# A Review of Codebook Models in Patch-Based Visual Object Recognition

**Amirthalingam Ramanan · Mahesan Niranjan**

**Abstract** The codebook model-based approach, while ignoring any structural aspect in vision, nonetheless provides state-of-the-art performances on current datasets. The key role of a visual codebook is to provide a way to map the low-level features into a fixed-length vector in histogram space to which standard classifiers can be directly applied. The discriminative power of such a visual codebook determines the quality of the codebook model, whereas the size of the codebook controls the complexity of the model. Thus, the construction of a codebook is an important step which is usually done by cluster analysis. However, clustering is a process that retains regions of high density in a distribution and it follows that the resulting codebook need not have discriminant properties. This is also recognised as a computational bottleneck of such systems. In our recent work, we proposed a resource-allocating codebook, to constructing a discriminant codebook in a one-pass design procedure that slightly outperforms more traditional approaches at drastically reduced computing times. In this review we survey several approaches that have been proposed over the last decade with their use of feature detectors, descriptors, codebook construction schemes, choice of classifiers in recognising objects, and datasets that were used in evaluating the proposed methods.

**Keywords** Bag-of-features · Cluster analysis · Object recognition · Visual codebook · SIFT

A. Ramanan · M. Niranjan
School of Electronics and Computer Science,
University of Southampton, Southampton, UK

M. Niranjan
e-mail: mn@ecs.soton.ac.uk

*Present Address:*
A. Ramanan (✉)
Department of Computer Science, University of Jaffna,
Jaffna, Sri Lanka
e-mail: a.ramanan@jfn.ac.lk

## 1 Introduction

An important problem in computer vision is to determine the presence or absence of a specific object class under a wide variety of conditions. Each three-dimensional object in the real world can cast an infinite number of different two-dimensional images onto the retina. Changes in pose, lighting, occlusion, clutter, intra-class differences, inner-class variances, deformations, background that varies relative to the viewer, large number of images and several object categories make this problem highly challenging.

The popular approach in visual object recognition is to use local information extracted at several points or patches in the image. Such local patch-based approaches have been shown to have benefits over global methods [30]. The assumption is, in different image classes, the statistical distribution of the patches is different. For instance, the patches showing spikes in the 'wheel' are more likely to appear in the images of vehicles than those of animals or persons. In the state-of-the-art visual object recognition systems, the visual codebook model has shown excellent categorisation performance in large evaluations (e.g. the PASCAL

**Figure 1** General framework of a visual object recognition system.
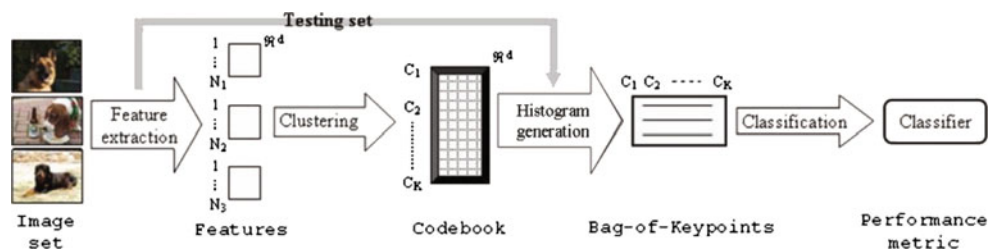


VOC Challenges).[1] Figure 1 shows the generic framework of such a codebook model.

Naturally, this framework ignores the spatial layout of features corresponding to overall shapes and sizes of objects, a limitation that will require community-wide attention in the future, that is outside the scope of this paper.

Desirable properties of a visual codebook are compactness, low computational complexity, and high accuracy of subsequent categorisation. Discriminative power of a visual codebook determines the quality of the codebook model, whereas the size of a codebook controls the complexity of the model. Thus, the construction of a codebook plays a central role that affects model complexity.

Several combinations of image patch detectors and descriptors, features, matching strategies, clustering methods and classification techniques have been proposed for codebook model-based visual object recognition. Assessing the overall performance of the individual components in such systems is difficult, since the computational requirements and the fine tuning of the different parts become crucial. However, a straightforward but effective approach lies in the use of the codebook model.

This review is organised as follows. In Section 2, we summarise the widely used visual descriptors, SIFT and SURF, in a patch-based visual object recognition framework. In Section 3, we present the bag-of-features approach that has proved to yield the state-of-the-art performance in large evaluations such as the PASCAL Visual Object Classes (VOC) Challenges. Section 4 provides various techniques that have been used in the literature in constructing visual codebook for object categorisation. The popular $K$-means method is also described in this section together with its drawbacks. In Section 5, the types of codebook models are discussed. In this section we provide a review of several codebook models that are prominent in the literature of object recognition or scene classification which have been proposed in the last decade. Section 6, discusses

a recent work which is free of a codebook model for visual object recognition. Finally, a summary of this paper is presented in Section 7.

## 2 Patch-Based Visual Descriptors

The feature extraction process in visual object recognition systems generally seeks for invariance properties that do not vary according to different conditions such as scale, rotation, affine and illumination changes. Usually images are composed of different set of colours, a mosaic of different texture regions, and different local features. Most previous studies have focused on using global visual features such as colour, texture, and shape that are important to describe image contents semantically to categorise objects in scenes. However, the introduction of powerful patch-based Scale-Invariant Feature Transform (SIFT) descriptors proposed by Lowe [33] had a significant impact on the popularity of local features. Interest points combined with local descriptors started to be used as a black box providing reliable and repeatable measurements from images for a wide range of applications such as object recognition, texture recognition, robot navigation and visual data mining. The local patch-based visual object recognition has several advantages that we list below:

– Local patch-based descriptors can robustly detect regions up to some extent which are translation, rotation and scale invariants addressing the problem of viewpoint changes [11, 48].
– Viewpoint invariant local descriptors provide a wide baseline matching [34].
– When objects to be recognised are partially occluded then global methods fails as it requires the outline of an object, but the patch-based method can cope well as the information is acquired at local point.
– Changes in the geometrical relation between image parts can be modelled in a flexible way [25, 41].
– The visual object classes do not need to be segmented prior to recognition [30].

---

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

Beside these advantages of the patch-based visual object recognition system, there are some known disadvantages.

- Although the interest points detected are significantly lower than the number of pixels in the image, the feature space suffers from the 'curse of dimensionality', i.e. each interest point detected by SIFT is described by a 128 dimensional vector.
- When using the bag-of-features approach with the patch-based object recognition systems, the physical location from where the patches were extracted gets discarded. In image scene classification, e.g. classification of 'sand' and 'sky', the performance may achieve better rates when spatial locations are preserved, i.e. in natural scenes 'sand' always appears at the bottom, whereas 'sky' always appears at the top. However, the usage of latent information makes the training of object recognition models more difficult.
- Interest points are detected when sharp changes happens in the intensities at any resolution of the image regions, e.g. the Difference of Gaussians (DoG) [34]. This causes the problem that those relevant parts of the object that were detected in the testing images may be missed in the training images due changes in resolution. If this is the case classification which relies on these parts is likely to fail.

In patch-based visual object recognition, SIFT is the most widely used descriptor due to good performances observed empirically. SIFT [33] detects interest points by filtering gray-value images at multiple scales that have sharp changes in local image intensities. The features are located at maxima and minima of a difference of Gaussian (DoG) function applied in scale space. Next, the descriptors are computed based on eight orientation histograms at a 4 × 4 sub region around

the interest point, resulting in a 128 dimensional vector. The SIFT algorithm can be summarised in four major stages: Scale-space extrema detection, keypoint localisation, orientation assignment and representation of a keypoint descriptor. Ke and Sukthankar [21] improved upon SIFT by replacing the smoothed weighted histograms with principal components analysis (PCA) at the final stage of the SIFT [21]. In this PCA-SIFT, the dimensionality of the feature space was reduced from 128 to 20 which requires less storage and increased speed in matching images. Recently, a colour image-based SIFT has been developed by Koen van de Sande et al. [56]. In this development, instead of using intensity gradients the color gradients were used into the Gaussian derivative framework.

More recently, Speeded-up Robust Features (SURF) are also becoming popular due to their faster performance with less number of interest points and dimension when compared to SIFT. The SURF [3] is partly inspired by SIFT that makes use of integral images. The scale space is analysed by up-scaling the integral image-based filter sizes in combination with a fast Hessian matrix-based approach. The detection of interest points is selected where the determinant of the Hessian matrix is maximum. The Hessian matrix is approximated using a set of box-type filters and no smoothing is applied when going from one scale to the next. Image convolutions with these box filters can be computed rapidly by using integral images independently of their size. The use of integral images drastically reduces the computation time. Next, the descriptors are computed based on orientation using 2D Haar wavelet responses calculated in a 4 × 4 sub region around each interest point, resulting in a 32 dimensional vector. When information about the polarity of the intensity changes is considered, this in turn results in a 64 dimensional vector. The extended version of SURF (e-SURF) has the same dimension as SIFT.

