A Probabilistic Multimedia Retrieval Model and its Evaluation

Thijs Westerveld[1], Arjen P. de Vries[1], Alex van Ballegooij[1], Franciska de Jong[2], and Djoerd Hiemstra[2]

[1] CWI
PO Box 94079, 1090 GB Amsterdam
The Netherlands
{thijs,arjen,alexb}@cwi.nl
[2] University of Twente
PO Box 217, 7500 AE Enschede
The Netherlands
{hiemstra,fdejong}@cs.utwente.nl

Abstract. In this paper we present a probabilistic model for the retrieval of multimodal documents. The model is based on Bayesian decision theory and combines models for text based search with models for visual search. The textual model is based on the language modelling approach to text retrieval and the visual information is modelled as a mixture of Gaussian densities. Both models have been proved successful on various standard retrieval tasks. We evaluate the multimodal model on the search task of TREC's video track. We found that the disclosure of video material based on visual information only is still too difficult. Even with purely visual information needs, text based retrieval still outperforms visual approaches. The probabilistic model is useful for text, visual and multimedia retrieval. Unfortunately, simplifying assumptions that reduce its computational complexity degrade retrieval effectiveness. Regarding the question whether the model can effectively combine information from different modalities, we conclude that whenever both modalities yield reasonable scores, a combined run outperforms the individual runs.

1 Introduction

Both image analysis and video motion processing have been unable to meet the requirements for the disclosing of content of large scale unstructured video archives. There appear to be two major unsolved problems in the indexing and retrieval of video material on the basis of these technologies, viz., (a) image and video-processing is still far away from understanding the content of a picture in the sense of a knowledge-based understanding, and (b) there is no effective query language (in the wider sense) for searching image and video databases. Unlike the target content in the field of text retrieval, the content of video archives is hard to capture at the conceptual level. An increasing number of developers that accept this analysis of the state-of-the art in the field have started

to use human language as the media interlingua, making the assumption that, as long as there is no possibility to carry out both a broad scale recognition of visual objects and an automatic mapping from such objects to linguistic representations, the detailed content of video material is best disclosed through the linguistic content (text) that may be associated with the images: speech transcripts, manually generated annotations, subtitles, captions, and so on [5].

Since the recent advances in automatic speech recognition, especially the potential role of speech transcripts in improving the disclosure of multimedia archives has been given a lot of attention. One of the insights gained by these investigations is that for the purpose of indexing and retrieval, perfect word recognition is not an indispensable condition, since not every word will have to make it into the index, relevant words are likely to occur more than once, and not every expression in the index is likely to be queried. Research into the differences between text retrieval and spoken document retrieval indicates that, given the current level of performance of information retrieval techniques, recognition errors do not add new problems for the retrieval task [7, 11].

The limitations inherent to the deployment of language features only have already lead to several attempts to deal with the requirements of video retrieval by more closer integration of human language technology and image processing. The notion of multimodal and even more ambitious: cross modal retrieval, have come in use to refer to the exploitation of the analysis of a variety of feature types in representing and indexing aspects of video documents [25, 26, 20, 21, 1, 3].

As indicated, many useful tools and techniques have become available from various research areas that have contributed to the domain of multimedia retrieval, but the integration of automatically generated multimodal metadata is most often done in an ad hoc manner. The various information modalities that play a role in video documents are each handled by different tools. How the various analyses affect the retrieval performance is hard to establish and it is impossible to give an explanation of performance results in terms of a formal retrieval model.

This paper describes an approach which employs both textual and image features and represents them in terms of one uniform theoretical framework. The output from various feature extraction tools is represented in probabilistic models based on Bayesian decision theory and the resulting model is a transparent combination of two similar models, one for textual features, based on language models for text and speech retrieval [9], and one for image features based on a mixture of Gaussian densities [22]. Initial deployment of the approach within the search tasks for the video retrieval tracks in TREC-2001 [15] and TREC-2002 [19] has demonstrated the possibility to use this model in retrieval experiments for unstructured video content. Additional experiments have taken place for smaller test collections.

Section 2 of this paper describes the general probabilistic retrieval model, its textual (Section 2.1) and visual constituents (Section 2.2). Section 3 presents the experimental setup followed by a number of experimental results to evaluate the effectiveness of the retrieval model. Finally, Section 4 summarises our main conclusions.

2 Probabilistic Retrieval Model

If we reformulate the information retrieval problem to one of pattern classification, the goal is to find the class to which the query belongs. Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_M\}$ be the set of classes underlying our document collection and Q be a query representation. Using the optimal Bayes or maximum a posteriori classifier, we can then find the class ω^* with minimal probability of classification error.

$$\omega^* = \arg\max_i \; P(\omega_i|Q) \tag{1}$$

In a retrieval setting, the best strategy is to rank classes by increasing probability of classification error. When no classification is available, we can simply let each document be a separate class. It is hard to estimate (1) directly, therefore, we reverse the probabilities using Bayes' rule.

$$\begin{aligned} \omega^* &= \arg\max_i \; \frac{P(Q|\omega_i)P(\omega_i)}{P(Q)} \\ &= \arg\max_i \; P(Q|\omega_i)P(\omega_i) \end{aligned} \tag{2}$$

If the a priori probabilities of all classes are equal (i.e. $P(\omega_i)$ is uniform), the maximum a posteriori classifier (2) reduces to the maximum likelihood classifier, which is approximated by the Kullback-Leibler (KL) divergence between query model and class model.

$$\omega^* = \arg\min_i \; KL[P_q(x)||P_i(x)]$$

The KL-divergence measures the amount of information there is to discriminate one model from another. The best matching document is the document with the model that is hardest to discriminate from the query model. Figure 1 illustrates the retrieval framework.[1] We build models for queries and documents and compare them using the KL-divergence between the models. The visual part is modelled as a mixture of Gaussians (see Section 2.2), for the textual part we use the language modelling approach in which documents are treated as *bags of words* (see Section 2.1). The KL-divergence between query model and document model is defined as follows.

