A Compression Framework for Content Analysis

Trish Keaton *
Dept. of Electrical Engineering (136-93) ∗ †
California Institute of Technology
Pasadena, CA 91125
{keaton,rogo}@micro.caltech.edu

Rodney Goodman †
Information Sciences Lab (RL69)∗
HRL Laboratories, LLC
Malibu, CA 90265
keaton@isl.hrl.hac.com

Abstract

We present a statistical coding framework that supports content analysis and retrieval in the compressed domain. An unsupervised learning approach based upon latent variable modeling is adopted to learn a collection, or mixture, of local linear subspaces that are designed for compression, while providing a probabilistic model of the source useful for inferring image content. The compressed bitstream is organized to enable the progressive decoding of the compressed data, such that the bitstream is only decompressed up to the level necessary to satisfy the query. We describe methods of extracting relevant features from the compressed representation that support query based on single and multiple example images, high level class categories such as people, and low-level features like particular colors and textures. Retrieval experiments have shown that this representation provides good inferencing with very little decompression.

1. Introduction

With the improvement of internet communications, digital image and video libraries are becoming more readily available, thereupon generating the need for fast and efficient methods to store, and retrieve the visual information. Until recently, the processes of data compression and content analysis were considered independently. Early efforts in the area of content-based retrieval attempted to extend traditional database techniques to support multi-media data storage, management and retrieval. Such systems stored meta-data in the form of text annotations or keywords assigned by a human operator, or relied upon the extraction of visual features (e.g. edges, color, etc.) stored in addition to the compressed imagery. These methods are inefficient from a compression perspective, since they entail full decompression of the data for image domain feature analysis, and their indexing schemes often produce a data expansion. From a recognition perspective, these methods restrict the types of queries that can be made by deriving the feature sets and descriptors *a priori*. Even if the set of image attributes extracted for indexing is rich, users of such systems are required to specify at the time of query which attributes are important and their desired range of values (e.g., percentage of the color red contained in the image). Clearly, since both compression and content indexing rely on efficient information extraction, an optimal solution to the problem of content-based retrieval should entail the joint optimization of both processes.

1.1. Previous Work

Prior methods of recognition or content-based retrieval in the compressed domain have generally taken two approaches. In the first approach, existing compression standards such as JPEG and MPEG are extended to support content-based analysis by deriving features and models from the transform coefficients [6]. This approach generally leads to an expansion of the data, and requires either partial or full decompression of the bitstream. An alternative approach explored by Gray *et al.* [4] involves a method of designing a vector quantizer for both classification and compression by incorporating a weighted Bayes risk component into the distortion measure used to design the code. Unfortunately at high vector dimensions (i.e., 8×8 blocks), the computation associated with using a vector quantizer grows exponentially, and thus its use is limited to very small vector dimensions (i.e., 2×2 blocks). In our work, we take an intermediate approach by using an existing compression algorithm that is well-suited for content analysis, and modify its design criteria to improve its retrieval performance.

Shannon's coding theory tells us how to design efficient codes when the distribution defining the information source is given. However, the source distribution is generally unknown, and instead must be estimated from the observed data. Therefore, coding efficiency will depend upon how

*T. Keaton is supported by a Doctoral Fellowship with the Information Sciences Lab, HRL Laboratories, LLC, Malibu, CA 90265.

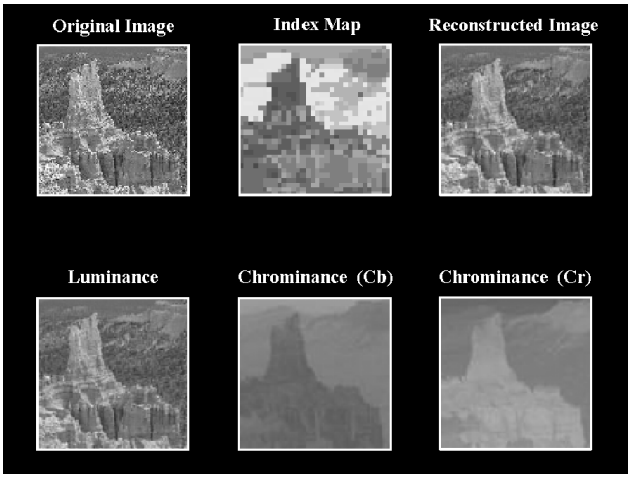

Figure 1. Reconstructed image that was compressed to 0.4 bpp.

