Informedia @ TRECVID 2009:
Analyzing Video Motions

Ming-yu Chen[1], Huan Li[2], and Alexander Hauptmann[1]

[1]School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213

[2]School of Computer Science and Engineering, BeiHang University, Beijing, PRC

Abstract

The Informedia team participated in the tasks of high-level feature extraction and event detection in surveillance video. This year, we especially put our focus on analyzing motions in videos. We developed a robust new descriptor called MoSIFT, which explicitly encodes appearance features together with motion information. For the high-level feature detection, we trained multi-modality classifiers which include traditional static features and MoSIFT. The experimental result shows that MoSIFT has solid performance on motion related concepts and is complementary to static features. For event detection, we trained event classifiers in sliding windows using a bag-of-video-word approach. To reduce the number of false alarms, we aggregated short positive windows to favor long segmentation and applied a cascade classifier approach. The performance shows dramatic improvement over last year on the event detection task.

1 MoSIFT

This section presents our MoSIFT[7] algorithm to detect and describe spatio-temporal interest points. In part-based methods, there are three major steps: detecting interest points, constructing a feature descriptor, and building a classifier. Detecting interest points reduces the whole video from a volume of pixels to compact but descriptive interest points. Therefore, we desire to develop a detection method, which detects a sufficient number of interest points containing the necessary information to recognize a human action. The MoSIFT algorithm detects spatially distinctive interest points with substantial motion. We first apply the well-known SIFT algorithm to find visually distinctive components in the spatial domain and detect spatio-temporal interest points with (temporal) motion constraints. The motion constraint consists of a 'sufficient' amount of optical flow around the distinctive points. Details of our algorithm are described in the following sections.

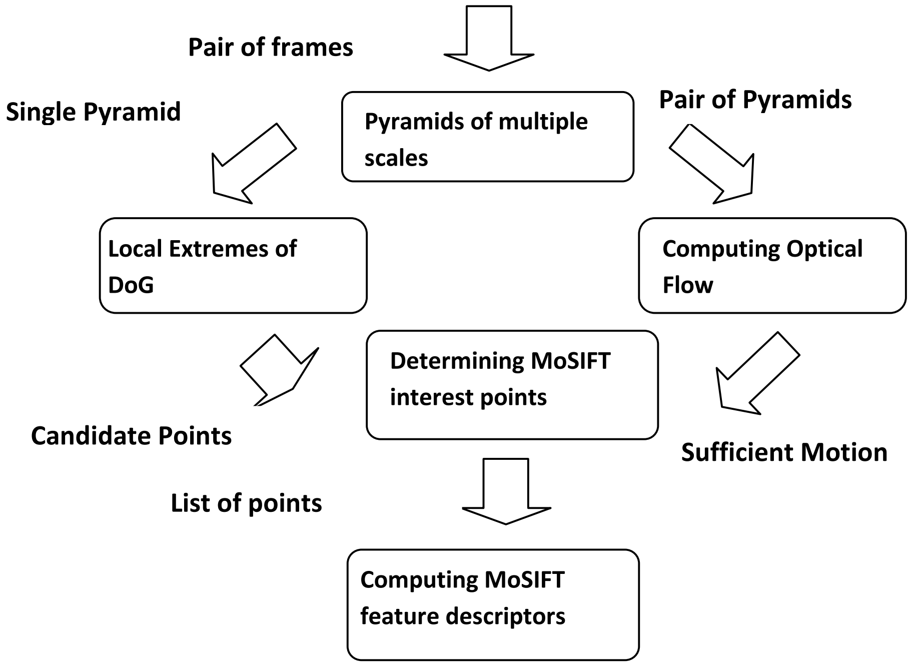


Figure 1: System flow graph of the MoSIFT algorithm. A pair of frames is the input. Local extremes of DoG and optical flow determine the MoSIFT points for which features are described.

1.1 MoSIFT interest point detection

Figure 1 demonstrates our MoSIFT algorithm. The algorithm takes a pair of video frames to find spatio-temporal interest points at multiple scales. Two major computations are applied: SIFT point detection and optical flow computation matching the scale of the SIFT points.

