Pattern Recognition 43 (2010) 650–662

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.de/locate/pr

Fixed partitioning and salient points with MPEG-7 cluster correlograms for image categorization

Azizi Abdullah [a,*], Remco C. Veltkamp [a], Marco A. Wiering [b]

[a] Department of Information and Computing Sciences, Utrecht University, Postbus 80089, 3508 TB Utrecht, The Netherlands
[b] Department of Artificial Intelligence, University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 16 July 2008
Received in revised form
29 August 2009
Accepted 7 September 2009

Keywords:
Cluster correlogram
Computer vision
Image indexing and retrieval

ABSTRACT

This paper compares fixed partitioning and salient points schemes for dividing an image into patches, in combination with low-level MPEG-7 visual descriptors to represent the patches with particular patterns. A clustering technique is applied to construct a compact representation by grouping similar patterns into a cluster codebook. The codebook will then be used to encode the patterns into visual keywords. In order to obtain high-level information about the relational context of an image, a correlogram is constructed from the spatial relations between visual keyword indices in an image. For classifying images a *k*-nearest neighbors (*k*-NN) and a support vector machine (SVM) algorithm are used and compared. The techniques are compared to other methods on two well-known datasets, namely Corel and PASCAL. To measure the performance of the proposed algorithms, average precision, a confusion matrix, and ROC-curves are used. The results show that the cluster correlogram outperforms the cluster histogram. The saliency based scheme performs similarly to the fixed partitioning scheme and the SVM significantly outperforms the *k*-NN classifier. Finally, we demonstrate the robustness to noise, photometric, and geometric distortions.

1. Introduction

The need for an efficient system to facilitate users in searching and organizing image collections in a large scale database is crucial. However, developing such systems is quite difficult, because an image is an ill-defined entity [26] consisting of complex and highly variable structures. In addition, digital images can be disturbed by geometric transformations, photometric transformations or other disturbance agents. Even though the images can be of a complex nature, it is not impossible to extract an approximation of the generic meaning from the complex data of images. One of the main issues addressed in finding images from large image collections is the quality of the retrieval results. It is common experience for the user to retrieve meaningless information from the query of digital images. Therefore, effective image representation and indexing in a large database are needed and so remain a challenge in computer vision research.

The most frequently cited image features found are color, texture and shape [29,6,4,16], but the most commonly used feature to represent images is color. The color histogram is the best known and most popularly used color feature in CBIR systems and is used in systems such as QBIC [14] and PhotoBook [23]. The color histogram is (almost) invariant to rotation, translation and scaling. This approach works well, especially in labeling the image content as a whole that the user is interested in (e.g., sunrises, sunsets, flowers, etc.), however it has problems when conveying image information that contains foreground and background objects and possible correlations between them. This is because computing the color histogram of the image and normalizing it destroys the spatial information aspect of texture patterns and only retains their brightness information, resulting in information loss and coarse indexing. Therefore, such indexing can potentially give false results on image queries, and sometimes two images with dramatically different semantics give rise to similar histograms.

To reduce the problem, Pass and Zabih [22] proposed a split histogram called color coherence vector (CCV). The results produced by this method are quite promising compared to a color histogram. Besides that, Huang et al. [16] proposed another kind of feature called the color correlogram that enables computation of the correlation between colors using spatial information in an image. The correlation is computed on the discrete domain, as a result the joint probability of certain sets of colors having certain values can be represented. However, these methods still could not fully solve the problem of fuzziness and primitiveness of the color features inherently exhibited in the color histogram. The color layout feature was also introduced to

* Corresponding author. Tel.: +31 30 253 9945; fax: +31 30 253 4619.
E-mail addresses: azizi@cs.uu.nl (A. Abdullah), Remco.Veltkamp@cs.uu.nl (R.C. Veltkamp), m.a.wiering@rug.nl (M.A. Wiering).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.09.007

overcome the drawbacks of a color histogram. In this method images are partitioned into several blocks and the average color of each block is calculated and combined [29]. However, the color layout is sensitive to shifting, cropping, scaling, and rotation, because images are represented by fixed blocks.

One way to overcome these problems is a technique that can localize and determine object positions in regions in an image. One region based approach tries to apply an image segmentation technique to extract regions from images [6]. Then, similarity between images are measured by calculating the correspondences between their regions. Typical examples of region-based retrieval systems include Blobworld [6], IRM [19], VisualSEEK [4], and SIMPLIcity [29]. However, it is quite difficult to achieve accurate segmentation in an image especially for images with less distinctive objects [4].

Besides image segmentation, another way to overcome the limitations of the global feature approach is to use the local appearance approach. This approach works by clustering feature vectors extracted from separate regions into similar group patterns. The approach shows remarkable performance in some applications as reported in [2,18,13,9]. The bag of words model is popular in this approach and works by computing a histogram for these patterns. However, this approach destroys spatial information and retains only their overall pattern distribution. Therefore, we believe that incorporating spatial information between patterns will enrich the semantic description of the visual recognition system. In addition, most of the recent studies are focusing on multiple image features for satisfactory results. Using multiple image features may help to recognize different semantics or structures of images efficiently. Furthermore, computing spatial information from multiple image features would be useful to increase the discriminative power of the recognition system. Following this, we propose and compare two different algorithms that can capture spatial information between patterns using the local appearance approach.

The contribution of our work is: (1) we present new methods to efficiently combine MPEG-7 descriptors with spatial information, (2) we compare the fixed partitioning scheme to the saliency-based scheme, (3) we demonstrate the effectiveness of combining the descriptors. Each MPEG-7 feature alone is not the best method to describe real world images, but an efficient combination of them can be, and (4) we compare two popular machine learning techniques for an automatic classification and categorization of real world images.

The rest of the paper is organized as follows. Section 2 describes fundamental principles of color and cluster correlograms. Section 3 describes our system for retrieving and categorizing images with focus on the MPEG-7 visual descriptors combined with cluster correlograms. Experimental results on the Corel and PASCAL datasets are shown in Section 4. Section 5 discusses the results, and Section 6 concludes the paper.

2. Fundamentals

2.1. Color correlogram

Spatial relations between colors are important for visual perception tasks. The color correlogram (or color co-occurrence matrix) is proposed by Huang et al. and they found that this feature is not only robust to variations in appearance, but can also tolerate some distortions of viewing position, background and occlusion [16]. The color correlogram enables computation of the correlation between colors by measuring the number of pairs of certain color pixels that occur at a certain distance and direction in an image.

Let $\mathcal{I}$ be an $N \times N$ image, where pixels are quantized into m colors $c_1, \ldots, c_m$. Let p be a pixel $p = (x, y) \in \mathcal{I}$, and let $p_1 \in \mathcal{I}_{c_i}$ mean that pixel p_1 is of color c_i. The color correlogram matrix $C^{\delta\varphi}$ of $\mathcal{I}$ is defined by the joint empirical probability on the image that a color c_i co-occurs with a color c_j at given distance δ and angle φ:

$$C^{\delta\varphi}(c_i, c_j) = \mathbf{P}(p_1 \in \mathcal{I}_{c_i} \wedge p_2 \in \mathcal{I}_{c_j} \wedge D(p_1, p_2) = (\delta, \varphi)). \quad (1)$$

Here $\mathbf{P}$ means probability, and $D(x, y)$ denotes a distance function using polar coordinates, where $\delta > 0$ and $\varphi \in [0, 2\pi]$. These two parameters are important to describe the coarseness of the micro textures [15]. Usually we take a small value for δ, since the correlation between pixels is more relevant for a small distance [25]. Finally, each image is indexed by a feature vector sized m^2 for each combination of δ and φ.

2.2. Cluster correlogram approach

The region-based approach is popular and widely used to represent local image content [6,4,29]. This approach is also believed to be efficient in terms of storage, complexity and effective in learning and indexing images. However, to be an effective method, a compact representation scheme that can describe all low-level visual features in the regions is required. By exploiting a vector quantization method such as a clustering algorithm, both compact and informative regions can be achieved. This algorithm is believed to be efficient in constructing a compact representation by grouping similar patterns into similar clusters or groups. Besides that, this information enables us to apply the correlogram approach to capture the statistical texture correlation with the distance and direction conditions in the image regions.

2.3. Fixed partitioning cluster correlogram

The fixed partitioning representation is described in [2,27]. In our fixed partitioning scheme, each image is divided into partitions of equal size as shown in Fig. 1 (left). We used this scheme because: (1) it is simple and needs less overhead of implementation and computation, (2) the spatial correlation between partitions can be incorporated to enrich the semantic description of the visual information, and (3) different fixed partitioning schemes such as 4×4, 8×8, etc. can be combined together to capture the different spatial correspondences in an image. Therefore, the spatial information from different schemes can be extracted for assessing the most informative description of the visual information.

After partitioning, low-level visual features are computed for each region. These features are quantized and clustered by k-means clustering [17]. Each region is represented by a cluster index, and a data structure similar to the color correlogram is used to capture the spatial relation between regions.

