IBM Research TRECVID-2010 Video Copy Detection and Multimedia Event Detection System

Apostol Natsev[*], John R. Smith[*], Matthew Hill[*], Gang Hua[*], Bert Huang[†], Michele Merler[†], Lexing Xie[*], Hua Ouyang[‡], Mingyuan Zhou[§]

Abstract

In this paper, we describe the system jointly developed by IBM Research and Columbia University for video copy detection and multimedia event detection applied to the TRECVID-2010 video retrieval benchmark.

A. Content-Based Copy Detection:

The focus of our copy detection system this year was fusing three types of complementary fingerprints: a keyframe-based color correlogram, SIFTogram (bag of visual words), and a GIST-based fingerprint. However, in our official submissions, we did not use the color correlogram component since our best results on the training set came from the GIST and SIFTogram components. A summary of our runs is listed below:

1. IBM.m.nofa.gistG: A run based on the grayscale GIST frame-level feature, with at most 1 result per query, except in the case of ties.
2. IBM.m.balanced.gistG: As in the above run, but with including more results per query, though on average still less than 2.
3. IBM.m.nofa.gistGC: The result of the nofa.gistG run, fused with results from GIST features extracted from the R,G,B color channels.
4. IBM.m.nofa.gistGCsift: The result of the nofa.gistGC run, fused with a SIFTogram result.

Overall, the grayscale GIST approach performed best. We found it produced excellent results when tested on the TRECVID-2009 data set, with an optimal NDCR that surpassed what we had achieved with SIFTogram previously. The "gistG" runs also outperformed our other runs on the 2010 data, although we changed the SIFT implementation we used this year which made it not directly comparable with our previous TRECVID results. Our system did not make use of any audio features.

B. Multimedia Event Detection:

Our MED system has three aspects to its design – a variety of global, local, and spatial-temporal descriptors; building detectors from a large-scale semantic basis, and designing temporal motif features:

1. IBM-CU_2010_MED_EVAL_cComboAll_1 : Combination of all classifiers.
2. IBM-CU_2010_MED_EVAL_pComboIBM+CU-HOF_1 : Combination of global image features, spatial-temporal interest points, audio features, and model vector classifiers.
3. IBM-CU_2010_MED_EVAL_cComboStatic_1 : Combination of global image features, and model vector classifiers.
4. IBM-CU_2010_MED_EVAL_cComboDynamic_1 : Combination of spatial-temporal interest points, audio features, temporal motif, and HMM classifiers.
5. IBM-CU_2010_MED_EVAL_cComboIBM+CU-HOF_2 :Combination of global image features, spatial-temporal interest points, audio features, and model vector classifiers.
6. IBM-CU_2010_MED_EVAL_cComboIBM-HOF_1 : Combination of global image features, spatial-temporal HOG points, and model vector classifiers.

[*]IBM T. J. Watson Research Center, Hawthorne, NY, USA
[†]Dept. of Computer Science, Columbia University
[‡]College of Computing, Georgia Tech
[§]Dept. of Electrical Engineering, Duke University

7. IBM-CU_2010_MED_EVAL_cComboIBM_1 : Combination of global image features, spatial-temporal interest points, and model vector classifiers.

8. IBM-CU_2010_MED_EVAL_cmodelVectorAvg_1 : Run with 272 semantic model vector features.

9. IBM-CU_2010_MED_EVAL_cTemporalMotifs_1 : Semantic model vector feature with sequential motifs.

10. IBM-CU_2010_MED_EVAL_cmvxhmm_1 : Semantic model vector feature with hierarchical HMM state histograms.

Overall, the semantic model vector is our best-performing single feature, while the combination of dynamic features outperforms the static features, and temporal motif and hierarchical HMMs show promising performance.

1 Introduction

This year the IBM team has participated in the TREC Video Retrieval Track, and submitted results for the Content-Based Copy Detection and Multimedia Event Detection tasks. This paper describes the IBM Research system and examines the approaches and results for both tasks.

For the Content-based Copy Detection (CCD) task, we focused our work on analysis of the video frames, as opposed to audio, which we did not use this year. Although local features such as SIFT have been shown in the past to be very useful for detecting video copies, as the collection size grows (as it has for TRECVID CCD since 2008) the local feature-matching approach suffers from scalability issues. So, we use exclusively frame-global features in our matching process, although in the case of SIFTogram, these frame-global features are derived from local features. The most important change we made this year was the addition of the GIST feature descriptor to the set we extracted and tested. We used two GIST descriptors - one computed from the grayscale version of the frames (averaged R.G, B) and another, three times the size, computed from the R,G and B channels independently. We found that the grayscale GIST feature performed the best, compared to our SIFTogram and color correlogram-based fingerprints.

For the MED task, we emphasize three aspects in exploring effective methods for multimedia event detection. Section 3 gives an overview and details on how using a large number of semantic detectors, covering scenes, objects, people, and various image types enhances event recognition.

2 Copy Detection System

Figure 1 gives an overview of our copy detection system. The two major flows are fingerprint extraction and fingerprint matching, which are explained further in the following sections.

2.1 Fingerprint Extraction and Indexing

In this section we describe our process for generating fingerprints from videos and indexing them. In section 2.2 we describe how query fingerprints are matched to reference video fingerprints.

2.1.1 Frame Sampling and Normalization

We sample frames from query and reference videos uniformly at a rate of one frame per second, and in the process, we detect and remove frames with low entropy, such as blanks. This eliminates useless feature vectors, and many false alarms due to trivial matches. For the GIST and color correlogram descriptors, we normalize frames in an attempt to calibrate color appearance. We median-filter in order to remove speckle noise, important in areas such as borders. We then detect and remove borders of homogeneous colors to avoid spikes in the color distribution. We resample the image to a size of 176x144 in order to normalize scale, aspect ratio, and reduce noise and compression artifacts. Finally, we perform contrast-limited histogram equalization, which normalizes brightness, contrast, and to some extent, gamma. The resulting image is color-quantized perceptually into a 166-dimensional HSV color space.