$$KL[P_q(x)||P_i(x)] = \int P(x|\omega_q) \log \frac{P(x|\omega_q)}{P(x|\omega_i)} \mathrm{d}x$$

$$= \int P(x|\omega_q) \log P(x|\omega_q) \mathrm{d}x - \int P(x|\omega_q) \log P(x|\omega_i) \mathrm{d}x,$$

The first integral is independent of ω_i and can be ignored, thus

$$\begin{aligned} \omega^* &= \arg\min_i \; KL[P_q(x)||P_i(x)] \\ &= \arg\max_i \int P(x|\omega_q) \log P(x|\omega_i) \mathrm{d}x \end{aligned} \tag{3}$$

[1] The query model is here, like the document models, represented as a Gaussian mixture model but it can also be represented as a *bag of blocks* (see Section 2.2).

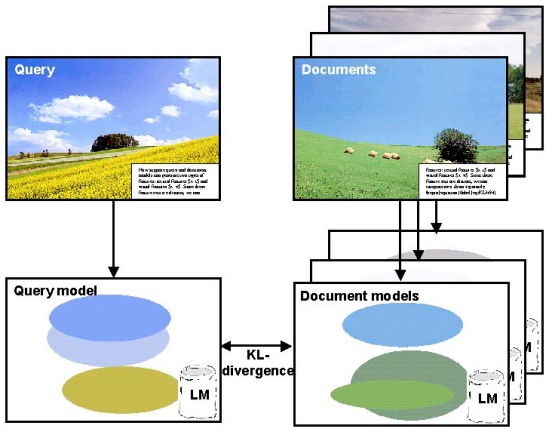


Fig. 1. Retrieval framework: Image represented as Gaussian Mixture, Text as Language Model ('bags of words')

When working with multimodal material like video, the documents in our collection contain features in different modalities. This means the classes underlying our document collection may contain different feature subclasses. The class conditional densities can thus be described as mixtures of feature densities.

$$P(x|\omega_i) = \sum_{f=1}^{F} P(x|\omega_{i,f})P(\omega_{i,f}),$$

where F is the number of underlying feature subclasses, $P(\omega_{i,f})$ is the probability of subclass f of class ω_i and $P(x|\omega_{i,f})$ is the subclass conditional density for this subclass. When we draw a random sample from class ω_i, we first select a feature subclass according to $P(\omega_{i,f})$ and then draw a sample from this subclass using $P(x|\omega_{i,f})$.

To arrive at a generic expression for similarity between mixture models, Vasconcelos [22] partitions the feature space into disjoint subspaces, where each point in the feature space is assigned to the subspace corresponding to the most probable feature subclass

$$\chi_k = \{x : P(\omega_{i,k}|x) \geq P(\omega_{i,l}|x) \; \forall \; l \neq k\}$$

Using this partition, Equation (3) can be rewritten to (proof given in [22]):

$$\int P(x|\omega_q)\log P(x|\omega_i)\mathrm{d}x =$$
$$\sum_{f,k} P(\omega_{q,f})\big[\log P(\omega_{i,k}) + \int_{\chi_k} P(x|\omega_{q,f}, x\in\chi_k)\log\frac{P(x|\omega_{i,k})}{P(\omega_{i,k}|x)}\mathrm{d}x\big]\int_{\chi_k} P(x|\omega_{q,f})\mathrm{d}x \tag{4}$$

When the subspaces χ_k form the same hard partioning of the features space for all query and document models, i.e. when

$$P(\omega_{i,k}|x) = P(\omega_{q,k}|x) = \begin{cases} 1, & \text{if } x\in\chi_k \\ 0, & \text{otherwise} \end{cases}$$

then

$$\int_{\chi_k} P(x|\omega_{q,f})\mathrm{d}x \begin{cases} 1, & \text{if } f=k \\ 0, & \text{otherwise} \end{cases}$$

and

$$P(\omega_{i,k}|x) = 1,\ \forall x\in\chi_k.$$

This reduces (4) to

$$\int P(x|\omega_q)\log P(x|\omega_i)\mathrm{d}x =$$
$$\sum_f P(\omega_{q,f})\log P(\omega_{i,f}) + \sum_f P(\omega_{q,f})\int_{\chi_f} P(x|\omega_{q,f})\log P(x|\omega_{i,f})\mathrm{d}x \tag{5}$$

This ranking formula is general and can *in principle* be used for any kind of multimodal document collection. In the rest of the paper, we limit ourselves to video collections represented by still frames and speech-recognized transcripts. The classes underlying our collection are defined through the shots in the videos. Furthermore, we assume we have two feature subclasses, namely a subclass generating textual features and one generating visual features. We can partition the feature space now in two distinct subspaces for textual and visual features: χ_t and χ_v. This partitioning is hard, i.e., a feature can be textual or visual but never both. Our ranking formula becomes:

$$\omega^* = \arg\max_i \int P(x|\omega_q)\log P(x|\omega_i)\mathrm{d}x =$$
$$= \arg\max_i \big[P(\omega_{q,t})\log P(\omega_{i,t}) + P(\omega_{q,t})\int_{\chi_t} P(x|\omega_{q,t})\log P(x|\omega_{i,t})\mathrm{d}x$$
$$+P(\omega_{q,v})\log P(\omega_{i,v}) + P(\omega_{q,v})\int_{\chi_v} P(x|\omega_{q,v})\log P(x|\omega_{i,v})\mathrm{d}x\big] \tag{6}$$