Let $\mathcal{B}$ be a $M \times M$ partitioning of an $N \times N$ image, where the feature vectors extracted from each region are quantized into m clusters $k_1, \ldots, k_m$. Let b be a partition $b = (x, y) \in \mathcal{B}$. Let $b_1 \in \mathcal{B}_{k_i}$ mean that block b_1 is of cluster k_i. The cluster correlogram matrix $C_f^{\delta\varphi}$ of $\mathcal{B}$ is defined by the joint empirical probability on the image that a cluster k_i co-occurs with a cluster k_j at given distance δ and angle φ as

$$C_f^{\delta\varphi}(k_i, k_j) = \mathbf{P}(b_1 \in \mathcal{B}_{k_i} \wedge b_2 \in \mathcal{B}_{k_j} \wedge D(b_1, b_2) = (\delta, \varphi)). \quad (2)$$

Similar to the color correlogram, the computation cost of the fixed partitioning cluster correlogram increases with increasing the number of clusters. However we found in our experiments that: (1) the number of quantized levels or clusters is generally small, i.e., less than the number of image colors. Therefore, it gives less

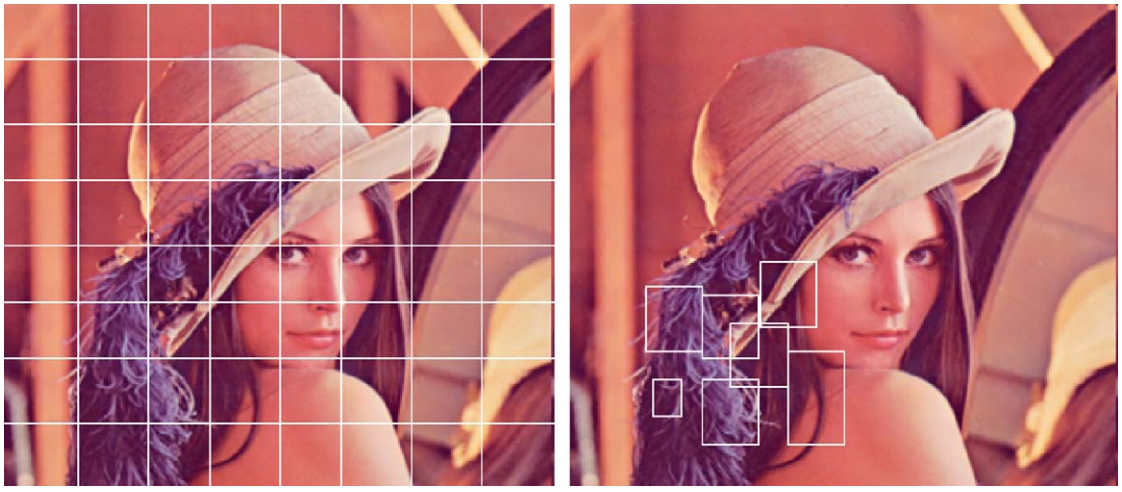

Fig. 1. Left: partitioned image and right: detected interest points.

computational time and it is efficient in storage and (2) usually, the size of the fixed partitioning scheme is much smaller than the size of the actual image. Thus, it gives less computational time to visit all partitions for calculating spatial information in the image. However, problems might arise if the fixed partitioning divides an important region in two or more parts. Therefore, a more recent technique named the saliency-based approach is proposed.

2.4. Saliency-based cluster correlogram

Images taken from scenes and objects usually have many variabilities such as viewpoint, clutter and occlusion. Most of these problems are quite difficult to handle with a global based approach like segmentation or fixed partitioning. There exists a technique that can cope with these problems named the saliency-based approach. The approach is claimed to be local and so it is robust to occlusion and clutter. Besides that, it is robust to photometric disturbances and therefore provides more distinctive and well localizable features, and it is also invariant to image transformations and illumination changes. Furthermore, the algorithm does not need prior segmentation of the images, but is based on the repeatable computation of local extrema points between the scale spaces of an image. The main idea of this approach is to find the most informative locations or salient points in an image. There are several algorithms to achieve this goal such as scale invariant feature transform (SIFT) [20] and speeded up robust features (SURF) [5].

In our experiments, the SURF algorithm is used to describe the salient points by dividing images into various informative rectangular regions or patches. The patches which are processed recursively, are composed of different size and location as shown in Fig. 1 (right). These patches are computed at some scale without orientation alignment to ease the MPEG-7 feature extractions. See [1] for detailed information of the implementation we used. Once the cluster index for each patch is calculated, the next step is to construct the cluster correlogram between patches. We found in our experiments, that it is difficult to find a pair of clusters that co-occurs at given distance δ and angle φ. The main reason why is that the location of salient points may vary according to image primitive types. For instance, the location of salient points in the edge primitive of an image object contains larger points than other primitive types. Therefore, the spatial relation is constructed by considering the nearest patches from a current patch point.

Let $\mathcal{S}$ be a set of n nearest patches from a current patch point of an $N \times N$ image. Each patch is quantized into m clusters $k_1, \ldots, k_m$. Let s be a patch and $s \in \mathcal{S}$. Let $s_1 \in \mathcal{S}_{k_i}$ mean that patch s_1 is of cluster k_i. The salient points cluster correlogram matrix C_s of $\mathcal{S}$ is defined by the joint empirical probability on the image that a cluster k_i co-occurs with a cluster k_j in set $\mathcal{S}$ as

$$C_s^{\delta_a \varphi_a}(k_i, k_j) = \mathbf{P}(s_1 \in \mathcal{S}_{k_i} \wedge s_2 \in \mathcal{S}_{k_j}). \tag{3}$$

Therefore, the salient points cluster correlogram measures the joint probability of all clusters of the image having the particular set of n-nearest patches. We used the number of nearest patches to describe the degree of globalness or localness of the micro-textures.

3. MPEG-7 correlogram indexing and categorization

It is often difficult to determine which image features are most useful to describe the information in an image. Good image features are crucial, because they can give a compact representation and help to discover meaningful patterns in the image. Recently, most studies are focusing on multiple image features for satisfactory recognition results. Using multiple image features may help to recognize different structures of images efficiently and enrich the semantic description of the visual information. Following this, there is a standard called MPEG-7, which provides a platform for indexing for multimedia content [21]. We will use this standard for computing different clusters in our system.

3.1. MPEG-7 image descriptors

The MPEG-7 standard defines a comprehensive, standardized set for effective searching, identifying, filtering, and browsing in multimedia contents such as images, videos, audios, and other digital or even analog materials [21]. To support various types of descriptors, MPEG-7 is organized into several groups. In our implementation, we have chosen four primitive MPEG-7 visual descriptors. MPEG-7 contains different primitive descriptors that enable to describe characteristics of real-world images. Instead of using them separately, it might be a good idea to combine the descriptors together, since this increases the amount of information about an image. We want to test the effectiveness of using MPEG-7 features in the cluster correlogram, because they are easy and fast to compute and have been shown to work well in practice

[2]. Finally, it gives an easy way to compare our algorithm with other systems that are based on the same standard. In our implementation, the color and texture descriptors are used for indexing images.

3.1.1. Color descriptors

Color is a very useful component in visual perception. It is the most instantaneous method of conveying message and meanings in an image. The following color descriptors are used to index images:

Scalable color—the descriptor contains information about color coefficients information in the HSV color space. First, the color histogram is quantized into a 256-bin histogram in one block—16 levels in H, four levels in S and four levels in V. After that, a Haar transform-based encoding scheme is applied to the color histogram. Our system uses 64 Haar coefficients to represent the block which are believed to provide a reasonably good performance.

Color layout—the main purpose of the color layout feature is to represent the spatial distribution of colors in an image. It is formed by dividing an image into 8×8 non-overlapping blocks and then the representative of the YCbCr color system for each block is obtained. A discrete cosine transform (DCT) is applied to each block and its coefficients are used as a descriptor. It should be noted that the representation of this descriptor is in the frequency domain. Thus, we have used 6, 3, 3 for the Y, Cb, Cr coefficients respectively. The descriptor with 12 coefficients was found to be the best value for retrieval performance.

Color structure—the main purpose is to represent local color features in an image. The image is quantized using the HMMD (hue, max, min, diff) color space. Next, a window is slided across the image and at each location the number of times a particular quantized color is contained in the window is counted and stored in a histogram called a color structuring element histogram. Then, the color structure histogram is constructed by incrementing the color present in the structuring element for each window. The color structure histogram is then re-quantized and normalized to construct a descriptor. The descriptor with 64 bins seems to work well to capture overall information about a region.

3.1.2. Texture descriptors

Texture is quite important to check homogeneity and non-homogeneity between images. Our system uses the following texture descriptor:

Edge histogram—instead of color information, the human is known to be sensitive to edge features. The edge histogram describes a non-homogeneous texture and captures a local spatial distribution of edges. First, an image is divided into 4×4 non-overlapping blocks. Then, using an edge detection algorithm, six different edge types (horizontal, vertical, 45°, 135°, non-directional, no-edge) are extracted. Finally, the descriptor with a 80-bin histogram for each image is constructed by excluding the no-edge information.

3.2. MPEG-7 cluster correlogram indexing

The bag of words model has been widely used and demonstrated impressive levels of performance in image classification and categorization applications [18,24,9]. However, because these methods disregard information about the spatial relation between local features, existing results still leave room for improvements. Here, we propose cluster correlograms using MPEG-7 primitive features to improve the indexing performance. Note that the proposed correlograms can work with any type of low-level visual descriptors.

