Semantic Representation, Search and Mining of Multimedia Content

Apostol (Paul) Natsev
natsev@us.ibm.com

Milind R. Naphade
naphade@us.ibm.com

John R. Smith
jsmith@us.ibm.com

IBM Thomas J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532

ABSTRACT

Semantic understanding of multimedia content is critical in enabling effective access to all forms of digital media data. By making large media repositories searchable, semantic content descriptions greatly enhance the value of such data. Automatic semantic understanding is a very challenging problem and most media databases resort to describing content in terms of low-level features or using manually ascribed annotations. Recent techniques focus on detecting semantic concepts in video, such as indoor, outdoor, face, people, nature, etc. This approach works for a fixed lexicon for which annotated training examples exist. In this paper we consider the problem of using such semantic concept detection to map the video clips into semantic spaces. This is done by constructing a *model vector* that acts as a compact semantic representation of the underlying content. We then present experiments in the semantic spaces leveraging such information for enhanced semantic retrieval, classification, visualization, and data mining purposes. We evaluate these ideas using a large video corpus and demonstrate significant performance gains in retrieval effectiveness.

Categories and Subject Descriptors:
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Algorithms, Design, Experimentation

Keywords: Semantic Indexing, Model Vectors, TRECVID

1. INTRODUCTION

The increasing growth of unstructured digital media content in the form of video, audio, images, graphics, and speech is driving the need for more effective methods for indexing, searching, categorizing and organizing such information. With the falling costs for media storage, higher bandwidth, and with the proliferation of affordable media production devices such as digital cameras and camcorders, media management is becoming increasingly important in a variety of consumer, scientific, and business applications. Tools for efficient storage and retrieval of multimedia content are absolutely essential for the utilization of raw content. Recent advances in content analysis, feature extraction and classification are improving capabilities for effectively searching and filtering of multimedia content. However, a significant gap remains between the low-level feature descriptions that can be automatically extracted, such as colors, textures, shapes, motions, etc., and the semantic descriptions of objects, events, scenes, people and concepts that users desire. This "semantic gap" between users' needs and systems' abilities has often been cited as the biggest stumbling block in the successful application of media management in real world problems. The problem of automatic semantic characterization of multimedia content is therefore an important research problem in data management and knowledge extraction. To enable efficient multimedia understanding it is necessary to be able to automatically tag and index content with meta data that spans a large number of concepts. We propose a scalable framework that relies on the definition of a lexicon of semantic concepts, learning model representations of these concepts, detecting these concepts in videos and then using these detection to create a semantic space. Processing in this space can lead to many new and exciting applications. We evaluate some applications in the context of the NIST TRECVID benchmark effort.

1.1 Proposed approach

We propose a framework that characterizes multimedia content using a semantic space and then permits search, retrieval, indexing, classification and clustering in the semantic space. This results from the following processing steps:

1. Concept Lexicon Design: The first step is to design the lexicon of concepts that need to be detected explicitly.
2. Concept Modeling: The next step is to learn low-level feature-based representations for the concepts in the lexicon. For this, we use machine learning with supervision in the form of annotated training examples.
3. Model Vector Construction: Once concept models covering the lexicon are learnt they can be applied to the detection of the concepts in a video shot, where each concept is detected with some level of confidence. These confidence values can then be concatenated to form what is hereby referred to as a *model vector*. The space spanned by the model vectors is the semantic space in which we then perform data processing.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

4. Model Vector Applications: Once a video shot is represented in terms of its model vector, several applications are made possible. This includes semantic similarity search, classification and clustering. In the case of search, both the query and target documents are identically mapped to that concept space. Thus the query can be carried out by searching the multi-dimensional model vector space and identifying nearest neighbors. In the case of classification and clustering the model vectors can be treated similarly to any other low level feature vectors such as color or texture features, etc. However since each dimension in the semantic space signifies a concept, the location of clusters in a subspace of this space assumes semantic significance and can help build new complex concepts based on the lexicon concepts. Benefits of processing in semantic space, include explicit semantic mining abilities, and fewer user interactions needed to achieve end results.