SIFT interest points are scale invariant and all scales of an image must be considered. Lowe [1] used a Gaussian function as a scale-space kernel to produce a scale space of the image. The whole scale space is divided into a sequence of octaves and each octave is divided into a sequence of intervals, where each interval is a scaled frame. The number of octaves and intervals is determined by the frame size. The size relationship between two adjacent octaves is in powers of 2. The first interval in the first octave is the original frame. In each octave, the first interval is denoted as $I(x,y)$. We can denote each interval as

$$L(x,y,k\delta) = G(x,y,k\delta) * I(x,y) \qquad (1)$$

where $*$ is the convolution operation in x and y, and $G(x,y,k\delta)$ is a Gaussian smoothing function. Difference of Gaussian (DoG) images are then computed by subtracting adjacent intervals

$$D(x,y,k\delta) = L(x,y,k\delta) - L(x,y,(k-1)\delta) \qquad (2)$$

Once the pyramid of DoG images has been obtained, the local extremes (minima/maxima) of the DoG images across adjacent scales are used as the interest points. This is done by comparing each pixel in the DoG images to its eight neighbors at the same interval and nine corresponding neighboring pixels in each of the neighboring intervals. The algorithm scans through each octave and interval in the DoG pyramid and detects all possible interest points at different scales.

However, SIFT is designed to detect distinctive interest points in a still image. The candidate points are distinctive in appearance, but they are independent of the motions or actions in video. For example, a cluttered background can produce many interest points unrelated to human actions. Clearly, only interest points with sufficient motion will provide the necessary information for action recognition. The widely used optical flow approach detects the movement of a region by calculating where a region moves in the image space by measuring temporal differences. Compared to video cuboids or volumes, optical flow explicitly captures the magnitude and direction of a motion, instead of implicitly modeling motion through appearance change over time. Our belief is that explicit motion measurement is essential for recognizing actions.

In the interest point detection part of the MoSIFT algorithm, optical flow pyramids are constructed over two Gaussian pyramids. Multiple-scale optical flows are calculated according to the SIFT scales. A local extreme from DoG pyramids can only become an interest point if it has sufficient motion in the optical flow pyramid. We assume that a complicated action can be represented by the combination of a reasonable number of interest points. Therefore, we do not assign strong constraints to spatio-temporal interest points. As long as a candidate interest point contains a minimal amount of movement, the algorithm will extract this point as a MoSIFT interest point. MoSIFT interest points are scale invariant in the spatial domain. However, they are not scale invariant in the temporal domain. Temporal scale invariance could be achieved by calculating optical flow on multiple scales in time. However, we want to select distinctive interest points with sufficient motion such that, ideally, humans could ‘recognize’ the action based on seeing these points, giving us reason to believe that machines should be able to learn a corresponding action model. Therefore, a small motion is sufficient at each interest point rather than imposing a complex motion constraint. Ultimately, this is still an open research topic.

1.2 MoSIFT feature description

In most current work on action recognition, much emphasis is placed on interest point detection and action model learning. However, the feature descriptor is an important component which is only given cursory attention. Most work [8,9,11,12] uses histograms of gradients to describe the appearance of interest volumes or cuboids. Some recent work [13,14] includes histograms of optical flow to boost performance.

Since MoSIFT point detection is based on DoG and optical flow, it is natural that our descriptor should leverage these two features. Instead of combining a complete HoF classifier with a complete HoG classifier, we build a single feature descriptor, which concatenates both HoG and HoF into one vector, which is also known as ‘early fusion’. We believe appearance and motion information together are the essential components for a classifier. Since an action is only

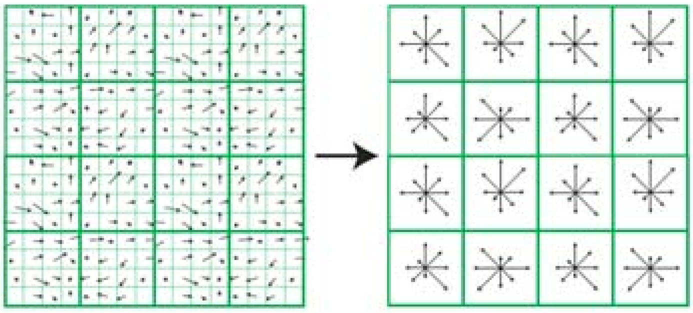


Figure 2: Grid aggregation for SIFT/MoSIFT feature descriptors. Pixels in a neighborhood are grouped into 4x4 regions. An orientation histogram with 8 bins is formed for each region resulting in a 128 element vector. MoSIFT concatenates aggregated grids for both appearance and motion for a 256 element descriptors vector.

represented by a set of spatio-temporal point descriptors, the descriptor features critically determine the information used in later recognition steps.

It is at times underappreciated that the original SIFT descriptor captures local appearance with an aggregated histogram of gradients from neighboring regions. This gives the SIFT descriptor better tolerance to partial occlusion and deformation. When an interest point is detected, a dominant orientation is calculated and all gradients in the neighborhood are rotated according to the dominant orientation to achieve rotation invariance. The magnitude and direction for the gradient are calculated for every pixel in a region around the interest point in the Gaussian-blurred image L. An orientation histogram with 8 bins is formed, with each bin covering 45 degrees. Each sample in the neighboring window is added to a histogram bin and weighted by its gradient magnitude and its distance from the interest point. Pixels in the neighboring region are normalized into 256 (16x16) elements. Elements are grouped as 16 (4x4) grids around the interest point. Each grid has its own orientation histogram to describe sub-region orientation. This leads to a SIFT feature vector with 128 dimensions (4x4x8 = 128). Each vector is normalized to enhance invariance to changes in illumination. Figure 2 illustrates the SIFT descriptor grid aggregation idea.

