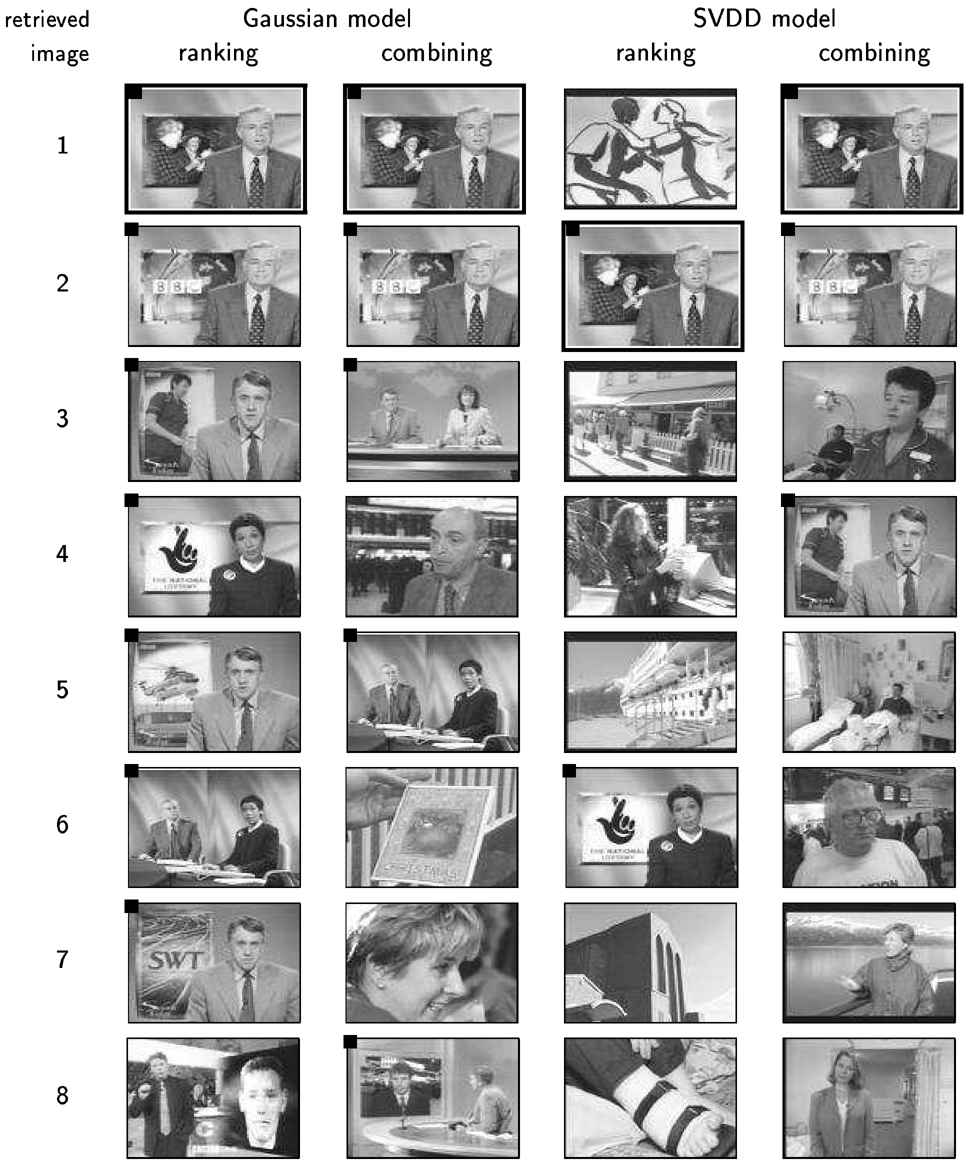


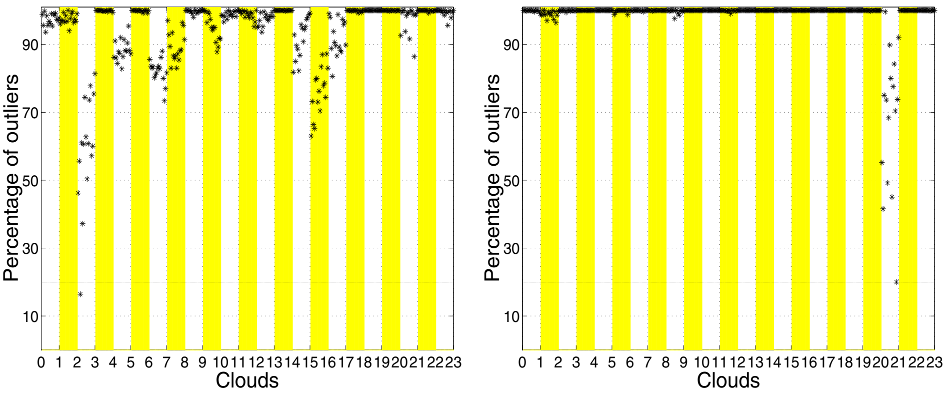
Fig. 9. Images retrieved by different methods from the Surrey database. The same image was used as a query in all four cases and was emphasized by the thick frame. Rows represent first eight retrieved images. The images, denoted by the black square (■) come from the same class as the query (`news`). Combining was performed with 100 randomly selected images in the profile.

two relevant images are found. Note that most other responses also show a similar pattern: a closeup of a person.

