Comparing compact codebooks for visual categorization

Jan C. van Gemert *, Cees G.M. Snoek, Cor J. Veenman, Arnold W.M. Smeulders, Jan-Mark Geusebroek

Intelligent Systems Lab Amsterdam, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

ABSTRACT

In the face of current large-scale video libraries, the practical applicability of content-based indexing algorithms is constrained by their efficiency. This paper strives for efficient large-scale video indexing by comparing various visual-based concept categorization techniques. In visual categorization, the popular codebook model has shown excellent categorization performance. The codebook model represents continuous visual features by discrete prototypes predefined in a vocabulary. The vocabulary size has a major impact on categorization efficiency, where a more compact vocabulary is more efficient. However, smaller vocabularies typically score lower on classification performance than larger vocabularies. This paper compares four approaches to achieve a compact codebook vocabulary while retaining categorization performance. For these four methods, we investigate the trade-off between codebook compactness and categorization performance. We evaluate the methods on more than 200 h of challenging video data with as many as 101 semantic concepts. The results allow us to create a taxonomy of the four methods based on their efficiency and categorization performance.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Today, digital video is ubiquitous. This omnipresence of digital video material spurs research in automatic content-based indexing. However, given the sheer quantity of available digital video, the applicability and quality of current video indexing algorithms severly depends on their efficiency [12,35]. One approach to achieve efficiency is by means of a compact, yet powerful representation of the visual data. To this end, this paper compares various methods which obtain compact and expressive models for video indexing.

As an instantiation of video indexing, we focus on automatic concept categorization [18,28,38,39,49]. Applications are mainly found in content-based retrieval and browsing. The goal of concept categorization is to rank shots according to their relevance to a set of predetermined semantic concepts. Some examples of these concepts are *airplane, beach, explosion, George Bush, people walking, etc.*

Many visual concepts are captured as a typical contextual arrangement of objects [2,15,20,27,30,42]. For example, consider an image of a beach, a city skyline, or a conference meeting. Such concepts are portrayed by a composition of the image as a whole, rather than characterized by one specific part in the image. Moreover, the background context of an object may provide considerable recognition cues. Consider Fig. 1 where an object is cut out of its surroundings. Without the background information, recognition becomes ambiguous even for humans. Alternatively, in Fig. 2a, a white patch is placed over the object, where the identity of a hidden object may be derived with high accuracy from the context and nothing but the context. Hence, the background context of an object can be more informative than the object itself. Therefore, in this paper we model the whole image for concept categorization, purposely including the context provided by the background.

We describe visual concepts in context with the codebook, or bag-of-visual-words, model. The codebook model is inspired by a word-document representation as used in text retrieval [34]. An schematic of the codebook model is given in Fig. 3. The codebook model treats an image as a distribution of local features, where each feature is labeled as a discrete visual prototype. These prototypes, or codewords, are defined beforehand in a given vocabulary, which may be obtained by unsupervised clustering [4,7,17,21,31, 33,36,41], or manual, supervised annotation [5,24,45,48]. Given a vocabulary, the codebook model allows visual categorization by representing an image by a histogram of codeword counts. The codebook model yields a distribution over codewords that models the whole image, making this model well-suited for describing context. This paper strives towards efficient concept categorization by investigating qualitative and compact codebooks.

1.1. Contribution

In this paper, we experimentally evaluate various codebook methods to obtain a small, compact, vocabulary that discriminates well between classes. The size of the vocabulary is linked to the discriminative power of the model. A too small vocabulary does not discriminate well between concept categories [47]. Hence, current state-of-the-art methods typically use several thousands of code-

---

* Corresponding author.
  E-mail address: J.C.vanGemert@gmail.com (J.C. van Gemert).

Fig. 1. Example of an object that is ambiguous without context.

words [44,22]. In a practical application, however, it may not be feasible to use such large number of codewords. Practical objections to a large vocabulary are its storage requirements, working memory usage, and the computation time to train a classifier. Moreover, it has recently been shown that a too large vocabulary severely deteriorates the performance of the codebook model [47]. Therefore, we selected four state-of-the-art methods that each individually focus on improving performance and evaluate these algorithms under a compactness constraint. The compactness constraint is typically ignored by systems who focus solely on performance. The four compacting methods consist of (1) global vocabulary clustering; (2) concept-specific vocabulary clustering; (3) annotating a semantic vocabulary; and (4) soft-assignment of image features to codewords. Methods 1–3 deal with vocabulary building, where method 2 is a variant of method 1. Method 4 is a generic approach to increase the expressive power of the codebook vocabulary. We evaluate each of these methods against each other, on a large shared dataset over two different feature types, and two different classifiers.

This paper is organized as follows. In the next section we give an overview of the related literature. In Section 3 we describe the four evaluated methods. We present our experimental setup in Section 4, whereas we highlight the results in Section 5. Section 6 concludes the paper.

2. Related work

Several techniques exist for efficiently retrieving high-dimensional image features in large image collections. Nistér and Stewénius [29] use hierarchical *k*-means clustering to quantize local image features in a vocabulary tree. This vocabulary tree demonstrates efficient feature retrieval in as many as 1 million images. A tree structure is also used by [23] who obtains efficiency gains by reducing the dimensionality of the features by a truncated Mahalanobis metric. Moreover, novel quantization method based on randomized trees is used by [32]. In contrast to a tree structure, Grauman and Darrell [11] present an approximate hashing scheme based on pyramid matching. The pyramid matching allows multi-resolution image matching while the hashing technique allows sub-linear retrieval in large collections of features. Hashing is also used by Kise et al. [19] who show that a simple binary representation of feature vectors can result in an efficient approximate nearest neighbor algorithm. Tree and hashing algorithms are well-suited for assigning features to extremely large vocabularies, with millions of centroids. These algorithms, however, do not consider categorization. They focus on recognition of (close to) exact image and feature matches. For categorization with the codebook model, a vocabulary of a million codewords is no longer practical when training a classifier, and a tree-structure does not help out there. The classifier is still left with storing a feature vector of a million codewords for each image. Therefore, we focus on compact vocabularies for efficiency.

A compact codebook model can be achieved by modeling codeword co-occurrence. Under the assumption that frequent

Fig. 2. Example showing the influence of context. (a) The surroundings of an object and (b) the whole image. Note that the category of the hidden object in (a) can easily be inferred from the context.
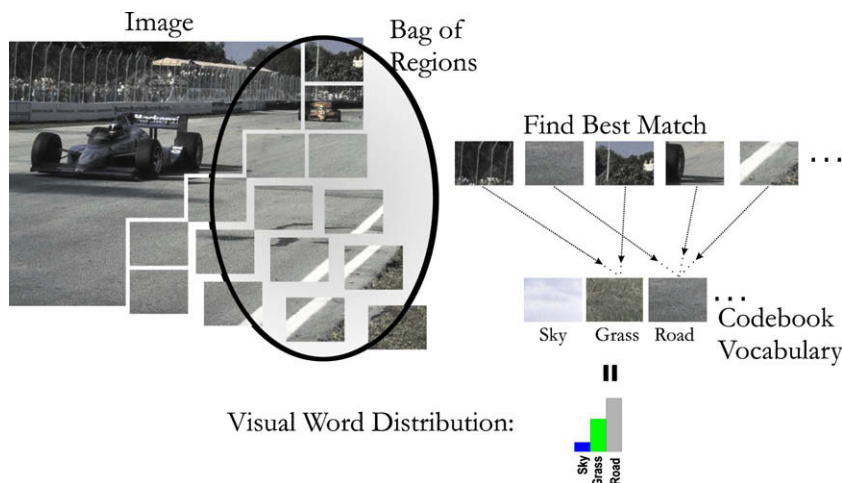

Fig. 3. An example of the visual word, or codebook model. An image is represented as a bag-of-regions where each region is represented by the best fitting codeword in the vocabulary. The distribution of the codeword-counts yields the image model.

co-occurring codewords describe similar information, the vocabulary size may be reduced by merging these codewords. Codeword co-occurrence is typically modeled by a generative probabilistic model [3,14]. To this end, Fei-Fei and Perona [7] introduce a Bayesian hierarchical model for scene categorization. Their goal is a generative model that best represents the distribution of codewords in each concept category. They improve on latent Dirichlet allocation [3] by introducing a category variable for classification. The proposed algorithm is tested on a dataset of 13 natural concept categories where it outperforms the traditional codebook model by nearly 30%. The work by Fei-Fei and Perona is extended by Quelhas et al. [33], who investigate the influence of training data size. Moreover, Bosch et al. [4] show that probabilistic latent semantic analysis improves upon latent dirichlet allocation. Further contributions using co-occurrence codebook models are by [41]. Typically, a generative model is built on top of a codebook model. Hence, the techniques proposed in this paper can easily be extended with co-occurrence modeling. The extra modeling step requires ample additional processing which is less practical for large datasets. Moreover, an additional step makes it harder to evaluate which part of our algorithm is responsible for what. Therefore, in this paper, we focus on compact codebook models, without introducing additional co-occurrence modeling steps.