**Figure 2** **a** Original Image **b** SIFT keypoints with magnitude and direction **c** SURF keypoints with magnitude and direction.
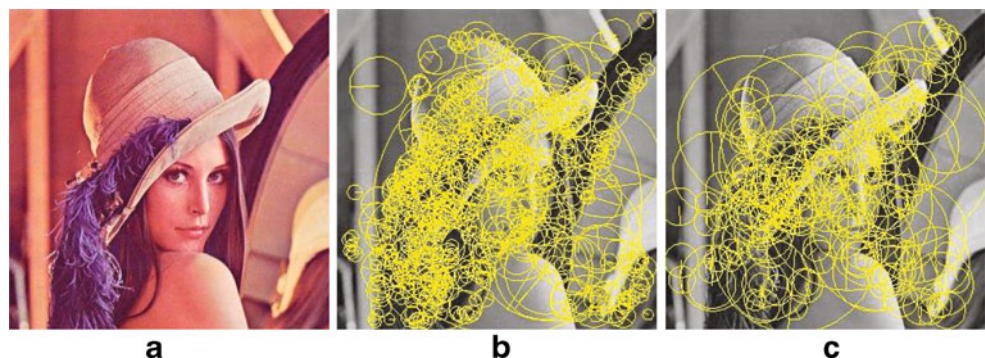
In contrast, SIFT when compared to PCA-SIFT and SURF, has shown better performance but it is slow and performs poorly at illumination changes [18]. A survey on invariant detectors, descriptors and implementation details can be found in [40, 55]. The SIFT and SURF features detected on the Lena's image are illustrated in Fig. 2b and c, respectively. Here the features detected are shown by centres of the circles, where the radius reflects magnitude and the direction reflects the orientation of the feature. The majority of the features are detected in the face, rim of the hat and mirror, and in other textured regions of the image. The SIFT detected $473 \times \mathbb{R}^{128}$ interest points while SURF detected $176 \times \mathbb{R}^{64}$ keypoints, resulting with 57 keypoints overlapping exactly.

## 3 Bag-of-Features

The bag-of-words (BOW) approach was originally used in text mining [17] and is now widely used in image scene classification [13, 48], retrieval of objects from a movie [52], and object classification [9, 19, 43, 50, 59, 64] tasks in computer vision. The bag-of-words in computer vision is normally referred as 'bag-of-features' or 'bag-of-keypoints'. The pseudocode of bag-of-features approach is given in Algorithm 1.

---

**Algorithm 1** Proccess of building a bag-of-feature (BOF) representation for images

---

**for all** *image* **do**
        interestPts ← detectPts(*image*)
        descriptors ← describePts(interestPts)
**end for**

codebook ← quantizePts(descriptors(*training_images*))
**for all** *image* **do**
        BOF ← computeHistogram(codebook, descriptors(*image*))
**end for**

---

Interest points or regions are detected in training images and a visual codebook is constructed by a vector quantization technique that groups similar features together. Each group is represented by the learnt cluster centres referred as 'visual words' or 'codewords'. Each interest keypoint of an image in the dataset is then quantized to its closest codeword in the codebook, such that it maps the entire patches of an image in to a fixed-length feature vector of frequency histograms, i.e. the visual codebook model treats an image as a distribution of local features. The size of the resulting histogram equals the size of the codebook and hence the number of clusters obtained from the clustering technique.

The aforementioned histogramming process can be mathematically expressed as follows. For each codeword $c$ in the visual codebook $\mathbf{C}$ the traditional codebook model constructs the distribution of codewords over an image by

$$H(c) = \sum_{c \in IR} \begin{cases} 1 & ; \text{ if } c = \underset{c \in \mathbf{C}}{\operatorname{argmin}} \, S(c, r) \\ 0 & ; \text{ otherwise} \end{cases} \quad (1)$$

where, $IR$ denotes the set of regions or patches in an image $I$ and $S(c, r)$ denotes the similarity (e.g. Euclidean distance) between a codeword $c$ and a region $r$. The mathematical expression in Eq. 1 of assigning a single codeword to a single image features is referred to as hard-assignment. Instead of hard-assignment, each region $r$, can be assigned to all codewords in a probabilistic manner, i.e. assign weights $w_c$ to neighbouring codewords. Hard-assignment becomes soft-assignment when Eq. 1 is replaced by

$$H(c) = \sum_{c \in IR} S(c, r) \times w_c \quad (2)$$

The traditional codebook approach makes use of the hard-assignment method. A soft-assigned method combines the spatial verification, in which each interest point in an image has more assigned codewords and can potentially match more features in the other image.

## 4 Clustering Algorithms Used in Codebook Construction

When local features are extracted from images of a particular class, the variability in images makes the number of detected features to vary. It is the difficulty of matching images by measuring a distance between them using variable number of features, which is circumvented by representing the statistics of these features by the bag-of-words approach. The codebook itself is constructed by clustering a large number of low-level feature descriptors extracted from training data. Based on the choice of a clustering algorithm, one might obtain different clustering solutions, some of which might be more suitable than others for object class recognition.

The popular approach to constructing a visual codebook is usually undertaken by applying the $K$-means method [9, 13, 32, 52, 53, 59]. Several other clustering techniques have been employed to construct visual codebooks:

– Agglomerative clustering [30]
– Gaussian mixture models [11, 12, 25, 43]

- Hierarchical *K*-means clustering [37, 42]
- Mean-shift based clustering [19]
- Randomised clustering forests [41]

We now brief the widely used *K*-means clustering technique. Given a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ (representing *N* points—rows—described with respect to *D* features—columns), then *K*-means clustering aims to partition the *N* points into *K* disjoint sets or clusters by minimizing an objective function, which is the squared error function, that minimizes the within-group sum of squared errors:

$$d_{ij} = \parallel X_i^{(j)} - C_j \parallel^2$$

$$\mathbf{X}_{\text{opt}} = \sum_{j=1}^{K} \sum_{i=1}^{N} d_{ij}$$

where $d_{ij}$ is a chosen distance measure between a data point $X_i^{(j)}$ and the cluster centre $C_j$, is an indicator of the distance of the *N* data points from their respective cluster centres.

The cluster centres obtained by *K*-means are the average of the points within their respective clusters that are useful only when mean is defined, but cannot be used in categorical data. *K*-means is unable to handle noisy data and outliers. It is also not suitable to discover clusters with non-convex shapes. Although it can be proved that the iterative procedure will always terminate, the *K*-means algorithm does not necessarily find the most optimal solution, corresponding to the global objective function that minimises the squared error within clusters [16, 29].

There are several other known difficulties with the use of *K*-means clustering, including the choice of a suitable value for *K* and the computational cost of clustering when the dataset is large. It is also significantly sensitive to the initial randomly selected cluster centres. The *K*-means algorithm can be run multiple times to reduce this effect, but that makes it computationally more expensive and might take several months or even years to cluster millions of data! The time complexity of the *K*-means method is O($NDKm$), where the symbols in parentheses represent number of data, dimensionality of features, the number of desired clusters and the number of iteration of the expectation-maximization (EM) algorithm.

It is important to note here that clustering by *K*-means and similar algorithms results in cluster centres which best represent the probability density of the space of features. There is no a priori reason to believe that preserving the density in this way should result in carving the space into partitions that capture rare

and informative visual keywords that help discriminate between image classes. As noted by Jurie and Triggs [19] such density preserving clustering will work well in homogeneous images such as textures, but with real world object recognition tasks we should expect a highly non-uniform distribution in feature space. Furthermore, clustering millions of data vectors of higher dimensions into thousands of cluster centres using the *K*-means or GMMs techniques is not straightforward to apply and need for novel approaches arises. Therefore, our own work on a novel one-pass algorithm [50] constructs a visual codebook resulting in a codebook that retains rare and novel features while drastically reducing computational costs.

## 5 Codebook Models

A codebook model yields a distribution over codewords that models the whole image, making this model well-suited for describing context. Unlike text, visual words are not intrinsic entities and different quantisation methods can lead to very different performances. The size of the codebooks that have been used in the literature range from $10^2$ to $10^4$, resulting in very high-dimensional histograms. A larger size of codebook increases the computational needs in terms of memory usage, storage requirements, the computational time to construct the codebook and to train a classifier. On the other hand, a smaller size of codebook lacks good representation of true distribution of features. Thus, the choice of the size of a codebook should be balanced between the recognition rate and computational needs. The compactness constraint is typically ignored by several systems who mainly focus on catergorisation performance.

Now, we provide a review of a selective research work from patch-based visual object recognition literature. In general, there are two types of codebook that are widely used in the literature: global and category-specific (or concept-specific) codebook. A global codebook may not be sufficient in its discriminative power but it is category-independent, whereas a category-specific codebook may be too sensitive to noise. The conventional approach to constructing either a global or category-specific codebook is achieved by cluster analysis, usually by the *K*-means method. The learnt cluster centres are not semantically meaningful since the clustering is based on appearance similarity only. However, another type of codebook is the semantic codebook approach that attempts to bring the semantic information into visual codebooks. This semantic codebook model is widely used in image scene categorisation.

5.1 Globally-Clustered Codebook

A globally-clustered codebook is usually constructed by clustering visual descriptors that are randomly chosen from each class of a training set. Thereafter, each image is represented as a feature vector by computing the frequency histograms with the learnt clusters. This mapping produces a bag-of-features representation. Several authors [9, 19, 37, 42, 43, 63] have used the globally-clustered codebook at some stage in their framework.

– Csurka et al. [9] used the Harris affine region detector [39] to identify the interest points in the images which are then described by SIFT descriptors. A visual codebook was constructed by clustering the extracted features using $K$-means method. Images are then described by histograms over the learnt codebook. The authors run the $K$-means several times over a selected size of $K$ and different sets of initial cluster centres. The reported results were the clusters that gave them the lowest empirical risk in classification. The size of the codebook used in reporting the results is 1000. The authors compared Naive Bayes and Support Vector Machine (SVM) classifiers in the learning task and found that the one-versus-all SVM with linear kernel gives a significantly (i.e. 13%) better performance. The proposed framework was mainly evaluated on their 'in-house' database that is currently known as 'Xerox7' image set containing 1,776 images in seven object categories. The overall error rate of the classification is 15% using SVMs. Our resource-allocating codebook (RAC) approach in [50] when applied on the Xerox7 image dataset performs slightly better than the authors' method but was achieved in a tiny fraction of computation time.
– Jurie and Triggs [19] proposed a mean-shift based clustering approach to construct codebooks in an undersampling framework. Our RAC approach which is briefly explained in this section, has strong similarities to this technique. The authors sub sample patches randomly from the feature set and allocate a new cluster centroid for a fixed-radius hypersphere by running a mean-shift estimator [8] on the subset. The mean-shift procedure is achieved by successively computing the mean-shift vector of the sample keypoints and translating a Gaussian kernel on them. In the next stage, visual descriptors that fall within the cluster are filtered out. This process is continued by monitoring the informativeness of the clusters or until a desired number of clusters is achieved.