Let us now discuss how the profile size influences the retrieval or classification performance. A general conclusion that we draw from our experiments is that using a smaller profile is preferable. In the retrieval experiment on the Surrey database, it appears that the performance is stable in a broad range of profile sizes for SVDD model. We have observed the same trend also for different settings of the SVDD model (not shown here). In case of Gaussian model, the performance is improved by using smaller profiles. This is in agreement with our previous finding that the profile is not much informative. Thus, reducing its size leads to a noise reduction and consequently to better retrieval performance.

Because the MIT database contains complete class information, we may use systematic reduction of the profile size instead of the random selection. As described in Sec. 4.1, systematic selection leads to a significant improvement of classification performance. The results of the systematic selection are presented in Fig. 8 which help us to evaluate how representative are individual classifiers. On the x-axis are listed all possible SVDDs in the database. 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 of the class *painting1* (with indices between 177 and 192) are never used.

Figure 10 shows two classifier profiles with a different combining performance. The x-axis represents all the clouds, grouped according to the class of similarity. The y-axis shows the fraction of outliers when the given SVDD is applied. Figure 10(a) corresponds to the classifier number 35 (class *building*) which was chosen in a systematic selection maximum number of times. It means that this classifier forms, together with other selected SVDDs, an informative profile. The classifier number 334 from the class *water* (Fig. 10(b)) was, on the other hand, selected just once. This



(a) SVDD from the class *building*

(b) SVDD from the class *water*

Fig. 10. Classifier profiles for two images from the MIT database. Each profile shows the fraction of outliers of all images (i.e. clouds) in the database applied to a single SVDD boundary.

classifier is thereby not informative for combining but perfect for direct ranking. The reason is that it rejects as outliers the clouds from all other classes. Such behavior is, however, exceptional as clouds of points are usually overlapping.

5.1. *Discussion on complexity issues*

Here, we will present rough estimates of the image retrieval complexity for the Gaussian and SVDD models. The following parameters are taken into account:

N	the number of images in the database (possibly large),
M	the number of points in the cloud representation,
K	the dimensionality of the feature space in which the points reside,
n_{repr}	the size of the reduced profile used in combining,
Mp_{rej}	the number of support vectors defining the boundary of the SVDD.

In the case of the Gaussian model, the mean vectors and the covariance matrices for N images in the database are stored. In the case of SVDD model with the cloud profiles, the SVDD boundaries based on Mp_{rej} vectors are precomputed.

In the following we assume that the query image is not present in the database. For the Gaussian model, the computation of the mean vector and the covariance matrix for the query cloud is required. Then, the Mahalanobis distances between the query cloud and N other image clouds have to be computed. A rough estimation of the retrieval complexity, assuming that $N > M$ is then $O(NK^3 + N\log N)$ for a direct ranking and $O(n_{\mathrm{repr}}K^3 + NK^2 + Nn_{\mathrm{repr}} + N\log N)$ by using the combined profiles. The K^3 component appears due to the inversion of the covariance matrix, and $N\log N$ is related to the sorting of N proximity responses. We also assume that for combining, only n_{repr} images are used to form a profile. The component Nn_{repr} corresponds to the computation of the Euclidean distances between the query and N other profiles. It is not yet clear to us whether n_{repr} is independent from N. If it is set to a fixed fraction of N, the estimated complexity for the combining case can be simplified to $O(NK^3 + N^2)$.

For the SVDD model, we analyze how this cloud fits to n_{repr} SVDD boundaries. This requires the computation of n_{repr} fractions of the query cloud being not accepted by the SVDD classifiers. The complexity of this operation is $O(n_{\mathrm{repr}}$ $M^2p_{\mathrm{rej}}K)$, which reduces to $O(n_{\mathrm{repr}}M^2)$, since $p_{\mathrm{rej}}K$ can be considered as negligible. Assuming that $K < M$, the rough estimation of the retrieval complexity is $O(NM^2 + N\log N)$ in case of ranking and $O(n_{\mathrm{repr}}M^2 + Nn_{\mathrm{repr}} + N\log N)$ in case of combining. Note that the Nn_{repr} term comes due to the computation of the distances between the profiles. If, as before, n_{repr} is considered to be in the order of N, then the latter expression simplifies to $O(NM^2 + N^2 + N\log N)$.