well the estimated model matches the distribution of the data-generating source. Universal coding theory attempts to improve practical code design by assuming that the data is generated by a distribution in a class of sources, rather than a specific source distribution. In practice, the approach employs a two-stage structure in which the single source code of traditional image compression systems is replaced with a family of codes designed to cover a large class of possible sources. In the first stage, the encoder describes the code from its collection considered to be the "best" from a compression standpoint. In the second stage, the encoder describes the data using the selected code. This general coding strategy was shown in [2] to give good compression performance in applications where the statistics of the source are not available at design time or may vary over time and space.

Pattern recognition systems also rely upon the design of good inferencing models derived from the statistics of the data. For the problem of content-based retrieval, we can expect image and video libraries to contain images of different types with varying statistics where different image clusters are better represented by a class of models. Thus, a new universal source coding technique modified to define a proper probability density model of the data would be well-suited for the design of efficient codes from which recognition in the compressed domain can be achieved.

In [2], Effros and Chou introduced a two-stage universal transform code called the Weighted Universal Transform Code (WUTC). The algorithm is based upon the Karhunen-Loeve Transform (KLT) which is a data-dependent transform that achieves optimal decorrelation and energy compaction. By replacing JPEG's single, non-optimal transform code with a collection of optimal transform codes, the WUTC algorithm achieved up to 3 db performance improvement over JPEG. In [5], we investigated the direct application of the WUTC algorithm for the efficient coding of image libraries, and discussed its inadequacies in terms of statistical modeling. The partitions formed using the expected error in reconstruction criterion are not optimal, since the clustering is done independently of the KLT projection. Proper clustering should include a distance within the subspace, and since the KLT does not define a proper density model for the data, it cannot model the off-subspace noise nor the in-subspace variability which are necessary for achieving good classification. Recently, projection methods based upon Gaussian latent variable modeling have been proposed [3], which overcome these deficiencies by deriving the transformation basis within a maximum-likelihood framework.

1.2. Our Approach

In this paper, we propose a universal statistical coding framework based upon Gaussian latent variable modeling for learning a collection of 8×8 block transform codes, derived from a training set comprised of representative samples from the database. For color imagery, a set of codes is learned separately for the luminance, and both chrominance bands of the YCbCr format. By allowing a collection of bases to be learned, each can become specialized to a larger variety of structures present in the data ensemble. We find that the final subspaces learned are color and texture selective, for example, skin-like regions are encoded using the same group of transform codes. Since the transform codes are derived within a maximum-likelihood framework, the partitioning of the data and estimation of the basis vectors are combined, where likelihood replaces the squared reconstruction error as the code selection criteria. Using this coding strategy, we achieved a 17% gain in classification performance over WUTC, with less than 1 db increase in distortion for all rates.

In local transform coding, we wish to allocate bits to different coefficients based on the variances of the components. This strategy effectively prunes components having a very low variance. We utilize different bit allocations for each transform basis. The quantized data is then entropy encoded using an arithmetic encoder. Figure 1 shows a reconstructed image after compression to 0.4 bits-per-pixel using our statistical coder.

The compressed representation is structured to permit the progressive independent decoding of the coefficients leading to efficient and successively refinable methods of query and retrieval. We utilize the information provided by the index map indicating which transform codes were used to encode each image block, to derive features useful for determining the similarity between images. Our coding strategy permits quick searches through large database populations without

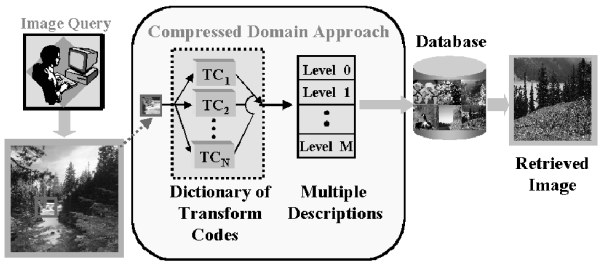

Figure 2. A block diagram of the system.

requiring the additional storage of content descriptors, or fully decompressing the bitstream, while still supporting conventional query techniques. A block diagram of the proposed system is shown in Figure 2.