MoSIFT adapts the idea of grid aggregation in SIFT to describe motions. Optical flow detects the magnitude and direction of a movement. Thus, optical flow has the same properties as appearance gradients. The same aggregation can be applied to optical flow in the neighborhood of interest points to increase robustness to occlusion and deformation. The main difference to appearance description is in the dominant orientation. Rotation invariance is important to appearance since it provides a standard to measure the similarity of two interest points. In surveillance video, rotation invariance of appearance remains important due to varying view angles and deformations. Since surveillance video is captured by a stationary camera, the direction of movement is actually an important (non-invariant) vector to help recognize an action. Therefore, we omit adjusting for orientation invariance in the MoSIFT motion descriptors. The two aggregated histograms (appearance and optical flow) are combined into the MoSIFT descriptor, which now has 256 dimensions.

2 High-Level Feature Extraction

This year, we submitted 6 runs to the high-level semantic features with 6 different low level features. Our low level features come from static image, motion and audio.

2.1 Description of 6 runs

- A_CMU1_1: SIFT feature alone, trained with x^2 kernel for each high-level feature.
- A_CMU2_2: MOSIFT feature alone, trained with x^2 kernel for each high-level feature.
- A_CMU3_3: Meta fusion of A_CMU1_1 and A_CMU2_2 for each high-level feature.
- A_CMU4_4: Meta fusion of A_CMU3_3 with Support Vector Machine (SVM) classification results of color feature and texture feature.
- A_CMU5_5: Meta fusion of A_CMU4_4 with SVM classification results of audio feature and face feature.
- A_CMU6_6: Select the best performing classifier (on the training data) for each high-level feature by using different feature combinations and late fusion strategies.

2.2 Low-level features

2.2.1 Grid-based color comments (GCM)

To generate the color moment feature, each image (key-frame) is divided into 5x5 grids, and each grid is described by the mean, standard deviation, and third root of the skewness of each color channel in the LUV color space. This results in a 225-dimension (5x5x3x3) color moment feature.

2.2.2 SIFT feature

The local feature of each image is computed from the local key points detected from the image. We use the key points using the DoG detector and depicted by Scale-invariant feature transform (SIFT) descriptors [1] which describes each key points by a 128-dimension vector. SIFT features are invariant to image scale and rotation, and are also robust to changes in illumination, noise, occlusion and minor changes in viewpoint. For each key frame, the number of extracted key points is different. Therefore, we try to use *bag-of-words* (BoW) to quantify SIFT feature to a fixed number vector feature of each key frame. We use K-means clustering to find the conceptual meaningful clusters and each cluster is treated as a visual word in BoW approach. All the visual words consist of a visual word vocabulary. Then key points in each key frame are assigned to clusters in the visual vocabulary which are their nearest neighbors. In the end, each key frame is presented by a visual word histogram feature. The performance of BoW in high-level feature detection in large-scale multimedia corpus is subject to several aspects, such as the size of the visual word vocabulary, visual word weighting scheme, etc. We discuss these factors below.

- **Size of vocabulary**

While text vocabulary size is relatively fixed in text information retrieval, the size of a visual words vocabulary is decided by the number of clusters generated by clustering process. Choosing vocabulary size is a trade-off between discrimination and generalization. A small vocabulary is less discriminative since two keypoints may be assigned into one cluster even if they are not similar to each other. On the other hand, a large vocabulary may lack of generalizability since similar keypoints may be assigned to different clusters. And it also increases the cost associated with clustering keypoints, assigning each keypoint to the cluster and running supervised learning with high dimension features. Our previous work shows using a moderate visual word vocabulary size lead a better performance. So we cluster the key points into 1000 clusters and at the same time we use distributed processing to reduce the computation time.

- **Soft cluster boundary**

Term weighting is a critical problem in text information retrieval. Term frequency (tf) and inverse document frequency (idf) are mostly used with BOW features. Essentially, this term weighting scheme assigns a key point to its nearest neighbor cluster without considering the relationship between this keypoint to other nearby clusters. This kind of assignments is called *hard boundary* which ignores the information of other nearest neighbors, e.g., the second nearest cluster.