In case the query comes only from the database, everything can be precomputed. This means that both the dissimilarity matrix consisting of the cloud profiles and the proximity matrix of e.g. Euclidean distances between the cloud profiles can be stored. In such a case, the retrieval process relies either on ranking of an entire

Table 3. Retrieval times in seconds for the experimental implementation (1GHz PC).

	Speed of Image Retrieval [sec]			
	Gaussian Model		SVDD Model	
Size of the Database	Ranking	Combining	Ranking	Combining
500 images	0.54	0.24	22.47	4.53
5,000 images	4.00	0.38	224.00	44.60

cloud profile or on the ranking of distances between the profiles, which both have the complexity of $O(N \log N)$, independently from the chosen model.

To give more concrete evidence on the actual computational demands of the investigated approaches, we measured the times of image retrieval on our experimental implementation (1GHz PC, Matlab). We consider a query image, not present in a dataset. Computing a feature representation takes about one second. The retrieval times are summarized in Table 3. For combined representations, we fixed the number of prototypes to 100. The results for a dataset with 5,000 images are extrapolated using the measurements on 500 images. Please note, that combining requires less time than direct ranking. It is because the most time consuming step is comparing image models (computing Mahalanobis distances or applying the SVDD boundary to a query cloud). For the direct ranking, the full set of dissimilarities must be computed from the query cloud to all images in a dataset. In case of combining, however, these expensive dissimilarities are computed only between the query cloud and a set of prototypes, here 100 images. The computationally cheaper cosine distance is then evaluated between the 100-dimensional profile and all the stored profiles in the database. The proposed combination strategy using cloud profiles is, therefore, attractive from a computational point of view.

6. Summary

The performance of an image retrieval system depends on an appropriate representation of image data. A possible rich description of images for the sake of retrieval is a cloud of points representation. In this study, we investigate the merits of combining several cloud representations.

In order to retrieve images, represented by clouds of points, a method for measuring the similarity between the clouds must be defined. The first of the two approached, studied by us, describes a cloud of points by the support vector data description (SVDD) method. In contrast to the density-based methods, the SVDD describes the data domain in the feature space. By this approach, images can be easily matched based on the fraction of points rejected by the description (the smaller, the better). The second method is based on a Gaussian cloud model. This representation assumes normally distributed data and measures the dissimilarity between the Gaussian models using the Mahalanobis distance. Inhomogeneous images may, however, violate the assumption of normality.

This study focuses on combining image representations for retrieval purposes. The responses of several cloud descriptions convey a pattern, specific for semantically similar images. Based on this observation, we proposed the combining strategy utilizing a set of these responses, which we call a profile. We have studied both the retrieval performance and properties of this combining approach compared to the direct ranking.

For this purpose, two different datasets are used. The first one contains 365 homogeneous images with known image classes. Therefore, experiments on these data treat the problem of image classification. The second database consists of 500 inhomogeneous images without known class information. For the sake of evaluation, we formed three semantically similar classes of 44 images in total. This allowed us to perform image retrieval experiments.

Our working hypothesis was that the Gaussian model would work better on homogeneous images (the MIT database), while the SVDD model would have outperformed it for inhomogeneous images (the Surrey database). It follows from our experiments that in case of the Gaussian model, the direct ranking gives very good results, which cannot be improved by combining. Surprisingly, the ranked SVDD model reaches poor results for inhomogeneous data. However, combining improves considerably its retrieval performance. The best overall results for homogeneous data (the MIT database) were reached by combining the SVDD models, while for inhomogeneous data, the direct ranking of the Mahalanobis distances was the best.

We observed that the representation of inhomogeneous images by the cloud of points leads to moderately multimodal clouds. The Gaussian model is able to describe these clouds very well by averaging out the effect of outliers and subclusters in clouds. The SVDD model captures the local shape of a cloud boundary, which makes it a weak classifier with a low ranking performance. This property makes it, however, suitable for combining.

Additionally, we have shown that reducing the profile size, even by a random selection, is beneficial for combining. It increases the retrieval performance and at the same time is computationally less intensive. These conclusions hold for relatively small image datasets, used in our experiments. How to select an informative profile for very large image databases is, however, a question for future research.

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, Proc. SPR'98 and SSPR'98* (IAPR, Springer-Verlag, Berlin, 1998) pp. 31–53.
2. C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, 1995).
3. T. Gevers, F. Aldershoff and A. W. M. Smeulders, Classification of images on internet by visual and textual information, in *Internet Imaging*, SPIE Vol. 3964, eds. G. B. Beretta and R. Schettini, 2000, pp. 16–27.
4. T. Huang, Y. Rui and S.-F. Chang, Image retrieval: past, present, and future, *Int. Symp. Multimedia Information Processing*, 1997.