The mixture probabilities for the textual and visual models $P(\omega_{i,t})$ and $P(\omega_{i,v})$ might be derived from background knowledge about the class ω_i. If, for example, we know ω_i is a class from a news broadcast, we

might assign a higher value to $P(\omega_{i,t})$, since the probability that there is text that helps us in finding relevant information is relatively high. On the other hand, if ω_i is from a documentary or a silent movie, we might gain less information from the text from ω_i and assign a lower value to $P(\omega_{i,t})$. At the moment however, we have no background information, therefore we do not distinguish between classes and use uniform mixture probabilities. This means, the first and third term from (6) are independent of ω_i and can be ignored.
Our final (general) ranking formula becomes:

$$\begin{aligned}\omega^* = \arg\max_i \; \big[P(t) \int_{\chi_t} P(x|\omega_{q,t}) \log P(x|\omega_{i,t}) \mathrm{d}x \\ + \; P(v) \int_{\chi_v} P(x|\omega_{q,v}) \log P(x|\omega_{i,v}) \mathrm{d}x\big]\end{aligned} \tag{7}$$

where $P(t)$ and $P(v)$ are the class-independent probabilities of drawing textual and visual features respectively.

2.1 Text Retrieval

For the textual part of our ranking function, we use statistical language models. A famous application of these models is Shannon's illustration of the implications of coding and information theory using models of letter sequences and word sequences [18]. In the 1970s, statistical language models were developed as a general natural language processing tool, first for automatic speech recognition [10] and later also for e.g. part-of-speech tagging [4] and machine translation [2]. Recently, statistical language models have been suggested for information retrieval by Ponte and Croft [16], Hiemstra [8] and Miller et al. [13].
The language modeling approach to information retrieval defines a simple unigram language model for each document in a collection. For each document $\omega_{i,t}$, the language model defines the probability $P(x_{t,1}, \cdots, x_{t,N_t}|\omega_{i,t})$ of a sequence of N_t textual features (i.e. words) $x_{t,1}, \cdots, x_{t,N_t}$ and the documents are ranked by that probability. The standard language modelling approach to information retrieval uses a linear interpolation of the document model $P(x_{t,j}|\omega_i)$ with a general collection model $P(x_{t,j})$ [8, 12–14]. As these models operate on discrete signals, the integral from (7) can be replaced by a sum. Furthermore, if we use the empirical distribution of the query as the query model, i.e. if we assume, then the standard textual part of (7) is:

$$\omega_t* \; = \; \arg\max_i \frac{1}{N_t} \sum_{j=1}^{N_t} \log[\lambda P(x_{t,j}|\omega_i) + (1-\lambda) P(x_{t,j})] \tag{8}$$

The linear combination needs a smoothing parameter λ which is set empirically on some test collection, or alternatively estimated by the expectation maximisation (EM)-algorithm [6] on a test collection. The probability of drawing textual feature $x_{t,j}$ from document ω_i ($P(x_{t,j}|\omega_i)$) is computed as follows: if the document contains 100 terms in total and the term $x_{t,j}$ occurs 2 times, this probability would simply be $2/100 =$

0.02. Similarly, $P(x_{t,j})$ is the probability of drawing $x_{t,j}$ from the entire document collection.
Using the statistical language modelling approach for video retrieval, we would like to exploit the hierarchical data model of video, in which a video is subdivided in scenes, which are subdivided in shots, which are in turn subdivided in frames. Statistical language models are particularly well-suited for modelling such complex representations of the data. We can simply extend the mixture to include the different levels of the hierarchy, with models for shots and scenes:[2]

$$\text{Shot}^* = \arg\max_i \frac{1}{N_t} \sum_{j=1}^{N_t} \log[\lambda_{\text{Shot}} P(x_{t,j}|\text{Shot}_i) + \lambda_{\text{Scene}} P(x_{t,j}|\text{Scene}_i) + \lambda_{\text{Coll}} P(x_{t,j})]$$
$$\text{with } \lambda_{\text{Coll}} = 1 - \lambda_{\text{Shot}} - \lambda_{\text{Scene}} \quad (9)$$

The main idea behind this approach is that a good shot contains the query terms and is part of a scene having more occurrences of the query terms. Also, by including scenes in the ranking function, we hope to retrieve the shot of interest, even if the video's speech describes the shot just before it begins or just after it is finished. Depending on the information need of the user, we might use a similar strategy to rank scenes or complete videos instead of shots, that is, the best scene might be a scene that contains a shot in which the query terms (co-)occur.

2.2 Image Retrieval

In order to specialise the visual part of our ranking formula (7), we need to estimate the class conditional densities for the visual features, $P(x_v|\omega_i)$. We follow Vasconcelos [22] and model them using Gaussian mixture models. The idea behind modelling shots as a mixture of Gaussians is that each shot contains a certain number of classes or components and that each sample from a shot (i.e., each block of 8 by 8 pixels extracted from a frame) was generated by one of these components. The class conditional densities for a Gaussian mixture model are defined as follows:

$$P(x_v|\omega_i) = \sum_{c=1}^{C} P(\theta_{i,c})\ \mathcal{G}(x_v, \mu_{i,c}, \Sigma_{i,c}), \quad (10)$$

where C is the number of components in the mixture model, $\theta_{i,c}$ is component c of class model ω_i and $\mathcal{G}(x, \mu, \Sigma)$ is the Gaussian density with mean vector μ and co-variance matrix Σ:

$$\mathcal{G}(x, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}\|x-\mu\|_\Sigma},$$

where n is the dimensionality of the feature space and

$$\|x - \mu\|_\Sigma = (x - \mu)^T \Sigma^{-1} (x - \mu).$$

[2] We assume each shot is a separate class and replace ω_i with Shot_i.