This paper is organized as follows. In section 2, we introduce the latent variable modeling framework for learning a mixture of local linear subspaces. Section 3 explains how we can perform content analysis and similarity matching from the resulting compressed representation. In Section 4, we describe our query processing engine which is based upon Bayesian evidential reasoning. The latter sections present preliminary experimental results and conclusions.

2. Latent Variable Modeling

Latent variable modeling for data reduction assumes that the high-dimensional observed space **x** is generated from an underlying low-dimensional process defined by a linear transformation of a small number of latent variables, or hidden causes, **z**, plus additive noise **u**: **x = Λ z + u**, where the columns of Λ are the basis functions. The latent variable model is specified by the prior distributions of the latent space **p(z)** and the noise model **p(u)**, and the linear mapping Λ from latent space to data space. The model concurrently performs the two steps of data partitioning and reduction by inferring the state of the latent variables, or transform coefficients, **z**, using a maximum *a posteriori* criterion, then adapting the basis to obtain a good model of the data space distribution. Maximum likelihood estimation is used to optimize the parameters of the model.

2.1. Mixtures of Factor Analyzers

Factor analysis [3] is a latent variable method for modeling the covariance structure of high dimensional data using a small number of latent variables called factors, where Λ is known as the factor loading matrix. The factors **z** are assumed to be independent and Gaussian distributed with zero-mean unit variance, $\mathbf{z} \sim \mathcal{N}$ **(0,I)**. The additive noise **u** is also normally distributed with zero-mean and a diagonal covariance matrix Ψ, $\mathbf{u} \sim \mathcal{N}$ **(0,Ψ)**. Hence, the observed variables are independent given the factors, and **x** is therefore distributed with zero mean and covariance $\Lambda'\Lambda + \Psi$. The goal of factor analysis is to find the Λ and Ψ that best model the covariance structure of **x**. The factor variables **z** model correlations between the elements of **x**, while the **u** variables account for independent noise in each element of **x**. Factor analysis defines a proper probability density model over the observed space, where different regions of the input space are locally modeled by assigning a different mean μ_j, and index ω_j (where j = 1,...,M), to each factor analyzer.

2.2. EM Learning of Model Parameters

The EM learning algorithm can be used to learn the model parameters without the explicit computation of the sample covariance which greatly reduces the algorithm's computational complexity:

E-Step: Compute the moments $h_{ij} = E[\omega_j|x_i]$, $E[z|x_i,\omega_j]$, and $E[zz'|x_i,\omega_j]$ for all data points *i* and mixture components *j* given the current parameter values Λ_j, and Ψ_j.

M-Step: This results in the following update equations for the parameters:

$$\widetilde{\Lambda}_j^{new} = \left(\sum_i h_{ij} x_i E[\widetilde{z}|x_i,\omega_j]'\right)\left(\sum_i h_{ij} E[\widetilde{z}\widetilde{z}'|x_i,\omega_j]\right)^{-1}$$

$$\widetilde{\Psi}_j^{new} = \frac{1}{n} diag\left\{\sum_{ij} h_{ij}(x_i - \widetilde{\Lambda}_j^{new} E[\widetilde{z}|x_i,\omega_j])x_i'\right\}$$

Details on the derivation of these update equations can be found in [3]. We iterate between the two steps until the model likelihood is maximized. Since our application is image coding, the final step assigns an image block to the mixture component yielding the lowest reconstruction error.

3. Similarity Matching Based on Transform Code Usage

We perform quick searches on the compressed imagery by only decompressing the information provided by the index map indicating which transform codes were used to encode each image block. Since we encode the luminance and chrominance bands separately, the index uniquely identifies which transform code was used to encode each band. Figure 1 shows the index map representation of an image. We see that our encoder has segmented the image into regions of similar color and texture composition, thus providing a useful representation for content-based retrieval. We can determine the similarity between images by computing first and second order statistical features derived from the index map representation.