Apart from co-occurrence modeling, a compact codebook may be achieved directly by reducing the vocabulary size or by carefully selecting the vocabulary elements. Such a careful selection can be achieved with a semantic vocabulary [5,45,24,48] that describes an image in meaningful codewords. A semantic vocabulary can be constructed by manually selecting image patches with meaningful labels, for example *sky, water* or *vegetation*. The idea of meaningful codewords, is that they allow a compact, discriminative, and semantic image representation. In contrast to annotating a vocabulary, Jurie and Triggs [17] compare clustering techniques to obtain a vocabulary. Specifically, they show that radius-based clustering outperforms the popular *k*-means clustering algorithm. Furthermore, Winn et al. [50] concentrate on a global codebook vocabulary, whereas Perronnin et al. [31] focus on concept-specific vocabularies.

In this paper, we concentrate on compact vocabulary construction while trying to retain the ability to discriminate well between concept categories. Note that this is more general than vocabularies that are built by a discriminative criterion [25]. Such methods assume that the discriminative ability of a single feature carries over to the whole vocabulary. Hence, a vocabulary created by discriminative criteria of single features also aims at a final vocabulary which is discriminative between concept categories.

Instead of reducing the size of a vocabulary, the expressive power of the vocabulary may be increased. With higher expressive power, a vocabulary needs less codewords to obtain similar performance which in turn leads to a more compact vocabulary. The expressive power can be increased by disposing of the hard-assign-ment of a single codeword to a single image features. Instead of using hard-assignment, some weight may be given to related codewords. To this end, Tuytelaars and Schmid [43] and Jiang et al. [16] assign weights to neighboring visual words. Whereas a visual word weighting scheme based on feature similarity is used in Agarwal and Triggs [1] and in our previous work [45,47]. This soft-assignment increases the expressiveness of a vocabulary. We will test the influence of soft-assignment on vocabulary compactness. In the next section we will present the details of the method.

3. Compact codebook models

In the codebook model, the vocabulary plays a central role. The expressive power of the vocabulary determines the quality of the model, whereas the size of the vocabulary controls the complexity of the model. Therefore, vocabulary construction directly influences model complexity. We identify two methods for constructing a vocabulary: a data-driven approach characterized by unsupervised clustering and a semantic approach which relies on annotation. Besides the construction of the vocabulary, the expressive power may be increased. To this end, we consider replacing the hard-assignment of codewords to image features with soft-assignment. This soft-assignment aims for a more powerful vocabulary, which in turn leads to a more compact model.

3.1. Codebook compactness by a clustered vocabulary

A codebook vocabulary consists of discrete visual codewords, which are described by high-dimensional features. In order to obtain discrete codewords, the continuous high-dimensional feature space needs to be discretized. A common approach to discretizing a continuous feature space is by uniform histogram binning. However, in a high-dimensional feature space a histogram with a fixed bin size for each dimension will create an exponentially large number of bins. Moreover, since feature spaces are rarely uniformly distributed, many of these bins will be empty [43]. We illustrate the partitioning of a continuous feature space with a uniform histogram in Fig. 4a.

An alternative to a uniform partitioning of the high-dimensional feature space is unsupervised clustering. The benefit of using clusters as codewords is a small vocabulary size without empty bins. A popular clustering approach for finding codewords is *k*-means [4,7,17,21,31,33,41]. *k*-Means is an unsupervised clustering algorithm that tries to minimize the variance between *k* clusters and the training data, where *k* is a parameter of the algorithm. The advantages of *k*-means are its simple and efficient implementation. However, the disadvantage of *k*-means is that the algorithm is variance-based. Thus, the algorithm will award more clusters to high-frequency areas of the feature space, leaving less clusters for the remaining areas. Since frequently occurring features are not necessarily informative, this over-sampling of dense regions is
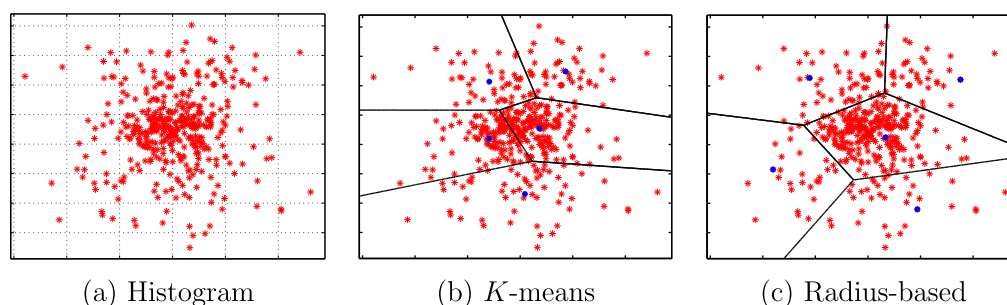

| (a) Histogram | (b) *K*-means | (c) Radius-based |

Fig. 4. Three examples of continuous space partitioning, using (a) a uniform histogram, (b) *k*-means clustering, and (c) radius-based clustering. Note the empty bins in the histogram, the cluster centers in densely populated areas of *k*-means, and the uniform partitioning of radius-based clustering.

inappropriate. For example, in analogy of text retrieval, the most frequent occurring words in English are the so called function words like *the*, *a*, and *it*, despite their high frequency these function words convey little information about the content of a document. Therefore a codebook vocabulary based on variance-based clustering may not be as expressive as it could be.

In contrast to variance-based clustering, Jurie and Triggs [17] argue that the codewords for a codebook vocabulary are better represented by radius-based clustering. Radius-based clustering assigns all features within a fixed radius of similarity *r* to one cluster, where *r* is a parameter of the algorithm. This radius denotes the maximum threshold between features that may be considered similar. As such, the radius determines whether two patches describe the same codeword. Hence, the influence of the radius parameter *r* on the codebook model is clear where the number of clusters, *k*, in *k*-means clustering is typically chosen arbitrary. The difference between radius-based clustering and *k*-means is illustrated in Fig. 4b and c. Note that the codewords found by *k*-means populate the densest part of the feature space, whereas the radius-based method finds codewords that each represent a distinct part of the feature space. Hence, radius-based clustering results in a non-empty, uniform sampling of a continuous feature space. Therefore, we will adopt radius-based clustering for data-driven codebook vocabulary creation.

3.1.1. Concept-specific vocabulary

A vocabulary formed by unsupervised clustering offers us the opportunity to construct a different, tuned, vocabulary for each concept [21,31]. This tuning endows each concept with its own unique vocabulary. For example, it might be beneficial to model the concept *boat* with a different vocabulary than the concept *office*, since scenes with a boat will contain water and sky, whereas office scenes hold tables and chairs. The idea behind concept-specific vocabularies is to obtain a reduced vocabulary, while retaining expressive power. We will experimentally compare the compactness and expressiveness of the concept-specific vocabularies against a global vocabulary obtained by clustering the whole feature space.

3.2. Codebook compactness by a semantic vocabulary

Whereas the previous section described a clustering approach for obtaining a codebook vocabulary, this section will focus on a semantic vocabulary. The use of semantic codewords builds on the principle of compositionality, stating that the meaning of an image can be derived from the meaning of the constituent parts of the image [5,24,45,48]. For example, an *outdoor* image is likely to contain *vegetation, water*, or *sky*. A semantic vocabulary consists of meaningful codewords. Therefore, the creation of the vocabulary requires a human annotator. This annotation step typically consists of drawing bounding boxes around a meaningful patch of pixels [45,48]. The rationale behind meaningful codewords is that local image semantics will propagate to the global codebook image model, leading to compact visual models.

Both the semantic vocabulary and the clustered vocabulary have specific advantages and disadvantages. The semantic vocabulary approach is based on manual selection of visually meaningful codewords. However, this approach has the underlying assumption that images can be decomposed in these semantic codewords, which may not hold for all images. For example, an *indoor* image is unlikely to contain any *sky* or *buildings*. In contrast to semantic labeling, clustering uses statistics to determine descriptive codewords. However, these codewords lack any meaningful interpretation. Such an interpretation may be important since humans typically decompose complex scenes into meaningful elements. Both approaches of acquiring a vocabulary of low-level descriptors

have their merits. We will experimentally compare both methods to determine their compactness and expressiveness.