Fig. 2 shows the overall process of the proposed cluster correlograms. The figure contains two main parts namely the fixed partitioning cluster correlogram (top-half) and saliency-based cluster correlogram (bottom-half). Each cluster correlogram algorithm consists of three main steps. The first step is building the visual features dataset that the k-means algorithm will work on. This is done by extracting the MPEG-7 low-level features from regions of images. For example, if the edge histogram descriptor is used, the resulting dataset is an array of 80-vectors for each region. In our implementation, we constructed four different feature datasets using four different MPEG-7 descriptors. After that the k-means algorithm is applied to each dataset, resulting in four different k cluster centers or cluster codebooks. Each codebook then is used to represent the k clusters for regions in the image. The second step is encoding, where each region in the image is represented using low-level visual features computed by an MPEG-7 descriptor. After that, the cluster codebook that belongs to this visual feature is used to retrieve the nearest cluster center for the region. The last step is the cluster correlogram construction. The two correlograms are constructed using Eqs. (3) and (4) for the fixed partitioning cluster correlogram and salient points cluster correlogram. Finally, these correlograms are used to index images.

3.3. Categorization

Once the feature vectors of all images are obtained, they can be used for machine learning algorithms to train classifiers for classifying test images. The feature vector of each signature is represented by a 2D $m \times m$ matrix where m is the number of clusters. Note that since we use four feature descriptors, we have four different signatures. The m value is varied and it depends on the number of clusters used in the clustering algorithm. If m clusters are used for all descriptors then the feature dimension size for each image is $4m^2$.

3.3.1. k-nearest neighbors classifier

First, the k-nearest neighbors (k-NN) algorithm is used to classify a given test image. The k-NN is a simple classifier based on the idea that similar observations belong to similar classes. This learning algorithm consists of a training phase and testing phase. In the training phase, a training dataset is constructed that is described by the set of examples $P = \{(a_1, c_1), (a_2, c_2), \ldots, (a_z, c_z)\}$ where a_i is a training pattern in the training data set, c_i is its corresponding class and $i = 1, \ldots, z$ is the number of training patterns. In the testing phase, the query starts at a given unlabeled point and the algorithm generates a list of the k nearest records from the entire set of training patterns. Then, the classification is done by a majority voting scheme to label the class of a test image. The similarity between two feature vectors is measured by using the Manhattan distance between two images described by four cluster correlograms. We chose the Manhattan distance because it gives the best performance in our experiments.

3.3.2. SVM classifier

Besides k-NN classification, we employ an SVM [28] to learn and classify images. Implementations of SVM to multiple class problems usually use the one vs. all or the one vs. one strategy. We apply the one vs. one strategy with a RBF kernel on Corel and with a linear kernel on PASCAL 2006. In the one vs. one approach each class is trained against each of the others resulting in $(K(K-1))/2$ models, where K is the number of classes. Each model is trained with $+1$ for images belonging to the right class and -1 for the imageas from a single other class. When testing, an image is given to all $(K(K-1))/2$ models and the class which most often wins

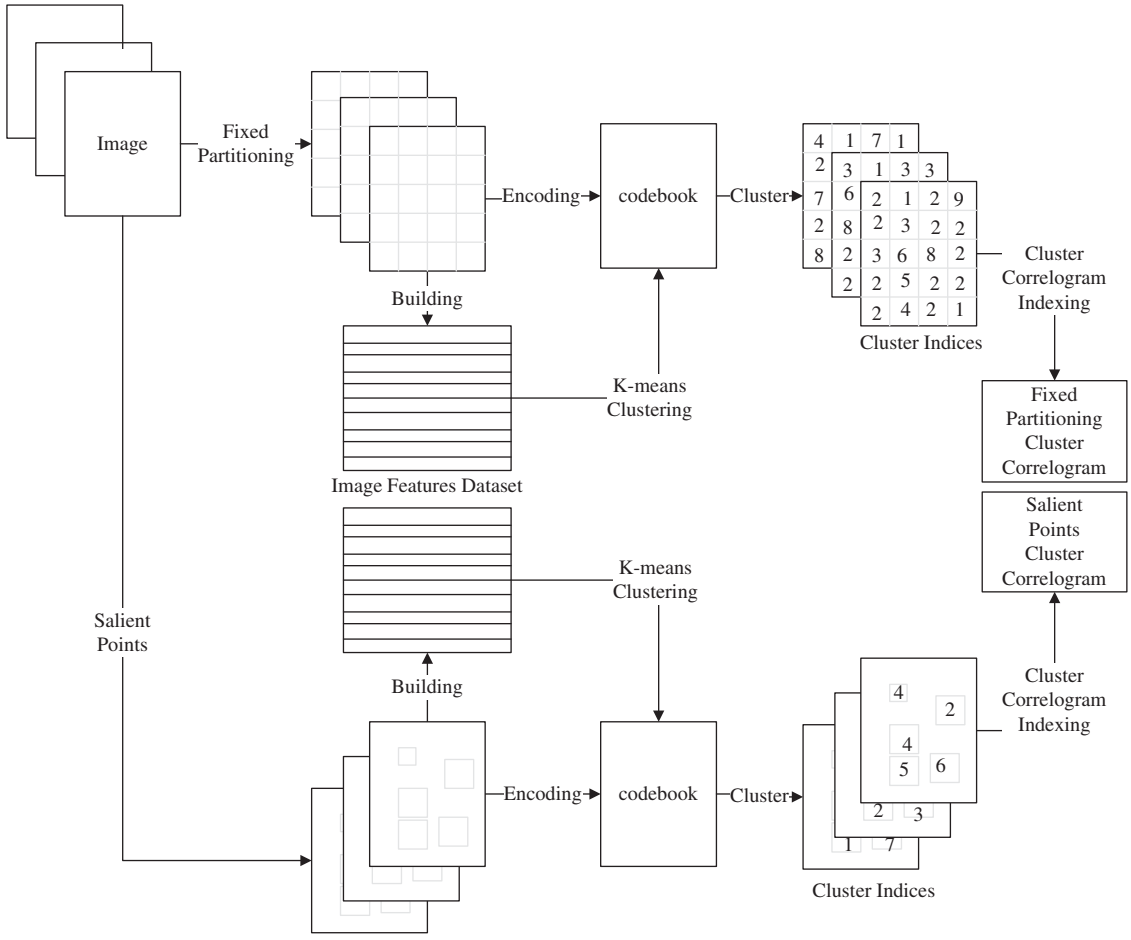

Fig. 2. Top-half: the fixed partitioning correlogram indexing and bottom-half: the saliency-based correlogram indexing.

against the other classes (it can win at most $(K-1)$ times), is the winner. For Corel we used the single class that won this competition, but for PASCAL 2006 we used the probability estimates from the SVM to compute the ROC curve. Finally, in PASCAL 2007, the number of classes in this dataset is large, i.e., 20 classes and there can also be multiple objects in an image. Therefore, a binary classification SVM with a linear kernel is employed. In this strategy there are K SVM models, one model for each class. Each model receives as training data $+1$ for images having the object inside and -1 for all images that do not have the object in it. Then all K models are trained. For testing, the test images are given to all class models, and models that have an output larger than 0 tell that their corresponding object is in the image. This is used to compute the average precision.

Initially, all attributes x_i in the training and testing were normalized to the interval $[-1,+1]$ by using this equation:

$$x_i' = \frac{2(x_i - min_i)}{(max_i - min_i)} - 1. \tag{4}$$

The normalization is used to avoid numerical difficulties during the calculation and to make sure the largest values do not dominate the smaller ones.

We also need to find the SVM parameters C and γ that perform best for the descriptors. To optimize the classification performance, the SVM parameters were determined by using the simple libsvm grid-search algorithm [7]. The C and γ values can be tried out exponentially to get the best accuracy performance. Therefore, we tried the following values $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$ for C and γ respectively. The values that gave the best accuracy performance are picked and used to train on the training set.

4. Experiments

For a more robust comparison between the proposed algorithms, some established datasets are needed. Therefore, to demonstrate the performance of our proposed algorithms, we have first used two well known datasets namely Corel[1] and PASCAL 2006.[2] These datasets contain various image sizes and were categorized into 10 different classes. We also did a small experiment with PASCAL 2007. We will explain the evaluation measures, the datasets, and the performance results of our proposed algorithms compared to other systems in the following subsections.

4.1. Experimental setup

We have implemented the cluster correlograms on three different datasets i.e., Corel, PASCAL 2006 and PASCAL 2007. The cluster correlograms are used to index all images. In the color

[1] The dataset is available from http://www.corel.com

[2] PASCAL stands for pattern analysis, statistical modeling and computational learning. The dataset is available from http://www.pascal-network.org/challanges/VOC/

Table 1
Parameters used in Corel dataset.

Visual descriptors	Fixed partitioning		Salient points	
	Size	Cluster	Neighbor	Cluster
Scalable color	16×16	24	8	24
Color layout	16×16	24	8	24
Color structure	24×24	32	8	24
Edge histogram	16×16	24	32	24

Table 2
Parameters used in PASCAL dataset.