The features used in their experiments are the gray level patches sampled densely from multi-scale pyramids with ten layers. Three different feature selection methods proposed by [5] were used in the experiments: maximisation of mutual information, odds of ratio, and training an initial linear SVM on the entire training set to select the features that have the highest weight. Two different ways of producing fixed-length feature vectors from the learnt codebook were used in the experiments: Binary indicator vectors which were produced by thresholding the frequency counts of the codeword in the image and the histograms. The proposed method was evaluated on three datasets: Side views of cars from [1], Xerox7 image dataset [9] and the ETH-80 dataset [30] containing four object categories (cars, horses, dogs and cows) each with 205 images. Naive Bayes and linear SVM classifiers were compared in all their experiments. The size of the codebook was 2,500. Based on the obtained experimental results, the authors conclude the following: (i) the initial training with linear SVM in feature selection was better however, full codebooks generally outperformed compact codebooks, (ii) mean-shift based codebooks outperformed $K$-means based codebooks, (iii) histogram representation performs better than binary indicators, and (iv) linear SVMs easily outperform Naive Bayes classifiers. The authors' mean-shift based clustering method is computationally intensive in determining the cluster centroid by mean-shift iterations at each of the sub samples. The convergence of such a recursive mean-shift procedure greatly depends on the nearest stationary point of the underlying density function and its utility in detecting the modes of the density. Also efficient computation of the mean-shift method requires the sub sampling of visual keypoints with a regular grid and the selection of the bandwidth. A technique that has many parameters can overfit data and generalise poorly [46]. In contrast, the RAC approach pursued in [50] has a single threshold that takes only one-pass through the entire data, making it computationally efficient.

– Nister and Stewenius [42] proposed a hierarchical $K$-means clustering that constructs a vocabulary tree in an offline training stage for image retrieval from a large database. Features were extracted using maximally stable extremal regions (MSERs) [36] which are then described by SIFT descriptors. SIFT features were then quantized with the vocabulary tree. The vocabulary tree is constructed by a hierarchical scoring scheme based

on the term frequency-inverse document frequency (*tf-idf*) score. First, an initial *K*-means process is run on the training data, defining *K* centroids. The training data is then partitioned into *K* groups, where each group consists of the features closest to a particular centroid. The second step is then recursively processed by quantizing each node into *K* new parts, where *K* defines the number of children of each node. The tree is constructed level-by-level up to a maximum number of levels. Following the recursive process, in the online phase, each visual descriptor is propagated down the vocabulary tree by coding the closest node at each level. The proposed technique was tested on a ground truth database containing 6,376 images in groups of four of the same object but under different conditions. From their experimental results, they found that larger vocabulary (between 1 and 16 million leaf nodes) improves retrieval performance. They claim that this methodology provides the ability to make fast searches on extremely large databases (i.e. one million images).

– Mikolajczyk et al. [37] find local features by extracting edges with a multi-scale Canny edge detector [7] with Laplacian-based automatic scale selection. For every feature, a geometry term gets determined, coding the distance and relative angle of the object centre to the interest point, according to the dominant gradient orientation and the scale of the interest point. These regions are then described with SIFT features that are reduced to 40 dimension via principal component analysis (PCA). The visual codebook is constructed by means of a hierarchical *K*-means clustering. Initially the features are clustered using *K*-means algorithm and then agglomerative clustering is performed to obtain compact feature clusters within each partition. Given a test image, the features were extracted and a tree structure is built using the hierarchical *K*-means clustering method in order to compare with the learnt model tree. Classification is done in a Bayesian manner computing the likelihood ratio. This test is done at local maxima of the likelihood function of the object being present. Some additional tests are applied to determine whether objects of different classes share similar clusters or whether overlapping objects exist. In this manner, the location, scale and orientation of multiple objects can be determined. Experiments were performed on a five class problem taken from the PASCAL VOC 2005 image dataset containing four classes and a RPG (rocket-propelled grenade) shooter that was collected from various sources.

– Wu and Rehg [60] showed that when the histogram intersection kernel (HIK) are used in clustering patch-based visual descriptors that are histograms, the codebooks constructed produce improved bag-of-features classifiers. The proposed method replaces *K*-means clustering that uses the $L_2$ distance measure with HIK for better performance when the choice of feature representation is histograms. When comparing *K*-means with *K*-median, the latter uses the $L_1$ distance measure. In the first step, features are extracted to construct a visual codebook of size 200. At the next step, an image or image sub-window is represented by a histogram of codewords in a specified image region. An image is represented by the concatenation of histograms from all 31 sub-windows that split an image into three levels, resulting in a histogram of dimension 6200. Spatial and edge informations are incorporated as an additional input, and histograms are concatenated from the original input and Sobel gradient image. The authors also propose a one-class SVM formulation using HIK that can be used to improve the effectiveness of the HIK-based codebook, by compact clusters in histogram feature space. The proposed methods are validated using three datasets: the dataset used in [28] containing 15 classes, a sports event dataset containing eight categories, and the Caltech-101 object recognition dataset. For the experiments performed with Caltech-101 image datasets, SIFT descriptors were used to describe the image patches and densely sampled features over grid. The census transform histogram (CENTRIST) descriptors [61] proposed by these authors was used with the other datasets. The original dimensionality of the CENTRIST descriptor is 256 which can be also reduced to 40 via PCA. One-versus-one SVM is used for classification with the histogram intersection kernel. The authors empirically show that the *K*-median codebook is a compromise between the HIK and *K*-means codebooks.

– In our recent work [50], we demonstrated a resource-allocating codebook (RAC) method to constructing a discriminant visual codebook that takes only one-pass through the entire data, inspired by the resource allocation network (RAN) family of algorithms [45]. The RAC starts by arbitrarily assigning the first data item as an entry in the codebook. When a subsequent data item is processed, its minimum distance to all entries in the current codebook is computed using an appropriate distance metric. If this distance is smaller than a predefined threshold (radius of the hypersphere),
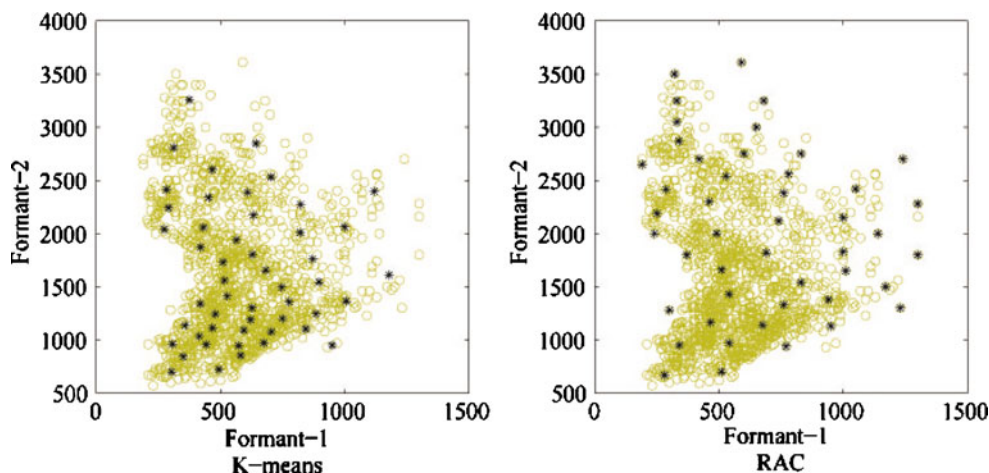
**Figure 3** An example of feature space partitioning using *K*-means clustering (*left*) and RAC approach (*right*) applied on the Peterson's dataset [44] that are vowel sounds characterised by the first two formant frequencies. The figure is a plot of 50 cluster centres found on the entire vowel dataset. It can be seen that the cluster centres found by *K*-means are around densely populated areas, whereas the centres of RAC are around lesser occurrence data points. Thus in RAC outlier (or less occurrence) data get included as part of the codebook.

the hypersphere that includes the processed data item is redefined by the weighted average of all its previous points and the new point. If the threshold is exceeded by the smallest distance to codewords, a new entry in the codebook is created by including the current data item as the additional entry. This process is continued until all data items are seen only once.

A comparison of RAC algorithm with *K*-means and a closely related mean-shift method of Jurie and Triggs [19] was tested on a set of binary classes selected from the PASCAL VOC Challenge 2007 dataset. The RAC strategy performs similar or slightly better than the methods compared in the experiments but achieved in a tiny fraction of computation time. RAC when applied on the Xerox7 image dataset, performs slightly better than the method in [9] with an error rate of 13.64%, but was achieved in a tiny fraction of computation time. That is, *K*-means clustered $105,000 \times \mathbb{R}^{128}$ SIFT descriptors into 1,000 clusters in an average time performing each fold of the ten-fold cross-validation in 149 h while RAC only needed an average time of 19 min on a cluster computer with a dual core Xeon running at 2.6 GHz and 48 GB of RAM, showing the drastic reduction in the computational needs. In all the experiments the authors have used SIFT features and the classification was performed using one-versus-all linear SVMs. In contrast, RAC looks for visual codebook that has a wider span of the feature space than that found by any density preserving clustering methods, such as K-means

algorithm. Figure 3 illustrates the partitioning of a feature space using the *K*-means and RAC techniques, respectively. Note that the cluster centres found by *K*-means populate the densest part of the feature space, whereas RAC finds centres that each represent a distinct part of the feature space.