Estimating Model Parameters The parameters of the models for a given shot can be estimated using the EM algorithm. This algorithm iterates between estimating the a posteriori class probabilities for each sample $P(\theta_c|x_v)$ (the E-step) and re-estimating the components parameters (μ_c, Σ_c and $P(\theta_c)$) based on the sample distribution (M-step).[3]

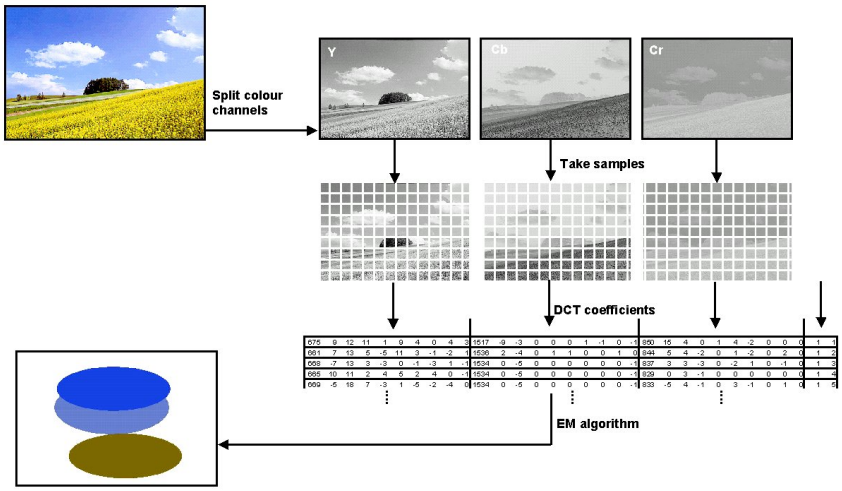


Fig. 2. Building a Gaussian Mixture Model from an Image

The approach is rather general: any kind of feature vectors can be used to describe samples. Our sampling process is as follows, illustrated in Figure 2. First, we convert the keyframe of a shot to the YCbCr color space. Then, we cut it in distinct blocks of 8 by 8 pixels. On these blocks we perform the discrete cosine transform (DCT) for each of the 3 colour channels. We now take the first 10 DCT-coefficients from the Y-channel and only the DC coefficient from both the Cb and the Cr channels to describe the samples. These feature vectors are then fed to the EM-algorithm to find the parameters (μ_c, Σ_c and $P(\theta_c)$). The EM algorithm first assigns each sample to a random component. Next, we compute the parameters (μ_c, Σ_c and $P(\theta_c)$) for each component, based on the samples assigned to that component[4]. We re-estimate the class assignments, i.e. we compute the posterior probabilities ($P(\theta_c|x), \forall c$). We iterate between estimating class assignments (expectation step) and estimating class pa-

[3] Looking at a single shot, we can drop the class subscripts i.

[4] In practice a sample does not always belong entirely to one component. In fact we compute means, covariances and priors on the weighted feature vectors, where the feature vectors are weighted by their proportion of belonging to the class under consideration

rameters (maximisation step) until the algorithm converges. Figure 3 shows a query image and the component assignments after different iterations of the EM algorithm. Instead of a random initialisation, we initially assigned the left-most part of the samples to component 1, the samples in the middle to component 2 and the right-most samples to component 3. This way it is clearly visible how the component assignments move about the image. Finally, after convergence of the EM-algorithm, we describe

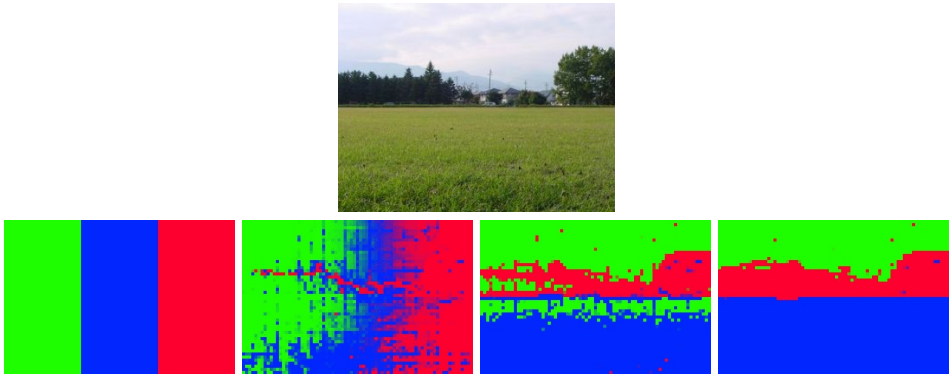


Fig. 3. Class assignments (3 classes) for the image at the top after different numbers of iterations

the position in the image plane of each component as a 2D-Gaussian with mean and covariance computed from the positions of the samples assigned to this component.

Bags of Blocks Just like in our textual approach, for the query model, we can simply take the empirical distribution of the query samples. If a query-image x_v consists of N_v samples: $x_v = (x_{v,1}, x_{v,2}, \ldots, x_{v,N_v})$ then $P(x_{v,i}|\omega_q) = \frac{1}{N_v}$. For the document model, we take a mixture of foreground and background probabilities, i.e. the (foreground) probability of drawing a query sample from the document's Gaussian mixture model, and the (background) probability of drawing it from any Gaussian mixture in the collection. In other words, the query image is viewed as a bag of blocks (BoB), and its probability is estimated as the joint probability of all its blocks. The BoB measure for query images then becomes:

$$\omega_v^* = \arg\max_i \frac{1}{N_v} \sum_{j=1}^{N_v} \log\left[\kappa P(x_{v,j}|\omega_i) + (1-\kappa)P(x_{v,j})\right], \qquad (11)$$

where κ is a mixing parameter and the background probability $P(x_{v,j})$ can be found by marginalising over all M documents in the collection:

$$P(x_{v,j}) = \sum_{i=1}^{M} P(x_{v,j}|\omega_i)P(\omega_i).$$

Again we assume uniform document priors ($P(\omega_i) = \frac{1}{M}$ for all i). In text retrieval, one of the reasons for mixing the document model with a collection model is to assign non-zero probabilities to words that are not observed in a document. Smoothing is not necessary in the visual case, since the documents are modelled as mixtures of Gaussians, having infinite support. Another motivation for mixing is to weight term importance: a common sample x (i.e., a sample that occurs frequently in the collection) has a relatively high probability $P(x)$ (equal for all documents), and therefore $P(x|\omega)$ has only little influence on the probability estimate. In other words, common terms and common blocks influence the final ranking only marginally.