Color histogram matching is one of the most popular image retrieval approaches because it has shown good recognition performance without the need for object extraction,

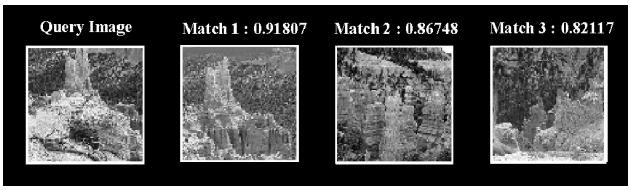

Figure 3. An example of a correct retrieval.

and for low dimensional spaces it is easy to compute. We adopt a similar approach for image matching by computing histograms of the indices of transform codes used to encode an image. Given a query image, its code usage histogram is computed and matched to histograms of database images. The best matching histogram is selected and its class label is considered the "recognized" classification. The histograms are normalized and compared using relative entropy or the Kullback-Leibler divergence which computes the distance between two distributions. Figure 3 shows an example of a correct retrieval result obtained using this method of classification. The performances obtained with this method are remarkably robust. Objects are recognized despite changes in size and orientation. The code usage histograms may be clustered, so that comparisons with the centroid distributions may be used to reduce the search space prior to image matching or to retrieve based upon a specified class label.

In addition to the first-order statistics, we can compute second-order statistics in the form of co-occurrence matrices, that describe the occurrence of some block coding spatial relationship. The co-occurrence statistics may be represented by a matrix of relative frequencies $P_{\phi,d}(a,b)$, where entries describe how frequently two transform codes a and b appear separated by a block distance d in a scan direction ϕ. The co-occurrence statistics can be used to support query by more than one example image.

4. Progressive Matching of Transform Coefficients

The factors of the transform code can be used to rank order the coefficients, thus permitting their independent compression and progressive decoding. We can group coefficients into multiple description levels based upon this ranking, for example, the level 1 description are the first coefficients of each block. By structuring the bitstream in this fashion we achieve a successively refinable description that supports query refinement.

As we progressively decode the transform coefficients, their matching distance must be included in the overall similarity measure between two images. We compute the coefficient matching using the modified Hausdorff distance, which is a distance defined between two sets of points that encodes an intuitive notion of the concept of *"looking similar"* without trying to build any one-to-one correspondences between the two sets of points.

Given two sets of coefficients that have been encoded using the same transform code, one set originating from query image $A = a_1, ..., a_p$ and the other from a model image $B = b_1, ..., b_q$ within the database. The modified Hausdorff distance [1] is defined as

$$H(A,B) = \max(h(A,B), h(B,A))$$
$$h(A,B) = \frac{1}{N_a} \sum_{a \in A} \min_{b \in B} \| a - b \|$$

$\| \cdot \|$ is defined as the Euclidean distance between the point sets, and N_a is the number of points in set A. As each description level is decoded, this distance is computed for all transform code classes.

5. Query Processing Thru Bayesian Evidential Reasoning

We treat the normalized coefficient distances between the query image and an image in the database, as evidence, E_i, pointing to the hypothesis, H, that the query image is similar to the model image in the database. Bayesian evidential reasoning is then used to aggregate the evidence into a single measure of similarity.

Bayesian evidence theory uses an “Odds-Likelihood Ratio" formulation of Bayes’ rule to aggregate the evidence from multiple sources. The likelihood of the evidence E_i, given that the hypothesis H is true, is

$$L(E_i|H) = \frac{P(E_i|H)}{P(E_i| \sim H)}.$$

where $P(E_i|H)$ are modeled by the normalized coefficient distances weighted by their significance defined by the code usage histogram. The formula for updating the odds (i.e., the a posteriori odds) of a hypothesis H, given the evidence observed, E_i, is

$$O(H|E_1, E_2, ..., E_n) = O(H) \prod_{i=1}^{n} L(E_i|H)$$

thus, the final measure of “similarity" between the query image and a model image is

$$P(H|E_1, E_2, ..., E_n) = \frac{O(H|E_1, E_2, ..., E_n)}{1 + O(H|E_1, E_2, ..., E_n)}.$$

The "best match" is chosen to be the model hypothesis H having the greatest probability given all of the accumulated evidence.