## 5.2 Category-Specific Clustered Codebook

A category-specific or concept-specific codebook is usually constructed by clustering the extracted features from images in a single class only. Sometimes, the features can also be extracted with a concept that covers different and independent regions of the same category or scene. This makes the resulting clusters depend on only that subset of the feature space which is relevant for the concept. The construction process of a codebook is identical to the globally-clustered codebook, and is carried out separately for each of the categories or concepts. Several authors [12, 23, 29, 52, 63] have used the category-specific clustered codebook at some stage in their framework.

– Sivic and Zisserman [52] proposed an approach to retrieve visual objects and scenes from a movie using a text retrieval approach. Local regions were extracted from each frame in the video in the following two different ways: One method is referred to as a shape-adapted (SA) region which surrounds an interest point by an elliptical shape. The second method is referred to as a maximally stable (MS) region which is constructed by intensity watershed

image segmentation. The SA regions are detected on corner like regions and the MS regions correspond to blobs of high contrast with respect to the surroundings. Both SA and MS regions are then described by SIFT descriptors. The authors were aware of the difficulty in clustering a very large scale of descriptors extracted from their movies, so instead they selected 10,000 frames which represent about 10% of all the frames in the movie, resulting in 200,000 averaged track descriptors to construct a codebook. A visual codebook is constructed using $K$-means clustering algorithm and Mahalanobis distance measure. The Mahalanobis distance function between two patch-based visual descriptors $\mathbf{x}$ and $\mathbf{y}$ of the same distribution with the covariance matrix $\Sigma$, is given by:

$$d\left(\mathbf{x} - \mathbf{y}\right) = \sqrt{\left(\mathbf{x} - \mathbf{y}\right)^T \Sigma^{-1} \left(\mathbf{x} - \mathbf{y}\right)} \quad (3)$$

The authors claim that the Mahalanobis distance enables more noisy components of the SIFT features to be weighted down and also decorrelates the components. It would not be appropriate to use the covariance matrix over the entire feature space, since it is mainly influenced by inter-class variations. $K$-means was run several times with different sets of initial cluster centres to maximise retrieval results. The codebook constructed using SA features was about 6,000 and the codebook using MS features was about 10,000. The ratio of the size of codebooks for each type is chosen nearly to the ratio of detected descriptors of each type. The collections of codewords are used in the term frequency-inverse document frequency (*tf-idf*) scoring of the relevance of an image to the query. The *tf-idf* scoring is used in information retrieval and text mining. The *tf* term measures the number of occurrences of a particular codeword in the example divided by the total number of patch-based features in the example. The *idf* term measures the distinctiveness of a particular codeword over different examples. The performance was evaluated on two feature films: 'Run Lola Run' and 'Groundhog Day'. The authors have constructed codebooks sufficient for two films in a very computationally expensive way, which makes it hard to apply by using the $K$-means method for a large number of films. The proposed system allows to reduce noise sensitivity in matching and to search efficiently through a given video for frames containing a particular object using inverted files. Furthermore, in such a system, better performance can be achieved by using a large number of visual words.

However, this high number of visual words leads to less compact models, which may be infeasible for large video sets.

– Leibe and Schiele [30] used the Harris interest point detector [15] to extract image patches. The pixel gray values of those patches are then clustered using the agglomerative clustering method to generate a visual codebook. The size of the learnt codebook was further reduced by merging the most similar clusters in a pair-wise manner when the similarity between clusters exceeds a predefined threshold $t$. The similarity between two clusters $C_1$ and $C_2$ was measured by the normalised grey-value correlation (NGC).

$$similarity\,(C_1, C_2) = \frac{\sum_{x \in C_1, y \in C_2} NGC(x, y)}{|C_1| \times |C_2|} \quad (4)$$

where,

$$NGC\,(x, y) = \frac{\sum_i (x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_i (x_i - \overline{x_i})^2 \sum_i (y_i - \overline{y_i})^2}} \quad (5)$$

Instead of assigning image patches to their nearest codeword in the learnt codebook, every patch casts probabilistic votes to the codebook using the NGC measure whose similarity is above $t$. For classification, a generalised Hough transform-like [33] voting scheme is applied. The proposed method was evaluated on a database of 137 images of scenes containing one car each in varying poses. The size of the codebook was around 2,500.

– Farquhar et al. [12] proposed alternatives to the scheme introduced by Csurka et al. [9]. The Gaussian mixture model (GMM) was proposed as a replacement of the $K$-means based codebook construction, and summed responsibility replacing bin membership for histogram generation. The GMMs were all trained for category-specific codebooks and were then combined into a single codebook. Features were extracted using multi-scale Harris affine region detector that are then described by SIFT descriptors. The features were pre-processed to reduce its dimensionality. The authors used two different methods to reduce dimensions: the PCA and partial least squares (PLS), and found that PLS improves classification performance over the PCA method for the same number of reduced dimensions. The proposed method was also tested on the Xerox7 image dataset used in [9]. The classification results were obtained by using one-versus-all SVM classifiers with linear kernel. Although about 2% of improvement was obtained over the original results of [9], the concatenation of category-specific

codebooks into a single codebook approach is impractical for a large number of visual object categories, as the size of the concatenated codebook grows linearly with the number of classes. When the number of classes increases, not only does it increase the computational cost but it also makes the classification of histograms challenging due to its diverse range in object classes.

– Zhang et al. [63] compare sets of local features in two different methods. Their first method involved clustering a set of patch-based descriptors in each image to form a representation of $(c_i, w_i)$ pairs, that they refer to as image *signature* where $c_i$ is the cluster centre and $w_i$ is the proportional size of the $i$th cluster. Cluster centres were obtained using $K$-means algorithm with $K = 40$. Earth Mover's Distance (EMD) [51] was the choice for measuring similarities between image representations. The EMD between two image signatures $S_1 = \{(p_1, u_1), \cdots, (p_m, u_m)\}$ and $S_2 = \{(q_1, w_1), \cdots, (q_n, w_n)\}$ is defined as:

$$D(\mathbf{S}_1, \mathbf{S}_2) = \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij}\, d\,(p_i, q_j)}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \qquad (6)$$

where, $f_{ij}$ is a flow value that is usually determined by solving a linear programming problem, and $d\,(p_i, q_j)$ is the ground distance (e.g. Euclidean distance) between cluster centres $p_i$ and $q_j$.

The second method was clustering the patch-based descriptors from a training set to build a global codebook by concatenating class-wise codebooks and then represent each image as a frequency histogram. The class-wise codebook was also constructed by $K$-means method. $\chi^2$ distance measure was used in this case to compare two histograms $\mathbf{x}$ and $\mathbf{y}$, which is defined as:

$$D(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^m \left[ \frac{(x_i - y_i)^2}{x_i + y_i} \right] \qquad (7)$$

Interest points were detected using the Harris and Laplacian detector, and were compared with different invariance properties: Scale invariance only, scale with rotation invariance, and affine invariance. The SIFT and/or SPIN [27] descriptors were used to describe the interest points found by different detectors as mentioned above. Each detector/descriptor pair is considered as a separate channel at the classifier stage. One-versus-one SVM classifiers were compared with three different kernels: linear, $\chi^2$, and the EMD. Their experimental evaluations were performed on four texture

(UIUCTex, KTH-TIPS, Brodatz, and CUReT) datasets and five object category (Xerox7, Caltech-6, Caltech-101, Graz, and PASCAL VOC 2005) datasets. Based on their experiments they conclude that the combination of Harris and Laplacian detectors with SIFT and SPIN descriptors is the preferable choice in terms of classification performance together with the choice of the $\chi^2$ kernel. The $\chi^2$ kernel performs better than the linear one and at the same time it is comparable with the EMD kernel.

### 5.3 Semantic Codebook

The semantic relationship between features is useful especially for scene understanding. The construction of a semantic codebook can be categorised as supervised and unsupervised approaches.

The supervised approach is achieved by image patch annotation or image annotation [41, 62, 64] that yield meaningful codewords making the codebook more compact and discriminative. The underlying phenomenon of selecting meaningful codewords is that the local image semantics will propagate to the global codebook image model. However, not all images can be decomposed into semantic codewords. For example, an indoor scene, say a house, is unlikely to contain sea, sky, rock, sand, and mountain. In addition, manually annotating local patches in large evaluations, especially when there are multiple object categories present in most of the images, becomes a time consuming process. Moreover, several other authors [31, 57, 59] use mutual information between the features and class labels to create the semantic codebook from an initial and relatively larger codebook constructed by the $K$-means method.

The unsupervised approach is typically used to model the visual codeword co-occurrences in object categories. Codeword co-occurrence is typically modeled by a generative probabilistic model [13, 25, 48, 53]. Typically, a generative model is built on top of a codebook model. In this approach, visual words are considered as generated from latent aspects (or topics). The model expresses images as combinations of specific distributions of topics that can essentially be a semantic codebook. In general, this approach involves many parameters to be estimated. The parameter estimation is much more time consuming and difficult to find the optimal values that yields better performance.

In the following section we present the codebook models that have been categorised according to some aspects, such as, discriminative power, compactness, probabilistic models, and unifying codebook construction with classifier learning.

### 5.3.1 Compact and Discriminative Codebook

– Winn et al. [59] optimised codebooks by hierarchically merging visual words in a pair-wise manner using the information bottleneck principle [54] from an initially constructed large codebook. The final visual words are represented by the GMMs of pixel appearance. Training images were convolved with different filter-banks made of Gaussians and Gabor kernels. The resulting filter responses were clustered by the $K$-means method with a large value of $K$ in the order of thousands. Mahalanobis distance between features is used during the clustering step. The learnt cluster centres and their associated covariances define a universal visual codebook. Following the construction of this large codebook, each region of the training images is processed to compute the histogram **h** over the initial codebook and the corresponding histogram **H** of target codewords. A mapping function $\mathbf{H} = \phi(\mathbf{h})$ is used to produce a much more compact visual codebook, where $\phi$ is the pair-wise merging operation that acts on the initial codewords. Classification results were obtained on photographs acquired by the authors, images from the web and a subset of 587 images in total that were selected from the PASCAL VOC challenge 2005 dataset containing four classes. Gaussian class models were compared with multi-modal nearest neighbours in classification. Their class models were learnt from a set of manually segmented photographs into object-defined regions. Even though the authors claim that the proposed technique is simple and extremely fast, the complex learning process i.e. the initial codebook construction based on $K$-means clustering and the merging of visual words make it harder to apply on large number of features. However, if two distinct visual words are initially grouped in the same cluster, they cannot be separated later. Also the vocabulary is tailored according to the categories under consideration, but it would require fully retraining the framework on the arrival of new object categories, whereas the RAC technique can cope with new object categories without retraining the whole system.