3.3. Codebook compactness by soft-assignment

In order to take the continuous nature of image patches into account, we have proposed [45] to base the codebook model on a degree of similarity between patches. Similarity between patches is a more suitable representation than assigning only one visual word to an image patch. Labeling an image patch with the single best visual word ignores all ambiguity regarding the meaning of the image patch. In contrast, assigning a degree of similarity to an image patch will model the inherent uncertainty of the image patch. For example, instead of labeling a blue pixel patch as *sky*, the patch is better represented by saying that its similarity to *sky* is 0.9, and its similarity to *water* is 0.8. By using soft-assignment to model the uncertainty of the meaning of an image patch, we foresee improved expressive and discriminative power while maintaining a constant vocabulary size [45]. To evaluate this claim we will test soft-assignment versus hard-assignment as used in the traditional codebook model. If this claim is sound, the vocabulary size may be reduced, which in turn yields a more compact codebook.

Soft-assignment is easily incorporated in the codebook model. For each codeword, or bin, *b* in the vocabulary *V* the traditional codebook model constructs the distribution of codewords over an image by

$$H(b) = \sum_{r \in R(im)} \begin{cases} 1 & \text{if } b = \arg \max_{v \in V}(S(v,r)), \\ 0 & \text{otherwise}. \end{cases} \tag{1}$$

Here, $R(im)$ denotes the set of regions in image *im*, and $S(v,r)$ is the similarity between a codeword *v* and region *r*. The similarity $S(b,r)$ is specific to the type of image features that are used. The similarities are given with the image features in Appendix A. The similarities allow replacing hard-assignment with soft-assignment by

$$H(b) = \sum_{r \in R(im)} S(b,r). \tag{2}$$

This soft-assignment weights each codeword according to the similarity of an image region to this codeword. Fig. 5 illustrates this advantage.

4. Experimental setup

The experiments focus on the relation between codebook compactness and codebook quality. Codebook compactness is given by the size of the vocabulary, whereas codebook quality is measured by its categorization performance. To reduce dependency on a single visual feature, we show results over two visual features (Wiccest features and Gabor features, see Appendix A). Furthermore, we investigate the effect of the linear and light-weight Fisher classifier against a computationally more intensive non-linear SVM classifier. We identify three experiments:

- *Experiment 1:* Soft-assignment versus hard-assignment.
- *Experiment 2:* Semantic vocabulary versus globally-clustered vocabulary.
- *Experiment 3:* Semantic vocabulary versus concept-specific clustered vocabulary.

The experiments are conduced on a large video dataset where each shot is annotated if a concept is present. This fixed ground-truth allows repeatable experiments.
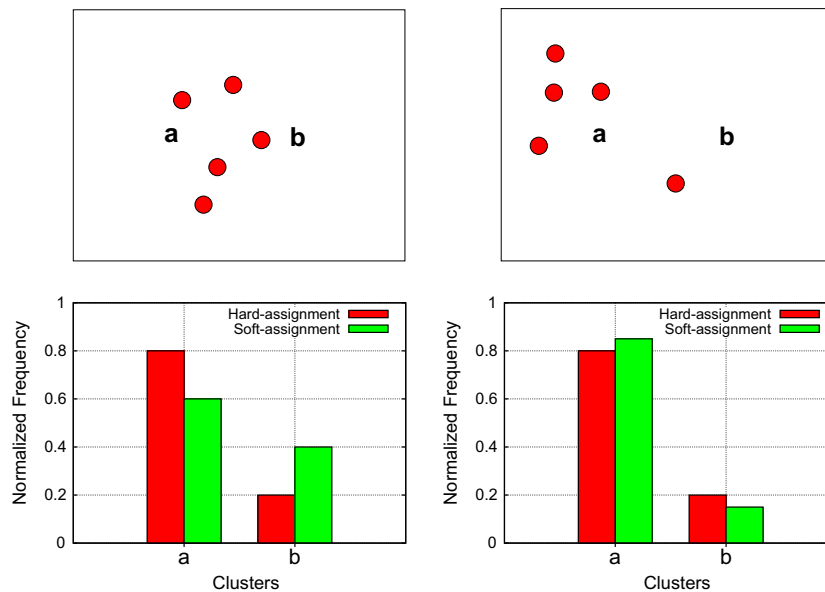
Fig. 5. Two examples indicating the difference between hard-assignment and soft-assignment of codewords to image features. The first row shows two images with each five samples (dots) around two codewords 'a' and 'b'. The second row displays the normalized occurence histograms of hard-assignment and soft-assignment for both images. Note that hard-assignment is identical for both examples, whereas soft-assignment is sensitive to the position of the samples.

4.1. Video datasets

The experiments are evaluated on the TREVID 2005 development set [37]. This video set contains nearly 85 h of English, Chinese and Arabic news video. In addition to the video data, we use the standard ground truth provided by the MediaMill Challenge [40]. This ground truth defines 101 semantic concepts with shot labels for each category, where the video data is split in 70% for training, and the remaining 30% for testing. In total there are 43,907 shots, where 30,630 are in the training set, and 13,277 in the testing set. The shots are indexed by their representative keyframe, as defined by the MediaMill Challenge. We selected the MediaMill Challenge because it is a realistic and challenging dataset with a shared ground truth, allowing repeatable experiments. In Fig. 6 we show some concepts defined by the MediaMill Challenge. Note the wide variety of concepts, i.e.: graphics (*drawing, maps, weather*), objects (*bird, chair, flag USA*), scenes (*duo-anchor, meeting, night fire, river, sky, splitscreen, studio, tennis*), persons (*anchor, Mr. Lahoud, prisoner*), and emotional (*entertainment*). The video data is a realistic subset of broadcast news, containing com-

mercials, e.g. (*bird, river*), and concepts with little variation in their appearance for this set, (e.g.*night fire,tennis, chair, weather, anchor*). In contrast to simplified datasets recorded in a laboratory setting [26], the MediaMill Challenge allows a more truthful extrapolation of our conclusions to other real-world datasets.

4.2. Visual categorization implementation

4.2.1. Image features

To evaluate if a method generalizes over visual features, we conduct all experiments with two different image features: Wiccest and Gabor. Wiccest features rely on natural image statistics which makes them well suited to describe natural images. On the other hand, Gabor features respond to regular textures and color planes, which is beneficial for man-made structures. Both these image features measure colored texture, where the Gabor features also takes non-textured color into account. Each feature is calculated on two scales, making them sensitive to differently scaled textures. We selected texture features because of their ability to
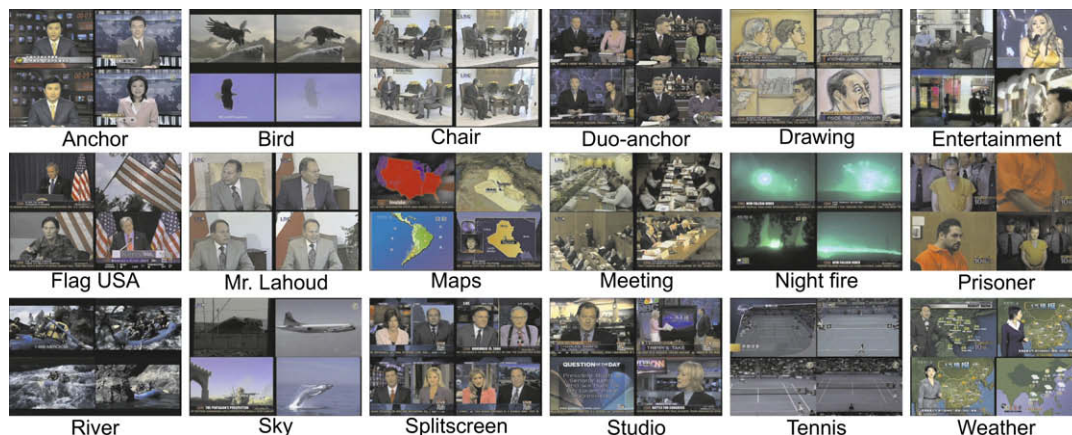

Fig. 6. Some examples of the concepts defined by the MediaMill Challenge, which we use to evaluate categorization performance.

describe the foreground as well as the contextual background of an image. More details about the image features are in Appendix A.

4.2.2. Image sampling

The codebook model represent an image as a distribution over codewords. To build this distribution, several regions are sampled from an image. Since grid-based sampling is shown to outperform interest points in scene categorization [7,17], we use a grid for region sampling. Specifically, this grid is constructed by dividing an image in several overlapping rectangular regions. The regions are uniformly sampled across the image, with a step size of half a region. We use two different region sizes, with ratios of $\frac{1}{2}$ and $\frac{1}{6}$ of both the x-dimension and y-dimension of the image.