The model vector approach can be applied to any domain as long as a lexicon can be defined over concepts in the domain that can be learnt and detected. For example, we used support vector machines (SVMs) [10] for modeling visual concepts while the same result could be achieved with any classifier that produces a confidence measure to help build the model vector. The models developed then help create the model vector representation which maps data items into a model vector space. The advantage of the model vector representation is that it captures the concept labeling broadly across an entire lexicon. It also captures the uncertainty of the labels by incorporating confidence scores rather than using simple binary labels. Finally, it provides a compact storage-efficient representation that enables efficient indexing using straightforward computation of model vector distances. Once model vectors are extracted, complex similarity operations can be replaced by simple and inexpensive vector operations in the semantic space, which enhances the scalability of the database system. For example, range queries or nearest-neighbor queries with respect to complex semantic similarity can be performed using the simple Euclidean distance between model vectors. The model vector space can further be indexed using data structures and methods designed specifically for optimized query execution in multi-dimensional spaces (see Section 1.2). This allows development of efficient and effective systems for similarity searching, relevance feedback, classification, clustering and filtering that operate at a semantic level.

1.2 Related work

The problem of multimedia semantic modeling has been addressed in a number of ways relying on manual, semi-automatic, or fully-automatic methods. The use of manual annotation tools allows humans to manually ascribe labels to multimedia documents. However, manual cataloging is a very expensive and time consuming process. It is also subjective leading to incomplete and inconsistent annotations. Fully-automatic approaches based on statistical modeling of low-level audio-visual features have also been investigated for detecting generic frequently observed semantic concepts such as indoors, outdoors, nature, man-made, faces, people, speech, music, etc. There are two distinct schools of thought for modeling these concepts. In one, each concept is treated uniquely and the modeling process draws heavily on domain knowledge, manually enforced constraints and other information that can only be applicable for the concept at hand. The low-level features extracted in this case are also very specific for each such concept. The literature in this area is rich. The other school of thought is to use generic machine learning algorithms coupled with standard off the shelf low-level media features and let the learning algorithm figure out the specific feature properties that help build the model for the concept [7]. The advantage with this latter approach is that it is scalable to a large number of concepts. Towards this end a variety of classification techniques have been investigated in this context based on the static or temporal nature of the underlying media features extracted and the concept characteristics [1, 4, 5, 6].

Indexing of multimedia documents for fast search and retrieval has also received much attention in the literature. Considerable amount of work has focused on efficient indexing in multi-dimensional vector spaces. Approaches include specialized data structures, such as the R-tree and its variants [3], as well as methods for indexing objects in general metric spaces [2, 9]).

In general, computational efficiency and fast query execution are notoriously difficult to achieve in high-dimensional spaces, such as the model vector space we propose but a combination of sampling, dimensionality reduction and multi-dimensional indexing techniques is typically an approach that achieves reasonable scalability. While we realize that scalability is a very important issue, we note that all of the above indexing approaches can be transparently applied once we have constructed our model vector representation, and are thus considered complementary, and out of scope, with respect to this paper.

1.3 Our contributions

In this paper, we propose a novel framework for describing and indexing multimedia documents in terms of their membership to a predetermined lexicon of semantic concepts. We investigate the extraction of semantic model vectors using statistical modeling approaches, and taking into account uncertainty, reliability, and relevance of the semantic detectors. We study the properties of the model vector representation and consider lexicon design and model vector normalization approaches. Finally, we explore the application of the model vector framework to semantic retrieval, classification and clustering. We validate the proposed approaches empirically and observe significant performance improvements compared to alternative state-of-the-art methods. To summarize, our specific contributions are as follows:

- A novel framework for semantic indexing of multimedia documents based on a *model vector representation* capturing semantics across a lexicon of concepts.
- A generic approach for modeling of lexicon concepts.
- Methods for automatic model vector construction given binary detectors for the lexicon concepts
- An intuitive semantic similarity measure which can be computed by simple operations on model vectors.
- An application of the proposed semantic indexing framework to the problems of similarity search, classification and clustering in multimedia databases.
- Experimental validation on a large video corpus showing that model vectors significantly improve retrieval effectiveness compared to content-based retrieval and allow for classification and clustering to be performed in more meaningful semantic spaces.