Asymptotic Likelihood Approximation A disadvantage of using the BoB measure is its computational complexity. In order to rank the collection given a query, we need to compute the posterior probability $P(x_v|\omega_i)$ of each image block x_v in the query for each document ω_i in the collection. For evaluating a retrieval method this is fine, but for an interactive retrieval system, optimisation is necessary.
An alternative is to represent the query image, like the document image, as a Gaussian model (instead of by its empirical distribution as a bag of blocks), and then compare these two models using the KL-divergence. Yet, if we use Gaussians to model the class conditional densities of the mixture components, there is no closed-from solution for the visual part of the resulting ranking formula (7). As a solution, Vasconcelos assumes that the Gaussians are well separated and derives an approximation, ignoring the overlap between the mixture components: the asymptotic likelihood approximation (ALA) [22]. Starting from (4) he arrives at:

$$\begin{aligned}
\omega_v^* &= \arg\max_i \int_{\chi_v} P(x_v|\omega_q)\ \log P(x_v|\omega_i)\mathrm{d}x_v \\
&\approx \arg\max_i ALA[P_q(x_v)||P_i(x_v)] \\
&= \arg\max_i \sum_c P(\theta_{q,c})\{\log\ P(\theta_{i,\alpha(c)}) + \log\ \mathcal{G}(\mu_{q,c}, \mu_{i,\alpha(c)}, \Sigma_{i,\alpha(c)}) \\
&- \frac{1}{2} trace[\Sigma_{i,\alpha(c)}^{-1}\Sigma_{q,c}]\}, \\
&\text{where } \alpha(c) = k \Leftrightarrow \|\mu_{q,c} - \mu_{i,k}\|_{\Sigma_{i,k}} < \|\mu_{q,c} - \mu_{i,l}\|_{\Sigma_{i,l}} \forall l \neq k
\end{aligned} \tag{12}$$

In this equation, subscripts indicate respectively classes and components (e.g. $\mu_{i,c}$ is the mean for component θ_c of class ω_i).

2.3 ALA assumptions

The main assumption behind the ALA is that the Gaussians for the components, θ_c, within a class model, ω_i, have small overlap; in fact, there are two parts to this [22]. The first assumption is that each image sample is assigned to one and only one of the mixture components. The second is that samples from the support set of a single query component are all assigned to the same document component. More formally:

Assumption A: For each sample, the component with maximum posterior probability has posterior probability one:

$$\forall \omega_i, x : \max_k P(\theta_{i,k}|x) = 1$$

Assumption B: For any document ω_j, the component with maximum posterior probability is the same for all samples in the support set of a single query component $\theta_{q,k}$:

$$\forall \theta_{q,k}, \omega_j \ \exists l^* \ \forall x : P(x|\theta_{q,k}) > 0 \implies \arg\max_l \ P(\theta_{j,l}|x) = l^*$$

We used Monte Carlo simulation to test these assumptions on our collection (the TREC-2002 video collection, see Section 3.1) as follows. First, we took a random document ω_i from the search collection and then a random mixture component $\theta_{i,k}$ from the mixture model of this document. We then drew 10,000 random samples from this component and for each sample x computed:

- $P(\theta_{i,l}|x)$, the posterior component assignment within document i for all components $\theta_{i,l}$;
- $P(\theta_{j,m}|x)$, the posterior component assignment in a different randomly chosen document j, for all components $\theta_{j,m}$.

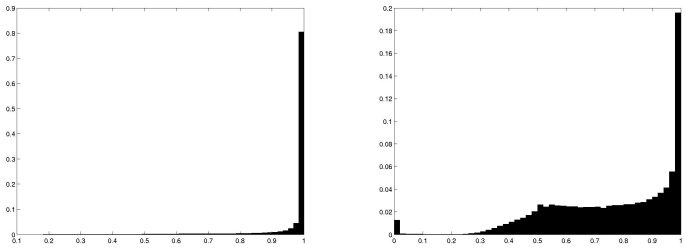
For the first measure we simply took the maximum posterior probability for each sample. We averaged the second measure over all 10,000 samples and took the maximum over all components to approximate the proportion of samples assigned to the most probable component (remember, there should be a component that explains all samples). We repeated this process 100,000 iterations for different documents and components selected at random, and histogrammed the results (Figure 4). Both measures should be close to 1, the first to satisfy assumption A, the second to satisfy assumption B.

As we can see from the plots in Figure 4, the first assumption appears reasonable, but the second does not hold.[5] We investigate the effect of this observation in the retrieval experiments below.

3 Experiments

We evaluated the model outlined above and the presented measures on the search task of the video track of the Text REtrieval Conference TREC-2002 [19].

[5] The bar at probability zero results from a truncation error in the Bayesian inversion to compute $P(\theta_{j,m}|x)$ from a (too small) probability $P(x|\theta_{j,m})$.



$\max_l P(\theta_{i,l}|x)$ $\max_m P(\theta_{j,m}|x)$

Fig. 4. Testing the ALA assumptions A (left) and B (right), samples x drawn from $P(x|\theta_{i,k})$.

3.1 TREC video track

TREC is a series of workshops for large scale evaluation of information retrieval technology [23, 24]. The goal is to test retrieval technology on realistic test collection using uniform and appropriate scoring procedures. The general procedure is as follows:

- A set of statements of information need (topic) is created;
- Participants search the collection and return the top N results for each topic;
- Returned documents are pooled and judged for relavance to the topic;
- Systems are evaluated using the relevance judgements.