In our experiment, we consider N nearest neighbors of one key point and assign different weights to clusters according to their distance rank. For each key point in an image, we select N (N=4) nearest neighbor clusters for it. These N nearest neighbor clusters are then assigned weights with their inverse rank value. The final weight of each cluster is the sum of inverse rank values calculated from all the keypoints in an image belong to it. Suppose we have a visual vocabulary of K visual words, we have a K-dimension feature vector $v=\{v_1,v_2,...,v_k\}$ for each image where each term represents the weight of *i*th visual word in the image,

$$v_k = \sum_{i=1}^{N}\sum_{j=1}^{B_i}\frac{1}{2^{i-1}} \qquad (3)$$

where B_i represents the number of key points whose *i*th nearest neighbor is *k.* In particular, normalization factor is significant since different images may have different number of key points. Even among images of the same size, the number of keypoints varies according to the complexity of the image content. Normalization eliminates such difference. So we normalize our term weight such as

$$v_k^{'} = \frac{v_k}{\sum_{m=1}^{M}\sum_{i=1}^{N}\frac{1}{2^{i-1}}} \qquad (4)$$

where M is the number of key points in an image. We apply both hard boundary and soft boundary to calculate the term weight. The result is shown in Table 1.

Assignment	Descriptor	Construction	Vocabulary size	MAP
Hard boundary	SIFT	K-MEANS	1000	0.068
Soft boundary	SIFT	K-MEANS	1000	0.089

Table 1: The Mean Average Precision (MAP) of two different weighting scheme

2.2.3 MOSIFT feature

MoSIFT is described in section 1. It is a descriptor which explicitly describes appearance and motion for a region which contains abundant information. MoSIFT is represented as a bag-of-word feature too for each shot. We also apply soft boundary to form the bag-of-word feature.

2.2.4 Texture feature

The texture features are obtained from the convolution of the image pixels with Gabor wavelet filters. We compute it in 7*7 image grids. In each grid we use the mean and variance of twelve oriented energy filters aligned in 30-degree intervals.

2.2.5 Face feature

Schneiderman's face detector [15] is used to detect the faces from the video key frames with a confidence score, pose, scale and location for each detected face in the keyframe.

2.2.6 Audio feature

Mel-frequency cepstral coefficients (MFCCs) [16] are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). The frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound, for example, in audio compression . We create a 20-dimensional MFCCs feature for each key frame based on this.

2.3 Kernel-based learning

Similar to previous years, we evaluate a set of SVMs with different kernels using different features and model parameters for each high-level feature. For this, we use the LIBSVM implementation [2] of SVM with probabilistic output [3, 4].

- **Cross-validation**

The parameters of SVM are well known to have a significant influence on performance. For each parameter combination, we compute its performance in TRECVID 2007 development data using a 2-fold cross-validation to prevent over-fitting. Performance is measured by average precision (AP). We select the parameter combination that yields the best performance to train SVM models for each high-level feature using TRECVID 2007 development data and test data. This results in models which we will then use for late fusion.

- **SVM kernels**

We divide all these local features and global features into two categories: histogram features and non-histogram features. Histogram features are features such as SIFT and MOSIFT features which are represented by frequencies of the visual words in an image. Non-histogram features are features such as the grid-based color moments feature which is a concatenation feature over all grids. For histogram features, we apply a x^2 kernel in SVM because it has been shown to be better for calculating histogram distances [5]. The x^2 kernel is defined as

$$K(x_i, x_j) = \exp\left(-\frac{1}{A} D(x_i, x_j)\right) \qquad (5)$$

where A is a scaling parameter that can be determined though cross-validation. $D(x_i, x_j)$ is the x^2 distance defined as:

$$D(x_i, x_j) = \frac{1}{2}\sum_{i=1}^{m} \frac{(u_i - w_i)^2}{u_i + w_i} \qquad (6)$$

with $x_i = (u_1, \ldots, u_m)$ and $x_j = (w_1, \ldots, w_m)$.
Radian basis kernel (RBF) in SVM is applied for non-histogram features.

2.4 Late fusion

It was shown by Snoek [6] that late fusion frequently has better performance for most high-level features than early fusion. Therefore, we use 2 kinds of late fusion strategies to combine the prediction results from different low-level features. One strategy is named Meta fusion which takes the component probability output as input and outputs an overall prediction. The other one is named Borda-rank which uses the value of the inverse rank instead of the probability output as input. For both strategies we use SVM to train the final prediction model. TRECVID2007 test data are used for a 2-fold cross validation with an RBF kernel, from which select the one with the best performance parameters. Since different low-level features have different prediction performance, we select different combinations of these features for late fusion. Meta fusion of the SIFT feature, MOSIFT feature, color feature and the texture feature exhibits the highest MAP which is 0.139 in our test evaluation using TRECVID2007 test data.