Visual descriptors	Fixed partitioning		Salient points	
	Size	Cluster	Neighbor	Cluster
Scalable color	28×28	32	30	32
Color layout	28×28	32	30	32
Color structure	28×28	32	30	32
Edge histogram	28×28	32	30	32

correlogram approach we tried out several number of colors i.e. 8, 16, 24, 32, 64 and 128 and we found that $m=64$ gives the best performance. In the fixed partitioning cluster correlogram approach, we tried out several numbers of partition schemes to get the best accuracy performance. Therefore, we tried out the following schemes: 8×8, 16×16, 24×24 and 32×32. The scheme that gave the best accuracy performance is used to train on training set. The size of the fixed partitioning scheme is different for each dataset as mentioned in Tables 1 and 2. Besides that, the orientation (φ) and distance (δ) attributes are also optimized. In our experiment we combined four different φ, i.e., 0, 45, 90, and 135 in one correlogram to enrich the spatial information between clusters. After that, this combination is tested on several δ namely 1, 2, 3, and 4. In our experiment $\delta=1$ gave the best performance.

In the salient points cluster correlogram, we tried out several numbers of nearest patches: 4, 8, 16, 24, 30 and 32. Similar to fixed partitioning, the number of n that gave the best accuracy performance is used to train on the training set. Besides this parameter, other important parameters in SURF are the σ and r values. These parameters influence the number of salient points in images. In our experiments, the default values of σ and r are used for all datasets.

Finally, the number of clusters is computed manually and we start the experiment with $m=4$, 8, 16, 24 and 32. The number of m which gave the best accuracy performance is used to construct the correlograms for each dataset. For the fixed partitioning and saliency-based methods the length of feature vectors is the sum of the size of all cluster correlograms, i.e., $m^2(ScalableColor)+m^2(ColorLayout)+m^2(ColorStructure)+m^2(EdgeHistogram)$. For the color correlogram, the length of feature vectors is $m^2(Color)$ and for the MPEG-7 approach, the length is determined by concatenating the size of feature vectors of all MPEG-7 primitive descriptors. In the Corel experiment, the length of feature vectors that we used for fixed partitioning is $24^2+24^2+32^2+24^2=2752$ feature values. The saliency-based scheme uses $24^2+24^2+24^2+24^2=2304$ feature values. In the PASCAL experiment, the length of feature vectors for the fixed partitioning and saliency-based cluster correlogram are both $32^2+32^2+32^2+32^2=4096$ feature values. The length of the feature vectors for the color correlogram and MPEG-7 are the same in both experiments, i.e., $64^2=4096$ and $64+12+64+80=220$ feature values, respectively.

4.2. Evaluation methods

In the experiments we have used three evaluation measures, namely the average precision, a confusion matrix and receiver operating characteristics curve (ROC-curve). The reason why we have chosen to use these measures is that they are standardized, and they will enable us to compare our proposed algorithms with other systems.

4.2.1. Average precision

For evaluating CIREC's retrieval performance, we compute the precision on the queries and average it over all queries to compute the average precision. In general, we want to have N images returned having the same category as the query image. In our comparison all images will be used one time as a query image. The precision is then computed as follows. Let $\mathbf{Rank}(Q,\mathcal{I}_i)\,\varepsilon\,[1,n]$ be the rank of retrieved image $\mathcal{I}_i$ from the database, where n is the number of images in a dataset and Q is a query image. The images having a rank below some number N may contain relevant and irrelevant images. Next, let $C(Q,\mathcal{I}_i)$ denote that the retrieved image $\mathcal{I}_i$ has the same category as the query image Q. The precision (P) of the first N retrieved images for a query Q is defined as

$$P(Q,N)=\frac{|\{\mathcal{I}_i|\mathbf{Rank}(Q,\mathcal{I}_i)\le N\ \wedge C(Q,\mathcal{I}_i)\}|}{N}. \quad (5)$$

We used it to compare our algorithms with other systems for the Corel dataset.

4.2.2. Confusion matrix

The confusion matrix is used to compute the accuracy of the classification models and it can also be used to visualize the errors on a given image category. A k-nearest neighbor classifier using majority voting of the retrieved images and an SVM are used to categorize a test image. For the k-NN classifier various values of k are tested. We have introduced a rule that says that when multiple categories have the same number of votes with a particular $k>1$, the query image is assigned to the category with the lowest index.

4.2.3. Receiver operating characteristic curve (ROC)

The ROC curve is measured by calculating the relationship between the sensitivity (true positive rate (TPR)) and the specificity (false positive rate (FPR)). To have a single performance measure, we compute the area under the ROC curve (AUC). This measure is used to compare our algorithms with other systems on the PASCAL dataset. We do not use accuracy here, since an image in the PASCAL dataset can consist of multiple objects that need to be recognized.

4.3. Evaluation on datasets

The Corel and PASCAL datasets are used to compare the two schemes to each other and some other algorithms like the MPEG-7 features and the color correlogram. We test the algorithms with different numbers of blocks, clusters, and neighbors using the k-NN classifier as mentioned in Tables 1 and 2. The values of the parameters were determined in the training session by a trial and error approach. Finally, the best parameters are used for the k-NN and the SVM classifiers. Fig. 3 shows some cluster correlogram patterns from the fixed partitioning and salient points schemes.

4.3.1. Corel dataset

The Corel dataset has become a de-facto standard in demonstrating the performance of CBIR systems [29,6]. In general, Corel contains a collection of more than 800 photo CDs and about 100 images for each theme. We used the first 10 categories and a total

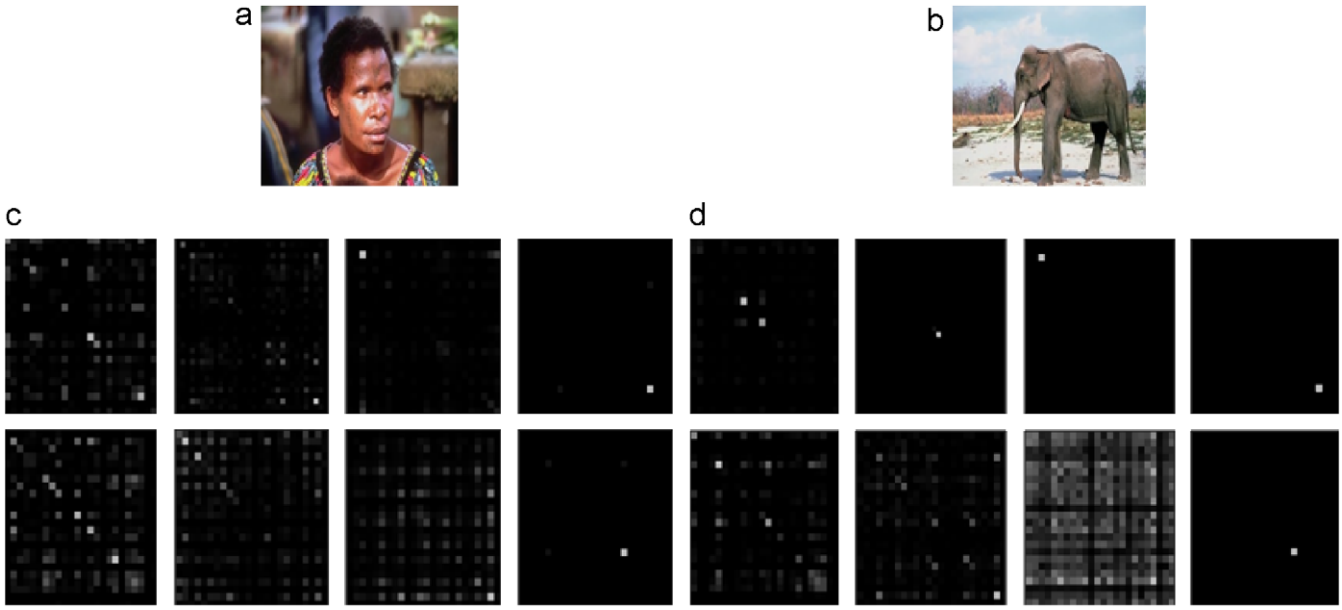

Fig. 3. (a, b) Two sample real world images, (c, d) the cluster correlogram patterns for these two images. The first row shows the cluster correlogram patterns of the fixed partitioning. The second rows shows the correlogram patterns of the salient points. Each cluster correlogram contains four different cluster primitives, i.e., color layout, color structure, edge histogram and scalable color. Each feature vector in the matrix is mapped into range [0,255] gray-level.

of $10 \times 100 = 1000$ images for evaluation, also known as the Wang dataset [29]. These images are all in JPEG format with size 384×256 or 256×384 and were categorized into 10 different groups, namely Africans, beaches, buildings, buses, dinosaurs, elephants, flowers, horses, mountains and foods. In this dataset, there is only one target object category for each image and its appearance looks consistently good. The position of the interest object is approximately centered or takes up most of the whole image size. Besides that, the pictures taken in each group tend to be similar in viewpoints and orientations. The images seem to be simple with little or no occlusion and clutter. Fig. 4 shows the ground truth for different groups in the Corel dataset.

For evaluating the fixed partitioning and saliency-based approach and further comparing it to using the color correlogram and global MPEG-7 features, we first compute the precision of the retrieved images on the queries. In our comparison all images are used one time as a query image. The precision is then computed using Eq. (4) and by averaging it over all query examples. Table 3 displays the average precision of the fixed partitioning, salient points, color correlogram and MPEG-7 visual descriptors over 10, 20, 30, 40 and 50 retrieved images for each group using a ranking scheme employing the Manhattan distance. The results clearly show that the cluster correlogram with the fixed partitioning and salient points schemes outperforms the other methods and that fixed partitioning performs slightly better than the salient points scheme.