The measures used in evaluation are usually precision and recall oriented. Precision and recall are defined as follows:

$$\text{precision} = \frac{\text{number of relevant shots retrieved}}{\text{total number of shots retrieved}}$$

$$\text{recall} = \frac{\text{number of relevant shots retrieved}}{\text{total number of relevant shots in collection}}$$

The video track was introduced at TREC-2001 to evaluate content-based retrieval from digital video [15]. Here, we use the data from the TREC-2002 video track [19]. The track defines three tasks: shot boundary detection, feature detection and general information search. The goal of the shot boundary task is to identify shot boundaries in a given video clip. In the feature detection task, one has to assign a set of predefined features to a shot, e.g. *indoor*, *outdoor*, *people* and *speech*. In the search task, the goal is to find relevant shots given a description of an information need, expressed by a multimedia topic. Both in the feature detection task and in the search task, a predefined set of shots is to be used. In our experiments we focus on the search task.

The collection to be searched in this task consists of approximately 40 hours of MPEG-1 encoded, video, in addition a set of 23 hours of training material is available. The topics consist of a textual description of the

information need, accompanied by images, video fragments and/or audio fragments illustrating what is needed. For each topic a system can return a ranked list of 100 video fragments. The top 50 returned shots of each run are then pooled and judged.

We report experimental results using the standard TREC measures, average precision and mean average precision (MAP):

Average precision: The average of the precision value obtained after each relevant document is retrieved (when a relevant document is not retrieved at all its precision is assumed to be 0)

MAP: The mean of the average precision values over all topics.

For the textual descriptions of the shots, we used speech transcripts kindly provided by LIMSI. These transcripts were aligned to the predefined video shots. We did not have or define a semantic division of the video into scenes, but defined scenes simply as overlapping windows of 5 consecutive shots.[6] We removed common words from the transcripts (stopping) and stemmed all terms using the Porter stemmer [17]. For the visual description we took keyframes from the common video shots and we used EM to find the parameters of Gaussian mixture models. Keyframe selection was straightforward: we simply used the middle frame from each shot as representative for the shot.

3.2 Estimating the Mixture Parameters

The model does not specify the value of mixing parameters λ, λ_{Shot}, λ_{Scene}, and κ. An optimal value can only be found a posteriori by evaluating retrieval performance for different values on a test collection; a priori, we must make an educated guess for the right values.

Figure 5 shows the mean average precision scores on the TREC-2002 video track search task for κ ranging from 0.0 to 1.0. We can see that retrieval results are insensitive to the value of the mixing parameter as long as we take both foreground and background into account. The plot has a similar shape as that found in Hiemstra's thesis for the λ parameter in the standard language model [9].

For the transcripts, we tried over thirty combinations of settings, using two sets of text queries (see also Section 3.4). For query set *Tlong*, this resulted in optimal settings for mean average precision with $\lambda_{\text{Shot}} = 0.090$, $\lambda_{\text{Scene}} = 0.210$, and $\lambda_{\text{Coll}} = 0.700$. Here, modelling the hierarchy in the video makes sense, because shot and scene both contribute to results in the ranking (λ_{Shot} and λ_{Scene} are larger than zero). For set *Tshort* however, the optimal settings had $\lambda_{\text{Shot}} = 0.000$ and the resulting model is identical to the original language model. Summarizing, ranking transcript units longer than shots is important, but we cannot conclude from these experiments whether modeling the hierarchy is really necessary.

In all experiments, the differences between the better parameter choices are not significant, but, a particularly bad choice may seriously degrade retrieval effectiveness. In the remainder of this work, we have used $\kappa = 0.9$, $\lambda_{\text{Shot}} = 0.090$, $\lambda_{\text{Scene}} = 0.210$, and $\lambda_{\text{Coll}} = 0.700$.

[6] In preliminary experiments on the TREC-2001 collection, when varying the window lengths, 5 shots was the optimum.

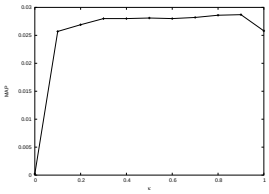


Fig. 5. MAP on video search task for different κ

3.3 Using All or Some Image Examples

In general, it is hard to guess what would be a good example image for a specific query. If we look for shots of the *Golden Gate bridge*, we might not care from what angle the bridge was filmed, or if the clip was filmed on a sunny or a cloudy day; visually however, such examples may be very different (Figure 6). If a user has presented three examples and

Fig. 6. Visual examples of the Golden Gate bridge.

no additional information, the best we can do is try to find documents that describe all example images well. Unfortunately, a document may be

ranked low even though it models the samples from one example image well, as it may not explain the samples from the other images.
For each topic, we computed which of the example images would have given the best results if it had been used as the only example for that topic. We compared these *best example* results to the *full topic* results in which we used all available visual examples. The experiment was done using both the ALA and the BoB measure. In the *full topic* case, the set of available topics was regarded as one large bag of samples. For the ALA measure, we built one mixture model to describe all available visual examples. For BoB, we ranked documents by their probability of generating all samples in all query images. For the single image queries in the *best example*, we built a separate mixture model from each example and used it for ALA-ranking. For BoB ranking, we used all samples from the single visual example. Since it is problematic to use multiple examples in a query, we wanted to see if it is possible to guess in advance what would be a good example for a specific topic. Therefore, for each topic we hand-picked also a single representative from the available examples and compared these *manual example* results to the other two result sets. The results for the different settings are listed in table 1. A first thing to notice is that all scores are rather low. When we take a closer look at the topics with higher average precision scores, we see that these mainly contain examples from the search collection. In other words, we can find similar shots from within the same video, but generalisation is a problem. Comparing BoB to ALA, we see that averaged over all topics, for each set of examples, BoB outperforms ALA. For some specific topics, the ALA gives higher scores, but again these are cases with examples from within the collection. In general, the BoB approach, which uses fewer assumptions performs better.
The fact that using the best image example outperforms the use of all examples shows that indeed combining results from different visual examples can degrade results. Looking at the results, manually selecting good examples seems a non-trivial task, but the drop in performance is partly due to the generalisation problem. If one of the image examples happens to come from the collection it scores high. If we fail to select that particular example, the score for the manual example run drops. Simply counting how often the manually selected example was the same as the best performing example, we see that this was the case for 8 out of 13 topics.[7]