3 Experimental Results

Figure 3 shows an overview of all high-level feature submitted runs. Our 6 runs are the 6 black bars. A_CMU2_2 run shows the best performance with MAP of 0.112 in our 6 runs. The MAP of the worst run A_CMU1_1 is much lower than the others. Figure 4 shows the top MAP, the average MAP of all the submitted runs and our best MAP for each high-level feature. Some high-level features such as People-dancing, Hand, Airplane_flying, Person-playing-soccer, our performance are quite close to the best result. But for some other features such as Chair, Classroom, and Singing, our performance is far from the best. An overview of our 6 runs is depicted in Figure 5. A_CMU2_2 and A_CMU5_5 did best for 8 high-level features in our 6 runs.

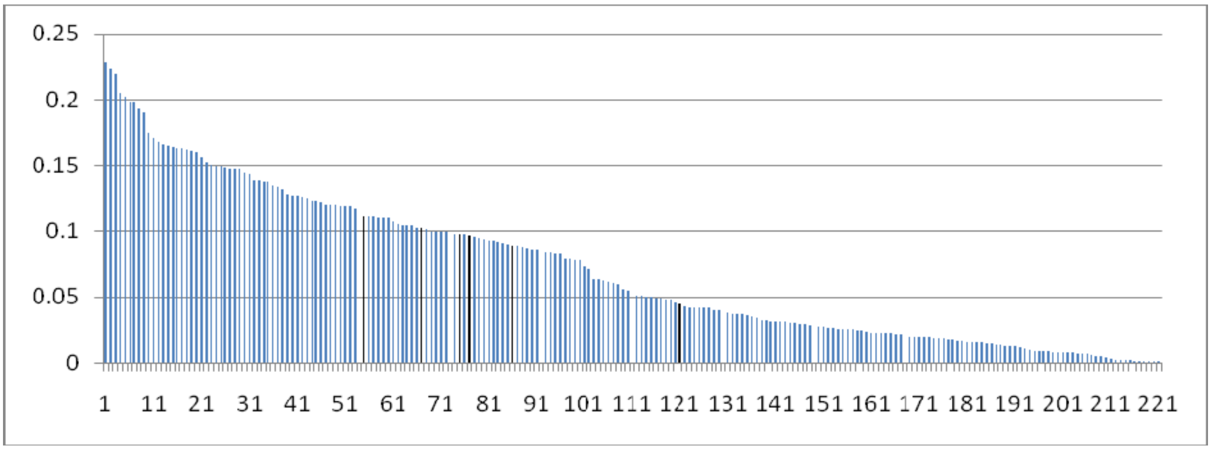


Figure 3: Performance of our 6 runs in all submitted runs

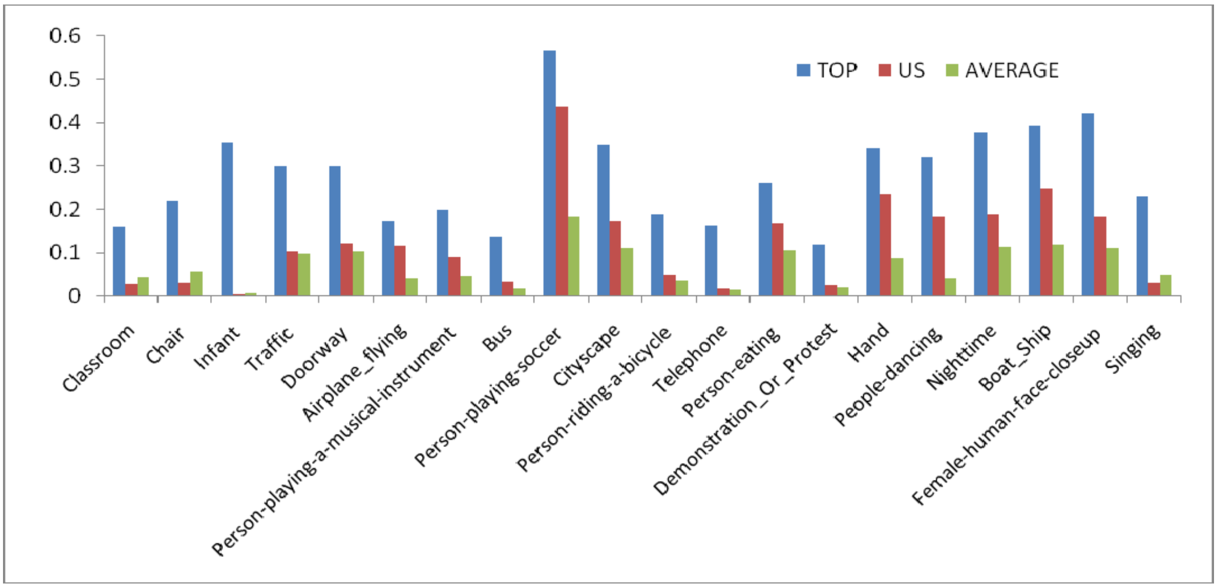


Figure 4: Best MAP, average MAP and our best MAP for each high-level feature

3.1 Conclusions

- Soft boundary helps.
  This year we used a soft boundary for assignment of key points to clusters and the performance increased of 31.4% compared to hard-boundary assignment.
- MOSIFT feature helps.
  A_CMU2_2 achieved the highest MAP and had the best performance for 8 high-level features in our submitted 6 runs. It solely used the MOSIFT feature.
- Reduced computation time.
  We pre-compute kernel matrix which reduces SVM computation time. A distributed, parallel computation of k-means reduced clustering computation time.