We have also compared our proposed algorithms with another CBIR system based on the wavelet correlogram [3]. In this comparison, the same methodology of evaluation is used to compute the average precision for every query image. When retrieving 10 images, the precision of the wavelet correlogram is 0.73, which is much lower than the performance of our proposed systems.

To measure the fixed partitioning and saliency-based performances for image categorization, we have first tested these schemes in combination with the k-nearest neighbor method (k-NN). Table 4 displays the overall image categorization performance of the fixed partitioning and saliency-based schemes using the k-nearest neighbors classifier. We have experimented with various values of k, namely $k = 1$, 3, 5, 7, 9, and 19. In this experiment, fixed partitioning gives the best performance with $k = 7$ and yields 89.7% correctly classified images. Other state-of-the-art categorization systems that have been applied to categorize images of the same Corel dataset are: (1) the use of a set of features and support vector machines (SVMs) [8], (2) invariant feature histogram [10], and (3) a system that combined five different features [11]. These systems scored 81.5%, 84.5%, and 87.3%, respectively on the same dataset. This indicates that the fixed partitioning with MPEG-7 correlograms performs very well and works well in combination with a simple k-NN classifier. One main problem with the comparison is that it is quite difficult to get the actual images for testing and training classifiers. Therefore, the popular data mining technique n-fold cross validation is employed to attain high confidence in the performance of the classifiers. In this case, 5-fold cross validation is used to measure the performance of the k-NN classifier. Thus, we obtained five subsets of equal size of training sets. After that, each of the five subsets is tested using the classifier trained on the remaining four subsets.

We have also done experiments with a support vector machine using the cluster correlogram and a bag of keywords (or cluster histogram). Table 5 displays the results for the experiment with the SVM. It is shown that the SVM significantly outperforms the k-NN. Furthermore, the cluster correlogram outperforms the cluster histogram even though we experimentally optimized the number of clusters for the cluster histogram. We have used 320 clusters when fixed partitioning was used and 256 clusters using salient points. The clustering using k-means clustering took much more computational time than the use of the small number of clusters that were used in the cluster correlogram. Therefore the results show that the cluster correlogram clearly has advantages for the Corel dataset compared to a cluster histogram.

We also show the results of using fixed partitioning for image categorization with an SVM in a confusion matrix in Table 6. The confusion matrix is a square matrix that shows the various classifications and misclassifications of the classifier. In the confusion matrix, numbers on the diagonal are correct classifications and off-diagonal numbers correspond to misclassifications. A detailed examination of the confusion

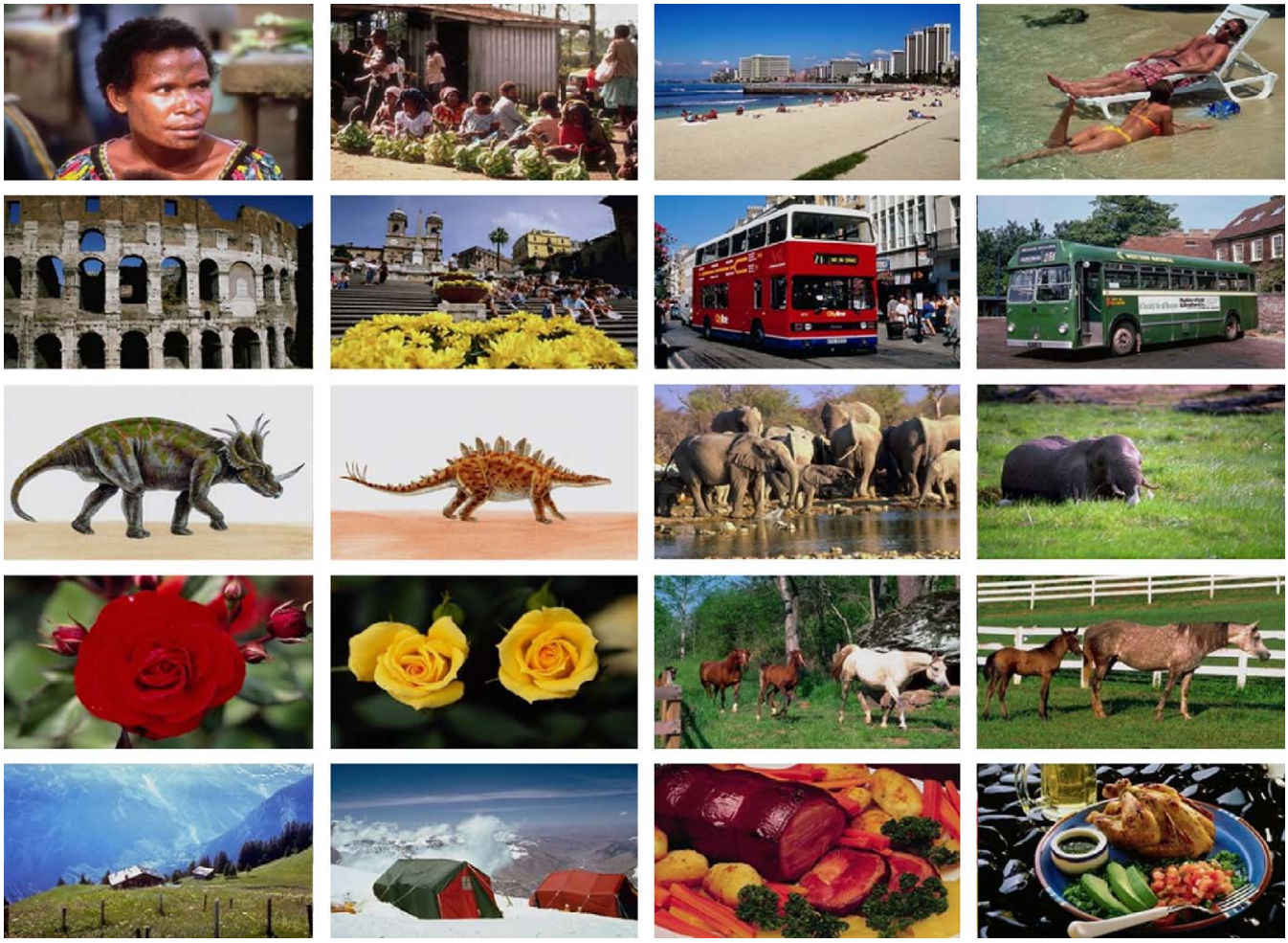

Fig. 4. Image examples for Corel with ground truth for different groups namely Africans, beaches, buildings, buses, dinosaurs, elephants, flowers, horses, mountains and foods, respectively.

Table 3
The average precision for the different methods on the Corel set.

Methods	Number of retrieved images				
	10	20	30	40	50
Fixed partitioning	0.80	0.76	0.73	0.70	0.67
Salient points	0.78	0.74	0.71	0.68	0.65
Color correlogram	0.71	0.65	0.61	0.58	0.56
MPEG-7	0.62	0.56	0.52	0.50	0.47

Table 4
The average categorization precision results using a *k*-nearest neighbors classifier on the Corel set.

Methods	*k*-nearest neighbors					
	1	3	5	7	9	19
Fixed partitioning	87.9	89.4	89.2	**89.7**	88.2	88.0
Salient points	85.4	86.5	87.4	**88.6**	87.5	87.1
Color correlogram	80.7	81.2	80.4	80.7	**81.5**	80.0
MPEG-7	71.4	74.8	**74.8**	74.5	73.7	72.8

The best result is marked in boldface.

Table 5
The average categorization precision results using an SVM on the Corel set.

Methods	Cluster correlogram	Cluster histogram
Fixed partitioning	93.4	92.9
Salient points	91.8	90.8

Table 6
The confusion matrix of image categorization using the fixed partitioning with SVM on the Corel set.

Categories	A	B	C	D	E	F	G	H	I	J
A	89	1	1	0	1	7	0	0	1	0
B	2	85	2	1	0	0	1	1	8	0
C	1	4	86	2	0	4	1	0	0	2
D	0	1	0	98	0	0	0	0	0	1
E	0	0	0	0	100	0	0	0	0	0
F	1	0	1	0	0	94	0	3	1	0
G	0	0	0	0	0	0	99	0	0	1
H	0	1	0	0	0	0	0	99	0	0
I	0	7	0	0	0	3	0	0	90	0
J	2	1	0	1	0	0	0	0	2	94

A = Africans, B = beaches, C = buildings, D = buses, E = dinosaurs, F = elephants, G = flowers, H = horses, I = mountains, and J = foods.