6. Retrieval Results

In the first experiment, we evaluated the performance of our system using a database derived from the MIT's Vision Texture (VisTex) image collection. Each of the 167 512x512 reference texture images was subdivided into 9 128x128 images, where 5 were randomly selected to comprise the training set consisting of 835 images, and the other 4 were included in the test set containing 668 images. A retrieval was considered to be correct if the query and the first retrieved image were from the same reference image. Compressing the images to an average rate of 0.4 bpp, we compared the retrieval results of our coding scheme with that achievable using WUTC designed for compression only. Matching only the histograms of the transform code usage maps, WUTC achieved a **78%** correct retrieval rate, while our method achieved **95%** correct retrieval with only an average of 1 dB difference in the distortion performance.

In the second experiment, we compared the retrieval performance of matching histograms of transform code usage, to the performance achieved in the uncompressed domain using color histograms. The database consisted of 5 complex image classes: birds, deserts, flowers, people, and water scenes, with 10 images per class. The classification accuracy of the color indexing method was **76%**, while matching the histograms derived from the compressed representation, coded at a rate of 0.4 bpp, achieved **72%**. Although the retrieval rates were nearly the same, our algorithm required that we only decode the index map information which on average is less than 35% of the bitstream, while the color indexing method required the full decompression of the image. Decompressing and matching the first 5 coefficients improved the retrieval accuracy to **94%**.

Finally, our third experiment involved video shot detection and frame retrieval. Using only the difference in code usage maps we correctly identified the representative shots in the video sequence shown in Figure 4. The number below each image identifies the video shot each frame was classified as. Our algorithm was also able to segment the sequence into key frames including transition frames. We achieved 96% correct frame retrieval on a database of 20 video clips decoding only the index map information.

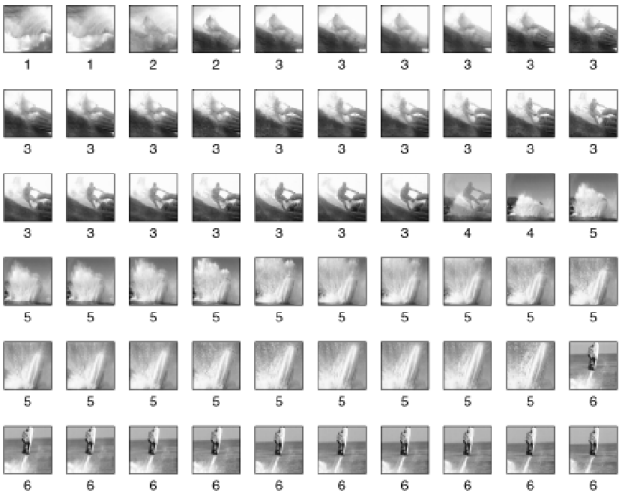

Figure 4. Video shot detection from code usage maps.

7 Conclusions

We introduced a universal statistical coder based upon a mixture of local linear Gaussian subspaces, which we have shown to be an efficient encoding scheme supporting content analysis in the compressed domain. The transformation basis was derived from the perspective of density estimation, thus offering the advantage of allowing Bayesian inferencing methods to be applied for image comparison. The compressed representation can be structured to permit the progressive independent decoding of the coefficients leading to successively refinable methods of query and retrieval. Our experiments showed that good retrieval rates can be achieved without full decompression of the data offering a substantial savings in space and time.

References

[1] M. Dubuisson, and A. K. Jain, "A modified Hausdorff distance for object matching", In *Proc. of the 12th Int'l Conf. on Pattern Recognition*, Jerusalem, Israel, 1994.

[2] M. Effros, and P. A. Chou, "Weighted universal transform coding: universal image compression with the Karhunen-Loeve transform", In *Proc. of the Int'l Conf. on Image Processing*, Washington, D.C., October 1995.

[3] Z. Ghahramani, and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers", University of Toronto Technical Report CRG-TR-96-1, 1996.

[4] K. Oehler, and R. Gray, "Combining image compression and classification using vector quantization", In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(5):461-471, 1995.

[5] T. Keaton, M. Effros, and R. Goodman, "Coding for image and video retrieval in the compressed domain", submitted to the *Int'l Conf. on Image Processing*, Kobe, Japan, October 1999.

[6] N. Vasconcelos, and A. Lippman, "Library-based coding: a representation for efficient video compression and retrieval", In *Proc. of the Data Compression Conference*, Snow Bird, UT, March 1997.