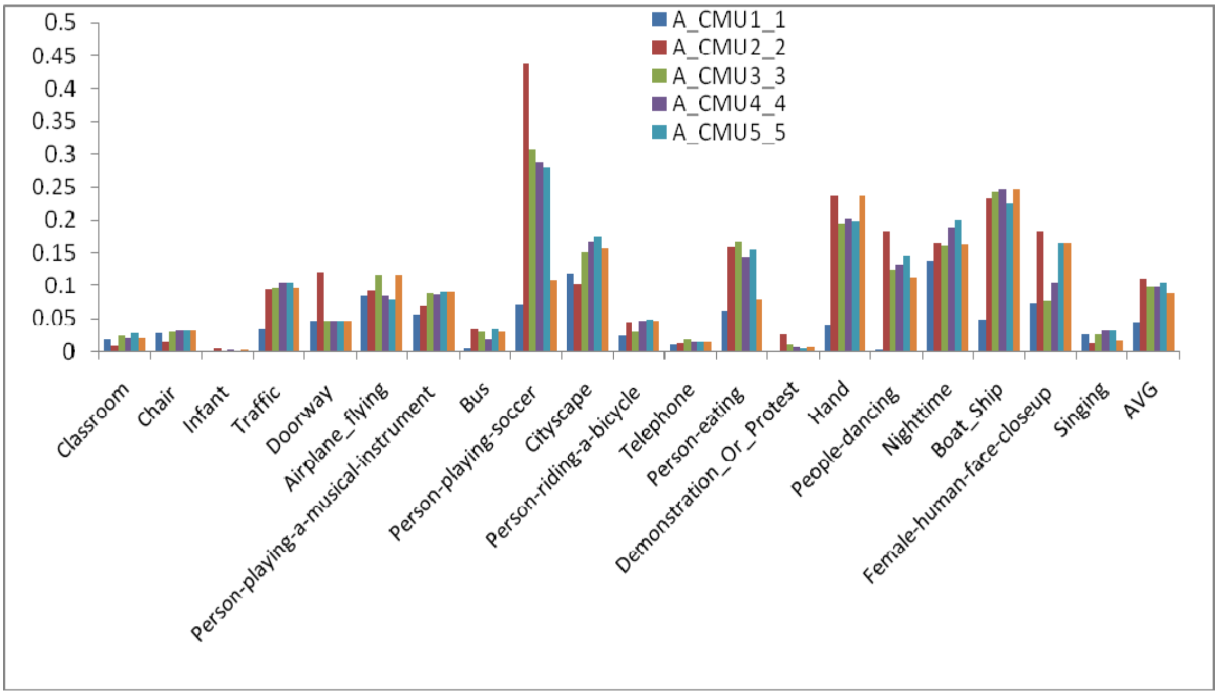


Figure 5: MAP of each high-level feature of our 6 runs

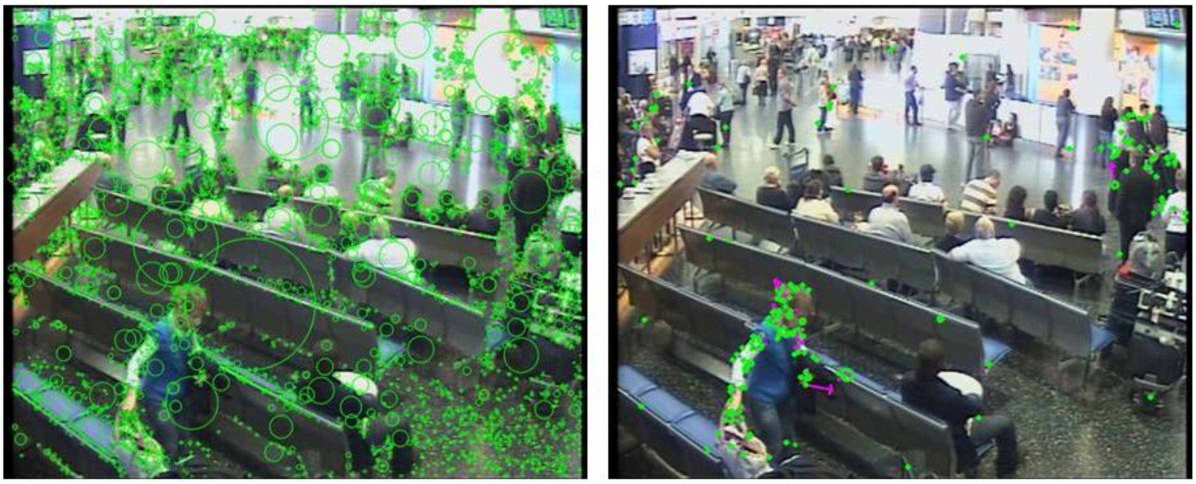


Figure 6: Interest points detected with SIFT (left) and MoSIFT (right). Green circles denote interest points at different scales while magenta arrows illustrate optical flow. Note that MoSIFT identifies distinctive regions that exhibit significant