3.4 Using Example Transcripts

We took two different approaches in building textual queries from the multimedia topics. The first set of textual queries, *Thsort*, was constructed simply by taking the textual description from the topic. In the second set of queries, *Tlong*, we augmented these with the speech transcripts from the video examples available for a topic. The assumption here is that relevant shots share a vocabulary with example shots, thus

[7] If we ignore the topics for which there is only one example and the ones for which the best example scored 0.

	full topic		best example		manual example	
Topic	BoB	ALA	BoB	ALA	BoB	ALA
vt075	0.0038	0.0100	**0.2438**	0.0591	**0.2438**	0.0560
vt076	**0.4854**	0.1117	0.4323	0.1327	0.1760	0.0958
vt077	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt078	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt079	0.0000	0.0000	**0.0040**	0.0015	0.0000	0.0000
vt080	0.0048	0.0020	**0.0977**	0.0007	**0.0977**	0.0007
vt081	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt082	**0.0330**	0.0203	0.0234	0.0022	0.0234	0.0022
vt083	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt084	**0.0046**	0.0000	**0.0046**	0.0000	**0.0046**	0.0000
vt085	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt086	0.0053	0.0000	**0.0704**	0.0149	**0.0704**	0.0005
vt087	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt088	0.0046	0.0000	0.0069	**0.0139**	0.0069	**0.0139**
vt089	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt090	0.0000	0.0000	**0.0305**	0.0003	**0.0305**	0.0003
vt091	0.0095	0.0000	**0.0095**	0.0000	**0.0095**	0.0000
vt092	0.0003	0.0000	0.0106	**0.0213**	0.0000	0.0000
vt093	0.0006	0.0000	**0.0006**	0.0003	0.0000	0.0000
vt094	0.0021	0.0004	**0.0021**	0.0013	**0.0021**	0.0013
vt095	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt096	0.0323	0.0000	0.0323	**0.0383**	0.0323	**0.0383**
vt097	0.1312	0.0002	**0.1408**	0.0496	0.0000	0.0000
vt098	0.0000	0.0000	0.0003	**0.0006**	0.0003	0.0000
vt099	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MAP	0.0287	0.0058	**0.0444**	0.0135	0.0279	0.0084

Table 1. MAP for Full Topics, Best Examples and Manual Examples

using example-transcripts might improve retrieval results. In both sets of queries we removed common words and stemmed all terms. We found that across topics, *Tlong* outperformed *Tshort* with a MAP of 0.1212 against 0.0916. For detailed per topic information see Table 2.

3.5 Combining Textual and Visual runs

We combined textual and visual runs using our combined ranking formula (7). Since we had no data to estimate the parameters for mixing textual and visual information we used $P(t) = P(v) = 0.5$. For the textual part we tried both short and long queries, for the visual part we used full queries and best-example queries. Table 2 shows the results for combinations with the BoB measure. We also experimented with combinations with the ALA measure, but we found that in the ALA case it is difficult to combine textual and visual scores, because they are on different scales. The BoB measure is closer to the KL-divergence and, on top of that, more similar to our textual approach, and thus easier to combine with the textual scores.
For most of the topics, textual runs give the best results, however for some topics using the visual examples is useful. This is mainly the case when either the topics come from the search collection or when the relevant documents are outliers in the collection. This illustrates how difficult it is to search a generic video collection using visual information only. We only succeed if the relevant documents are either highly similar to the examples provided or very dissimilar from the other documents in the collection (and therefore relatively similar to the query examples). When both textual and visual runs have reasonable scores, combining the runs can improve on the individual runs, however, when one of them has inferior performance, a combination only adds noise and lowers the scores.

4 Conclusions

We presented a probabilistic framework for multimodal retrieval in which textual and visual retrieval models are integrated seamlessly and evaluated the framework using the search task from the TREC-2002 video track. We found that even though the topics were specifically designed for content-based retrieval, and relevance was defined visually, a textual search outperforms visual search for most topics. As we have seen before [20], standard image retrieval techniques can not readily be applied to satisfying a variety of information requests from a generic video collection. Future work has to show how incorporating different sources of additional information (e.g. contextual frames, the movement in video or user interaction) can help improve results.
In the text-only experiments, we saw that using the transcripts from the example videos in queries improves results. We also found that it is useful to take transcripts from surrounding shots into account to describe a shot. However, it is still unclear whether a hierarchical description of scenes and shots is necessary.