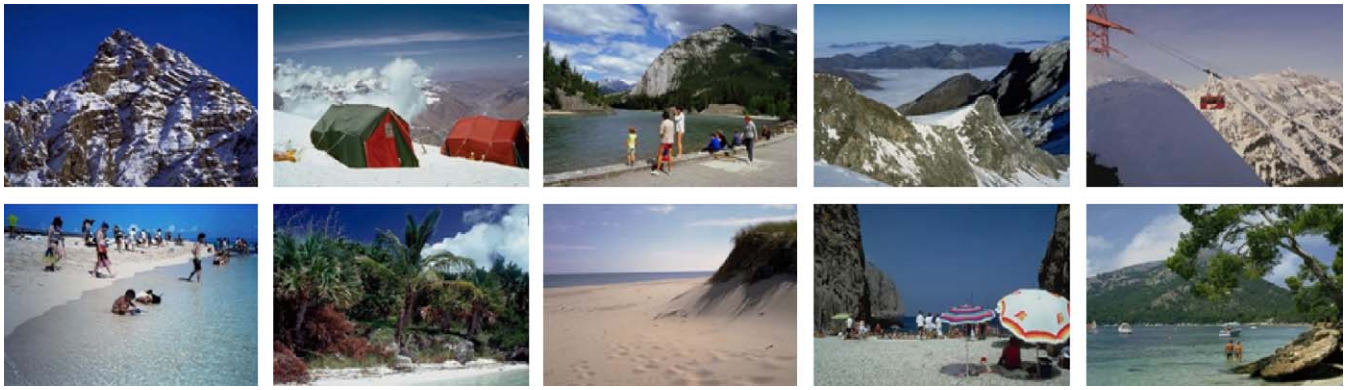

Fig. 5. Some sample images are misclassified. The first row is misclassified as "Beaches" and the second row as "Mountains (with glaciers)". The first and second rows should be classified as "Mountains (with glaciers)" and "Beaches".

matrix shows that there are two distinct misclassifications (the underlined numbers in Table 6). The model is slightly confused to make distinctions between "Beaches" and "Mountains (with glaciers)". The difficulty of distinguishing between these two categories has also been noted in other studies. Fig. 5 shows misclassified images from both categories.

4.3.2. PASCAL dataset

This dataset is used to compare our algorithms with other systems as reported in the 2006 PASCAL challenge. The dataset is designed to recognize objects from a number of visual object classes in realistic scenes. Ten object classes are provided in the dataset namely bicycle, bus, car, motorbike, cat, cow, dog, horse, sheep and person. Each category has a different number of photos and these have various image sizes. The images are collected from the photo-sharing web-site "flickr"[3] and some are provided by Microsoft Research Cambridge.[4] In total there are 5304 images that contain 9507 annotated objects in the dataset. The dataset is quite complicated and sometimes quite difficult for recognition purposes. The images are taken from different points of view and orientations and objects do not take up most of the image. Many objects are occluded and there is background clutter with unwanted objects. Besides that, the quality of the images is not as good as in the Corel dataset. Fig. 6 shows the ground truth for different groups in the PASCAL dataset.

In the PASCAL challenge [12], there were three types of image sets provided to be used in the classification task, namely training data, validation data and test data. The dataset is split into 2618 images for training or validation and 2686 images for testing. Here the k-NN and the SVM algorithms are used on this dataset to measure the performance of the cluster correlogram and cluster histogram with the fixed partitioning and saliency-based approaches. In the cluster histogram, we used 200 visual keywords by clustering MPEG-7 features with k-means. Note that, the clustering algorithm takes a long time to obtain 200 visual codewords from the training or validation images. Therefore, we saved time by choosing only 50 images for clustering from each group. In total, we used 500 images to construct the visual codewords. After that, we represent each image as the histogram of visual keywords. For the cluster correlogram we used the parameters of Table 2 (so we only used 32 clusters for each MPEG-7 descriptor). Table 7 displays the overall image categorization performance of the k-NN and SVM classifiers and different approaches. For k-NN, we have tested the classifier with various values of k. We found that $k=21$, 35, 41, 45, 49 and 35 gave the best performance for M1, M2, M3, M4, M5 and M6, respectively.

The best result as measured by the ROC curve is underlined. In contrast to the previous experiment, this time the saliency based approach outperforms the fixed partitioning scheme in many categories. The cluster correlogram clearly outperforms the cluster histogram, color correlogram, and the use of MPEG-7 features alone. The SVM outperforms the k-NN. The system clearly has most difficulties with recognizing persons. Finally, we have compared our approaches with other experimental results using the average ROC curve values on 10 categories. In the first round of the PASCAL 2006 challenge, the best team QMULLSPCH achieved an average AUC of 0.936, whereas the lowest ranked team (at place 18), AP06Batra, achieved an AUC of 0.702. The fixed partitioning and salient points approaches would be ranked top ten (at places 6 and 7) in the competition and therefore seem to perform reasonably well on this dataset. In contrast with the best result in this challenge, our methods are based on indexing on whole images. This indicates that the cluster correlogram is quite well without using a time-consuming detector to search for objects in an image.

Finally, we have tested one of the cluster correlograms using the PASCAL 2007 dataset. In PASCAL 2007, there are 9,963 images from 20 different image classes. In this experiment, the salient point cluster correlogram with the same configuration settings as in PASCAL 2006 were used to train the binary SVM classifiers. The dataset is far more challenging than PASCAL 2006 because it contains: (1) more image classes, then the probability to get the correct class is lower, (2) the pictures taken in each group tend to be more diverse in viewpoints, orientations, occlusion and clutter. (3) a last difficulty of this dataset is its large inter-class variability and background information seems to be less informative to describe object categories. Therefore, the dataset will place a challenging task for object recognition systems and require algorithms to detect the most informative parts of images. In this dataset, a different average precision measure is used as the performance metric for determining the accuracy for each category. This average precision averages precision over the entire range of recall. Thus, a good score requires both high recall and high precision. However, the salient points cluster correlogram with MPEG-7 descriptors did not perform well in this dataset. It gives 52% average precision for the aeroplane image class, while other techniques have reported a performance between 49% and 77%. This confirms our insight that the color and texture primitives of the MPEG-7 standard perform better in scene classification than in object classification.

[3] The photos can be accessed at http://www.flickr.com/.
[4] http://research.microsoft.com/cambridge/

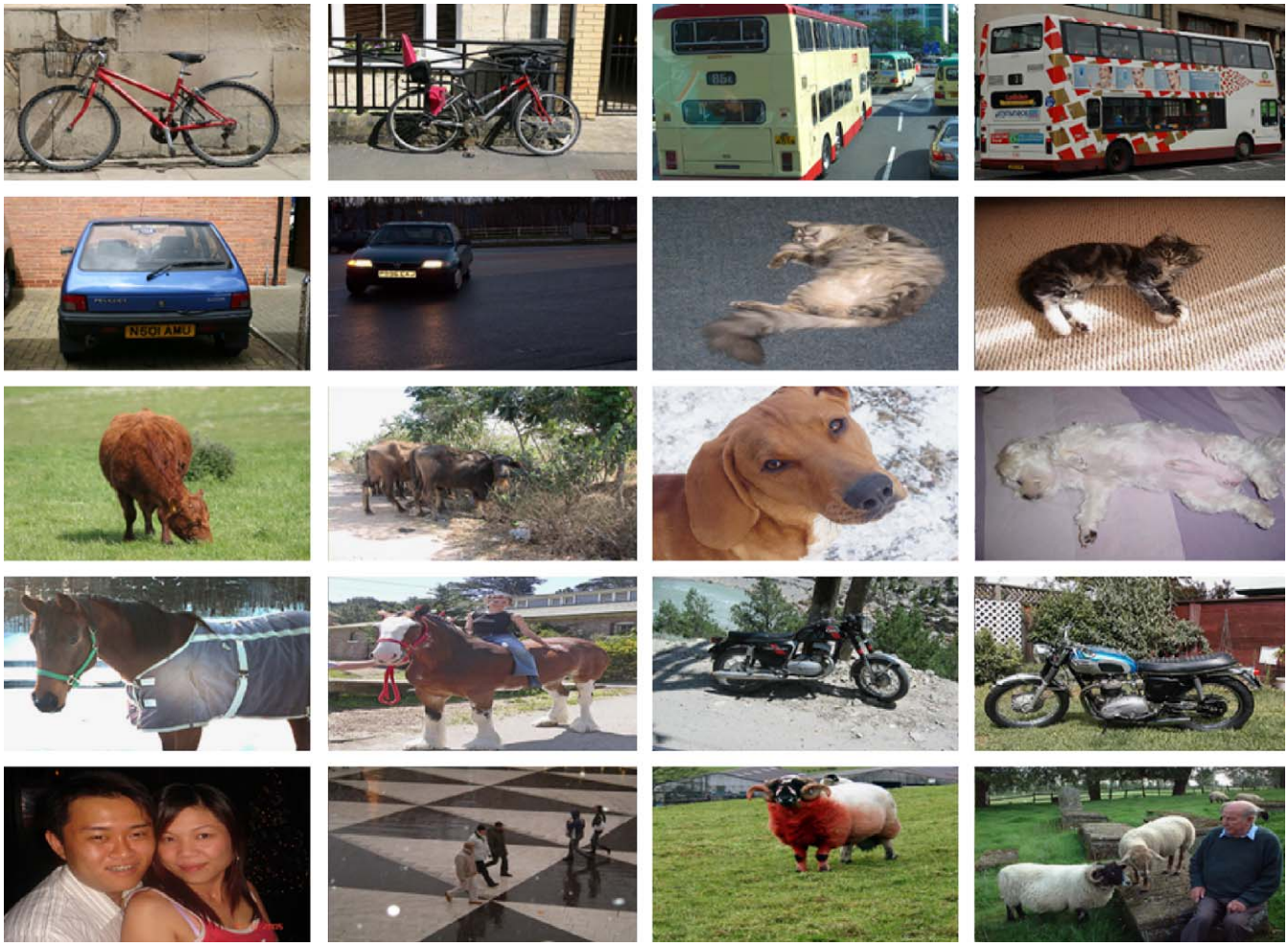

Fig. 6. Image examples for PASCAL 2006 with ground truth for different groups, namely bicycles, buses, cars, cats, cows, dogs, horses, motorbikes, persons, and sheep.