2. MODEL VECTOR REPRESENTATION

2.1 Notation and definitions

Let $\mathcal{C} = \{l_1, l_2, \ldots, l_K\}$ denote a lexicon of K concepts, where l_m is the label of the m^{th} concept. We refer to lexicon $\mathcal{C}$ also as *semantic basis*. Let $\mathcal{D}$ be the set of K concept detectors, where d_m is the detector corresponding to concept l_m in lexicon $\mathcal{C}$. Let $c_m[j]$ give the confidence of detection of concept l_m by detector d_m in item j, where without loss of generality we assume that $c_m \in [0, 1]$, and $c_m = 1$ gives highest value of confidence of detection of l_m. We then define *model vectors* as follows:

DEFINITION 1. Given a semantic basis $\mathcal{C}$ and corresponding detectors $\mathcal{D}$, defined as above, we define the K-dimensional *model vector* for item j with respect to lexicon $\mathcal{C}$ as

$$m_j = \langle c_1[j], c_2[j], \ldots, c_K[j] \rangle \quad (1)$$

DEFINITION 2. Given two items A and B and a semantic basis $\mathcal{C}$, we define the semantic similarity between A and B with respect to $\mathcal{C}$ to be the fraction of semantic classes that both A and B belong to, or alternatively:[1]

$$Sim_{\mathcal{C}}(A, B) = \sum_{m=1}^{K} Pr(A, B \mid S_m) Pr(S_m),$$

where $Pr(A \mid S_m)$ denotes the probability that item A belongs to the m-th semantic class denoted by S_m.

If we assume item independence and interpret the semantic detection scores $c_m[A]$ as $Pr(A \mid S_m)$, then the semantic similarity reduces simply to inner products of model vectors, as follows:

$$\begin{aligned} Sim_{\mathcal{C}}(A, B) &= \sum_{m=1}^{K} Pr(A \mid S_m) Pr(B \mid S_m) P(S_m) \\ &= \sum_{m=1}^{K} (c_m[A])(c_m[B]) \\ &= m_A \cdot m_B \end{aligned} \quad (2)$$

2.2 Model vector advantages

We now list several properties and advantages of the model vector representation:

- Model vectors index multimedia data items in a semantic space
- Model vectors reduce dimensionality and provide a compact, time and space cost-efficient representation
- They capture semantics effectively (see Section 5)
- They can be used for indexing in general metric spaces when vector space embeddings for the database objects are not available (e.g., semantics, audio, shapes)
- They are computationally efficient and replace complex similarity measures with simple vector space operations (e.g., eliminate expensive statistical model evaluations during on-line query phase)

[1]There is a slight abuse of notation here since in effect A and B denote both items and random variables. However, in general the semantic similarity is well defined, and applies both to individual objects, such as multimedia documents, and to entire classes of objects, such as semantic categories

3. MODEL VECTOR EXTRACTION

The generation of model vectors involves two stages of processing: (1) *a priori* learning of detectors and (2) concept detection and score mapping to produce model vectors. The output of the detectors is transformed in a mapping process to produce the model vectors.

3.1 Learning Concept Models

The generic framework for modeling semantic concepts from multimedia features [5] includes an annotation interface, a learning framework for building models and a detection module for ranking unseen content based on detection confidence for the models Positive examples for interesting semantic concepts are usually rare. The concept learning process uses ground-truth labeled examples as training data for building statistical models for detecting semantic concepts. We construct a set of K binary detectors, each corresponding to the presence or absence of a distinct concept from the lexicon. We have experimented with different classification algorithms and found support vector machine classifiers to perform better for video concept modeling. Concepts in our lexicon occur at global or image levels or subframe levels i.e. regions. For this we extract the following set of features from the image as well as up to 5 most dominant regions in the image marked automatically by bounding boxes proceeding image segmentation[2] We used the following descriptors: color correlogram (166-D), co-occurrence texture (96-D), edge histogram (64-D), and moment invariants for shape (6-D). For more details, see [5].