– Wang [57] proposed the construction of a discriminant codebook at a multi-resolution level using a hierarchical clustering technique and then use a boosting feature selection method to select the discriminant codewords. Features were extracted using the Harris affine interest point detector and SIFT descriptor. The extracted patch descriptors are clustered into a sufficiently large number of clusters (e.g. 2000). These clusters are then hierarchically clustered in a bottom-up way to generate new clusters in each level. Centroids of these clusters form a multi-resolution codebook that is usually very large as it includes more resolution levels. To reduce the size of the codebook, discriminant codewords are selected by a threshold-based boosting feature selection technique. To do this, frequency histograms of the training images are sorted according to a histogram feature. Using the threshold through the sorted list, the weak classifier giving the minimal training error is selected and the corresponding codeword in the codebook is indicated to be inactive. The choice of classifier was the Kernel Fisher Discriminant Analysis (KFDA) with the RBF kernel. Their method is evaluated against a selected four class problem (motorbikes, airplanes, faces_easy, and background_Google) from the Caltech-101 image dataset. However, this method involves greater computation and suffers from the difficulty in identifying the optimal value of the size of an initial codebook.

– Kim and Kweon [23] proposed a technique to reduce the size of a codebook and enhance its discriminative power by eliminating some visual codes from the codebook using an entropy-based minimum description length (MDL) criterion. This process involves the construction of intra-class and inter-class codebooks. The intra-class codebook is initially constructed for each object category using an agglomerative $K$-means clustering method. The MDL of each category-specific codebook is then computed. If the MDL is not minimum then the codebook that has the lowest entropy is removed. Following this step, the inter-class entropy of a codebook that has large entropy is removed from the intra-class codebook yielding the inter-class codebook. The authors used their own feature that they refer to as the generalised robust invariant feature (G-RIF) [22]. The 189 dimensional G-RIF was reduced to five dimensions via PCA. In their first experiment, an intra-class codebook was used to compare SVMs with nearest neighbour classifiers using different distance measures: KL-divergence, $\chi^2$, histogram intersection and Euclidean distances. Based on the experimental results, they found that the NN with KL-divergence gives better performance than SVMs. This might be the case as a small set of 15 training samples from each category was used in training the SVMs. Also it is reported that the directed acyclic graph (DAG) SVMs [47] for multi-class classification performed worse than

one-versus-all SVMs. A selected ten object category from the Caltech-101 image dataset was used in their first experiment. In the second experiment, an inter-class codebook was used to evaluate the classification performance of the entire Caltech-101 image dataset (15 training and 15 images for testing) using nearest neighbour with KL-divergence distance metric. However, there is a large amount of computation involved in constructing both the intra and inter-class codebooks and the resized codebook is not optimally compact.

–   Moosmann et al. [41] introduced extremely randomised clustering (ERC) forests to construct a visual semantic codebook. Initially a tree is built using random forests [6]. This tree is used as a spatial partitioning method by assigning each leaf of each tree a visual word, which is how a semantic visual codebook is constructed, instead of using it as a classifier. Compared to random forests using C4.5 [49], extremely randomised trees are faster to construct. Different types of features were used in their experiments: an HSL (hue, saturation, and lightness) colour descriptor of 768 dimensions ($16 \times 16$ pixels $\times$ 3), a Haar wavelet-based colour descriptor that transforms this into another 768 dimensions, and SIFT descriptor of 128 dimension. A detailed experimental piece of work was carried out with a Graz-02 image dataset[2] containing three object categories (bicycles, cars and persons) and negatives (i.e. none of the three object categories are present). The PASCAL VOC challenge 2005 image dataset and a horse database[3] were also used in evaluating their method. The sizes of the codebooks used with Graz-02 and PASCAL VOC 2005 are 5,000 and 30,000, respectively. A linear SVM classifier was employed in the classification tasks. However, this approach creates a very large codebook which has difficulty in coping with large datasets. In addition, it can lead to overfitting.

–   Li et al. [32] proposed the construction of a discriminant codebook in a similar fashion to that proposed in [58] and [59], i.e. constructing a compact codebook through selecting a subset of codes from an initially learnt large codebook. An initial codebook was constructed using $K$-means clustering algorithm. Each codeword in this codebook is then modelled by a spherical Gaussian function through which an intermediate representation for each training image is obtained. A Gaussian model

for every object category is learnt based on this intermediate representation. Following this step, an optimal codebook is constructed by selecting discriminant codes according to the learnt Gaussian model. The discriminative capability is measured either by likelihood ratio or by Fisher score. Interest points in their experiments were detected by the DoG detector and were described by SIFT descriptors. Classification was performed using SVM classifiers with RBF kernel. The authors claim that the likelihood ratio performs better than the Fisher score as it fits their classification problem. This method was evaluated on the Caltech-4 object dataset containing four object categories within a total of 2,876 images and a background class with 450 images. All the images for training or test were scaled to 300 pixels in width. They also carried out experiments with different codebook sizes using the algorithm proposed in [9] using the Caltech-4 dataset. The highest classification rate achieved was 91.5% with a codebook size 900. The best performance of Li et al.'s method was 90.5%, that was achieved with a more compact codebook of size 100, where the optimal codes were selected from an initial codebook of size 1,400. Although effective, it still suffers from the disadvantages caused by $K$-means clustering in the construction of an initial large codebook.

### 5.3.2 Unified Codebook Construction with Classifier Learning

–   Yang et al. [62] proposed a unified codebook generation that is integrated with classifier training. Unlike clustering approaches that associate each image's low-level features with a single codeword in their approach (see Fig. 4) images are represented
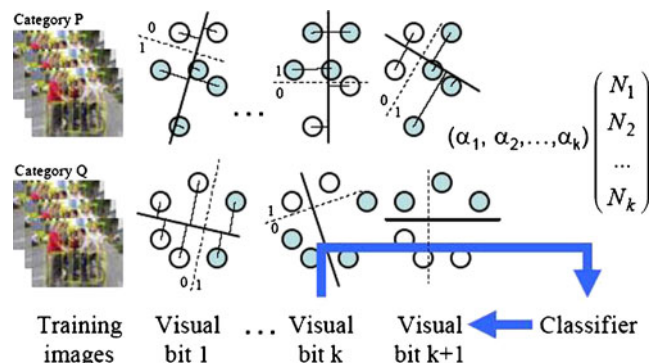
**Figure 4** Overview of the unified visual bit generation and classification process. This diagram adapted from [62] with permission.

by means of visual bits associated with different categories, i.e. an image which can contain objects from multiple categories is represented using aggregates of visual bits for each category that constitute the semantic vocabulary. If a feature is considered to better describe an image category, then its visual bit is '1', otherwise '0'. Each visual bit is a linear/RBF kernel classifier that maps the features to a binary bit for classification. These visual bits are augmented iteratively to refine visual words based on the learning performance of the classifier. The iterative process is carried out until a desired performance is achieved. Harris Laplace corner detectors [38] were used in detecting interest points and were described by SIFT descriptors. The authors compare their technique with the $K$-means based codebook of size 1,000 followed by an SVM classifier that uses the $\chi^2$ kernel, and with a codebook constructed using the extremely-random classification forest algorithm [41]. Evaluations were performed on the PASCAL VOC Challenge 2006 image dataset that contains ten classes of total 5,304 images.

– Zhang et al. [64] proposed an iterative non-redundant codebook construction process by means of a weighted voting scheme of the AdaBoost procedure that is integrated with classifier learning. The authors applied this framework in visual object recognition and document classification domains with different experimental setups. However, in this paper, the visual object recognition part is described for clarity. The following steps are iterated for a pre-defined number of iterations $T$:

1. a base codebook is learnt from a bag-of-features that are associated with a set of weights. The weights are initialised to be uniform over the training set.
2. training images are then mapped to fixed-length vectors using the *tf-idf* weight. A classifier is then learnt from the fixed-length feature vectors.
3. the predictions of the classifier in step 2 are used to update the weights using the AdaBoost procedure to the next iteration from step 1.

Different feature detectors: Hessian affine, the Kadir and Brady Salient regions, and the principal curvature-based region (PCBR) detector [10] described by SIFT descriptors are compared in their experiments. A base codebook is constructed using the $K$-means clustering algorithm with different weighted sampling techniques. A separate codebook for each detector is constructed with $K = 100$ and then concatenated to form a global codebook. The number of boosting iterations $T$ is set to 30. This straightaway increases the model's complexity by $T \times K$, making it difficult to cope with a large number of images and a large number of object categories. Evaluations are made on the Stonefly image dataset [24] containing 3,826 images of nine different species. An ensemble of 50 unpruned C4.5 decision trees [49] was employed in each boosting iteration.