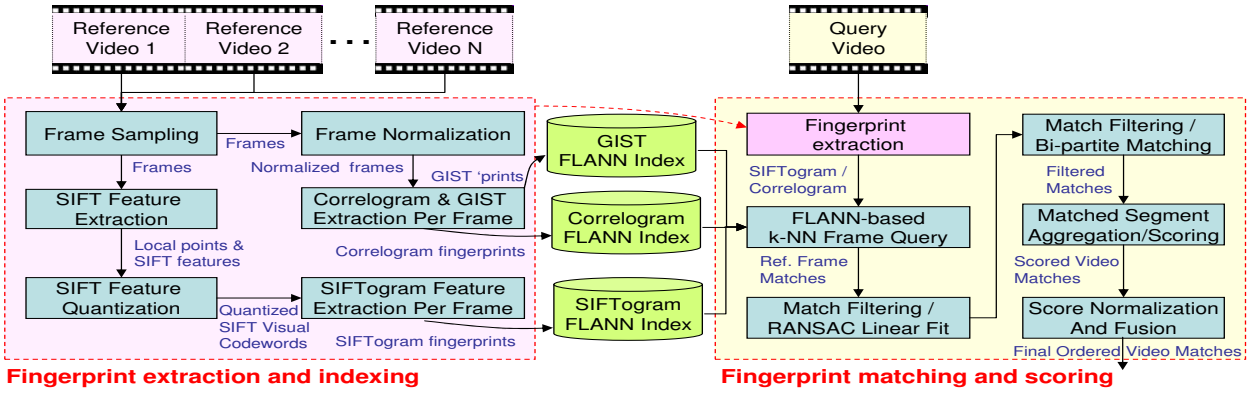

Figure 1: System overview illustrating fingerprint extraction/indexing (left) and fingerprint matching (right).

2.1.2 GIST descriptor

The new descriptor we added to our system this year was the GIST feature [10], which concatenates together the spatially pooled Gabor filter responses in different scales and orientations in different spatial blocks of the image to characterize a scene. It is largely invariant to changes in color, re-encodings of the video, missing frames, and quite robust to pattern insertion as well. However, as with many frame-global descriptors, it can be ineffective on transforms that change the nature of the entire frame, such as flipping, PIP, or heavy cropping. To extract the 320-dimensional feature, we used code derived from Torralba's published version [10].

2.1.3 SIFTogram descriptor

We use the "bag-of-words" approach [11] to leverage the retrieval power and invariance of SIFT features to color, rotation, shift, and scale, while balancing computation time. Following this method, we apply the Harris-Laplace interest point detector and extract SIFT local point features from all sampled frames. We use a sample of 1M SIFT features from a training set of reference videos to generate a codebook of 1000 representative clusters. The centroids of these clusters become *visual codewords*, which are used to quantize any SIFT feature into a discrete visual word. For each frame, we then compute a histogram of the codewords, making a global feature from the set of local ones, which discards feature locations but preserves feature co-occurrences and distributions. The number of codewords is the dimensionality of the feature vector, in our case, 1000. We used soft bin assignment and a sigma parameter of 90. This "SIFTogram" feature is robust to changes in colors, gamma, rotation, scale, shift, and added borders.

2.1.4 Color correlogram descriptor

The third descriptor we considered was the color correlogram [4], which captures the local spatial correlation of pairs of colors, and is a second-order statistic on the color distribution. The color correlogram is rotation-, scale-, and to some extent, viewpoint-invariant. It was designed to tolerate moderate changes in appearance and shape due to viewpoint changes, camera zoom, noise, compression, and to a smaller degree, shifts, crops, and aspect ratio changes. We extract an auto correlogram in a 166-dimensional perceptually quantized HSV color space, resulting in a 166-dimensional descriptor length for the baseline correlogram feature vector. The correlogram fingerprint performs well against mild to moderate geometric transforms but does not handle gamma correction changes or hue/saturation transforms. Its sensitivity to color makes it complementary to the SIFTogram and the grayscale GIST feature.

2.1.5 Cross spatial layout

We adopt a "cross"-layout formulation of the correlogram, which extracts the descriptor from two central im-

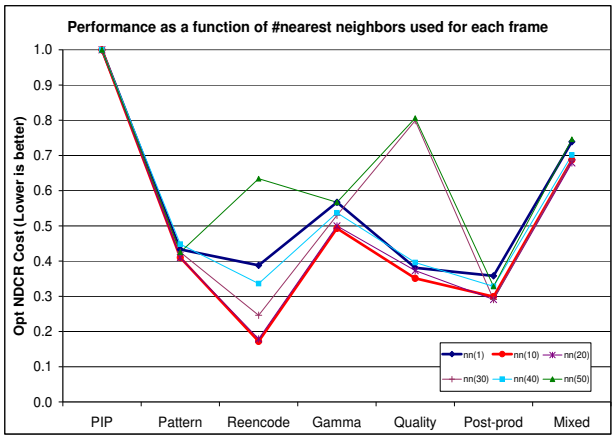

Figure 2: k-NN parameter selection

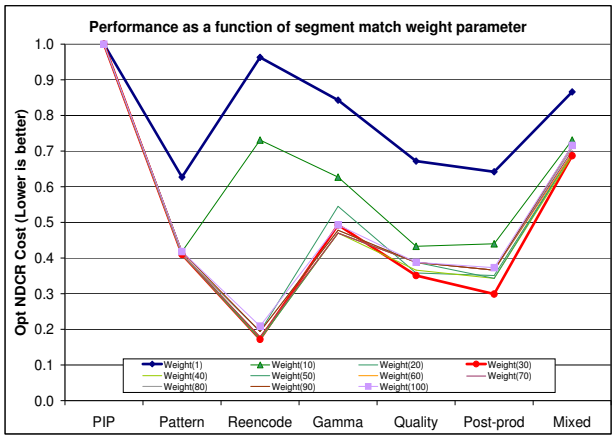

Figure 3: Scoring weight parameter selection

age stripes, one horizontal and one vertical, thereby emphasizing the center portion of the image and disregarding the corner regions. The cross layout improves robustness with respect to text/logo overlay, borders, small crops and shifts, etc. It is invariant to horizontal or vertical flips, while capturing some spatial information.

2.1.6 Fingerprint indexing

Once the two types of fingerprints are extracted for all reference videos, we create an index for fast nearest neighbor lookup. We used the Fast Library for Approximate Nearest Neighbor (FLANN)[8] to index all fingerprints. We found that FLANN could speed up query times by a factor of over 100, as compared to exact nearest-neighbor search, without affecting accuracy substantially.

2.2 Fingerprint Matching and Scoring

To process a query video, we extract the SIFTogram, GIST and color correlogram descriptors from all sampled query video frames. For each query frame q_i, we use the corresponding descriptors to retrieve a set of k nearest neighbors from the corresponding SIFTogram, GIST or correlogram FLANN index. The resulting matches for all query frames are grouped per video, forming a short list of candidate video matches, each with a set of $\langle q_i, r_i \rangle$ matching pairs of query and reference video frames. We then filter the candidate matches in two ways before scoring them.