3.2 Modeling concepts using SVM classifiers

We use a training set with manually annotated and marked regions to learn the SVM models. For the experiments in this paper we have reported results using the radial basis kernel function defined in Equation 3:

$$K(x_i, x_j) = e^{\frac{-||x_i - x_j||^2}{2\sigma^2}} \quad (3)$$

Assuming that we extract features for color, texture, shape, structure etc. it is important to fuse information from across these feature types. One way is to build models for each feature type including color, structure, texture and shape and combine their confidence scores post-detection. We also experiment with early feature fusion by combining multiple feature types at an early stage to construct a single model across different features. Alternately we can simply concatenate one or more of these feature types (appropriately normalized). Different combinations can then be used to construct models and the validation set is used to choose the optimal combination. This is feature selection at the coarse level of feature types. Based on our experimentation of early feature fusion [5] we chose to combine the features mentioned in 3.1. Performance of SVM classifiers can vary significantly with variation in parameters of the models. For parameter tuning and validation purposes, we use average precision to measure the retrieval effectiveness and detection performance. Let R be the number of true relevant documents in a set of size S; L the ranked list of documents returned. At any given index j let R_j be the number of relevant documents in the top j documents. Let $I_j = 1$ if the

[2]In this paper we only report experiments for visual concept models, learnt from visual features extracted from keyframes

Concept	f	v	Concept	f	v	Concept	f	v
Airplane	185	0.1296	Animal	596	0.0156	Beach	164	0.1218
Bill Clinton	192	0.0516	Building	1273	0.0509	Car	1565	0.0487
Cartoon	262	0.0294	Cityscape	427	0.0123	Cloud	282	0.0971
Crowd	899	0.2003	Desert	79	0.0880	Face	26670	0.4776
Female Face	4010	0.1788	Fire	88	0.0032	Flower	172	0.0147
Graphics & Text	20716	0.7642	Graphics	24304	0.2727	Human	27730	0.6232
Indoors	10068	0.3742	Land	199	0.0038	Male Face	6152	0.1702
Man Made Scene	3047	0.1753	Mountain	218	0.1682	Nature Vegetation	2417	0.3731
Newt Gingrich	25	0.0017	Non-Studio Setting	23754	0.1565	Outdoors	11745	0.3839
People Event	756	0.1045	People	4773	0.1571	Person	18884	0.0969
Physical Violence	225	0.0014	Podium	80	0.0094	Riot	18	0.0079
Road	808	0.0293	Rock	124	0.0497	Sky	1333	0.1619
Smoke	55	0.0034	Snow	314	0.0183	Sport Event	1232	0.3774
Studio Setting	25696	0.8010	Text Overlay	13447	0.2626	Transportation	2738	0.0892
Tree	700	0.0547	Truck	181	0.0025	Water Body	574	0.0618
Weather News	208	0.8454						

Table 1: Lexicon of 46 concepts for which visual models are built using SVMs. For each concept the number of training samples (f) in the training set and the non-interpolated average precision ((Eq. 4) on a validation set at the depth of 1000 documents is also reported as the validity v.

j^{th} document is relevant and 0 otherwise. Assuming $R < S$, the non-interpolated average precision **(AP)** is defined as

$$\frac{1}{R}\sum_{j=1}^{S}\frac{R_j}{j} * I_j \tag{4}$$

We reduce parameter sensitivity and dependence on any single feature type but data dependency is harder to deal with.

3.2.1 Lexicon

Table 1 shows the lexicon of 46 concepts that we use in the experiments reported in this paper listed alphabetically. An important consideration for choosing the concepts in the lexicon is that when modeled, their performance should be acceptable. Another aspect to consider is the effect of lexicon size and model vector dimensionality on retrieval effectiveness (see Section 5). In many situations, it is not feasible to model and evaluate a large number of detectors and the size of the model vector lexicon is constrained from practical considerations, such as computational time or storage requirements. In such scenarios, it is desirable to form a small lexicon while maximizing its effectiveness. We therefore consider methods for studying the model vector space for the purposes of reducing overlap among the semantic concepts and maximizing their utility. We would like to preserve the semantic meaning of the model vector space, and for this it is more appropriate to do model selection, rather than dimensionality reduction techniques such as PCA.

We consider two approaches for prioritizing dimensions in the model vector space so that the more important ones can be selected. The first approach, *frequent model selection* advocates selecting the models for the most frequently occurring concepts. The frequency can be measured as the number of relevant examples for the given concept in an annotated training set. The second approach, *Robust model selection*, associates the priority for a given model with its performance reliability, which can be measured on an independent validation set. The motivation is that the high-performing models should be preserved.

