CONTENT-BASED PRE-INDEXED VIDEO

Nuno Vasconcelos *Andrew Lippman*

MIT Media Laboratory
{nuno,lip}@media.mit.edu

ABSTRACT

The viability of large distributed image databases is strongly dependent on the development of new image representations capable of providing support for extended functionality, directly in the compressed domain. We have recently introduced one such representation (Library-based coding) which we now augment with statistical pre-indexing schemes, automatically built at the time of encoding, that provide several layers of content description allowing efficient content-based retrieval and summarization.

1. INTRODUCTION

The explosion in availability of image and video content, due to the high interconnectivity of the new digital media and recent developments in multimedia technology, demands the formulation of powerful paradigms for automated content-based retrieval using image cues. The practical success of such paradigms requires, however, that they lead to algorithms characterized by high retrieval accuracy and computation efficiency. Because pictorial content is typically stored in a compressed format, efficiency is improved if content queries can be performed directly in the compressed domain. On the other hand, because sophisticated content characterization is required for both accurate retrieval and efficient compression, it is only natural to expect that each of these tasks can benefit from the modeling effort required by the other.

It is, therefore, our belief that the two problems should be viewed as components of a larger goal: the design of image representations capable of supporting high level functionality (such as content-based retrieval, or interactive image manipulation) directly in the compressed domain, without significant cost in terms of compression efficiency. With this objective in mind we have recently introduced *Library-based Coding* (LBC), a representation based on the idea of embedding a probabilistic description of the source in the compressed bitstream. This not only allows retrieval by statistical inference without decompression but, being closely related to standard *vector quantization* (VQ) compression techniques, also leads to a bandwidth efficient representation.

The foundations of LBC were already presented in [3, 5, 4]. In this paper, we exploit the statistical nature of the representation to develop pre-indexing schemes for video which are automatically built at the time of encoding. These schemes are based on hierarchical video information structures which provide several layers of content description, leading to computationally efficient query procedures and enabling scalable queries where the amount of decoding is proportional to the coarseness desired for retrieval. As a demonstration of this capability we are currently developing schemes for video *cross-indexing* by scene similarity, which can be used for applications such as video browsing or monitoring of broadcast channels.

2. LIBRARY-BASED CODING AND RETRIEVAL

From an image retrieval perspective, embedding a probabilistic description of an image source in its compressed bitstream has various advantages. First, it enables retrieval through sound statistical procedures, such as those based on Bayesian inference. For example, given the densities associated with M image sources $P(\mathbf{x}|\mathcal{S}_i)$, $i = 1, \ldots, M$, the source probabilities $P(\mathcal{S}_i)$, and a query image $\mathbf{x}_e$, example-based retrieval can be achieved by determining which source is most likely given the query, i.e. finding

$$\mathcal{S}_i = \arg\max_i P(\mathcal{S}_i|\mathbf{x}_e) = \arg\max_i P(\mathbf{x}_e|\mathcal{S}_i)P(\mathcal{S}_i). \quad (1)$$

Also, if the probabilistic descriptions are compact, only a small fraction of the bitstream must be decoded for retrieval, allowing computationally efficient queries. Furthermore, because estimating probabilities is in general helpful for the coding process itself, support for retrieval can be attained without compromise of coding efficiency. LBC exploits the relationships between mixture density estimation and VQ to achieve these objectives.

2.1. Gaussian mixtures, the EM algorithm, and vector quantization

Mixture densities are a class of parametric probabilistic models capable of approximating any probability density. In this work, we consider the particular class of Gaussian mixtures, characterized by

$$P(\mathbf{x}) = \sum_{i=1}^{C} \frac{p_i}{K_i} e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mu_i)}, \quad (2)$$

where p_i is the *prior* probability of mixture component, or class i, μ_i and $\mathbf{\Sigma}_i$ the associated mean and variance, and $K_i = \sqrt{(2\pi)^n|\Sigma_i|}$. Given a dataset $\mathbf{x}^m$, $m = 1, \ldots, M$ the parameters of the mixture model can be estimated through the Expectation-Maximization (EM) [1] algorithm, which alternates between the computation of the *posterior* probabilities of the mixture components given the data and current parameter estimates (E-step), and the update of these estimates (M-step). For the Gaussian case, assuming a priori equally likely classes ($p_i = 1/C$) and unit covariances, these steps resume to