4.3. Compact codebook models implementation

4.3.1. Semantic vocabulary

A semantic vocabulary consists of meaningful elements, obtained by annotation. We use the semantic vocabulary by [45]. This vocabulary consists of 15 different codewords, namely: building (321), car (192), charts (52), crowd (270), sand/rock (82), fire (67), flag USA (98), maps (44), mountain (41), road (143), sky (291), smoke (64), snow (24), vegetation (242), water (108), where the number in brackets indicates the number of annotation samples of that concept. We use the train set as a basis for selecting relevant shots containing the codewords. In those shots, we annotate rectangular regions where the codeword is visible for at least 20 frames. Note that a vocabulary of 15 codewords, evaluated for two scales and two region sizes will yield a descriptor of $4 \times 15 = 60$ elements.

4.3.2. Globally-clustered vocabulary

A globally-clustered vocabulary is created on all image features in the train set. We build a such a global vocabulary by radius-based clustering. Radius-based clustering aims to cover the feature space with clusters of a fixed similarity radius. Hence, the algorithm yields an even distribution of visual words over the feature space and has been shown to outperform the popular k-means algorithm [17]. Whereas Jurie and Triggs [17] use mean-shift with a Gaussian kernel to find the densest-point, we maximize the number of features within its radius r for efficiency reasons.

Since each image features is calculated at two scales for two region sizes there are four image descriptors per feature. We cluster each descriptor separately, yielding four different clustering steps. The final vocabulary consists of the resulting clusters for a single radius as found by all these four clustering steps. Note that the number of clusters may vary per scale and region size combination.

4.3.3. Concept-specific clustered vocabulary

A concept-specific vocabulary is designed for a single concept. Such a specific vocabulary may be found by limiting the radius-based clustering algorithm to images in a single class only. This makes the resulting clusters depend on only that subset of the feature space which is relevant for the concept. Note that the images are labeled globally, whereas the clustering is based on local codewords. The clustering step itself is identical to the globally-clustered vocabulary, and is performed separately for each of the four feature scale and region size combinations.

4.4. Supervised machine learning implementation

Automatic concept categorization in video requires machine learning techniques. For each semantic concept, we aim for a ranking of shots relevant to this concept. To evaluate this ranking, we employ two classifiers: a strong and computationally intensive SVM classifier and a weak but fast Fisher classifier. Fisher's linear discriminant [8] projects high-dimensional features to a one-dimensional line that aims to maximize class separation. The most important reason why we use Fisher's linear discriminant is its fair categorization performance with high efficiency. This efficiency is mostly due to its linearity and the benefit that this classifier has no parameters to tune. The other classifier is the popular discriminative maximum-margin SVM classifier. The reason for choosing an SVM is because it generally gives good results on this type of data [40]. For the SVM we use a non-linear χ^2 kernel, where we use episode constrained cross-validation [46] to tune the best C-slack parameter.

4.5. Evaluation criteria

We evaluate compactness and categorization performance. Compactness is measured in by the size of the codebook vocabulary. For measuring categorization performance, we adopt average precision from the Challenge framework. Average precision is a single-valued measure that summarizes the recall–precision curve. If $L_k = \{s_1, s_2, \ldots, s_k\}$ are the top k ranked elements from the retrieved results set L, and let R denote the set of all relevant items, then average precision (AP) is defined as

$$AP(L) = \frac{1}{|R|} \sum_{k=1}^{|L|} \frac{|L_k \cap R|}{k} I_R(s_k), \tag{3}$$

where $|\cdot|$ denotes set cardinality and $I_R(s_k) = 1$ if $s_k \in R$ and 0 otherwise. In our experiments we compute AP over the whole result set.

Average precision measures the categorization performance for a single concept. The MediaMill Challenge, however, defines 101 concepts. As the performance measure over multiple concepts, we report the mean average precision (MAP), given by the average precision averaged over all concepts.

5. Experimental results

5.1. Experiment 1: soft-assignment versus hard-assignment

The first experiment compares soft-assignment with hard-assignment in the codebook model for a semantic vocabulary over two classifiers and over the two visual features. In Appendix A we detail both features and their respective soft-assignment functions. In Fig. 7 we show the results for the Wiccest and Gabor features. The figure illustrates that performance for nearly all concepts improves by using soft-assignment. This improvement is in line with the expectations in [45]. In the few cases where soft-assignment is outperformed by hard-assignment, the performance difference is marginal. On average over the two features and two classifiers there are 92 ± 2.71 concepts that increase and 8.75 ± 2.87 concepts that decrease. Over both features and both classifiers there are 78 of the 101 concepts that always improve. In contrast, there is no concept whose performance always decreases. For the four feature-classifier combinations, there are 28 concepts that decrease in performance for at least one of these combinations. Note that this is the absolute worst-case performance. In contrast, all 101 concepts are found to increase at least once or more in the four feature-classifier combinations. The average performance over all 101 concepts for the two visual features is shown in Table 1. The table shows that using soft-assignment improves performance for both feature types and for both classifiers.

The difference per concept between soft-assignment and hard-assignment is given in Fig. 8. Here we show the five most increasing concepts and the five most decreasing concepts by replacing hard-assignment with soft-assignment. Note that the performance gain by the improving concepts is several magnitudes higher than the decrease in performance. There are four concepts that
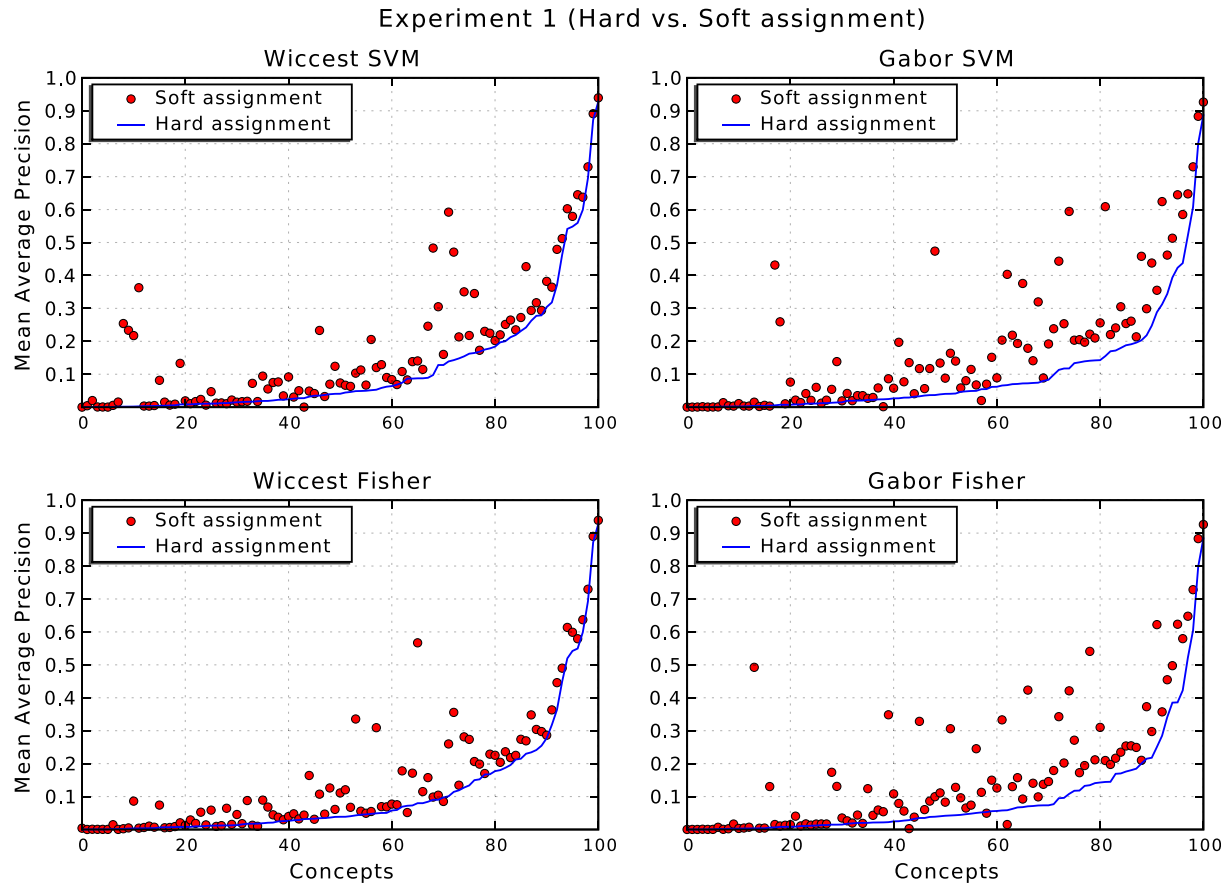
Experiment 1 (Hard vs. Soft assignment)