### 5.3.3 Probabilistic Latent Model

– Fei-Fei and Perona [13] proposed a Bayesian hierarchical model that represents the distribution of codewords in each category of natural scenes as a mixture of aspects. Each aspect is defined by a multinomial distribution over the quantized local descriptors. Their method is modified on the latent Dirichlet allocation (LDA) model [4] by introducing a category variable for classification, which explicitly requires each image example to be
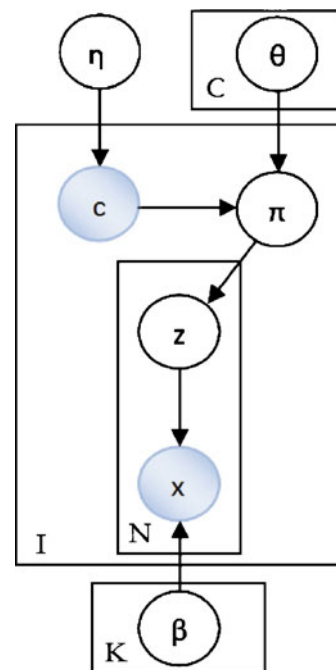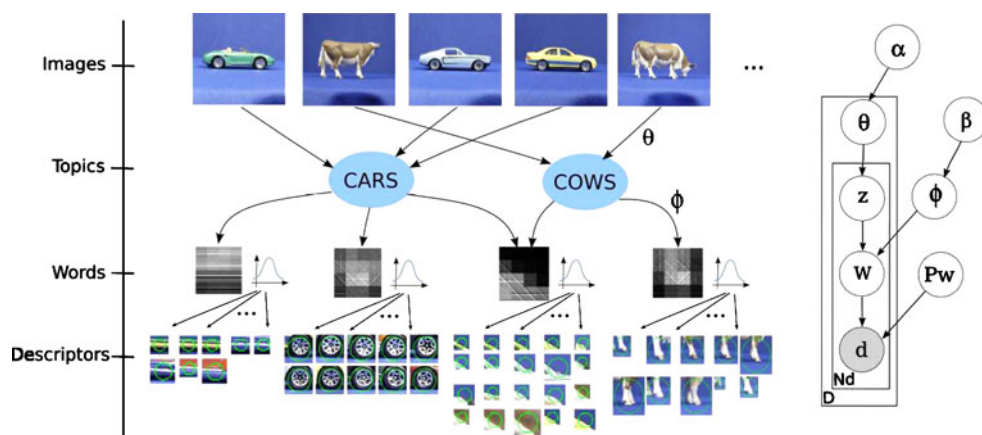


**Figure 5** The graphical representation of the modified latent Dirichlet allocation model. An image $I$ consists of $N$ patches denoted by **x**. The total number of object categories is $C$. $\eta$ is a $C$-dimensional vector of a multinomial distribution, and $\pi$ is the parameter of a multinomial distribution. $K$ is the total number of themes. $\theta$ is a parameter conditioned on the category $c$. **x** and **c** are observed variables.

**Figure 6** Overview of the latent mixture codebook model and the corresponding graphical model representation. This diagram adapted from [25] with permission.



labelled during the learning process. A graphical representation of the modified LDA is depicted in Fig. 5. The authors tested four different ways of extracting local regions: evenly sampled grid, random sampling, the Kadir and Brady Saliency detector [20] and difference of Gaussian (DoG) detector [33]. The patches are then described by two different methods: normalised $11 \times 11$ pixel gray values or SIFT. Features extracted from all training images of all categories were clustered by the $K$-means algorithm. Following the construction of the codebook, clusters with too small number of members were pruned out. The dataset they used for evaluation contained 13 categories of natural scenes with 3,859 images that were collected from a mixture of COREL images, Google image search engine and personal photographs. Based on their experimental results, they report that the SIFT representation is more robust than the pixel gray value representation. Furthermore, the evenly sampled grid-based SIFT approach out performs the random, saliency, and DoG based SIFT approaches by 4.5, 12.1 and 12.7%, respectively.

– Larlus and Jurie [25] proposed a generative model based on latent aspects that represent images at low-level feature descriptors. The construction of a visual codebook is achieved by an object model that embeds visual words as a component of the learning process. In their model, images are treated as distributions of topics, topics are considered as distributions of visual words, and visual words considered as Gaussian mixtures over SIFT descriptors. Figure 6 depicts the proposed model. This latent variable model of Larlus and Jurie [25] is a form of Gaussian-Multinomial latent Dirichlet allocation (GM-LDA). Topic distributions over words are sampled from a Dirichlet distribution.

Compared to the model in [13], GM-LDA has an extra layer responsible for the generation of visual descriptors conditional to visual words that allows for learning the visual codebook. The model parameters are estimated by an iterative technique called Gibbs sampling. Experiments were carried out on two datasets: a subset of the ETH-80 dataset [30] containing four object categories and the Bird dataset [26] containing six categories each with 100 images. Local descriptors were extracted on a dense grid at different scales and each patch was represented by SIFT descriptor. The experiments using the proposed model under different settings compare image categorisation based on the latent topics and visual features in a bag-of-features framework. Also the standard codebook model using $K$-means and the standard LDA model are compared with their model. The topic-based classification was compared with SVM classifiers and a Bayesian type classifier in which the authors note that both of the classifiers perform equally. The bag-of-features based classification employs SVM classifiers, from which the authors conclude that the GM-LDA is better than the $K$-means based method, and the bag-of-features approach compared to topic-based classification performs much better. As the proposed model has four parameters, its estimation is much more time consuming than a standard LDA or $K$-means clustering method.

– Quelhas et al. [48] have extended the work of Fei-Fei and Perona [13] for scene classification that integrates scale-invariant feature extraction and probabilistic latent semantic analysis (PLSA)-based clustering of images. Images are modelled as mixtures of aspects in an unsupervised way. The distribution over aspects serves as image representation that is inferred from new images and

then used for classification. The visual codebook was constructed by the $K$-means algorithm with a desired choice of $K$, typically $K = 1,000$. Following the construction of the codebook, the authors use the PLSA model to capture co-occurrence information between elements in the bag-of-features representation. The parameters of the PLSA model are estimated using the maximum likelihood principle. They compare different feature detectors: DoG, multi-scale Harris affine, multi-scale Harris, and a fixed $15 \times 20$ grid; three different descriptors: SIFT, complex filters, and a $11 \times 11$ pixel sample of the area defined by the detector were used in paired combinations. The main experiments were tested on two datasets, one used in [13] and the other on six natural scene classes containing a total of 700 images. The classification results were obtained by one-versus-all SVMs with Gaussian kernel. The authors' experimental results confirm that in practice DoG+SIFT constitutes a reasonable choice for image scene classification.

– Sudderth et al. [53] developed a family of hierarchical probabilistic models for object recognition in natural scenes. Visual objects are modelled as a set of parts with an expected appearance and position, in an object-centred coordinate frame. The authors started developing models for images with single objects, and models which share parts among related categories, and finally turned to multiple object scenes through the use of Dirichlet processes. They extracted interest regions from images using three different criteria: Harris affine invariant regions, Laplacian of Gaussian operator [34] and the maximally stable extremal regions (MSERs) [36] algorithms that were then described by SIFT descriptors. Edge-based features were also extracted using the Canny detector [7]. $K$-means clustering was used to construct a visual codebook of size 1,000, where the $K$ was set by cross-validation. Each of the three different feature types is then mapped to a disjointed set of visual words. An expanded codebook then jointly encodes the appearance and coarse shape of each feature. The parameters of the models are learnt via a Gibbs sampler which uses a graphical model to analytically average over many parameters. They evaluated the model on a collection of 16 categories containing seven animal faces, five animal profiles and four wheeled vehicles as object categories and also evaluated the model on a simple street scene containing three object categories (buildings, cars, and roads). Classification is undertaken using the likelihood ratio. The ap-

proach only works for images with roughly aligned objects, as in the Caltech 101 object database.

– Perronnin [43] characterised images using a set of category-specific histograms generated one per object category, where each histogram describes whether the content can be best modelled by a universal vocabulary or by its corresponding category-specific codebook. A universal codebook describes the visual content of all the considered categories that are trained with data from all classes under consideration and a codebook is represented by GMMs using maximum likelihood estimation. On the other hand, category-specific codebooks are obtained by adapting the universal codebook using the class training data and a form of Bayesian adaptation based on the maximum a posteriori (MAP) criterion. The maximum number of Gaussians in the universal codebook was set to 2,048. An image is then characterised by a set of histograms called bipartite as they can be split into two equal parts. Each part describes how well one codebook accounts for an image compared to the other codebook. Local patches were extracted from regular grids at five different scales. Each patch is then described by SIFT and colour features. PCA was applied to reduce the dimensionality of SIFT from 128 to 50, and the RGB colour channels from 96 to 50. Evaluations were performed on their own in-house database containing 19 classes of object categories and scenes, and the PASCAL VOC 2006 image dataset containing ten classes. Classification was performed using one-versus-all linear SVMs and a logistic regression with a Laplacian prior. However, in this method the memory requirement is high due to the storage of large number of Gaussians for each adapted codebook. Furthermore, in this approach, if two visual object classes are visually close, there is no guarantee that a distinctive visual word will be obtained. On the other hand, the process that generates bipartite histograms is computationally expensive.

## 6 Codebook-Free Model

Martínez-Muñoz et al. [35] recently proposed a framework that is free from the use of a codebook for categorising objects in images. The dictionary-free categorisation is achieved by learning an initial random forest of trees, followed by the construction of a second-level ('stacking') training set, and learning through a stacked classifier. Bootstrap samples of images are

drawn with replacements from the training set to create an initial random forest using a modified version of C4.5 [49]. A histogram of the training examples belonging to each class is stored at each leaf of the decision tree. The purpose of the second-level training set is to consider the images that were not used to build the initial tree. For each image, its descriptors are dropped through each tree and their histograms are concatenated to obtain the feature vector for the stacking example. The authors extracted several features with the use of different combinations of detectors and descriptors. A random forest is associated with each and every combination of the detector and descriptor. Experiments were carried out with the Stonefly-9 [24] image dataset containing 3,826 images of nine different species, and the PASCAL VOC Challenge 2006 image dataset containing ten classes. For the PASCAL06 image set, interest points in each image were detected using Harris, Hessian and PCBR detector [10] and regularly sampled image patches. These interest points are then independently described by three different descriptors: SIFT, Colour SIFT [56], and the filter-bank descriptor employed by Winn et al. [59]. For the Stonefly-9 dataset, interest points were found using Hessian, Kadir and Brady salient region, and PCBR detectors; each of them was then described using SIFT descriptors. Edges were extracted using the Canny edge detector. The classifier is a boosted decision tree. Although they claim that the proposed method is simple and elegant, they were unable to grow any single tree on all the extracted descriptors as they drew a random sample of the descriptors. Despite this, they have to determine the minimum number of training examples in each leaf node, the minimum number of trees in each random forest, and the number of boosting iterations for the stacked classifier.