E-step: **M-step:**

$$h_i^m = \frac{e^{-\frac{1}{2}||\mathbf{x}^m - \mu_i||^2}}{\sum_{k=1}^N e^{-\frac{1}{2}||\mathbf{x}^m - \mu_k||^2}}, \quad (3) \qquad \mu_i^{new} = \frac{\sum_m h_i^m \mathbf{x}^m}{\sum_m h_i^m}, \quad (4)$$

where h_i^m is the *posterior* probability that the m^{th} sample belongs to class i.

If the *soft-decisions* of equation (3) are replaced by the greedy *hard-decision* of choosing the mixture with higher posterior probability h_i^m, the expectation step becomes

$$h_i^m = \begin{cases} 1, & \text{if } ||\mathbf{x}^m - \mu_i||^2 \leq ||\mathbf{x}^m - \mu_j||^2, \ \forall j \neq i \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

and, consequently, the maximization step simply computes the centroids of the samples assigned to each of the classes. This is the well known *LBG* [2] algorithm for VQ design which can, therefore, be seen as a greedy version of the EM algorithm. Thus, in practice, VQ codebooks provide approximate estimates for the parameters of the Gaussian mixture model[1]. The significance of this result is that it allows the combination of the compression efficiency of VQ, with the ability to support efficient retrieval by embedding an explicit probabilistic description of the source in the compressed bitstream.

2.2. Library-based coding

LBC builds on the relationships between EM and VQ design to achieve high efficiency in both the tasks of compression and retrieval. For each frame, a codebook is designed by either the EM or the LBG algorithm and transmitted to the receiver. This codebook is then used as an extra prediction mode for a standard MPEG coder. Each block is vector quantized and, if the resulting quantization error is smaller than the prediction error due to motion compensation, the vector quantized block is used as prediction. Even though VQ could be used by itself, we prefer embedding it in a predictive structure as this increases coding efficiency, and maintains the representation closer to existing standards [3, 5].

For retrieval, we simply compare the library of the query frame to the libraries of each of the frames in the bitstream. A number of metrics can be used for this comparison, including metrics of the distance between the associated probability densities (such as the Kulback-Leibler divergence), or the likelihood of the entries in the query library being originated by the source of the frame in the bitstream. We use the latter because 1) it approximates $P(\mathbf{x}_e|\mathcal{S}_i)$ in equation (1) and, therefore, provides support for Bayesian inference, and 2) it is computationally efficient. In fact, it can be shown that, under the assumption of small overlap between the mixture components, this metric is simply the mean square distance between libraries [4] and, in fact, this was the metric that we used in the experiments reported on section 5.

[1]See [4] for a more detailed derivation of this relationship which includes the treatment of Gaussian mixtures with unrestricted covariances and prior class probabilities.

3. PRE-INDEXED VIDEO

Because library-encoded bitstreams contain an embedded compact description of their content (which is decodable independently of the bulk of the bitstream and, therefore, searchable on its own), LBC can be seen as a way of *pre-indexing* the compressed bitstreams for retrieval purposes. Making an analogy to text-based retrieval systems, libraries can be seen as *image keywords*, and our implementation of library-based retrieval as search by *keyword matching*. From the standpoint of computational efficiency this type of retrieval is significantly more effective than approaches which require image decoding.

Nevertheless, the simple pre-indexing strategy of assigning a library to each frame, is still unsatisfactory for video applications. Reasons for this are that

- there are simply too many frames to search from on a typical video sequence, making a frame-based query overwhelming even if individual queries are efficient;
- given a query frame, one is generally satisfied by, or even only interested in, retrieving the shot that most likely contains it, not the isolated frame that best matches it;
- one is typically more interested in providing as example query a video shot or group of shots than a single frame.

Efficient video-indexing procedures therefore need to rely on more sophisticated indexing structures, capable of modeling the temporal coherence of video-based content: that nearby frames typically belong to the same shot, nearby shots to the same scene, etc.