Fig. 7. Comparing hard-assignment versus soft-assignment for all 101 concepts, over two different visual features with a semantic vocabulary.

consistently decrease in the bottom five. The concepts *prisonerperson, HassanNasrallah, bicycle* are found in the bottom five of the Gabor features for both the Fisher as the SVM classifier. These concepts are sensitive to exact color matching. The *bicycle* concept is a sparse but repetitive commercial, and the *prisonerperson, HassanNasrallah* concepts contain shots of highly discriminative colors, like an orange prisoner uniform. Since the Gabor features take the color of an image patch into account, these features are more effected than the Wiccest features. The six concepts *bird, river, DuoNewsAnchorPersons, graphicalmap, EmileLahoud, splitscreen* consistently increase in the top five. Of these six concepts the concepts *graphicalmap* and *EmileLahoud* are found in the Gabor features top five for both the Fisher as the SVM classifier. In this case the concepts are again typically colorful, such as the many variations of a *graphicalmap*, or a colorful flag in the background of Mr. *EmileLahoud*. In this case, however, performance increases. We deem that this is the case because there is significant variation in the colors. By using soft-assignment this variation is better modeled. The concept *DuoNewsAnchorPersons* increases for the Wiccest

features in both the SVM as in the Fisher classifier. Again, we attribute the gain of soft-assignment to slight variation between the examples. With slight variation in the images, hard-assignment may choose complete different visual words, whereas soft-assignment proves robust. The concept *splitscreen* is found in the top five of three feature-classifier combinations. Only the Gabor-Fisher does not have this concept in the top five. This concept is characterized by a strong artificial edge in the middle of the screen.

Besides this edge, there is some variation on the people present in the screens. Again, soft-assignment seems to be able do deal better with this variation. The concept *bird* improves for Wiccest-Fisher and for Gabor-SVM. This concept is a repetitive commercial. We attribute the reason why static or near-copies benefit most to the fact that minor changes in the image content results in minor changes in the soft-assignment approach. In contrast, minor image content changes in the traditional codebook model may give rise to completely different codewords stemming from the hard-assignment in this method. In Fig. 6 we show example images for some concepts.

5.2. Experiment 2: semantic vocabulary versus globally-clustered vocabulary

As a second experiment, we focus on the difference between a semantic vocabulary and a clustered vocabulary. In Fig. 9 we show the results with hard-assignment and soft-assignment over the two features and over the two classifiers. This figure shows that increasing the number of visual words increases the performance. Moreover, the figure shows a clear advantage of using an SVM classifier over the Fisher classifier. Nevertheless, for Gabor features with a vocabulary of 1480 codewords the Fisher classifier proves

Table 1
The mean average precision over all 101 concepts in experiment 1. Results are shown for hard-assignment versus soft-assignment for Wiccest features and Gabor features and the Fisher and SVM classifier, using a semantic vocabulary. Note that soft-assignment outperforms hard-assignment for both feature types and for both classifiers.

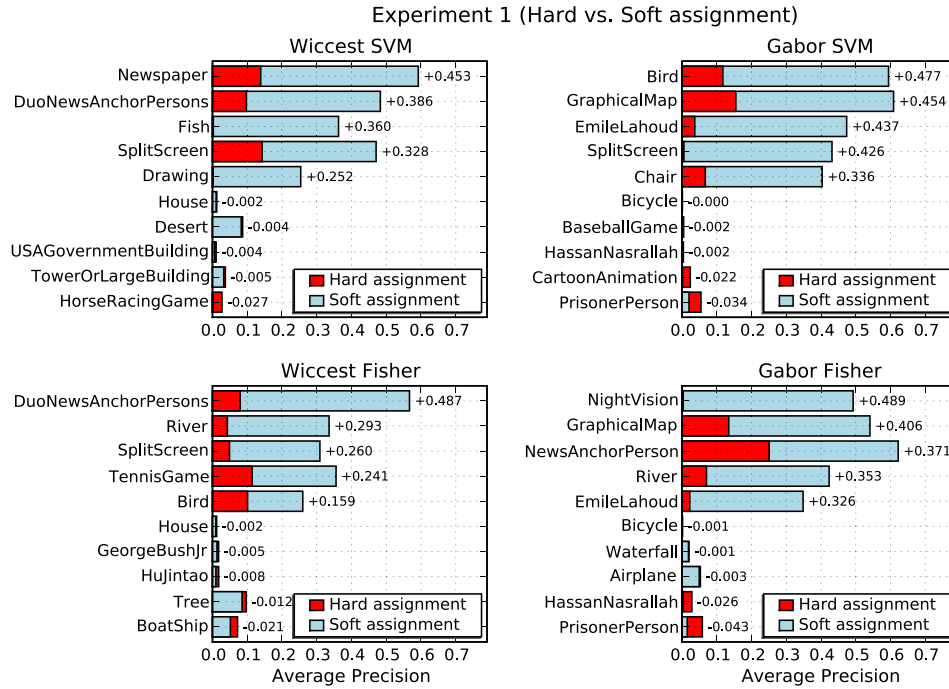| Experiment 1	Wiccest		Gabor	
	SVM	Fisher	SVM	Fisher
Hard-assignment	0.120	0.113	0.100	0.097
Soft-assignment	0.179	0.157	0.187	0.175

Fig. 8. The difference between soft-assignment and hard-assignment for the top and bottom five concepts in experiment 1.
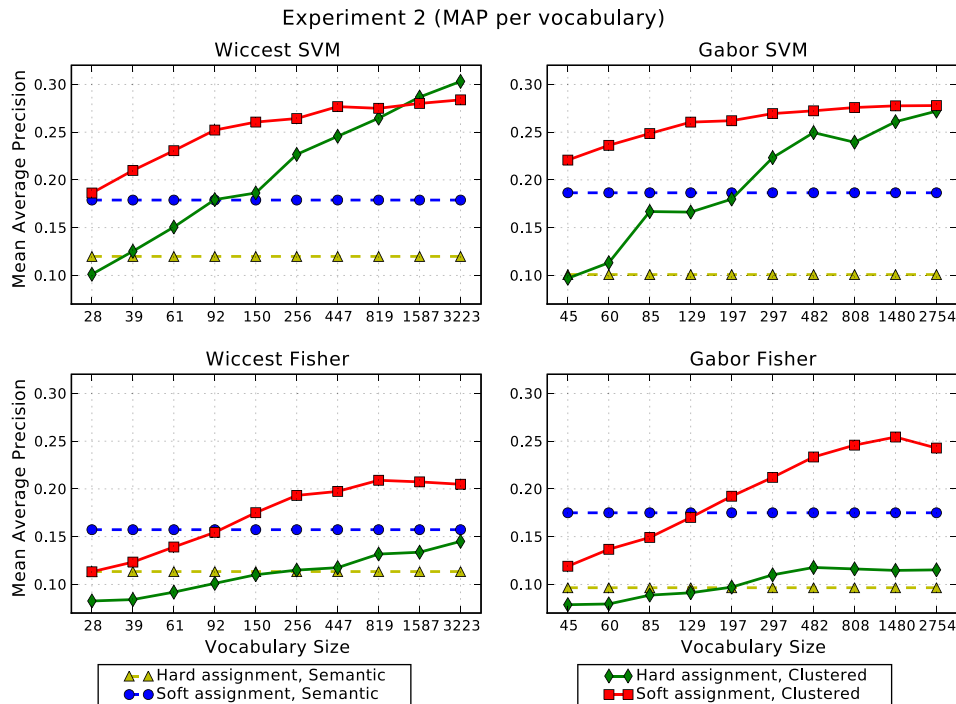

Fig. 9. Comparing a semantic codebook vocabulary with a globally-clustered codebook vocabulary for hard-assignment and soft-assignment. Results are shown in mean average precision over 101 concepts. The semantic vocabulary is the same as in experiment 1. Note that the Wiccest and the Gabor features have different vocabulary sizes. This is the case, because the number of clusters depends on the similarity function of the visual features (see Appendix A).