## 7 Summary

This paper provides a review of the literature on the codebook model-based approach to visual object recognition. The approach, while ignoring any structural aspect in vision, nonetheless provides state-of-the-art performances on current datasets. This is impressive because we are simply modelling the statistical distributions of low-level image features. As in any review, the coverage here is not exhaustive in visual object recognition. However, as our focus in this paper is on the design of the codebook, we have attempted to provide an exhaustive coverage of the different codebook design strategies different authors have adopted. A summary of those several approaches that have been proposed over the last decade with their use of different feature detectors, descriptors, codebook construction schemes, choice of classifiers in recognising objects is depicted in Figs. 7 and 8.

Many models have been proposed using conventional methods such as $K$-means, balanced clustering such as agglomerative or mean-shift, and even more complex models such as the latent Dirichlet allocation (LDA), probabilistic latent semantic analysis (PLSA), Gaussian mixture models (GMMs), or random forests. The vast majority of methods in the literature relating to the construction of codebooks are either $K$-means or GMMs, in which the obtained cluster centres are those that have high probability density. These codewords are not necessarily the most discriminative. GMM has better representative power than a single cluster. However, it requires more computational power.

In addition, recent studies have started to explore the construction of visual codebook leading to an improved categorisation performance in terms of discriminative power, compactness, and inclusion of spatial information. Winn et al. [59], Kim and Kweon [23], Wang et al. [58], Li et al. [32] and Yang et al. [62] have focused on both compactness and discriminative power of visual codebooks, whereas Larlus and Jurie [25], Lazebnik et al. [28] and Moosmann et al. [41] have focused on incorporating spatial information in the codebook model. Moreover, Grauman and Darrell [14], Nister and Stewenius [42], Agarwal and Triggs [2], Wang [57] and Zhang et al. [64] have focused on constructing multi-resolution codebooks.

The practical difficulty and bottleneck in a bag-of-keypoints approach is the construction of a visual vocabulary with huge number of keypoints using the traditional clustering methods such as the K-means algorithm. It is worth noting that in their work, Sivic and Zisserman [52] used $200,000 \times \mathbb{R}^{128}$ features which represent about 10% of the original dataset that was clustered into 10,000 and 6,000 clusters for each type of detector used in constructing a codebook. Winn et al. [59] used a subset of the PASCAL VOC Challenge 2005 dataset (587 images) to construct an initial codebook of size 1,200. Furthermore, Moosmann et al. [41] in their experiments with the PASCAL VOC Challenge 2005 dataset, used 50,000 patches in total over the 648 images (73 patches per image) to construct a codebook of size 30,000. These examples show the major bottleneck occurs in handling the massive scale of the datasets and the patch-based descriptors in constructing a visual codebook. While several approaches were explored, there has been very little attempt at a large scale clustering of patch-based descriptors. The methods that we reviewed in this paper are mostly

**Figure 7** Summary of codebook models in the literature that employed several combinations of image patch **a** detectors and **b** descriptors for visual object recognition.
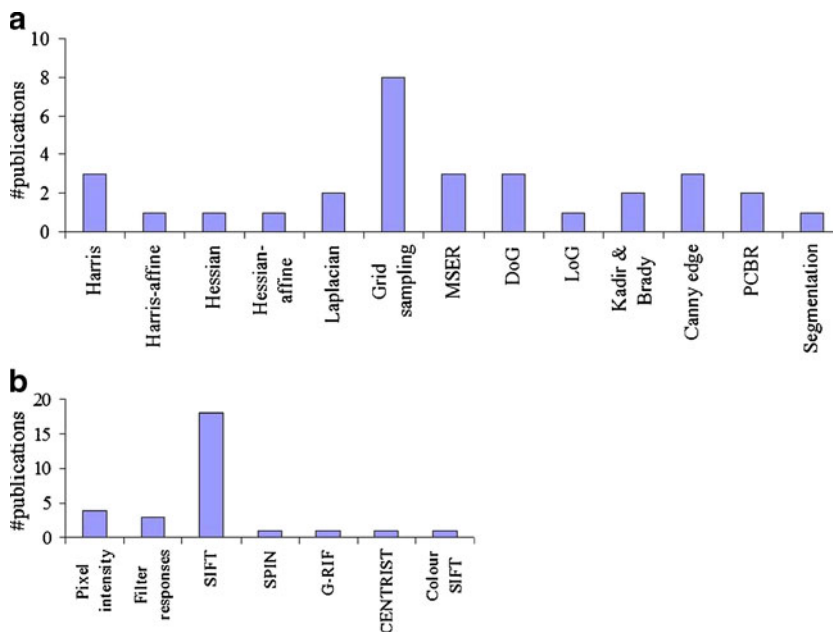


**Figure 8** Summary of codebook models in the literature that employed **a** different clustering techniques and **b** various classifiers in patch-based visual object recognition. It can be noticed that majority of the visual codebook construction involves *K*-means or its combination with hierarchical clustering. Another popular approach is the use of Gaussian Mixture Models (GMMs) in constructing a codebook. Both of these techniques constructs a codebook in such a way that the obtained cluster centres are those that have high probability density. In the classification step, the choice of SVMs are quite straightforward as they are naturally designed to perform classification in high dimensional spaces.
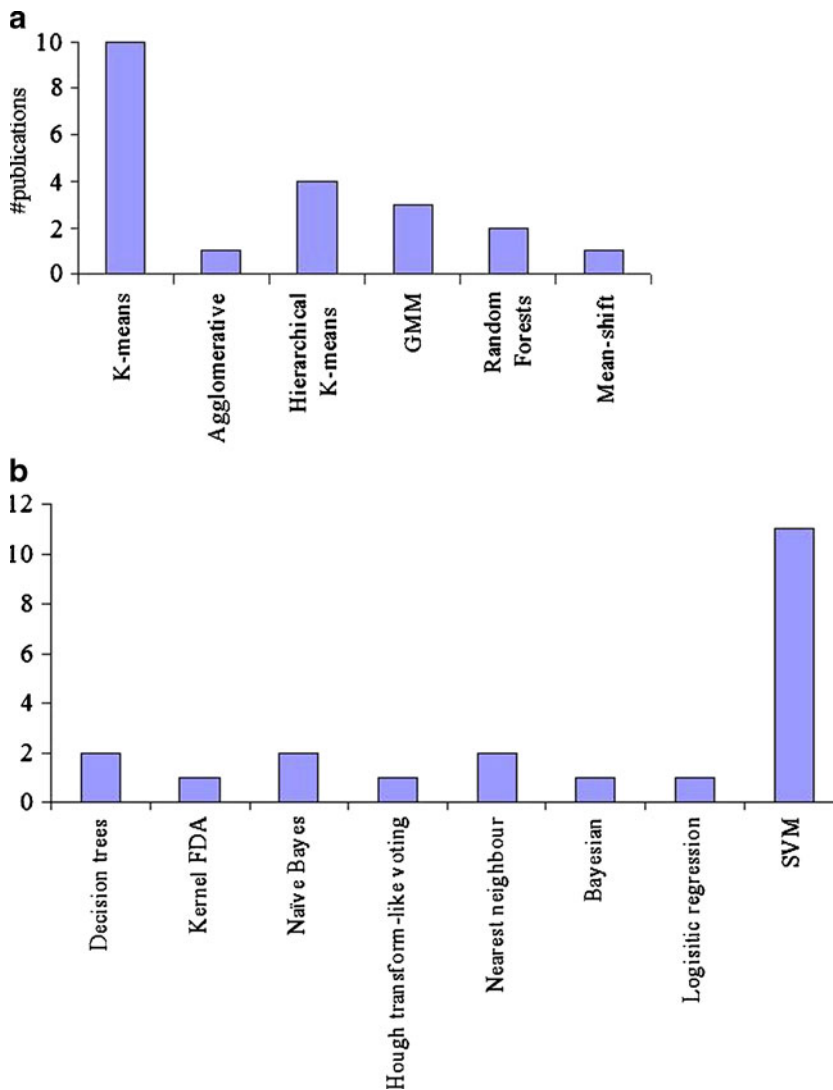
applied to modest size problems. Recently, many traditional approaches have gained favour as machines have become fast enough to make them practical in constructing a codebook on relatively large scale descriptors. As the size of training sets is increasing, the size of codebook and complexity of construction will increase. Future research should focus on designing more effective online clustering approaches in constructing highly discriminative and compact histograms rather than focusing on methods tuned to achieve high performance in classification. Our own contribution, the resource-allocating clustering approach is fundamentally different from traditional approaches in that it is not the density of detected patches one needs to retain in the codebook but the coverage across the feature space.

## References

1. Agarwal, S., Awan, A., & Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*, 1475–1490.

2. Agarwal, A., & Triggs, B. (2006). Hyperfeatures—Multilevel local coding for visual recognition. In *Proceedings of the European conference on computer vision (ECCV'06)* (pp. 30–43). Berlin: Springer.

3. Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). SURF: Speeded up robust features. *Computer Vision and Image Understanding, 110*, 346–359.

4. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*(3), 993–1022.

5. Brank, J., Grobelnik, M., Milic-Frayling, N., & Mladenic, D. (2002). Interaction of feature selection methods and linear classification models. In *Proceedings of the ICML-02 workshop on text learning*, forthcoming.

6. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

7. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8*, 544–567.

8. Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 603–619.

9. Csurka, G., Dance, C. R., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV'04* (pp. 1–22).

10. Deng, H., Zhang, W., Mortensen, E., Dietterich, T., & Shapiro, L. (2007). Principal curvature-based region detector for object recognition. In *Proceedings of the conference on computer vision and pattern recognition (CVPR'07)* (pp. 1–8).

11. Dorko, G., Schmid, C., & Lear, P. (2005). Object class recognition using discriminative local features. Technical report, RR-5497, INRIA-Rhone-Alpes.

12. Farquhar, J. D. R., Szedmak, S., Meng, H., & Shawe-Taylor, J. (2005). Improving "bag-of-keypoints" image categorisation: Generative models and PDF-kernels. In *LAVA report*. U.K.: University of Southampton.

13. Fei-Fei, L., & Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'05)* (Vol. 2, pp. 524–531).

14. Grauman, K., & Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of the tenth IEEE international conference on computer vision (ICCV'05)* (Vol. 2, pp. 458–1465).

15. Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of Alvey vision conference* (pp. 147–151).

16. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*, 264–323.

17. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European conference on machine learning (ECML'98)* (pp. 137–142).

18. Juan, L., & Gwun, O. (2009). A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing, 3*, 143–152.

19. Jurie, F., & Triggs, B. (2005). Creating efficient codebooks for visual recognition. In *Proceedings of the tenth IEEE international conference on computer vision (ICCV'05)* (Vol. 1, pp. 604–610).

20. Kadir, T., & Brady, M. (2001). Saliency, scale and image description. *International Journal of Computer Vision, 45*(2), 83–105.

21. Ke, Y., & Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the conference on computer vision and pattern recognition (CVPR'04)* (pp. 511–517).

22. Kim, S., & Kweon, I.-S. (2006). Biologically motivated perceptual feature: Generalized robust invariant feature. In *Proceedings of the Asian conference on computer vision (ACCV'06)* (pp. 305–314).

23. Kim, S., & Kweon, I.-S. (2007). Object categorization robust to surface markings using entropy-guided codebook. In *IEEE workshop on applications of computer vision* (p. 22).

24. Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., et al. (2007). Automated insect identification through concatenated histograms of local appearance features. *Machine Vision and Applications, 19*(2), 105–123.

25. Larlus, D., & Jurie, F. (2006). Latent mixture vocabularies for object categorization. In *Proceedings of the British machine vision conference (BMVC'06)* (pp. 959–968).

26. Lazebnik, S., Schmid, C., & Ponce, J. (2005). A maximum entropy framework for part-based texture and object recognition. In *Proceedings of the IEEE international conference on computer vision (ICCV'05)* (Vol. 1, pp. 832–838).

27. Lazebnik, S., Schmid, C., & Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transac-

*tions on Pattern Analysis and Machine Intelligence, 27*, 1265–1278.

28. Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE conference on computer vision and pattern recognition (CVPR'06)* (pp. 2169–2178).

29. Leibe, B., Leonardis, A., & Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision, 77*(1–3), 259–289.

30. Leibe, B., & Schiele, B. (2003). Interleaved object categorization and segmentation. In *Proceedings of the British machine vision conference (BMVC'03)* (pp. 759–768).

31. Li, X., Chen, S.-C., Shyu, M.-L., Li, S.-T., & Furht, B. (2002). A novel hierarchical approach to image retrieval using color and spatial information. In *Proceedings of the third IEEE Pacific rim conference on multimedia: PCM '02* (pp. 175–182). London: Springer.

32. Li, T., Mei, T., & Kweon, I.-S. (2008). Learning optimal compact codebook for efficient object categorization. In *IEEE workshop on applications of computer vision* (pp. 1–6).

33. Lowe, D. (1999). Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on computer vision (ICCV'99)* (Vol. 2, pp. 1150–1157).

34. Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*, 91–110.

35. Martínez-Muñoz, G., Zhang, W., Payet, N., Todorovic, S., Larios, N., Yamamuro, A., et al. (2009). Dictionary-free categorization of very similar objects via stacked evidence trees. In *Proceedings of the IEEE conference in computer vision and pattern recognition (CVPR'09)* (pp. 549–556).

36. Matas, J., Chum, O., Urban, M., & Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *British machine vision conference (BMVC'02)* (Vol. 1, pp. 384–393).

37. Mikolajczyk, K., Leibe, B., & Schiele, B. (2006). Multiple object class detection with a generative model. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'06)* (Vol. 1, pp. 26–36).

38. Mikolajczyk, K., & Schmid, C. (2001). Indexing based on scale invariant interest points. In *Proceedings of the 8th IEEE international conference on computer vision (ICCV'01)* (pp. 525–531).

39. Mikolajczyk, K., & Schmid, C. (2002). An affine invariant interest point detector. In *Proceedings of the European conference on computer vision (ECCV'02)* (pp. 128–142). Berlin: Springer.

40. Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*, 1615–1630.

41. Moosmann, F., Triggs, B., & Jurie, F. (2007). Fast discriminative visual codebooks using randomized clustering forests. In *Neural information processing systems (NIPS'07)* (pp. 985–992).

42. Nister, D., & Stewenius, H. (2006). Scalable recognition with a vocabulary tree. *IEEE Computer Society Conference on Computer, 2*, 2161–2168.

43. Perronnin, F. (2008). Universal and adapted vocabularies for generic visual categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*, 1243–1256.

44. Peterson, G., & Barney, H. (1952). Control methods used in a study of the vowels. *Journal of the Acoustical Society of America, 24*, 175–184.

45. Platt, J. C. (1991). A resource-allocating network for function interpolation. *Neural Computation, 3*, 213–225.

46. Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers* (pp. 61–74). Cambridge: MIT Press.

47. Platt, J. C., Cristianini, N., & Shawe-Taylor, J. (2000). Large margin DAGs for multiclass classification. In *Advances in neural information processing systems (NIPS'00)* (Vol. 12, pp. 547–553).

48. Quelhas, P., Monay, F., Odobez, J.-M., Gatica-Perez, D., & Tuytelaars, T. (2007). A thousand words in a scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(9), 1575–1589.

49. Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo: Morgan Kaufmann.

50. Ramanan, A., & Niranjan, M. (2010). A one-pass resource-allocating codebook for patch-based visual object recognition. In *IEEE international workshop on machine learning for signal processing (MLSP'10)* (pp. 35–40).

51. Rubner, Y., Tomasi, C., & Guibas, J. (1998). A metric for distributions with applications to image databases. In *Proceedings of the IEEE international conference on computer vision (ICCV'98)*.

52. Sivic, J., & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Proceedings of the ninth IEEE international conference on computer vision (ICCV'03)* (pp. 1470–1478).

53. Sudderth, E. B., Torralba, A., Freeman, W. T., & Willsky, A. S. (2008). Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision, 77*(1–3), 291–330.

54. Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *The 37th annual allerton conference on communication, control, and computing* (pp. 368–377).

55. Tuytelaars, T., & Mikolajczyk, K. (2008). *Local invariant feature detectors: A survey*. Now Publishers Inc.

56. van de Sande, K. E. A., Gevers, T., & Snoek, C. G. M. (2008). Evaluation of color descriptors for object and scene recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'08)*.

57. Wang, L. (2007). Toward a discriminative codebook: Codeword selection across multi-resolution. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR'07)* (pp. 1–8).

58. Wang, L., Zhou, L., & Shen, C. (2008). A fast algorithm for creating a compact and discriminative visual codebook. In *Proceedings of the tenth European conference on computer vision (ECCV'08)* (pp. 719–732). LNCS 5305.

59. Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In *Proceedings of the tenth IEEE international conference on computer vision (ICCV'05)* (Vol. 2, pp. 1800–1807).

60. Wu, J., & Rehg, J. (2009). Beyond the Euclidean dstance: Creating effective visual codebooks using the histogram intersection kernel. In *Proceedings of the IEEE international conference on computer vision (ICCV'09)*.

61. Wu, J., & Rehg, J. (2009). CENTRIST: A visual descriptor for scene categorization. Technical report, GIT-GVU-09-05, GVU Center, Georgia Institute of Technology.

62. Yang, L., Jin, R., Sukthankar, R., & Jurie, F. (2008). Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'08)* (pp. 1–8).

63. Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision, 73*, 213–238.
64. Zhang, W., Surve, A., Fern, X., & Dietterich, T. (2009). Learning non-redundant codebooks for classifying complex objects. In *Proceedings of the 26th international conference on machine learning (ICML'09)*.



**Amirthalingam Ramanan** is a Senior Lecturer at the Department of Computer Science at the University of Jaffna. He received his BSc Honours in Computer Science (2002) from the University of Jaffna, Sri Lanka and his PhD from the University of Southampton, United Kingdom (2010). His research interests include visual object recognition, cluster analysis in large-scale benchmark evaluations.



**Mahesan Niranjan** is Professor of Electronics and Computer Science at the University of Southampton, where he was head of the Information: Signals, Images and Systems (ISIS) research group. Prior to this appointment in February 2008, he has held a professorship in the University of Sheffield (1999–2008) and a lectureship in the University of Cambridge (1990–1998). At Sheffield he has served as Head of Computer Science (2002–2004) and Dean of the Faculty of Engineering (2006–2008). He received his BSc from the University of Peradeniya, Sri Lanka (1982), MEE from Eindhoven, The Netherlands (1985), both in Electronics Engineering, and his PhD from the University of Cambridge (1990). His research interests are in the algorithmic and applied aspects of Machine Learning, and he has authored or co-authored about 100 papers in peer reviewed journals and conferences. He has been Program Chair of several international workshops and has acted as a co-organizer of a six month program on Neural Networks and Machine Learning at the Isaac Newton Institute for Mathematical Sciences, Cambridge.