motion, which corresponds well to human activity while SIFT fires strongly on the cluttered background.

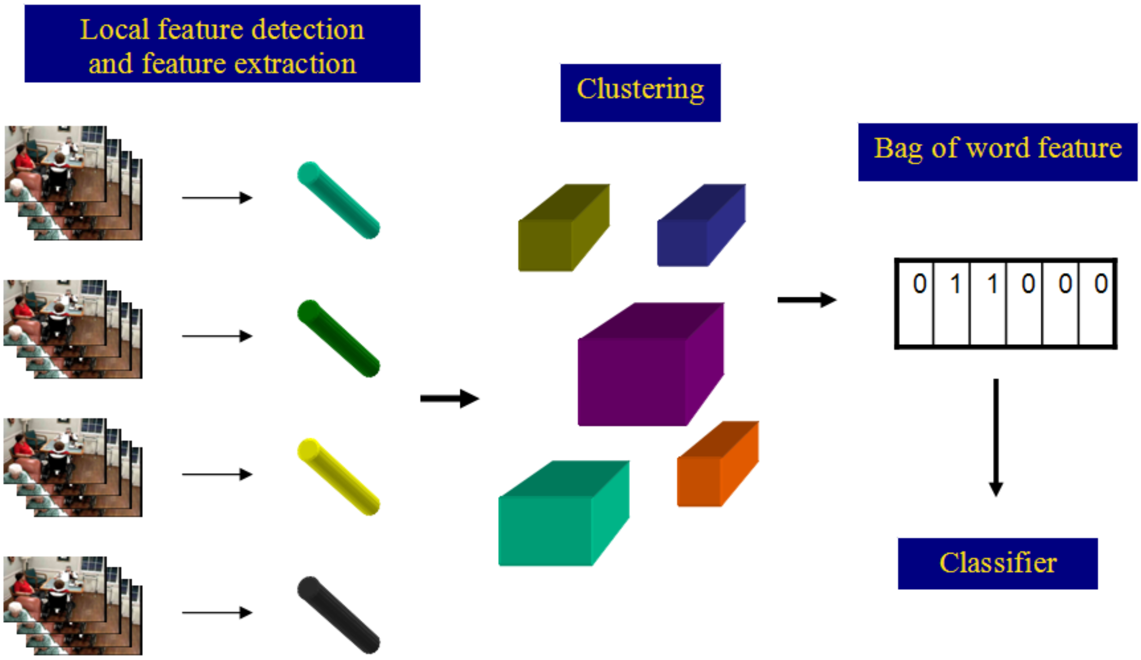


Figure 7: Framework of the proposed event recognition. It includes 3 major stages: (1) MoSIFT feature detection and extraction, (2) clustering and bag-of-words representation based on video codebooks, and (3) classification to achieve event recognition.