competitive to an SVM classifier. Note that a larger vocabulary not always yields the best results. For example, for the Fisher classifier with soft-assignment, the largest vocabulary is not the best performing one. Furthermore, the figure shows that for Wiccest features and a Fisher classifier the performance difference between a semantic and a clustered vocabulary is only slightly in favor of the semantic vocabulary when both vocabularies have an equal

number of visual words (±60). In contrast, for Gabor features a semantic vocabulary is more beneficial, yielding a higher performance for a lower number of codewords. We credit this difference between the Wiccest and the Gabor features to the difference in dimensionality between the features. The Wiccest features use only 12 numbers, whereas the Gabor features consist of histograms of 101 bins. Since the feature-space of the Gabor descriptor is much

higher in dimensionality, it is harder to fill this space, let alone find discriminative visual words. In contrast to clustering, a semantic vocabulary is given by manual annotation. This annotation step introduces meaningful visual words without the need to partition a high-dimensional feature space. Nevertheless, a fixed sized semantic vocabulary is outperformed by a clustered vocabulary for both features. This performance gain comes at a price, paid by an exponentially growing visual word vocabulary, leading to a more complex, and therefore less compact model. Comparing the results of a semantic vocabulary and a clustered vocabulary for the SVM classifier, shows a clear advantage for a clustered vocabulary. The clustered vocabulary already outperforms a semantic vocabulary with half the number of codewords in the case of Wiccest features. Moreover, for the Wiccest features the hard-assignment method outperforms the soft-assignment method for large vocabularies. In the case of the Gabor features, the hard-assignment performance equal to soft-assignment for large vocabularies. Nevertheless, for an SVM classifier, soft-assignment proves robust over the size of the vocabulary. Soft-assignment clearly outperforms hard-assignment for compact vocabularies.

In Fig. 10 we show per concept the vocabulary size which gives the best performance. Moreover, we show the contours of the areas that perform within 90% of the best score. When comparing soft-assignment versus hard-assignment, it can be seen that for soft-assignment there are more areas where the performance is within 90% of the best score. Hence, soft-assignment seems more robust to the size of the vocabulary. Furthermore, the figure shows that soft-assignment has more variation in the size of the best vocabulary than hard-assignment. Hence, soft-assignment seems the better choice for compact vocabularies. Moreover, as the variation in the size of the best vocabulary suggests, it may prove beneficial to tune a vocabulary per concept, instead of using a global vocabulary. This tuning per concept is explored in the next section.

5.3. Experiment 3: semantic vocabulary versus concept-specific clustered vocabulary

In an attempt to create more compact vocabularies while keeping performance on par, we evaluate individual vocabularies that are tuned to the specific concept at hand. These concept-specific
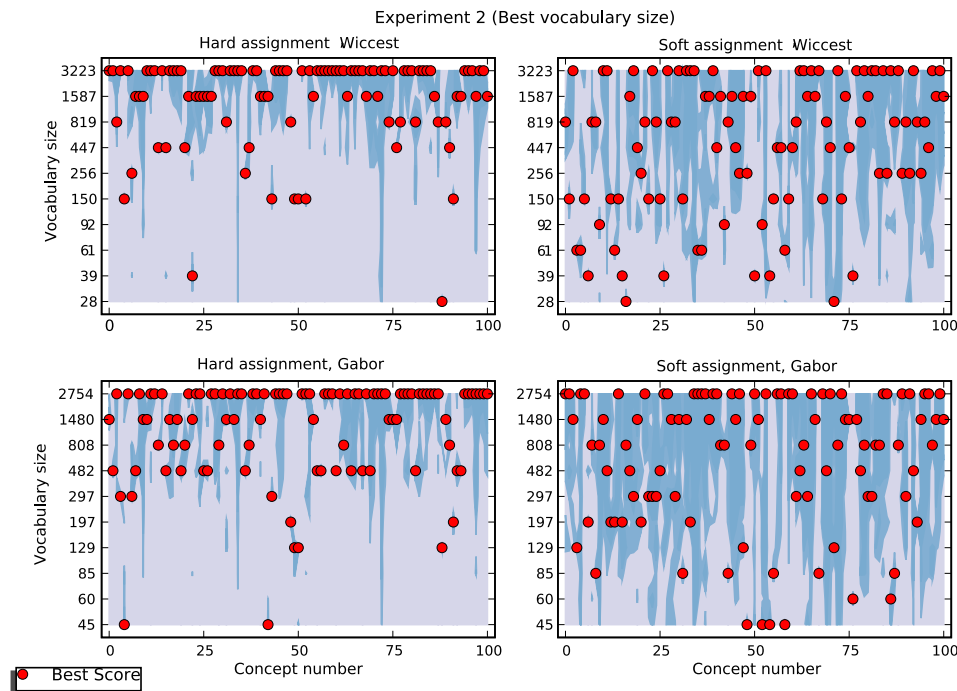

Fig. 10. The red dots indicate the best performing vocabulary size for each concept. The contours highlight the area within 90% of the best performance. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
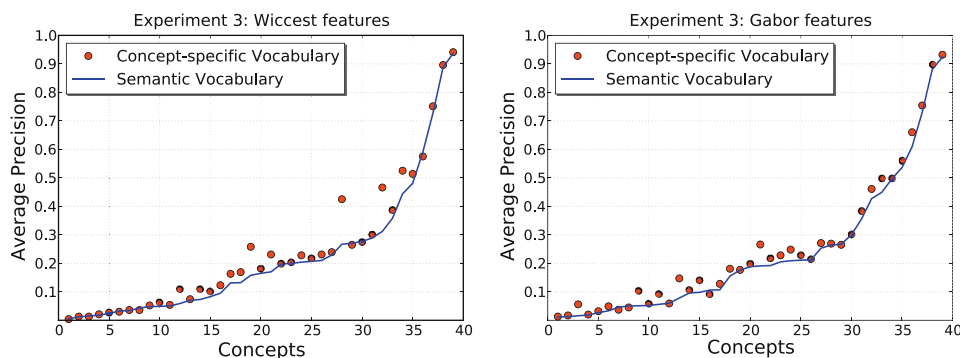

Fig. 11. Comparing a semantic vocabulary with a concept-specific vocabulary, both using soft-assignment.

vocabularies are created by restricting the radius-based clustering algorithm to the positive examples of a semantic concept. To constrain the computations, we limit this experiment to the Fisher classifier only and to the 39 concepts that were used in the TREC-VID 2006 benchmark. Moreover, we select a fixed radius for the clustering algorithm: $r = 1.2$ for the Wiccest features and $r = 4.5$ for the Gabor features. These radii are selected with the intention to closely match the performance of the semantic vocabulary.

The performance differences between the semantic vocabulary and the concept-specific vocabularies for the Wiccest and Gabor features using soft-assignment are shown in Fig. 11. Note that the performance of both methods is closely aligned. Nevertheless, there are a few concepts that perform better with a concept-specific vocabulary. The top ten of the concepts that increase most are shown in Fig. 12. Some video frames containing these concepts are shown in Fig. 6. In the top ten, there are three concepts (*animal, weather, sky*) that increase for both features. The other features that improve per visual feature seem related to the feature type. The Wiccest features are related to edge statistics as found in natural images, and the concepts that improve are related to natural scenes (*animal, mountain, waterbody, desert, sports, sky, crowd*). Furthermore, it is striking that seven concepts out of the top ten for the Wiccest features consist of elements that are also used in the semantic vocabulary (*mountain, waterbody, desert, charts, maps,*

sky, crowd). We speculate that this is the case because the improved concepts for the Wiccest features focus on natural images, and the semantic vocabulary consists mainly of naturally occurring codewords. In the case of Gabor features, that are more related to color and texture frequency, the concepts that improve may rely on colored texture for discrimination (*prisoner, flag USA, meeting, entertainment, weather, studio*). Nevertheless, disregarding those few outliers who outperform the semantic vocabulary, both vocabulary types perform more or less equal, as intended.

In Table 2 we show the number of codewords used to achieve more or less the same performance. The number of codewords for the concept-specific vocabulary was found by increasing the radius of the clustering algorithm, until the performance of the concept-specific clustered vocabulary was reached. The results show that an annotated vocabulary has the most compact descriptor, with only 60 visual words. In contrast, the globally-clustered vocabulary requires at least three times more visual words than a semantic vocabulary. The individually clustered concept-specific vocabularies require two times the number of codewords than a semantic vocabulary. However, those concept-specific vocabularies are still only half the size of a globally clustered vocabulary. Hence, while a semantic vocabulary proves the most descriptive, the concept-specific clustered vocabularies yield a more powerful descriptor than a globally-clustered vocabulary (see Table 3).
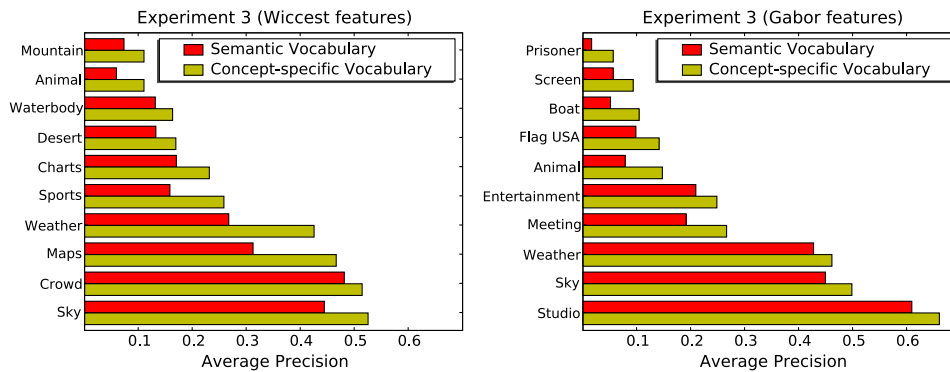

Fig. 12. The 10 concepts that benefit most from a concept-specific vocabulary over a semantic vocabulary.