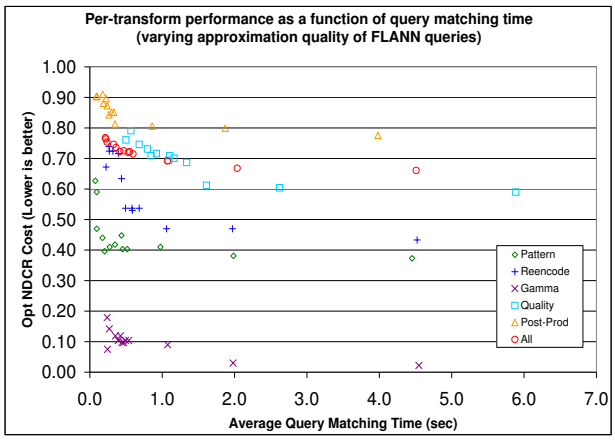

Figure 4: NDCR vs. time for TV2008 data

2.2.1 Linear fit filtering

We use RANSAC [3] to estimate the best linear fit (matching offset and slope) between the video times of query frames and matching reference frames:

$$Time(r_i) = offset + slope * Time(q_i), \forall i$$

We constrain the slope to be in the $[0.8, 1.2]$ range, allowing up to 20% frame drops, speed-up or slow-down of the query videos. Once the linear fit parameters are estimated, we filter all reference frame matches that deviate by more than 4sec from the estimated position in the reference video based on the corresponding query frame position. This soft filtering eliminates false matches while

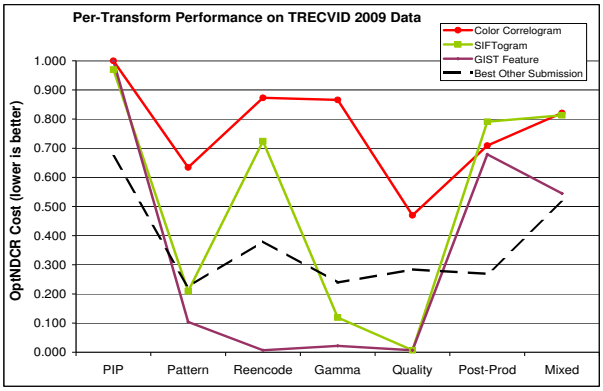

Figure 6: Detection on TV09 (video-only data)

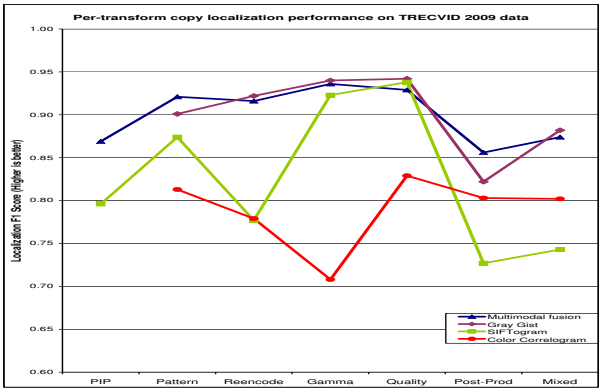

Figure 7: Localization on TV09 (video-only data)

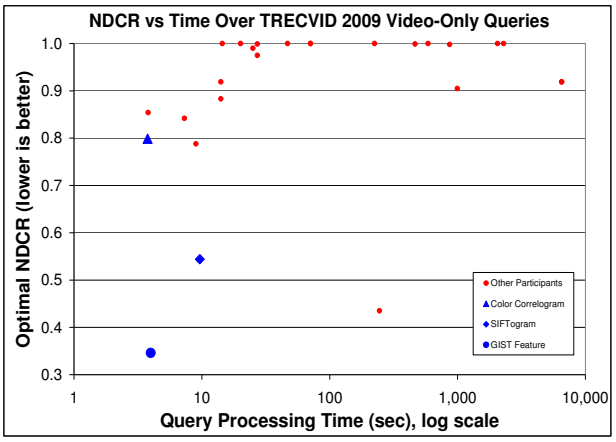

Figure 5: Processing time vs NDCR on TV2009 data

allowing matches from the general shot neighborhood of the estimated match position.

2.2.2 Bi-partite match filtering

We further filter candidate matches by allowing each query frame to match at most one reference frame, and vice versa. This is essentially the problem of computing a maximum weighted bi-partite matching, when considering the pairwise frame similarities as edge weights. While there are exact solutions to this problem, such as the Hungarian algorithm, they can be expensive computationally. We use a simple greedy heuristic that iteratively picks the edge with highest weight (i.e., highest frame match score), removes other edges sharing a vertex (i.e., same query or reference frame), and repeats until all edges are examined. This algorithm runs in $\mathcal{O}(|E| \log |E|)$ time, where $|E|$ is the number of edges, or candidate frame matches.

2.2.3 Matched segment aggregation and scoring

The remaining frame matches are aggregated into matching segments and scored. In principle, there can be multiple valid segment matches in the same reference video but the TRECVID CBCD task evaluation considers at most one segment match as valid, and penalizes the rest as false alarms, even if they overlap a true copied segment. For the purposes of this evaluation, we therefore force a single matching segment per reference video, by taking the union of all matching segments and allowing gaps in between. The final score is then a weighted combination of the matched segment duration and density:

$$S(Q,R) = 100\frac{|q_i|}{|Q|} + w\sum_i sim(q_i, r_i), \qquad (1)$$

where Q and R are the given query and reference videos, $S(Q,R)$ is the matching score between Q and R, $|Q|$ is the total number of sample query frames (matched and un-matched), $\{q_i\} \subseteq Q$ are the subset of matched query frames, $\{r_i\} \subseteq R$ are the corresponding matched reference frames. The left portion of Eq. 1 represents the density of the matched segment, accounting for potential gaps

between matched frames. It is also normalized with respect to video duration so it is comparable across different query videos. The right portion captures the "strength" of the match by accumulating the pairwise similarity scores over all matched frames. This is essentially the number of matched frames, $|q_i|$, weighted by the confidence of each frame match. The density score favors short query videos, due to smaller $|Q|$, while the right component favors longer match durations and therefore longer query videos. We balance the two factors using a weighted combination of the scores, where the weight w of the match strength is expressed relative to a weight of 100 for the match density. The parameter w is tuned empirically, and we used $w = 30$ in these runs.

2.2.4 Score Normalization and Fusion

Eq. 1 shows the overall match score for a single fingerprint descriptor, either SIFTogram, GIST or correlogram. The final detection result for a given query can be a combination of the normalized scores from the different fingerprints. We use linear range normalization, dividing all scores by the maximum observed score (per descriptor) on a training set, to map all scores into the $[0, 1]$ range. We then fuse the two normalized scores using simple score averaging.

We also leverage the information from multiple detectors to improve the copy localization performance of the system. Specifically, if two systems assert matched segments that are compatible (e.g., have similar linear fit parameters), we take the union of the asserted matched segments as the final matched segment. If the asserted matched segments are conflicting, we use localization information only from the detector with the highest normalized confidence score.

2.3 CCD Experiments

We performed several experiments on the prior years' data to select parameter values used in our system. Figures 2-3 show performance as a function of the number of nearest neighbors, k, retrieved for each frame using FLANN, and the weight, w, of the match strength relative to match density from Eq. 1. The results show that performance is quite stable with respect to both: values of k between 10 and 30, and w between 30 and 100, produce near-optimal results for our features.