4.4. Robustness evaluation

We have performed extensive experiments to test the robustness of the proposed method against several transformations. The popular transformation algorithms namely Gaussian noise, photometric and geometric disturbances were used on the fixed partitioning cluster correlogram and color correlogram methods with an SVM classifier.

In all experiments, we used 10 testing images from Corel for each image class. Thus, in total, $10 \times 10 = 100$ images were used to test the robustness of the system. To compute the performances of the different methods, we choose five times different training and test images randomly from a set of candidate images in the 10 classes of the Corel dataset. We report the fixed partitioning performance in terms of the mean and standard deviation of the classification accuracy.

Fig. 7 shows one type of transformation algorithm. This type uses Gaussian noise disturbance to test the methods. Table 8 displays results on the Gaussian test for the fixed partitioning cluster correlogram and color correlogram. In fixed partitioning we used the scaling vectors $[-1, +1]$ for all noise types and for the color correlogram we used the scaling vectors $[-1, +1]$ for STD $= 0$ and $[0, +1]$ for others. The results show that the cluster correlogram can tolerate well small amount of noise. However, it becomes worse when the STD of the Gaussian noise is increased.

Table 9 displays results with the photometric transformations for the fixed partitioning cluster correlogram and color correlogram. We test the algorithms with several disturbance

Table 7
Results of different classifiers and approaches on the PASCAL set, as measured by the area under the ROC curve (AUC).

	Categories	M1	M2	M3	M4	M5	M6
k-NN	Bicycle	0.860	0.862	0.768	0.764	0.851	0.845
	Bus	0.896	0.919	0.796	0.809	0.870	0.883
	Car	0.933	0.939	0.834	0.855	0.905	0.917
	Cat	0.841	0.837	0.790	0.751	0.851	0.845
	cow	0.878	0.881	0.839	0.787	0.892	0.899
	Dog	0.784	0.798	0.723	0.717	0.803	0.817
	Horse	0.814	0.773	0.717	0.670	0.808	0.773
	Motorbike	0.848	0.898	0.740	0.762	0.838	0.874
	Person	0.742	0.748	0.646	0.639	0.677	0.692
	Sheep	0.881	0.892	0.860	0.810	0.896	0.903
	Average	0.848	0.855	0.771	0.756	0.839	0.845
SVM	Bicycle	0.886	<u>0.909</u>	0.825	0.876	0.845	0.847
	Bus	0.950	<u>0.951</u>	0.877	0.913	0.896	0.899
	Car	0.949	<u>0.953</u>	0.846	0.934	0.905	0.918
	Cat	<u>0.876</u>	0.875	0.817	0.861	0.864	0.855
	Cow	0.908	<u>0.911</u>	0.860	0.896	0.881	0.885
	Dog	<u>0.817</u>	0.814	0.752	0.810	0.816	0.807
	Horse	0.845	<u>0.850</u>	0.742	0.837	0.789	0.784
	Motorbike	0.924	<u>0.940</u>	0.839	0.894	0.854	0.868
	Person	0.771	<u>0.778</u>	0.706	0.762	0.660	0.678
	Sheep	0.908	<u>0.913</u>	0.879	0.910	0.876	0.906
	Average	0.883	<u>0.889</u>	0.814	0.869	0.839	0.845

The best result is underlined.
M1 = cluster correlogram with fixed partitioning,
M2 = cluster correlogram with salient points, M3 = color correlogram,
M4 = MPEG−7, M5 = cluster histogram with fixed partitioning, and
M6 = cluster histogram with salient points.

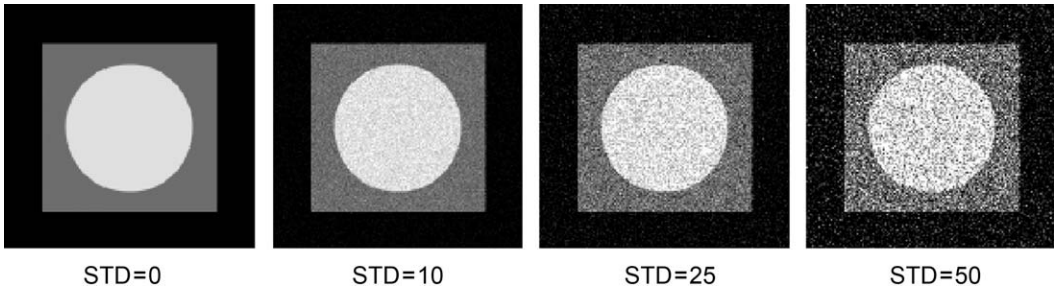

Fig. 7. Four different Gaussian noise disturbances applied on a pattern.

Table 8
Classification accuracy showing robustness to Gaussian noise.

Methods	STD = 0	STD = 10	STD = 25	STD = 50
Fixed partitioning	92.4 ± 1.67	81.8 ± 3.63	50.2 ± 3.63	39.6 ± 3.65
Color correlogram	88.2 ± 3.35	71.0 ± 12.02	64.6 ± 11.33	47.2 ± 10.26

Table 9
Classification accuracy showing robustness to photometric distortion.

Methods	Levels	P1	P2	P3	P4	P5
Fixed partitioning	L1	76.4 ± 5.18	52.4 ± 6.86	86.8 ± 3.11	87.0 ± 4.74	86.6 ± 2.88
	L2	65.6 ± 3.78	26.2 ± 4.32	72.0 ± 3.16	82.6 ± 7.23	82.8 ± 4.60
	L3	59.8 ± 5.63	16.0 ± 1.58	53.4 ± 5.81	80.2 ± 6.22	78.4 ± 5.64
Color correlogram	L1	60.4 ± 9.71	37.0 ± 12.65	58.2 ± 12.28	68.6 ± 11.10	70.0 ± 9.35
	L2	48.0 ± 11.42	11.6 ± 3.58	15.8 ± 4.97	64.2 ± 9.73	55.8 ± 14.31
	L3	38.4 ± 9.89	10.0 ± 0.00	11.8 ± 2.05	58.6 ± 11.72	43.0 ± 16.36

P1 = brighten, P2 = darken, P3 = blur, P4 = sharpen, and P5 = saturation.

Table 10
Classification accuracy showing robustness to geometric distortion.

Methods	Levels	Twirling	Rippling
Fixed partitioning	L1	83.2 ± 3.56	89.8 ± 3.11
	L2	81.6 ± 3.51	86.8 ± 3.56
	L3	81.0 ± 3.32	56.0 ± 5.79
Color correlogram	L1	70.2 ± 16.77	74.6 ± 11.76
	L2	69.6 ± 15.29	71.8 ± 12.79
	L3	69.2 ± 14.91	62.2 ± 11.50

levels. The following settings are used in our experiments: (1) for the brightening test, we increase the brightness component with 1.5, 2.0 and 2.5 for the labels L1, L2 and L3 respectively, (2) for darkening, we decrease the darkeners component with 0.8, 0.6 and 0.4, (3) for saturation, we increase the color information with 1.5, 2.0 and 2.5, (4) for blurring, the Gaussian blur with $r = 2$, 3 and 4 are used to convolve the test images, (5) and finally for the sharpening test, we sharpen the test images using a technique called unsharp masking with $r = 2$, 4 and 6.

Table 10 displays results with the geometric transformations for the fixed partitioning cluster correlogram and color correlogram. For the geometric test, we distort the geometrical structure of test images using two geometrical filters namely twirl filter and ripple filter. In the twirl filter, the angles of 1, 2 and 3 are used for L1, L2 and L3, respectively. For the ripple filter, we used the wavelength of ripple in the *x*- direction of 15, 5 and 1 for L1, L2 and L3, respectively. Both tests show that the proposed method can tolerate quite well with minimum to medium distortions.

4.5. Computing times

The fixed partitioning cluster correlogram is developed using Java (JDK 1.6.0 beta 2) in the Windows 2000 platform. Before the experiment is conducted, the indexing process needs to be performed on each image in the database. The indexing process take some time and it depends on the number of images, number of features used, and system configuration. We have indexed the algorithm on a pentium IV 2.4 GHz CPU with 533 MB memory. The indexing process has two stages: (1) first we cluster the features computed from regions of all images into a set of clusters. For the 1000 images dataset it takes 8 hours to complete and (2) second, we use the cluster and region topology to construct the cluster correlogram for all features we used. This takes 10–15 min for 1000 images in the dataset. Once all images are indexed, it takes about 47 ms to predict a 384×256 pixels test image. However, if the test image is not in the dataset, it takes about one extra second to extract all features from the test image.

5. Discussion

The proposed cluster correlogram with MPEG-7 features can deal very well with large objects or natural scenes where background information is informative. This system clearly outperforms other state of the art systems for the Corel dataset. However, since our system categorizes the whole image, it performs a bit worse for recognizing small objects as needed for the PASCAL dataset. Besides that, the cluster correlogram matrix approach is particularly suitable for describing microtextures. It is not suitable for textures comprising large area primitives since it does not capture shape properties. As a result, fixed partitioning performs better on the Corel dataset and the salient points method performs better on the PASCAL dataset. Especially for recognizing objects, there can be an advantage for the salient points scheme.