The probabilistic nature of the content description provided by the libraries makes them natural building blocks for the construction of such indexing structures. After all, by merging the blocks of several frames and computing the associated library, one still obtains a valid probabilistic description in the space of image blocks.

While individual frame libraries simply describe how the blocks in the associated frames populate the space, libraries computed from groups of frames also capture the coherence of the paths followed by these blocks in the space as the sequence progresses. One can, therefore, compute *shot-libraries* or *scene-libraries* with the procedures used for the computation of frame libraries, and use these libraries and the library matching methods of section 2.2 as a basis for *shot-retrieval* or *scene-retrieval*.

Furthermore, because all the layers in this library hierarchy are probabilistic descriptors in the same space, it is possible to combine information from various layers to increase the efficiency or the functionality of retrieval. Because there is no requirement in equation (1) that $\mathbf{x}_e$ and $\mathcal{S}_i$ belong to the same level in the hierarchy, one can ask questions such as "what is the most likely shot or scene source to have originated this image?". Answers to these types of questions not only expand the vocabulary of the retrieval system, but also enable significantly more efficient generic searches that progress from the higher to the lowest layers in the hierarchy.

4. HIERARCHICAL PRE-INDEXING

Our video pre-indexing scheme builds on the observations of the previous section. We start by defining a library hierarchy, where libraries in the upper levels provide a description of the input space that is coarser than that associated with their children. Fig. 1 presents the tree structure which characterizes this hierarchy. While the leaves of the tree consist of the frame libraries of section 2.2, higher levels are composed by *macro-libraries* obtained from the composition of these libraries.

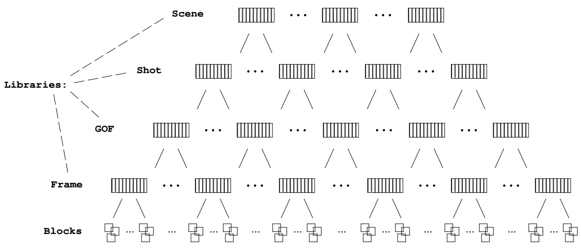

Figure 1: Tree structure which characterizes our hierarchical video pre-indexing.

Macro-libraries of different levels are content descriptors for nested partitions of the video stream: group-of-frames (GOF) libraries are obtained by composing frame libraries, shot libraries by composing GOF libraries, scene libraries by composing shot libraries, and so on. While this structure can theoretically be extended to arbitrarily high-level scene groupings, we currently stop at the shot level. To avoid both the massive storage requirements and complexity associated with the design of libraries based on all the image blocks in a GOF or shot, we rely on an approximation which consists in using as dataset for the design of the libraries of a given level only the blocks in the libraries of the level immediately below.

5. RETRIEVAL EXPERIMENTS

In this section we report results of a series of experiments designed to test the retrieval efficiency of a system based on hierarchical pre-indexing. In these experiments we addressed two particular issues

- would the computational efficiency achieved through the hierarchical search lead to a significant degradation in retrieval accuracy?
- how good would be the characterization of the video content provided by the GOF and shot libraries?

These questions are, in fact, related because if the characterization provided by the higher-level libraries were good, one would expect only a marginal decrease associated with hierarchical retrieval.

In all the experiments, libraries were designed with a variant of the LBG algorithm [3], typical GOFs contained 15 pictures, and shot boundaries were detected using a simple strategy that signaled a boundary if the number of blocks for which motion compensated prediction was not chosen was below a pre-defined threshold. GOFs were not allowed to span across shot boundaries, i.e. a new shot always implied the start of a new GOF.

All experiments were conducted on a dataset containing 1660 frames of video extracted from a trailer of the movie "Terminal Velocity", where 69 different shots were identified, leading to 148 GOFs. In addition to the natural grouping of GOFs into shots, the GOFs were also manually clustered into groups that were perceptually similar. The definition of similar was quite strict, basically only shots that reported to the same scene conditions were clustered into the same group[2].

5.1. Computational efficiency

Table 1 presents the retrieval accuracy and computational savings obtained through hierarchical searching on an experiment where each frame in the dataset was used as example query. Because the query libraries are included in the retrieval database, simple library-based retrieval (LBR) will always return the correct match. Hence, the percentage of correct matches using hierarchical retrieval is a measure of how much of the retrieval efficiency is retained by pre-indexing.