3.3 Model vector construction

Once concept models are constructed for all the concepts in the lexicon multimedia documents can now be analyzed, classified and scored using each concept model. We base the scoring on the confidence of detection of each concept. Additionally, we allow the incorporation of detector correlations in the mapping process and detector reliability and concept relevance score in the matching process (through score normalization and weighting).

For each of the K detectors a confidence score $s_k \in [0 \ldots 1]$ is produced for each multimedia document that measures the degree of certainty of detection of concept c_k. We base the confidence score on proximity to the decision boundary for each detector, where high confidence score is given for documents far from the decision boundary and low scores given when close to the boundary. Additionally, a validity score indicates how reliable the detector is for detecting its respective concept. The validity is calculated as the average precision on a separate validation set which is different from the one used to select optimal parameters.

The confidence scores c_k corresponding to the models d_k are mapped to produce the model vectors.

$$m_j = \langle c_1[j], c_2[j], \ldots, c_K[j] \rangle\,,\ \forall j \in [1, J]. \tag{5}$$

This is followed by appropriate normalization to remove bias and optionally by validity weighting to capture relative concept importance.

4. MODEL VECTOR APPLICATIONS

Once documents are indexed into the multi-dimensional model vector space, this enables document processing at a semantic level while using fully automated vector-space processing techniques. For example, documents can be compared, searched, classified, clustered, visualized, or mined by using the corresponding vector-space techniques. The benefits of performing some of the above operations in the semantic model vector space, as opposed to the original low-level feature space, are validated empirically in Section 5.

4.1 Semantic matching

The first application includes the problem of semantic matching for similarity-based retrieval of multimedia documents. In particular, the distance between model vectors is based on the similarity of the videos with respect to the detector confidence scores. Neural Networks or Gaussian Mixture Models, for example, have a natural interpretation as posterior probabilities. Alternatively, we can consider distance measures based on simple or weighted Euclidean distance of model vectors. Weights can be used to capture relative importance of concepts, reliability of concept detectors, or individual user preferences.

4.2 Classification and Clustering

Once the model vector space is constructed, it can be treated like any other feature space and we can apply the same classification techniques that were used to learn the models that created the model vector, recursively to the model vector features. This will result in models learnt in model vector spaces. We have used such methods for rare class classification [8] and context modeling and enforcement [7]. The main advantage of classification in model vector space is improved classification system scalability, especially with respect to large lexicons. This comes from a reduction in supervision requirements, reduction in storage, computational requirements, and through leveraging of inter-conceptual relationships.

Unlike clusters in low-level feature spaces, clusters in model vector spaces signify semantic homogeneity. Our clustering in this space led to the discovery of a cluster of *news anchors*, *sport events*, *outdoor crowds*, etc.

5. EXPERIMENTS

The experiments in this section were performed using the TRECVID 2003 Concept Detection Benchmark[3] corpus provided by the National Institute of Standards and Technology (NIST). This contains a development data set of approximately 60 hours of MPEG video consisting of CNN, ABC, and C-SPAN news broadcasts. We split this into one training set of 36 hours and 3 validation sets: validation set I of 6 hours, validation set II of another 6 hours and finally validation set III of 12 hours. In this paper, the concept models were trained using the training set, optimized for parametric settings using validation set I of 6 hours and the validity measured over validation set II of another 6 hours. Finally, all of the model vector experiments in this section report results on the unseen validation set III. As part of the TRECVID 2003 effort, the entire development set was annotated collaboratively by over 100 researchers using a lexicon of more than 100 primary concepts. Of those we modeled 46 visual concepts that had sufficient support in the training set in terms of number of relevant shots.

5.1 Evaluation methodology

The experiments below use average precision (as defined in Eq. 4) as the performance measure for evaluation. In the retrieval experiments, for each query topic, each relevant video clip is used in turn to query the database. The candidate images are ranked according to their similarity to the query image, and then average precision, AP, is calculated at full retrieval depth, thus effectively measuring the area under the precision-recall curve for a given query. The AP numbers are then averaged over all queries for a given topic, yielding an overall average precision score for the topic. Mean Average Precision (MAP) can then be calculated as the average AP score across several query topics.