Table 2
The number of codewords used to obtain the same performance over three types of vocabularies: semantic (experiment 1), clustered (experiment 2), and concept-specific (experiment 3). The size of the codeword vocabulary is shown, with the mean average precision in brackets for Wiccest features and Gabor features using soft-assignment. In the case of the concept-specific vocabulary, we show the average number of codewords, since this varies per concept.

Feature	Experiment 1: semantic		Experiment 2: clustered		Experiment 3: concept-specific	
	Size	MAP	Size	MAP	Size	MAP
Wiccest	60	0.219	205	0.251	128.7	0.244
Gabor	60	0.235	249	0.270	118.5	0.254

Table 3
Summary of the four evaluated methods to obtain a compact and expressive codebook. We indicate if a method requires manual annotation effort, computation effort, and if the method yields compact models, with good performance. We distinguish between a strong classifier such as an SVM and a weak classifier such as Fisher's linear discriminant. A + denotes good, − indicates bad, and ± is medium. Note that soft-assignment is performed after vocabulary creation, thus it is not affected by annotation nor clustering.

Method	Manual	Computational		Compact		Performance	
		Strong	Weak	Strong	Weak	Strong	Weak
Semantic	−	±	+	−	+	−	±
Globally clustered	+	−	±	+	−	+	+
Concept-specific clustered	+	−	−	+	+	+	+

5.4. Summary of experimental results

We summarize our results in Table 3. The first observation we can make is that soft-assignment typically outperforms hard-assignment in the codebook method. This improvement has been shown for two different visual features and for both a semantic vocabulary and a clustered vocabulary over two classifiers. Only for a very large vocabulary and an SVM classifier hard-assignment may improve over soft-assignment. Furthermore, the semantic vocabulary which requires manual annotation work has been shown to provide a competitive vocabulary when a weak classifier is used. In the case of the Fisher classifier it yields excellent performance with a minimum number of visual words leading to compact and expressive codebooks. For the Fisher classifier, a clustered vocabulary outperforms a semantic vocabulary when the number of visual words is high enough. However, this high number of visual words leads to less compact models, which may be infeasible for large video datasets. In the case of a strong classifier, the results show that clustered vocabularies outperform a semantic vocabulary. However, an SVM classifier takes more effort to train, with additional complication with cross-validation for parameter tuning [46]. Additional results indicate that the number of visual words in a clustered vocabulary may be reduced by tuning this vocabulary to each concept. These tuned vocabularies retain categorization performance while maintaining a reasonably compact vocabulary.

6. Conclusions

Given the vast amount of visual information available today, the applicability of automatic visual indexing algorithms is constrained by their efficiency. Accordingly, this paper focuses on compact, and thus efficient, models for visual concept categorization. We considered the codebook algorithm where model complexity is determined by the size of the vocabulary. We structurally compared four approaches that lead to compact and expressive codebooks. Specifically, we compared three methods to create a compact vocabulary: (1) global clustering, (2) concept-specific clustering, and (3) a semantic vocabulary. The fourth approach increases expressive power by soft-assignment of codewords to image features. We experimentally compared these four methods on a large and standard video collection. The results show that soft-assignment improves the expressive power of the vocabulary, leading to increased categorization performance without sacrificing vocabulary compactness. Further experiments showed that a semantic vocabulary leads to compact vocabularies, while retaining reasonable categorization performance. A concept-specific vocabulary leads to reasonable compact vocabularies, while providing fair visual categorization performance. Given these results, the best method depends at the application at hand. In this paper we presented a guideline for selecting a method given the size of the video dataset, the desirability of manual annotation, the amount of available computing power and the desired categorization performance.
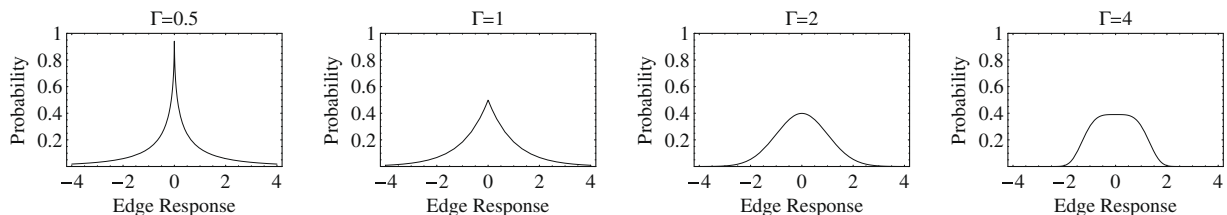
Appendix A. Image features

A.1. Wiccest features

Wiccest features [9] utilize natural image statistics to effectively model texture information. Texture may be described by the distribution of edges at a certain region in an image. Hence, a histogram of a Gaussian derivative filter is used to represent the edge statistics. The histogram describes image statistics in natural textures, which are well modeled with an integrated Weibull distribution [9]. This distribution is given by

$$f(r) = \frac{\gamma}{2\gamma^{\frac{1}{\gamma}}\beta\Gamma(\frac{1}{\gamma})} \exp\left\{ -\frac{1}{\gamma}\left|\frac{r-\mu}{\beta}\right|^{\gamma} \right\}, \tag{A.1}$$

where r is the edge response to the Gaussian derivative filter and $\Gamma(\cdot)$ is the complete Gamma function, $\Gamma(x) = \int_0^\infty t^{x-1}e^{-1}dt$. The parameter β denotes the width of the distribution, γ represents the 'peakness' of the distribution, and μ denotes the mode of the distribution. See Fig. A.1 for examples of the integrated Weibull distribution.

The Wiccest features for an image region consist of the Weibull parameters for the illumination invariant edges in the region at $\sigma = 1$ and $\sigma = 3$ of the Gaussian filter [45]. The β and γ values for the x-edges and y-edges of the three opponent color channels normalized by the intensity [10] yields a 12-dimensional descriptor. The similarity, $S_{\mathcal{W}}$, between two Wiccest features is given by the accumulated fraction between the respective β and γ parameters

$$S_{\mathcal{W}}(F, G) = \sum \left(\frac{\min(\beta_F, \beta_G)}{\max(\beta_F, \beta_G)} \frac{\min(\gamma_F, \gamma_G)}{\max(\gamma_F, \gamma_G)} \right), \tag{A.2}$$

where F and G are Wiccest features.

A.2. Color Gabor features

As an alternative to Wiccest features, one may use the popular Gabor filters. Gabor filters may be used to measure perceptual surface texture in an image [6]. Specifically, Gabor filters respond to regular patterns in a given orientation on a given scale and frequency. A 2D Gabor filter is given by

$$\widetilde{G}(x, y) = G_\sigma(x, y) \exp\left\{ 2\pi i \begin{pmatrix} \Omega_{x_0} \\ \Omega_{y_0} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \right\}, \quad i^2 = -1, \tag{A.3}$$

where $G_\sigma(x, y)$ is a Gaussian with a scale σ, $\sqrt{\Omega_{x_0}^2 + \Omega_{y_0}^2}$ is the radial center frequency and $\tan^{-1}\left(\frac{\Omega_{y_0}}{\Omega_{x_0}}\right)$ the orientation. Note that a zero-frequency Gabor filter reduces to a Gaussian filter. An example of color Gabor filters is shown in Fig. A.2. Illumination invariance is obtained by normalizing each Gabor filtered opponent-color channel by the intensity [13]. A histogram is constructed for each Gabor filtered color channel, where the Gabor similarity measure, $S_{\mathcal{G}}$, is given by histogram intersection,
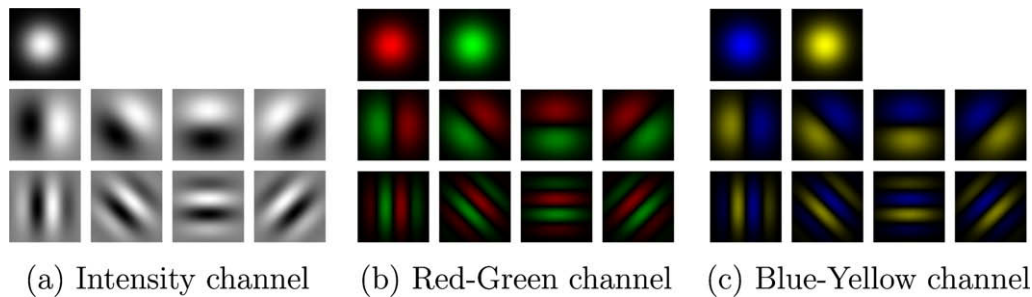

Fig. A.1. Some examples of the integrated Weibull distribution for $\beta = 1, \mu = 0$, varying values for $\gamma \in \{\frac{1}{2}, 1, 2, 4\}$.