Search Mode	Comp. Savings	Retrieval Accuracy
LBR	0 %	100 %
Hier 1	89.8 %	99.5 %
Hier 2	94.5 %	92.4 %

Table 1: Computational savings and retrieval accuracy achieved with hierarchical retrieval.

Two hierarchical search modes were defined, differing on the level of the library hierarchy at which the search begins. For mode 1, the example library is first matched to the GOF libraries and the query is then matched to the libraries of the frames in the closest GOF. Mode 2 uses the three library layers, considering first shot libraries, then GOF libraries, and finally frame libraries. The results in the table show that hierarchical search enables significant computational savings with only marginal decrease in retrieval accuracy. The accuracy drop from mode 1 to mode 2 is due mostly to incorrect shot segmentation. Whenever a shot boundary is missed, a single shot library is designed for two adjacent shots leading to a poor characterization of their content. While this is a problem that cannot be completely eliminated (as 100 % correct segmentation is unlikely to be achieved), the accuracy of mode 2 can be improved through the use of more sophisticated shot segmentation algorithms.

5.2. Effectiveness of higher order libraries

A second set of experiments was designed to determine how well the higher-level libraries characterize the video content

[2]The objective of this step was to achieve a ground truth for retrieval that would lead to the same retrieval results if two (or more) types of shots were interleaved (as commonly occurs when, for example, the image switches between close-ups of two subjects, during a dialog).

of the associated GOFs, shots and groups of GOFs. For this, each GOF library was used as example query, and the best M matches determined. The following metric,

$$\mathcal{R} = \frac{\#\text{ queries returning} \geq r \text{ GOFs in correct group}}{\#\text{ GOFs in groups with} \geq r+1 \text{ libraries}}$$

was then computed[3]. In this metric, r determines how stringent is the criteria for declaring a successful query, and the metric itself indicates the degree of success for the ensemble of all queries. E.g. $r = 1$ indicates that returning one library in the same group as the query is enough, while $r = M$ indicates that success is declared only when all the M matches are in the right group.

Table 2 presents the retrieval efficiency as a function of r and M, for three levels of strictness. As the table shows,

	Number of Matches (M)				
r	1	3	5	7	9
1	85.8	92.2	93.6	94.3	95.0
0.5 **M**	85.8	92.2	78.7	79.5	89.6
0.75 **M**	85.8	77.2	77.7	67.0	54.1

Table 2: Percentage of queries satisfying different levels of strictness (**r**) for success in **M** matches, as a function of **M**.

if the best three (or more) matches are retrieved from the database, there is at least a 92.2% chance of finding a match from the same group as the query GOF. On the other hand, if nine or more matches are retrieved from the database, at least four of these will belong to the correct group in 89.6% (or more) of the trials. The high accuracy demonstrated by these results supports the conclusion that the GOF libraries provide a good model of the GOF content, for retrieval purposes. Unfortunately, there are not enough shots in our database to perform a similar analysis at the shot or higher levels[4].We are currently in the process of applying LBC to larger movie databases, where such experiments will be possible.

6. REFERENCES

[1] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from Incomplete Data via the EM Algorithm. *J. of the Royal Statistical Society*, B-39, 1977.

[2] Y. Linde, A. Buzo, and R. Gray. An Algorithm for Vector Quantizer Design. *IEEE Trans. on Communications*, Vol. 28, January 1980.

[3] N. Vasconcelos. Library-based Image Coding using Vector Quantization of the Prediction Space. Master's thesis, Massachusetts Institute of Technology, 1993.

[4] N. Vasconcelos and A. Lippman. Library-based Coding: a Representation for Efficient Video Compression and Retrieval. In *Proc. Data Compression Conference, Snowbird, Utah*, 1997.

[5] N. Vasconcelos and A. Lippman. Libray-based Image Coding. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Adelaide, Australia, 1994.

[3]In this expression # means "number of" and $\geq$ "more than".

[4]Even at the GOF level, the sudden performance decrease for $\mathbf{M} \geq 7$ and $\mathbf{r} = 0.75\mathbf{M}$ indicates that, in these cases, we are near the region where our sample becomes to small to have statistical significance.