[3]http://www-nlpir.nist.gov/projects/tv2003/

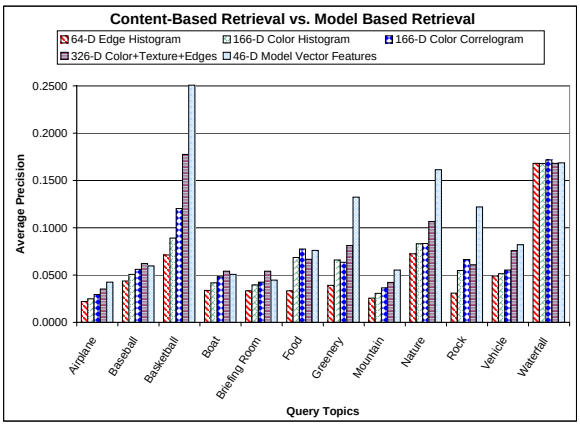


Figure 1: Retrieval effectiveness for 12 query topics using content-based retrieval *vs.* model based retrieval.

For evaluation, we consider 12 query topics, including objects, sites, and events, and a range of frequent to rare concepts. The majority are outside of the 46 concepts modeled explicitly but we have also included some overlapping ones for comparison purposes. In particular, the set of concepts used as query topics is as follows (numbers in parenthesis indicate number of relevant items in validation set III):

- Airplane (96), Baseball (26), Basketball (92), Boat (37), Briefing_Room (43), Food (247), Greenery (250), Mountain (70), Nature Vegetation (521), Rock (48), Vehicle (415), and Waterfall (6).

Pairwise item similarity is computed as an inner product in the feature vector space, corresponding to either low-level visual features or model vectors (see Sections 2.1 and 4.1).

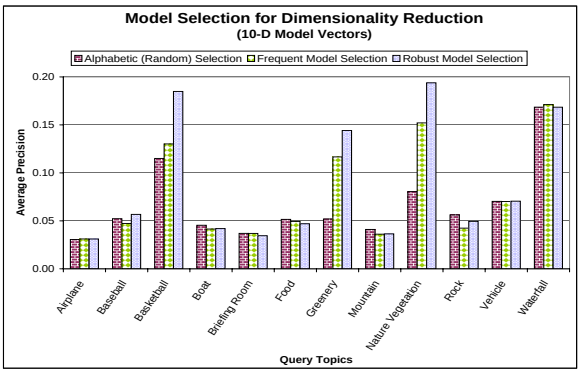
5.2 Experiment 1: MBR *vs.* CBR

The first experiment compares model vector based retrieval (MBR) to content-based retrieval (CBR). Overall, the following descriptors are compared: 46-dimensional model vectors, 64-dimensional edge histograms, 166-dimensional color histograms, 166-dimensional color correlograms, and 332-dimensional visual features derived by concatenating and normalizing color correlogram, co-occurrence texture, and edge histogram features. Figure 1 plots the Average Precision (AP) computed for each of the 12 topics, and the overall Mean Average Precision scores are shown in Table 2.

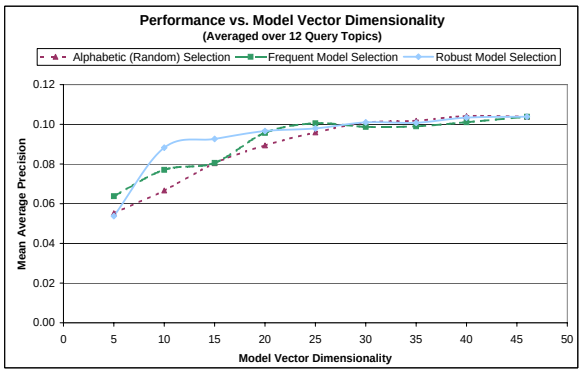
Query Topic	Edge Histogram	Color Correlogram	Correlogram+ Texture+Edge	Model Vectors
MAP	0.0519	0.0709	0.0820	0.1039

Table 2: Mean Average Precision over 12 topics for model vector-based *vs.* content-based retrieval.

Generally, model vectors outperform all visual features and result in a significantly higher Mean Average Precision. The performance gain in MAP ranges from 25% over the high performing 332-D visual feature to over 100% as compared to the simpler edge histogram features.