We also performed experiments relating accuracy and processing time. Figure 4 shows the per-transform NDCR cost as a function of the query matching time, computed by varying the approximation quality of the FLANN-based nearest-neighbor queries on the TRECVID 2008 dataset. FLANN allows each approximate k-NN query to specify the maximum number of "node checks" to be performed, which controls the approximation quality of the k-NN results. We used values corresponding to scanning between 0.01% and 1% of the reference index in order to produce the operating points plotted in Figure 4. Performance improves as node checks increase but flattens out beyond a certain point. In other experiments, we have used 1000 FLANN node checks per query, which corresponds to scanning $\sim$$0.15\%$ and 0.07% of the 2008 and 2009 reference sets, and maps to an average query time of about 1sec on the 2008 dataset and 5sec on the 2009 dataset (which includes $\sim$100 individual FLANN frame queries per query video).

Figure 5 shows a comparison of the tested approaches against all TRECVID 2009 official submissions in terms of NDCR and mean query processing time. Overall NDCR is computed by aggregating all queries, from all transform types, into a single transform class, and computing the best NDCR score for each submission across all queries. The plot shows our GIST run achieves the lowest overall NDCR and the fastest execution time compared to the 2009 system results we were working with. The next most accurate 2009 system in NDCR is more than 50x slower than the GIST-based system. The total query processing time, including query video decoding, fingerprint extraction, and matching, was $\sim$4 sec/query for the GIST system running on a 4-core processor, or more than 20x faster than real-time (comparable to the correlogram system); the SIFTogram system was $\sim$10 sec/query, or 9x real-time; and the fusion system's time was $\sim$13 sec/query, or 7x real-time.

2.4 CCD 2010 Results

Figure 8 shows the results of the runs we submitted for the three NOFA runs which we submitted. The fourth run which we submitted was our gistG balanced run which had scores indistinguishable from the gistG NOFA run in

figure 8. Since we did not use an audio feature descriptor, we have aggregated the results by video transform in this figure. The components of these submitted runs are listed in the abstract. We chose these runs for submission based on the performance of the component features on the TV-2009 data, shown in figures 6 and 7. Since we did not see an improvement in last year's NDCR when fusing with color correlogram, so we chose to rely on GIST and SIFTogram for 2010. The SIFTogram features we generated this year, however, used a different implementation which did not replicate the same feature values, rendering the results not directly comparable to our previous ones, and also, we believe, less effective. For reference we show the best performing submitted TRECVID run in figure 8 as well. Looking at the relative performance of the GIST feature on the 2009 data versus its performance with this year's data, we see an obvious drop. We believe this is the case due to the more varied nature of the year's data. We observed in the results that several other participants, like us, had the same results for all audio transforms of a video query, indicating that they also chose not to process audio. Relative to those runs, our video-only methods seem to have performed well, but relative to the entire set of submissions, there is a gap. We also note that in 2009, excellent performance was achieved by other participants such as CRIM, in large part due to audio processing. We take the lesson that audio features are important, and will investigate how they can be incorporated in our system in the future.

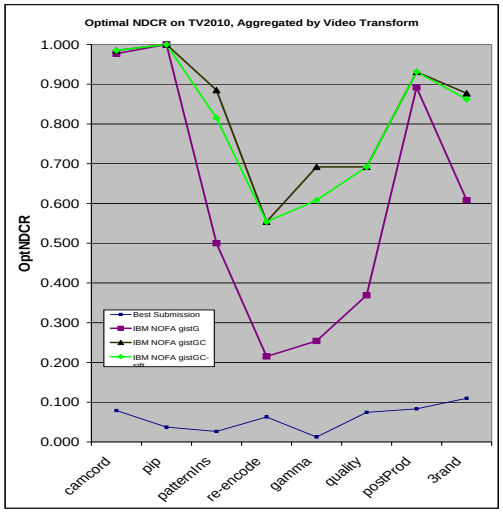

Figure 8: Submission run results on TV-2010 data

3 Multimedia Event Detection

We emphasize three aspects in exploring effective methods for multimedia event detection. Our recognition system incorporates information from a wide range of static and dynamic visual features. In particular, we study event recognition using a large number of semantic detectors, covering scenes, objects, people, and various image types. We find a semantic concept base descriptor to be the best-performing single feature in our comparisons. We analyze the temporal dimension of each video by testing different sampling methods, applying frame-to-video aggregation techniques both at the feature and at the prediction level, and by exploring capturing short-term temporal motifs in video events via sequential itemset mining, and clusters of hierarchical hidden Markov models.

An overview of our event detection framework is shown in Fig. 9. There are three main parts for processing and learning, presented left-to-right in the figure: video processing / feature extraction, model learning and decision aggregation. The rest of this section will discuss each part in detail.

3.1 Video processing

Each input video is processed a number of different ways in order to extract frame-based and dynamic visual features. Our system has three different modes to prepare a video for feature extraction.

- **Uniformly sampled frames.** We decode the video clip, and uniformly save one frame every two seconds. These frames are later used to extract static visual descriptors: local (SIFT), GIST, Global and Semantic Model Vectors.

- **Adaptively sampled keyframes**. We perform shot boundary detection using color histogram differences in adjacent frames, we then take one frame