Topic	Tshort	Tlong	BoBfull	BoBbest	BoBfull +Tshort	BoBfull +Tlong	BoBbest +Tshort	BoBbest +Tlong
vt075	0.0000	0.0082	0.0038	0.2438	0.0189	0.0569	0.2405	**0.3537**
vt076	0.4075	0.6242	0.4854	0.4323	0.5931	**0.7039**	0.5757	0.6820
vt077	0.1225	**0.5556**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt078	0.1083	**0.2778**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt079	0.0003	0.0006	0.0000	0.0040	0.0003	0.0000	**0.0063**	0.0050
vt080	0.0000	0.0000	0.0048	**0.0977**	0.0066	0.0059	0.0845	0.0931
vt081	0.0154	0.**0333**	0.0000	0.0000	0.0037	0.0000	0.0000	0.0000
vt082	0.0080	0.0262	0.0330	0.0234	0.0181	**0.0335**	0.0145	0.0210
vt083	**0.1669**	**0.1669**	0.0000	0.0000	0.0962	0.0962	0.0078	0.0078
vt084	**0.7500**	**0.7500**	0.0046	0.0046	0.6875	0.6875	0.6875	0.6875
vt085	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
vt086	0.0554	0.0676	0.0053	0.0704	0.0536	0.0215	**0.0791**	0.0600
vt087	**0.0591**	0.0295	0.0000	0.0000	0.0052	0.0003	0.0052	0.0003
vt088	**0.0148**	0.0005	0.0046	0.0069	0.0052	0.0046	0.0069	0.0069
vt089	**0.0764**	**0.0764**	0.0000	0.0000	0.0503	0.0503	0.0045	0.0045
vt090	0.0229	0.0473	0.0000	0.0305	0.0006	0.0075	0.0356	**0.0477**
vt091	0.0000	0.0000	**0.0095**	**0.0095**	0.0000	0.0086	0.0000	0.0086
vt092	0.0627	**0.0687**	0.0003	0.0106	0.0191	0.0010	0.0078	0.0106
vt093	**0.1977**	0.1147	0.0006	0.0006	0.0099	0.0021	0.0071	0.0012
vt094	0.0232	**0.0252**	0.0021	0.0021	0.0122	0.0036	0.0122	0.0036
vt095	**0.0034**	0.0021	0.0000	0.0000	0.0008	0.0012	0.0011	0.0010
vt096	0.0000	0.0000	**0.0323**	**0.0323**	0.0161	0.0161	**0.0323**	**0.0323**
vt097	0.1002	0.0853	0.1312	0.1408	0.1228	**0.1752**	0.1521	0.1474
vt098	**0.0225**	0.0086	0.0000	0.0003	0.0068	0.0000	0.0004	0.0003
vt099	**0.0726**	0.0606	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MAP	0.0916	**0.1212**	0.0287	0.0444	0.0691	0.0750	0.0784	0.0870

Table 2. Average precision per topic, for Textual runs, BoB runs and combined runs

In our visual experiments, we found that the general probabilistic framework is useful for image retrieval. However, we found that one of the assumptions underlying the Asymptotic Likelihood approximation of the KL-divergence does not hold for the generic video collection we used. This was reflected in the difference in performance of the ALA and the Bag of Blocks model. Unfortunately, computing the joint block probabilities in the BoB model is computationally expensive and unsuitable for an interactive retrieval system. Future work will investigate ways to speed up the process.
Furthermore, we noticed generalisation problems. The visual models only gave satisfying results if the relevant documents were either highly similar to the query image(s) (i.e. the query images came from the collection), or highly dissimilar to the rest of the collection (i.e. the relevant documents were outliers in the collection).
When either textual or visual results are poor, combining them, thus adding noise, seems to degrade the scores. However, when both modalities yield reasonable scores, a combined run outperforms the individual runs.

References

1. K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *International Conference on Computer Vision (ICCV 2001)*, pages 408–415, 2001.
2. P. F. Brown, J. C. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.
3. M. L. Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-base d image retrieval on the world wide web. In *Proceedings*, pages 24–28, 1998.
4. D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Proceedings of Applied Natural Language Processing*, pages 133–140, 1992.
5. F. de Jong, J.-L. Gauvain, D. Hiemstra, and K. Netter. Language-based multimedia information retrieval. In *Content-Based Multimedia Information Access, RIAO 2000 Conference Proceedings*, pages 713–722, 2000.
6. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, series B*, 39(1):1–38, 1977.
7. J.-L. Gauvain, L. Lamel, and G. Adda. Transcribing broadcast news for audio and video indexing. *Communications of the ACM*, 43, 2000.
8. D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 569–584, 1998.
9. D. Hiemstra. *Using language models for information retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001.

10. F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.
11. G. Jones, J. Foote, K. S. Jones, and S. Young. The video mail retrieval project: experiences in retrieving spoken documents. In M. T. Maybury, editor, *Intelligent Multimedia Information Retrieval*. AAAI press, 1997.
12. J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*, pages 111–119, 2001.
13. D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 214–221, 1999.
14. K. Ng. A maximum likelihood ratio information retrieval model. In *Proceedings of the eighth Text Retrieval Conference, TREC-8*. NIST Special Publications, 2000.
15. P. Over and R. Taban. The trec-2001 video track framework. In Voorhees and Harman [24].
16. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 275–281, 1998.
17. M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
18. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.
19. A. F. Smeaton and P. Over. The trec-2002 video track report. In Voorhees and Harman [23].
20. The Lowlands team. Lazy users and automatic video retrieval tools in (the) lowlands. In *The 10th Text Retrieval Conference (TREC-2001)*, 2002.
21. T.Westerveld. Image retrieval: Content versus context. In *Content-Based Multimedia Information Access, RIAO 2000 Conference Proceedings*, pages 276–284, 2000.
22. N. Vasconcelos. *Bayesian Models for Visual Information Retrieval*. PhD thesis, Massachusetts Institut of Technology, 2000.
23. E. M. Voorhees and D. K. Harman, editors. *The Eleventh Text REtrieval Conference (TREC-2002)*, 2002.
24. E. M. Voorhees and D. K. Harman, editors. *The Tenth Text Retrieval Conference (TREC-2001)*, volume 10. National Institute of Standards and Technology, NIST, 2002, to appear.
25. T. Westerveld. Probabilistic multimedia retrieval. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR2002)*, pages 437–438, 2002.
26. T. Westerveld, A. P. de Vries, and A. van Ballegooij. CWI at the TREC-2002 Video Track. In Voorhees and Harman [23].