5. L. I. Kuncheva, J. C. Bezdek and R. P. W. Duin, Decision templates for multiple classifier fusion: an experimental comparison, *Patt. Recogn.* **34**(2) (2001) 299–314.
6. C. Lai, D. M. J. Tax, R. P. W. Duin, El. Pękalska and P. Paclík, On combining one-class classifiers for image database retrieval, in *Multiple Classifier Systems*, *Proc. Third International Workshop MCS 2002*, Cagliari, Italy, June 24–26, eds. F. Roli and J. Kittler, 2002, Lecture Notes in Computer Science, Vol. 2364 (Springer-Verlag, Berlin).
7. M. S. Lew, *Principles of Visual Information Retrieval* (Springer-Verlag, London, 2001).
8. O. Maron and A. L. Ratan, Multiple-instance learning for natural scene classification, in *Proc. 15th Int. Conf. Machine Learning* (Morgan Kaufmann, San Francisco, CA, 1998), pp. 341–349.
9. K. Messer, Automatic image database retrieval system using adaptive colour and texture descriptors, Ph.D. thesis, University of Surrey, Guildford, 1999.
10. K. Messer and J. Kittler, A region-based image database system using colour and texture, *Patt. Recogn. Lett.* **20** (1999) 1323–1330.
11. El. Pękalska and R. P. W. Duin, Dissimilarity representations allow for building good classifiers, *Patt. Recogn. Lett.* **23**(8) (2002) 943–956.
12. Y. Rui, T. Huang and S. Mehrotra, Content-based image retrieval with relevance feedback in MARS, in *Proc. IEEE Int. Conf. Image Proc.*, 1997, pp. 815–818.
13. B. Schölkopf, P. Bartlett, A. J. Smola and R. Williamson, Shrinking the tube: a new support vector regression algorithm, *Advances in Neural Information Processing Systems*, eds. M. S. Kearns, S. A. Solla and D. A. Cohn, 1999.
14. A. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain, Content-based retrieval image retrieval at the end of the early years, *IEEE Trans. Patt. Anal. Mach. Intel.* **22**(12) (2000) 1349–1380.
15. D. M. J. Tax, One-class classification, Ph.D. thesis, Delft University of Technology, http://www.ph.tn.tudelft.nl/˜davidt/thesis.pdf, June 2001.
16. D. M. J. Tax and R. P. W. Duin, Support vector domain description, *Patt. Recogn. Lett.* **20**(11–13) (1999) 1191–1199.
17. V. N. Vapnik, *Statistical Learning Theory* (John Wiley, 1998).

Carmen Lai received the M.Sc. degree in electronic engineering from Cagliari University, Italy, in 2002. Currently, she is a Ph.D. student in pattern recognition applied in bioinformatics at the Delft University of Technology, the Netherlands.

Robert P. W. Duin studied applied physics at the Delft University of Technology, the Netherlands. In 1978, he received the Ph.D. for a thesis on the accuracy of statistical pattern recognizers. In his research, he included various aspects of the automatic interpretation of measurements, learning systems and classifiers. Between 1980 and 1990, he studied and developed hardware architectures and software configurations for interactive image analysis. At present, he is an associate professor at the Faculty of Applied Sciences of Delft University of Technology.

His current research interests include the design and evaluation of learning algorithms for pattern recognition applications. He runs several projects on applied and fundamental research in pattern recognition.

Pavel Paclík received his M.Sc. in transportation engineering from the Czech Technical University in Prague in 1998. He began work towards his Ph.D. in 1998, focusing on pattern recognition methods applied in intelligent transportation systems. Since 2000, he has been working as a research fellow in the Pattern Recognition Group at the Delft University of Technology in the Netherlands.

His interests are in pattern recognition methods applied in image analysis, especially in the design and computational aspects of data representations.

David M. J. Tax received the M.Sc. degree in physics from the University of Nijmegen, the Netherlands in 1996. In 2001, he received the Ph.D. at the Delft University of Technology, the Netherlands, for a thesis on the problem of one-class classification or novelty detection. Currently, he is working on a European Community Marie Curie Fellowship called "One-class classification" in the Intelligent Data Analysis group of Fraunhofer FIRST, Berlin, in close collaboration with the Delft University of Technology.

His research interests include pattern recognition and machine learning with a focus on outlier detection and novelty detection, the feature selection for and the evaluation of one-class classifiers.

Elżbieta Pękalska received the M.Sc. degree in computer science from Wroclaw University, Poland, in 1996. In 1997, she became a research fellow at the Delft University of Technology, the Netherlands, where in 1999, she started her Ph.D. research on dissimilarity-based pattern learning methods. She will be obtaining her Ph.D. in early 2005.

Her current research interests include various learning aspects in relation to data representation, nonlinear mappings and statistical and structural models, in particular.