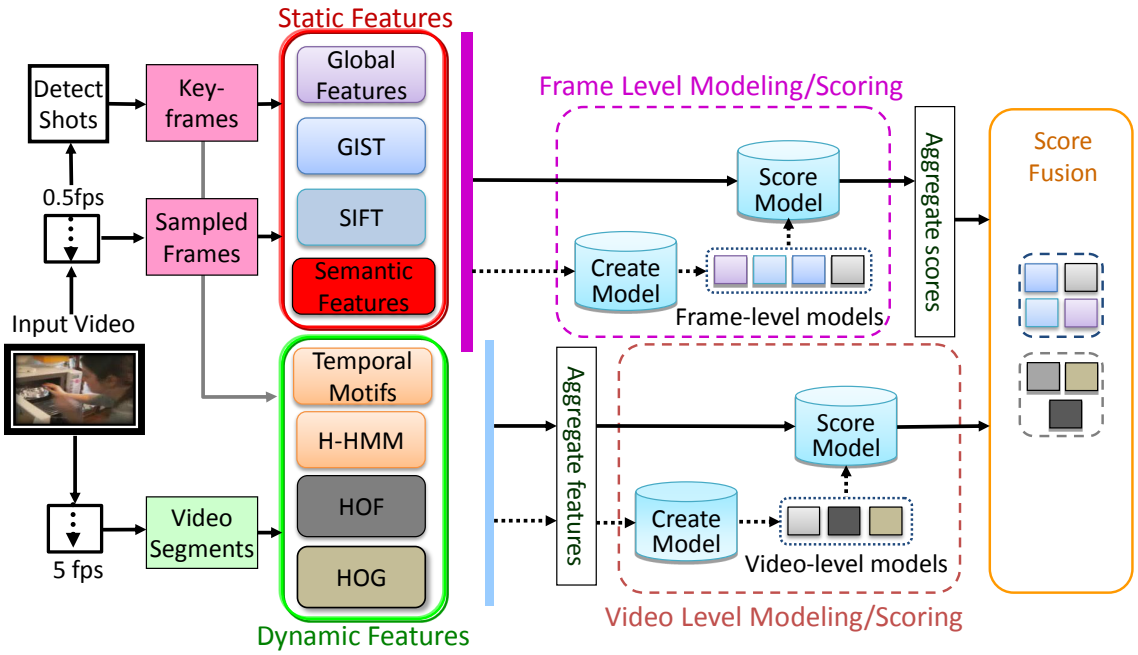

Figure 9: System framework adopted for video event recognition. We investigated multiple layers of operation/representation: video vs frame level, static vs dynamic features, early vs. late aggregation (fusion).

per shot. This frame sampling scheme produces less shots for the event videos since amateur videos tend to have long and unsteady shots. By being temporally adaptive this scheme may decrease overall appearance diversity in the frames, yet it avoids oversampling from long shots.

- **Down-sampled short video segments**. We keep short video segments for extracting spatial temporal features (Sec. 3.3). The video sequence is downsampled to five frames per second to reduce computational time, and the spatial temporal features are extracted within windows of four seconds each.

3.2 Static Features

We extract a large number of static image features from the sampled frames/keyframes. These features capture a wide range of image information including color, texture, edge, local appearances and scene characteristics. We build upon these features to extract the Semantic Model Vectors (Sec 3.4) and carry out a comprehensive comparison of state-of-the-art features for classification.

3.2.1 Local Descriptors

Local descriptors are extracted as SIFT [7] features with dense spatial sampling for keyframes – we use 16 pixels per grid, resulting in approximately 12,000 points per image, and Harris Laplace interest point detection for uniformly sampled frames. Each keypoint is described with a 128-dimensional vector containing oriented gradients. We obtain a "visual keyword" dictionary of size 1000 (for keyframes) and 4000 (for uniformly sampled frames) by running K-means clustering on a random sample of approximately 300K Interest point features, we then represent each frame with a histogram of visual words. For keyframes we used soft assignment following Van Gemert et al. [14] using $\sigma = 90$.
In our combination runs we also included local features

computed by Columbia University, which extracted SIFT with DoG and Hessian detectors at the sampled frames, employed 500-d codebooks, and adopted spatial pyramid matching for the full frame + 4 quadrants, obtaining a 5000-D total feature length.

3.2.2 GIST

The GIST descriptor [10] describes the dominant spatial structure of a scene in a low dimensional representation, estimated using spectral and coarsely localized information. We extract a 512 dimensional representation by dividing the image into a 4x4 grid, we also extract histograms of the outputs of steerable filter banks on 8 orientations and 4 scales.

3.2.3 Global Descriptors

In addition to the SIFT bag-of-words and GIST descriptors, we extracted 13 different visual descriptors on 8 granularities and spatial divisions. SVMs are trained on each feature and subsequently linearly combined in an ensemble classifier. We include a summary of the main descriptors and granularities. Details on features and ensemble classifier training can be found in our prior report [2].

- **Color Histogram**: global color distribution represented as a 166-dimensional histogram in HSV color space.
- **Color Correlogram**: global color and structure represented as a 166-dimensional single-banded autocorrelogram in HSV space using 8 radii depths.
- **Color Moments**: localized color extracted from a 5x5 grid and represented by the first 3 moments for each grid region in Lab color space as a normalized 225-dimensional vector.
- **Wavelet Texture**: localized texture extracted from a 3x3 grid and represented by the normalized 108-dimensional vector of the normalized variances in 12 Haar wavelet sub-bands for each grid region.
- **Edge Histogram**: global edge histograms with 8 edge direction bins and 8 edge magnitude bins, based on a Sobel filter (64-dimensional).

Having a large diversity of visual descriptors is important for capturing different semantics and dynamics in the scene, as so far no single descriptor can dominate across a large vocabulary of visual concepts and events, and using a collection like this has shown robust performance [2, 13]. The spatial granularities include global, center, cross, grid, horizontal parts, horizontal center, vertical parts and vertical center – each of which is a fixed division of the image frame into square blocks (numbering from 1 up to 25), and then concatenating the descriptor vectors from each block. Such spatial divisions has been repeatedly shown robust performance in image/video retrieval benchmarks such as TRECVID [12].

3.3 Dynamic Features

3.3.1 Spatial-Temporal Features

We detect spatial-temporal interest points (STIP) [6] over the down-sampled video segments (Sec. 3.1), within temporal windows of 20 frames (four seconds). We then compute histogram of gradients (HOG) and histogram of flow (HOF) features from spatio-temporal regions localized around each STIP. For both HOG and HOF features we generated a codebook of 1000 words by clustering a data sample of approximately 300K points. We then computed bag-of-words histograms similar to those for the SIFT features in Section 3.2, with soft assignment.

We explored three aggregation methods both for HOG and HOF. The first is to build a single BoW histogram directly for the entire video, resulting in a 1000 dimensional descriptor (named HOG(F)_Pyr0). The second employs the Temporal Pyramid Matching scheme [17], with the video temporally split into 2 and 4 segments. A BoW histogram is computed for each shot, and the descriptors are concatenated and weighted according to the temporal level at which they were computed (0.25 for levels 0 and 1, 0.5 for level 2). As reported in Figure 13, we tested two different pyramidal configurations: HOG(F)_Pyr1x2 (3000 dimensional, with whole video and two halves segments concatenated) and HOG(F)_Pyr1x2x2 (7000 dimensional, with whole video, two halves and four quarters segments concatenated). Since multiple STIP can be detected in the same frame, we also explored computing a BoW histogram for each frame where STIP were found. We then aggregated from frame level to video level using

the same methods employed for the static features and introduced in the next Section, thus obtaining 1000 dimensional vectors. We named descriptors obtained with this third aggregation method simply HOG and HOF.

Columbia University also computed HOG and HOF features following the same bag of words framework. For each descriptor a 4000-D codebook was adopted, and finally HOG and HOF were concatenated in a single descriptor.

3.3.2 Temporal motifs

Intuitively, an event consists of temporal and relational combinations of individual semantic entities. For example, in "two men on a vast ice ground, constructing an igloo with chisels", the individual visual concepts include people, ice, tools, shelter, etc. We propose to use temporal and co-occurrence of concepts to enrich low-level and semantic features. We take the ModelVector stream (extracted at keyframes) for each video, and extract temporal (A followed by B and C) and concurrence (A appears together with B and C) patterns using the sequential pattern mining tool SPAM [1].