(a) 10-D model vector performance across 12 topics



(b) Dimensionality effect on averaged performance over 12 topics

Figure 2: Performance of model selection methods for dimensionality reduction.

5.3 Experiment 2: Model selection

In the second experiment we investigate the effect of model vector dimensionality on retrieval effectiveness. As described in Section 3.2.1, we consider two main strategies for model selection—*frequent model selection* and *robust model selection*. For comparison purposes, however, we also consider the approach of selecting a random subset of the models, to serve as a baseline reference point. In the experiments below, the pseudo-random prioritization is achieved by alphabetical ordering of the concepts.

Figure 2 (a) compares the three model selection strategies for dimensionality reduction purposes. Each strategy is used to generate a 10-dimensional model vector from the original 46-dimensional one, and the Average Precision is plotted for the 12 query topics. The robust model selection strategy generally gives best performance, outperforming frequent model selection by 15% in MAP over the 12 topics, and outperforming pseudo-random selection by 30%.

Figure 2 (b) on the other hand shows the effect on retrieval performance (averaged over the 12 topics) as a function of model vector dimensionality. It is interesting to note that the two primary model selection strategies seem to get saturated performance at about half of the original dimensionality. Robust selection performs best again, and reaches 90% of top performance with merely 30% of the dimensionality.[4]

6. CONCLUSIONS

We investigated a novel framework for capturing and leveraging semantics in multimedia databases. The model vector approach uses a concept lexicon as a basis to provide a semantic descriptor that can be used in a variety of ways for multimedia indexing, including similarity-based retrieval, relevance feedback search and semantic concept classification.

[4]We should note that robust/frequent model selection methods have their caveats as well. In particular, both methods tend to favor models for generic concepts which occur most frequently. These concepts generally have the best performing models but may not have the best discriminatory power, or may not be the most relevant, for a given query topic. This can perhaps explain why the random model selection approach slightly outperformed the other two approaches for a few specific query topics (e.g., boat, rock, and food). In practice, we have found that model reliability is crucial for good performance but model relevance to the query topics is just as important, and a balance between the two is therefore essential to the success of the approach.

We performed an extensive empirical study to validate the proposed model vector construction and application approaches, emphasizing significant retrieval performance advantages as compared to processing in the low-level visual feature domain. We also considered the problem of semantic-preserving dimensionality reduction for model vectors and studied the effect of dimensionality on retrieval performance. Future work includes investigating other applications of semantic model vectors, as well as automatic methods for semantic basis selection.

7. ACKNOWLEDGMENTS

The IBM TRECVID 2003 team (shot segmentation, region segmentation, and annotation).

8. REFERENCES

[1] I. Buciu, C. Kotropoulos, and I. Pita. On the stability of support vector machines for face detection. In *IEEE Intl. Conf. on Image Processing (ICIP)*, 2002.

[2] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroqu´in. Searching in metric spaces. *ACM ACM Computing Surveys*, 33(3):273–321, Sep. 2001.

[3] V. Gaede and O. Gunther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[4] A. Gupta, T. E. Weymouth, and R. Jain. Semantic queries with pictures: the VIMSYS model. In *Intl. Conf. on Very Large Databases (VLDB)*, pages 69–70, Sep. 1991.

[5] M. Naphade and J. Smith. Learning visual models of semantic concepts. In *Proc. IEEE International Conference on Image Processing*, Sep 2003.

[6] M. R. Naphade, S. Basu, J. Smith, C. Y. Lin, and B. Tseng. Modeling semantic concepts to support query by keywords in video. In *Proc. IEEE Intl. Conference on Image Processing (ICIP '02)*, Rochester, NY, Sep. 2002.

[7] M. R. Naphade, I. Kozintsev, and T. S. Huang. A factor graph framework for semantic video indexing. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(1):40–52, Jan 2002.

[8] A. Natsev, M. Naphade, and J. Smith. Exploring semantic dependencies for scalable concept detection. In *IEEE International Conference on Image Processing*, Barcelona, Spain, Sep. 2003.

[9] A. Natsev and J. R. Smith. New anchor selection methods for image retrieval. In *Proc. SPIE Electronic Imaging: Storage and Retrieval for Media Databases*, San Jose, CA, Jan. 2003.

[10] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.