(a) Intensity channel   (b) Red-Green channel   (c) Blue-Yellow channel

Fig. A.2. Some examples of the color Gabor filter with the chosen orientations, scales and frequencies.

$$S_{\mathscr{G}}(I, M) = \sum_{j=1}^{n} \min(I_j, M_j), \qquad (A.4)$$

where I_j is bin j of the n-dimensional histogram of image I.

In the case of a Gabor filter, its parameters consist of orientation, scale and frequency. We follow Hoang et al. [13] and use four orientations, $0°, 45°, 90°, 135°$, and two fixed (scale and frequency) pairs: (2.828 and 0.720), (1.414 and 2.094), where we append zero frequency color to each scale. Furthermore, the histogram representation of the Gabor filters uses 101 bins for each Gabor filtered color channel.

References

[1] A. Agarwal, B. Triggs, Multilevel image coding with hyperfeatures, International Journal of Computer Vision 78 (1) (2008) 15–27.

[2] M. Bar, Visual objects in context, Nature Reviews: Neuroscience 5 (2004) 617–629.

[3] D. Blei, A. Ng, M. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[4] A. Bosch, A. Zisserman, X. Munoz, Scene classification using a hybrid generative/discriminative approach, IEEE Pattern Analysis and Machine Intelligence 30 (4) (2008) 712–727.

[5] M. Boutell, J. Luo, C. Brown, Factor-graphs for region-based whole-scene classification, in: CVPR SLAM Workshop, 2006.

[6] A.C. Bovik, M. Clark, W.S. Geisler, Multichannel texture analysis using localized spatial filters, IEEE Pattern Analysis and Machine Intelligence 12 (1) (1990) 55–73.

[7] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: CVPR, 2005.

[8] R. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188.

[9] J.M. Geusebroek, Compact object descriptors from local colour invariant histograms, in: BMVC, 2006.

[10] J.M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, H. Geerts, Color invariance, IEEE Pattern Analysis and Machine Intelligence 23 (12) (2001) 1338–1350.

[11] K. Grauman, T. Darrell, Pyramid match hashing: sub-linear time indexing over partial correspondences, in: Proceedings of CVPR 2007, 2007.

[12] A.G. Hauptmann, M.-Y. Chen, M. Christel, W.-H. Lin, J. Yang, A hybrid approach to improving semantic extraction of news video, in: ICSC, 2007.

[13] M.A. Hoang, J.M. Geusebroek, A.W.M. Smeulders, Color texture measurement and segmentation, Signal Processing 85 (2) (2005) 265–275.

[14] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Machine Learning 42 (1/2) (2001) 177–196.

[15] D. Hoiem, A. Efros, M. Hebert, Putting objects in perspective, in: CVPR, 2006.

[16] Y.-G. Jiang, C.-W. Ngo, J. Yang, Towards optimal bag-of-features for object categorization and semantic video retrieval, in: CIVR, 2007.

[17] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: ICCV, 2005.

[18] L. Kennedy, S.-F. Chang, A Reranking Approach for Context-based Concept Fusion in Video Indexing and Retrieval, in: CIVR, 2007.

[19] K. Kise, K. Noguchi, M. Iwamura, Simple representation and approximate search of feature vectors for large-scale object recognition, in: BMVC07, 2007.

[20] S. Kumar, M. Hebert, A hierarchical field framework for unified context-based classification, in: ICCV, 2005.

[21] D. Larlus, F. Jurie, Latent mixture vocabularies for object categorization, in: BMVC, 2006.

[22] M. Marszałek, C. Schmid, H. Harzallah, J. van de Weijer, Learning object representations for visual object class recognition, in: Visual Recognition Challenge Workshop, in Conjunction with ICCV, 2007.

[23] K. Mikolajczyk, J. Matas, Improving descriptors for fast tree matching by optimal linear projection, in: Proceedings of IEEE International Conference on Computer Vision, 2007.

[24] A. Mojsilović, J. Gomes, B. Rogowitz, Semantic-friendly indexing and quering of images based on the extraction of the objective semantic cues, International Journal of Computer Vision 56 (1–2) (2004) 79–107.

[25] F. Moosmann, B. Triggs, F. Jurie, Fast discriminative visual codebooks using randomized clustering forests, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, vol. 19, MIT Press, Cambridge, MA, 2006, pp. 985–992.

[26] H. Müller, S. Marchand-Maillet, T. Pun, The truth about corel-evaluation in image retrieval, in: CIVR, 2002.

[27] K. Murphy, A. Torralba, W. Freeman, Using the forest to see the trees: a graphical model for recognizing scenes and objects, in: NIPS 16, 2004.

[28] M. Naphade, T. Huang, A probabilistic framework for semantic video indexing, filtering, and retrieval, IEEE Transactions on Multimedia 3 (1) (2001) 141–151.

[29] D. Nistér, H. Stewénius, Scalable recognition with a vocabulary tree, in: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 2006.

[30] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.

[31] F. Perronnin, Universal and adapted vocabularies for generic visual categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (7) (2008) 1243–1256.

[32] J. Philbin, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[33] P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, T. Tuytelaars, A thousand words in a scene, IEEE Pattern Analysis and Machine Intelligence 29 (9) (2007) 1575–1589.

[34] G. Salton, M. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

[35] F.J. Seinstra, J.M. Geusebroek, D.C. Koelma, C.G.M. Snoek, M. Worring, A.W.M. Smeulders, High-performance distributed video content analysis with parallel-horus, IEEE Multimedia 14 (4) (2007) 64–75.

[36] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: ICCV, vol. 2, 2003.

[37] A. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and trecvid, in: MIR, 2006.

[38] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content based image retrieval at the end of the early years, IEEE Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.

[39] C.G.M. Snoek, M. Worring, D.C. Koelma, A.W.M. Smeulders, A learned lexicon-driven paradigm for interactive video retrieval, IEEE Transactions on Multimedia 9 (2) (2007) 280–292.

[40] C.G.M. Snoek, M. Worring, J.C. van Gemert, J.M. Geusebroek, A.W.M. Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: ACM Multimedia, 2006.

[41] E. Sudderth, A. Torralba, W. Freeman, A. Willsky, Describing visual scenes using transformed objects and parts, International Journal of Computer Vision 77 (1–3) (2008) 291–330.

[42] A. Torralba, Contextual priming for object detection, International Journal of Computer Vision 53 (2) (2003) 169–191.

[43] T. Tuytelaars, C. Schmid, Vector quantizing feature space with a regular lattice, in: ICCV, 2007.

[44] K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, in press.

[45] J.C. van Gemert, J.M. Geusebroek, C.J. Veenman, C.G.M. Snoek, A.W.M. Smeulders, Robust scene categorization by learning image statistics in context, in: CVPR-SLAM, 2006.

[46] J.C. van Gemert, C.J. Veenman, J.M. Geusebroek, Episode-constrained cross-validation in video concept retrieval, IEEE Transactions on Multimedia 11 (4) (2009) 780–785.

[47] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, J.M. Geusebroek, Visual word ambiguity, IEEE Transactions on Pattern Analysis and Machine Intelligence, in press.

[48] J. Vogel, B. Schiele, Semantic modeling of natural scenes for content-based image retrieval, International Journal of Computer Vision 72 (2) (2007) 133–157.

[49] H. Wactlar, T. Kanade, M. Smith, S. Stevens, Intelligent access to digital video: the informedia project, IEEE Computer 29 (5) (1996) 46–52.

[50] J. Winn, A. Criminisi, T. Minka, Object categorization by learned universal visual dictionary, in: ICCV, 2005.