3.3.3 Temporal patterns with hierarchical HMMs

In addition to temporal motifs, we'd also like to devise ways to probabilistically represent temporal relationships among individual semantic concepts. Dynamic graphical models are natural tools for this purpose. And localized nonlinear descriptors have recently been shown to be effective tools for recognition tasks in images. We adopt hierarchical hidden Markov models (HHMM) to encode temporal relations, because of its previous success in discovering meaningful features [16, 15] and that there are efficient learning algorithms which are applicable to large datasets. We use an automatic model selection algorithm to obtain the models, then we concatenate the hard- and soft- state assignment for each stream and use a histogram vector to represent a video.

3.3.4 Columbia Audio

Our fusion also included audio features extracted by Columbia University using audio keywords, detected as short-term sound onsets (every 32ms) and described using MFCC [5]. A Bag-of-audio-words model was employed based on a 4000-d audio word codebook.

3.4 Semantic Model Vectors

Intuitively, complex temporal events can be described using a combination of elementary visual concepts, their relationships and evolutions. To this end, we propose an intermediate semantic layer between low-level features and high-level event concepts. This representation, named Semantic Model Vectors, consists of hundreds of discriminative semantic detectors, each coming from an ensemble of SVMs trained from a separate collection of thousands of labeled web images, and from a common collection of global visual features as described in Section 3.2 and prior report [9, 18]. These semantic descriptors cover scenes, objects, people, and various image types. Each of these semantic dimensions provide the ability to discriminate among low-level and mid-level visual cues, even if such discrimination is highly noisy and imperfect for a new data domain. Our hypothesis is that the combination and temporal aggregation of the semantic concepts maps closely to complex video events, for example, *making_cake* event is likely to include *food* in a *kitchen* described with a *hand* closeup. The final Semantic Model Vector descriptor results from the concatenation of the 280 semantic detectors for each frame. Note that this representation (after being aggregated from frame level to video level) is a lot more compact than most descriptors introduced in Sections 3.2 and 3.3, as shown in Figure 11.

3.5 Model learning

One vs all SVMs with RBF kernel were trained, independently for each category, based on each descriptor. During training for one category, all the videos from the other categories (including the *random* one) were used as negative examples. Parameters C and γ were computed though grid search on a 5-fold cross validation, with a 70% training and 30% validation random splits on both positive and negative examples of the development set. Once the best parameters were determined, the SVM were retrained on the whole development set.

Either sampling approach seen in Section 3.1 typically produces multiple frames per video; this yields several

features vectors per video for each descriptor (excluding the Pyramid versions of the HOG and HOF features). Given that the problem we investigate consists in classifying whole videos and not individual frames, an aggregation process from frame level to video level is necessary.

We performed such aggregation both at feature level (early fusion) and at prediction level (late fusion). For all features besides Global, the descriptors extracted from the individual frames were combined through average or max into a single descriptor, representative of the whole video.

We also tested aggregation at prediction level, meaning training a classifier at the frame level and then combining the predictions on the individual frames of a test video into a final score. Such approach was used for the Global descriptor, for which we took the predictions of the ensemble classifier on the frames of a video and averaged them to obtain the score for the video itself.

Finally, we performed late fusion to combine the predictions of models trained on different descriptors, which offer complementary information. First we grouped static features and dynamic features separately, using linear combinations with uniform weights. We then performed late fusion involving all the descriptors in two ways: hierarchical, as a combination of the static and dynamic sets, and horizontal, as a linear combination of all the features.

3.6 Experimental Results and Discussion

In the following we discuss in detail the results emerging from the experiments in terms of Mean Average Precision (MAP).

3.6.1 Individual Descriptors Performance

First we compare the performance of event classifiers based on individual descriptors. As reported in Figure 14, performances vary across categories, with AP rates ranging from 0.15 to 0.3 for *Assembling_Shelter* and *Making_cake*, while *Batting_in_run* is easier to recognize, with AP rates from 0.49 to 0.62. However, some general conclusions can be drawn from the MAP rates. From the results presented in Figure 10 emerges that for any static descriptor, feature extraction on frames obtained by uniform sampling provides better MAP rates than adaptive sampling. Uniform sampling generates a significantly larger number of frames, thus providing richer information to the classifiers. Interestingly, the proposed Semantic Model Vectors outperforms all the other features in terms of Mean Average Precision (0.392), independently from the sampling adopted.

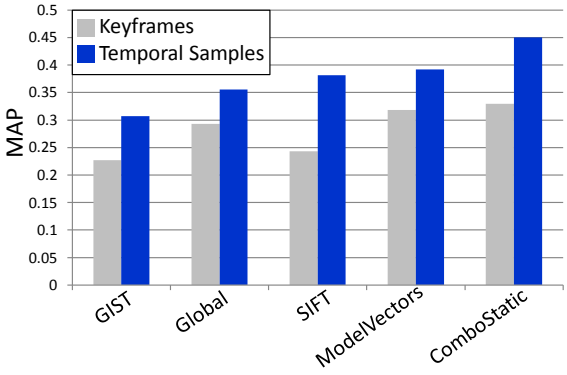

Figure 10: Mean Average Precision comparison between the keyframe and uniform temporal sampling frame selection methods. For each static descriptor we registered a significant improvement when using temporal sampling, and Semantic Model Vectors were the best single descriptor in both cases.

Considering the large scale nature of the video event recognition problem at hand, the space occupied by the feature representation of each video is crucial. In Figure 11 are reported the number of kilobytes necessary to represent each video (after the feature frames to video aggregation), for each descriptor. Semantic Model Vectors can represent an entire video with its 280 dimensional feature vector, making it not only the best performing descriptor in terms of MAP, but also the most compact. SIFT, which is the second best performing descriptor in term of MAP, occupies approximately 15 times the space required by Semantic Model Vectors. The Global descriptor, being an ensemble of multiple descriptors, occupies the largest amount of kilobytes.

3.6.2 Frame to Video Aggregation

As we discussed in Section 3.5, since each feature is extracted at the frame level, we must aggregate them to

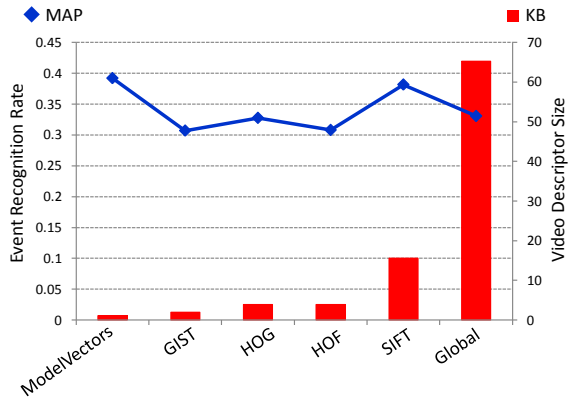

Figure 11: Mean Average Precision vs. Video descriptor size (in kilobytes) based on individual video descriptors. Semantic Model Vectors offer the most compact representation as well as the best recognition performance.