Generally, there are four factors that influence the correlogram indexing, namely number of boxes, number of clusters, number of neighbors and size of boxes. This will indirectly affect the retrieval and categorization performance. For instance, in Fig. 8, it is clearly shown why fixed partitioning outperforms the salient points scheme in the beaches category of the Corel dataset. The number

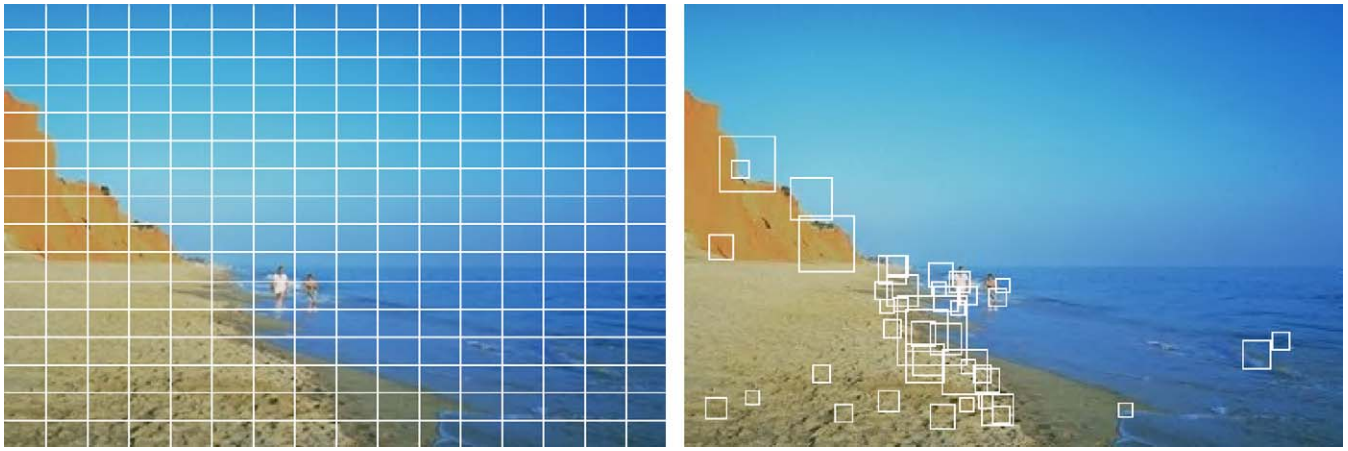

Fig. 8. The effect of the number of patches in correlogram and histogram construction. Salient points (right) would result in coarse indexing when the number of salient points is small. This problem is not happening when using the fixed partitioning scheme (left).

of salient points only covers a very small portion of the beach scene. As a result there is information loss and less distinctive indexing of an image.

6. Conclusions

Two methods of region indexing for image retrieval and categorization based on visual keywords and a correlogram were proposed. The primitives of MPEG-7 visual descriptors are used to extract and group similar patterns into a keyword index. The *k*-NN and SVM algorithms are used to classify the test images. The experiments show that the proposed methods provide useful information to represent images. The results show that the cluster correlogram outperforms the cluster histogram, a color correlogram and MPEG-7 features alone, and the SVM significantly outperforms the *k*-NN classifier. Our experimental results on real world datasets show that our system that uses MPEG-7 visual descriptors in a clustering algorithm achieves very good results on the Corel dataset, but performs a bit worse on the more difficult PASCAL dataset. Therefore, it would be interesting to model visual objects in the PASCAL dataset more explicitly and rely less on background information.

Acknowledgments

The authors want to thank the peer reviewers, the providers of the MPEG-7 feature extractors from http://www.semanticmetadata.net/features/, and Hado van Hasselt for helpful comments. The first author also wants to thank the government of Malaysia for the Ph.D. Grant.

References

[1] A. Abdullah, R.C. Veltkamp, M.A. Wiering, Fixed partitioning and salient points with MPEG-7 cluster correlograms for image categorization, Technical Report UU-CS-2009-008, Department of Information and Computing Sciences, Utrecht University, The Netherlands, 2009.
[2] A. Abdullah, M.A. Wiering, CIREC: cluster correlogram image retrieval and categorization using MPEG-7 descriptors, IEEE Symposium on Computational Intelligence in Image and Signal Processing (2007) 431–437.
[3] H. Abrishami, A.H. Roohi, T. Taghizadeh, Wavelet correlogram: a new approach for image indexing and retrieval, The Journal of the Pattern Recognition Society 38 (12) (2005) 2506–2518.
[4] J.R. Smith, S.F. Chang, Visualseek: a fully automated content-based image query system, in: Proceedings of ACM Multimedia, 1996, pp. 87–98.
[5] H. Bay, T. Tuytelaars, L.J. van Gool, SURF: speeded up robust features, in: Proceedings of the Ninth European Conference on Computer Vision (ECCV), vol. III, 2006, pp. 404–417.
[6] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, J. Malik, Blobworld: a system for region-based image indexing and retrieval, in: VISUAL, 1999, pp. 509–516.
[7] C. Chang, C. Lin, Libsvm: a library for support vector machines. 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm〉, 2001.
[8] Y. Chen, J.Z. Wang, Image categorization by learning and reasoning with regions, Journal of Machine Learning Research 5 (2004) 913–939.
[9] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision ECCV (2004) 1–22.
[10] T. Deselaers, D. Keysers, H. Ney, Features for image retrieval: a quantitative comparison, Lecture Notes in Computer Science 2021 (2004) 40–45.
[11] T. Deselaers, D. Keysers, H. Ney, Classification error rate for quantitative evaluation of content-based image retrieval systems, in: Proceedings of the 17th International Conference on Pattern Recognition, (ICPR'04), vol. 2, IEEE Computer Society, 2004, pp. 505–508.
[12] M. Everingham, A. Zisserman, C.K.I. Williams, L. Van Gool, The PASCAL visual object classes challenge 2006 (VOC2006) results. 〈http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf〉, 2006.
[13] V. Ferrari, T. Tuytelaars, L. van Gool, Simultaneous object recognition and segmentation by image exploration, Lecture Notes in Computer Science 4170 (2006) 145–169.
[14] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image and video content: the QBIC system, Computer 28 (9) (1995) 23–32.
[15] R.M. Haralick, Statistical and structural approaches to texture, in: Proceedings of the IEEE, vol. 67, 1979, pp. 786–804.
[16] J. Huang, S. Ravi Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, p. 762.
[17] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[18] F.-F. Li, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, 2005, pp. 524–531.
[19] J. Li, J.Z. Wang, G. Wiederhold, IRM: integrated region matching for image retrieval, in: Proceedings of the Eighth ACM International Conference on Multimedia, 2000, pp. 147–156.
[20] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[21] B.S. Manjunath, P. Salembier, T. Sikora, Introduction to MPEG 7: Multimedia Content Description Language, Wiley, New York, 2001.
[22] G. Pass, R. Zabih, Histogram refinement for content-based image retrieval, In: IEEE Workshop on Applications of Computer Vision, 1996, pp. 96–102.
[23] A. Pentland, R. Picard, S. Sclaroff, Photobook: content-based manipulation of image databases, International Journal of Computer Vision (1996) 233–254.
[24] F. Perronnin, C. Dance, G. Csurka, M. Bressan, Adapted vocabularies for generic visual categorization, in: Proceedings of ECCV, vol. 4, 2006, pp. 464–475.
[25] T.R. Reed, J.M. Hans du Buf, A review of recent texture segmentation and feature extraction techniques, CVGIP: Image Understanding 57 (3) (1993) 359–372.
[26] S. Santini, R. Jain, Beyond query by example, in: IEEE Second Workshop on Multimedia Signal Processing, 1998, pp. 3–8.
[27] I.K. Sethi, I. Coman, B. Day, F. Jiang, D. Li, J. Segovia-Juarez, G. Wei, B. You, Color-wise: a system for image similarity retrieval using color, in:

Proceedings of Storage and Retrieval for Image and Video Databases (SPIE), vol. 3312, 1998, pp. 140–149.
[28] V. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, 1995.
[29] J.Z. Wang, J. Li, G. Wiederhold, SIMPLIcity: semantics-sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (9) (2001) 947–963.

About the Author—AZIZI ABDULLAH received the BS degree (with honor) in Computer Science from the National University of Malaysia in 1996, and the Masters of Software Engineering at the University of Malaya, Malaysia in 1999. Currently, he is doing his PhD at the Department of Computer Science, University of Utrecht. His research interest includes machine vision, image indexing and retrieval.

About the Author—REMCO C. VELTKAMP studied Computer Science at Leiden University and completed the MS Thesis at IBM Scientific Centre, Paris, France. He received the PhD Degree from Erasmus University, Rotterdam, in 1992. He is currently a Professor in the Department of Information and Computing Sciences, Utrecht University. His current research focuses on the algorithmic aspects of multimedia retrieval, like the algorithmic design and analysis, and experimental verification.

About the Author—MARCO A. WIERING finished his PhD with the topic reinforcement learning in 1999 at the University of Amsterdam. From January 2000 until September 2007 he was an Assistant Professor at University Utrecht. He is now pursuing a enure track at the university of Groningen in the field of cognitive robotics. He is mostly researching the fields of machine learning, especially reinforcement learning, robotics, and machine vision.