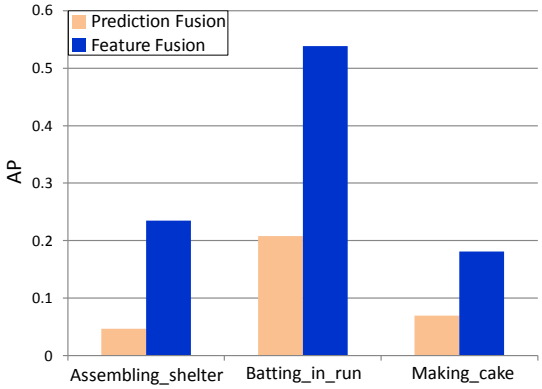

Figure 12: Semantic Model Vectors are extracted at every keyframe, thus require a fusion from frame level to video level. Fusing features from keyframes into a single descriptor per video and learning a classifier on top of it performs significantly better than learning a classifier directly on the frames and then aggregating the predictions from all the frames into a score for the entire video.

determine a single score for each video. We performed an experiment on the Semantic Model Vectors, which is the single best performing descriptor, to determine which aggregation strategy works best. We compare feature level versus SVM predictions aggregation. Aggregation is achieved with an average or max operation.

The AP results outlined in Figure 12 clearly suggest using feature level aggregation for all three categories. Hence we employed this early fusion strategy for all the individual descriptors. The results reported in all the other Figures in this Section besides Figure 12 follow this framework as well.

This result corroborates the initial intuition about the complexity of the events we are examining. Classification on a single or very few keyframes, whose influence would weight considerably in the SVM prediction aggregation stage, is not sufficient to correctly recognize these complex video events. A broader context must be inspected instead. Early fusion (or aggregation at the feature level), allows each frame to contribute significantly to the final video representation, therefore providing a more comprehensive description of the whole event.

3.6.3 Dynamic Features: Temporal Granularity Analysis

As explained in detail in Section 3.3, when considering the bag of words approach for spatial-temporal features, there are different options for building the histogram of codebook words occurrences in a video: to bin all interest points descriptors in a single histogram representing the whole video (HOG(F)_Pyr0), to employ a temporal pyramid matching framework to compute and weight separate histograms for temporally separated segments (HOG(F)_Pyr1x2 and HOG(F)_Pyr1x2x2), or to generate a histogram per spatio-temporal volume where STIPs have been detected, and then perform a frame (where the spatio-temporal cube is centered) to video aggregation similar to what has been done for static features and described in Section 3.6.2 (HOG and HOF).

We compare the MAP performances of such options in Figure 13. The results show a significant predominance of the frame to video aggregation, performed by averaging BoW histograms computed around each frame centered spatio-temporal cube (HOG and HOF, in dark blue). This result reflect the intuition that the inspected videos

are too long and complex to rely on a histogram count over the entire video, or even segments of it which are still too large (one half, on quarter). Such representations tend to weaken the contribution of codebook words which are discriminative for a particular event among the distribution of thousands of keypoint descriptors, many of which are noisy from the point of view of the discriminative representation needed for recognition. In particular, codeword distributions over short (four seconds in our experiments) but potentially significant/discriminative sequences in a video have a reduced weight in standard BoW representations, while they retain a higher weight in the frame to video aggregation.

In order to alleviate this effect in the pyramid type representations, one could think of increasing their granularity by adding further levels in the temporal pyramid. Such an idea has two major limitations. The first is the size of the descriptor: already with a pyramid of depth 2, a 7000 dimensional vector is needed (against the fixed 1000 dimensions of the frame to video aggregation which was used in Figure 11). The second lies in the hierarchical weight given to higher levels in the pyramid. Finer scale matches are weighted much more than coarse matches. This is desirable if the consecutive, uniformly sampled video sequences describing an event are aligned. The large variety both in appearance and length of the inspected videos suggest that this is not necessarily always the case. This could explain why we observed degrading performances for the HOG descriptor as the number of levels in the pyramid increased.

Temporal motifs and HMMM features are extracted starting from the Semantic Model Vector outputs at the keyframe level. Using the temporal motifs as described in Section 3.3.2, we append the binary motif presence/absence to the original ModelVector, and we observed that the detection accuracy improves significantly in batt_in_run, a highly progressive event. Using the methods described in Section 3.3.3, we extract the HHMM features from streams of model vectors. We partition the 272 input dimensions into 15 different clusters, resulting in about 800-900 states among all models. We train an SVM on these redundant state histogram vectors, the MAP (0.25) is comparable to other individual runs, and is used one of the dynamic constituent runs.

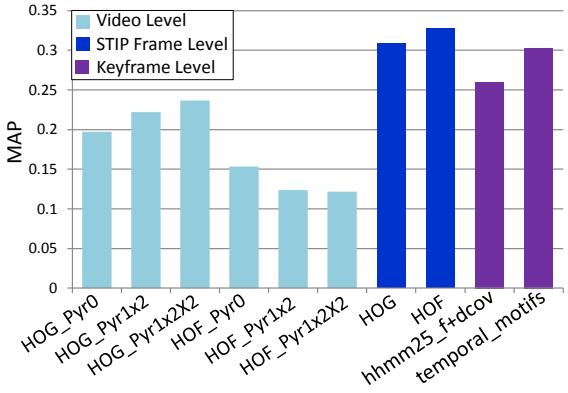

Figure 13: Mean Average Precision Retrieval performances of Dynamic Features: the HOG and HOF extracted at video-level or at frames where STIP keypoints where detected. HMM and temporal motifs operate on ModelVetcors extracted at keyframes. A denser sampling along the time dimension (STIP points) provided the best performances.

3.6.4 Features Fusion

Our baseline approach consisted in training RBF kernel SVMs based on individual descriptors. However, we notice that such descriptor are inherently complementary under different perspectives:

- Semantic Model Vectors operate on a higher semantic level with respect to all the other ones.
- GIST, Global, SIFT, and Semantic Model Vectors are inherently static, as they operate on individual frames, while HOG and HOF are dynamic, as they analyze spatio-temporal volumes within the videos.
- GIST, Global and Semantic Model Vectors are global features that analyze a whole image, while SIFT, HOG and HOF model patches localized around local interest points.

Therefore we applied ensemble late fusion methods to combine all event detection hypotheses generated by the different approaches. We ensured that the scores from all approached were compatible for fusion by applying sigmoid normalization on the non-probabilistic predictors.

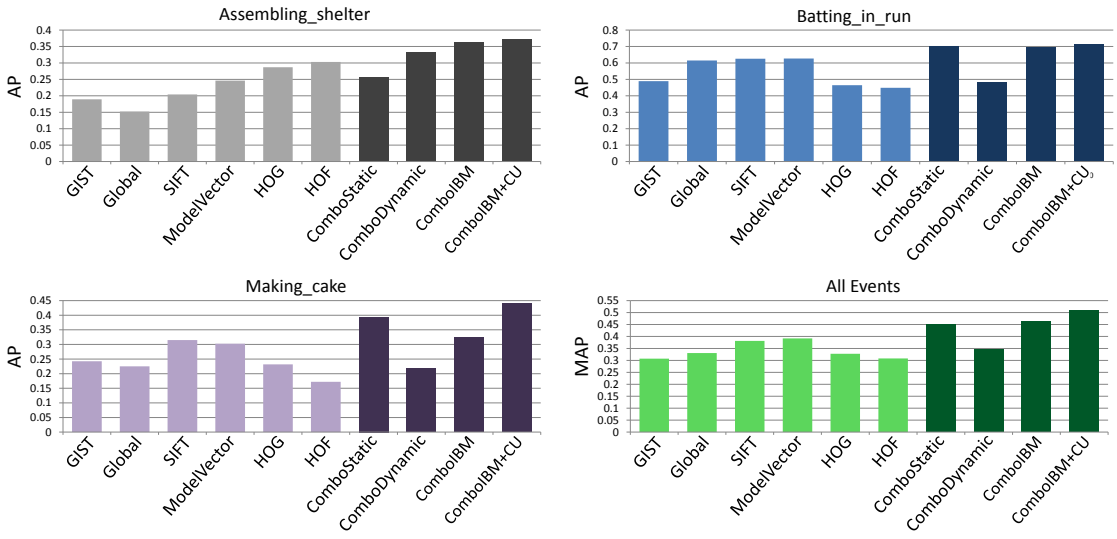

Figure 14: Retrieval performances of different event recognition approaches based on individual features (lighter colors) and their combinations (darker colors). Average precision computed for each category and MAP over the whole TRECVID MED dataset.

Fusion was performed by averaging prediction scores. The mean average precision (MAP) scores are reported in Figure 14.

We observed that a combination of static features (ComboStatic, with Global, SIFT, GIST, Semantic Model Vectors) better models events where intra class variation of iconic objects/settings visual appearance is relatively limited (a cake for *Making_cake*, the baseball field for and players outfits (including helmet and bat) *Batting_in_run*), while combining dynamic ones (ComboDynamic, with HOG, HOG_Pyr1x2, HOF, HOF_Pyr1x2) performs better for ones where the evolution of actions and appearance is more relevant then a single iconic image (*Assembling_shelter*).

The combination proved to boost Average Precision rates with respect to the individual descriptors for all the events. The performance behavior of static and dynamic features appears to be complementary across event categories. Hence we applied a hierarchical fusion (ComboIBM), which combines ComboStatic and ComboDynamic predictions. Such fusion further improved the MAP rate, confirming that complementary nature of static and dynamic features. We also experimented with an aggregation of all the feature prediction directly, without grouping them into subclasses. However, we did not register significant performance differences with respect to the hierarchical fusion.

In all the combination cases inspected, late fusion of multiple descriptors resulted in a boost of MAP with respect to the individual descriptors for all the events in the dataset, thus confirming the complementary nature of such features. The best MAP performance of 0.46 was achieved by fusing all the features. Finally, integrating also the runs produced by Columbia University (audio, additional SIFT and HOG+HOF), we registered an further boost in MAP performances over all events to 0.51.

4 Conclusions

In content based copy detection, we attempted to leverage three types of complementary fingerprints: a keyframe-based color correlogram, SIFTogram (bag of visual words), and a GIST-based fingerprint. Although we did not use audio features in this year's system, we found that

GIST alone performed quite well, better than our other descriptors.

For the MED task overall, the semantic model vector is our best-performing single feature, the dynamic features combination outperform the static features, and temporal motif and hierarchical HMMs shows promising performance. Our best performance was achieved by fusing these complementary methods together.

References

[1] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 429–435, New York, NY, USA, 2002. ACM.

[2] Murray Campbell, Alexander Haubold, Ming Liu, Apostol Natsev, John R. Smith, Jelena Tesic, Lexing Xie, Rong Yan, and Jun Yang. Ibm research trecvid-2007 video retrieval system. *Proc. NIST TRECVID Workshop*, 2007.

[3] M. Fischler and R. Bolles. Random sample consensus... *Communications of the ACM*, 24(6), 1981.

[4] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Spatial color indexing and applications. *International Journal of Computer Vision*, 35(3), December 1999.

[5] Wei Jiang, Courtenay Cotton, Shih-Fu Chang, Dan Ellis, and Alexander C. Loui. Short-term audio-visual atoms for generic video concept classification. In *Proc. ACM Multimedia*, 2009.

[6] Ivan Laptev. On space-time interest points. *Intl Jnl of Computer Vision*, 64(2):107–123, 2005.

[7] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[8] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications (VISAPP'09)*, 2009.

[9] Apostol Natsev, Matthew Hill, John R. Smith, Lexing Xie, Rong, Yan Shenghua Bao, Michele Merler, and Yi Zhang. IBM Research TRECVID-2009 video retrieval system. *Proc. TRECVID Workshop*, 2009.

[10] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

[11] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering object categories in image collections. In *ICCV*, 2005.

[12] Alan F. Smeaton, Paul Over, and Wessel Kraaij. High level feature detection from video in trecvid: a 5-year retrospective of achievements. *Multimedia Content Analysis*, pages 151–174, 2009.

[13] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press, 2010.

[14] Jan C. van Gemert, Cees G. M. Snoek, Cor J. Veenman, Arnold W. M. Smeulders, and Jan-Mark Geusebroek. Comparing compact codebooks for visual categorization. *Computer Vision and Image Understanding*, 2010. In press.

[15] Lexing Xie and Shih-Fu Chang. Pattern mining in visual concept streams. In *Proc. Intl. Conf. on Multimedia and Expo (ICME)*, Toronto, Canada, July 2006.

[16] Lexing Xie, Shih-Fu Chang, Ajay Divakaran, and Huifang Sun. *Unsupervised Mining of Statistical Temporal Structures in Video*, chapter 10. Kluwer Academic Publishers, 2003.

[17] Dong Xu and Shih-Fu Chang. Video event recognition using kernel methods with multilevel temporal alignment. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 30:1985 – 1997, 2008.

[18] Rong Yan, Jelena Tesic, and John R. Smith. Model-shared subspace boosting for multi-label classification. *Proc. ACM SIGKDD Intl Conf on Knowledge Discovery and Data mining*, pages 834 – 843, 2007.