4 Event Detection

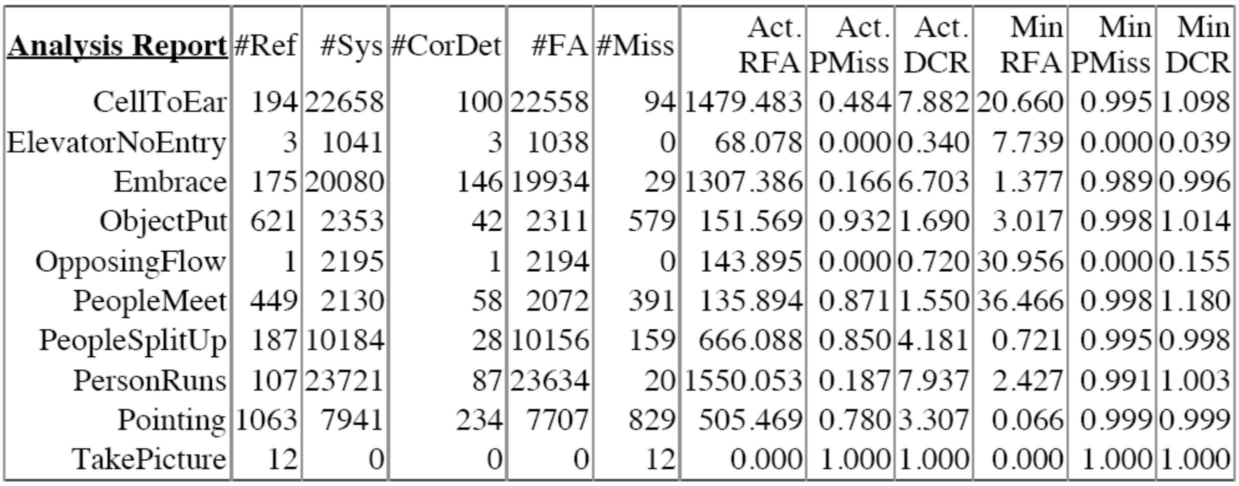
Our event detection uses a sliding window framework is applied to extend the MoSIFT recognition algorithm to a detection task. Our submission started with MoSIFT interest points in each window, clustered them into visual keywords, and used a classifier to detect events based on trained SVM models. Figure 6 shows our MoSIFT features in a Gatwick video key frame. It shows that MoSIFT features is able to clearly focus on areas with human activity.

4.1 Event recognition

We characterize events in surveillance video through the use of MoSIFT features. Each MoSIFT feature captures a small but informative motion in the video. We assume that an event can be described though a combination of these different types of small motions. MoSIFT is a scale-invariant local feature which is less affected by global appearance, posture, illumination and occlusion. Figure 7 illustrates our framework for event recognition. Similar to high-level semantic feature extraction, we apply a soft boundary to form our bag-of-word features. We also apply a x^2 kernel SVM and one-against-all strategy to construct action models.

4.2 Event detection

We use a sliding window to accomplish detection task. The size of the window is 25 frames (1 second) and it repeats every 5 frames. In the training set, annotations are distributed to each window to mark it as positive or negative. This creates a highly unbalanced dataset (positive windows are much less frequent than negative windows). Therefore, we build a two layer cascade classifier to overcome this imbalance in the data and reduce false alarms. For each layer, we choose an equal ratio of (positive v.s. negative) training data to build a classifier to favors to positive examples. This leads the classifier with high detection rates. By cascading two layers of these high detection rate classifiers, we can efficiently eliminate a good amount of false positives without losing too many detections. The computational expense prevented us from computing more than two layers. We also aggregate consecutive positive predictions to achieve multi-resolution. The detection result is in the table 2. From table 2, five of ten events are less than 1 in MinDCR, which is informally equivalent to random performance. Compared with our result from last year, MoSIFT and the cascade classifier significantly improved our performance.



Analysis Report	#Ref	#Sys	#CorDet	#FA	#Miss	Act. RFA	Act. PMiss	Act. DCR	Min RFA	Min PMiss	Min DCR
CellToEar	194	22658	100	22558	94	1479.483	0.484	7.882	20.660	0.995	1.098
ElevatorNoEntry	3	1041	3	1038	0	68.078	0.000	0.340	7.739	0.000	0.039
Embrace	175	20080	146	19934	29	1307.386	0.166	6.703	1.377	0.989	0.996
ObjectPut	621	2353	42	2311	579	151.569	0.932	1.690	3.017	0.998	1.014
OpposingFlow	1	2195	1	2194	0	143.895	0.000	0.720	30.956	0.000	0.155
PeopleMeet	449	2130	58	2072	391	135.894	0.871	1.550	36.466	0.998	1.180
PeopleSplitUp	187	10184	28	10156	159	666.088	0.850	4.181	0.721	0.995	0.998
PersonRuns	107	23721	87	23634	20	1550.053	0.187	7.937	2.427	0.991	1.003
Pointing	1063	7941	234	7707	829	505.469	0.780	3.307	0.066	0.999	0.999
TakePicture	12	0	0	0	12	0.000	1.000	1.000	0.000	1.000	1.000

Table 2: RFA denotes Rates of False Alarms. PMiss denotes probability of missed event. DCR denotes Detection Cost Rate.

5 Acknowledgments

This work was supported by the Nation Science Foundation under Grant No. IIS-0205219 and Grant No. IIS-0705491.

Reference:

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[3] H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on Platt's probabilistic outputs for support vector machines. ML, 68(3):267–276, 2007.

[4] J. C. Platt. Probabilities for SV machines. In A. J. Smola, P. L. Bartlett, B. Scholkopf, and D. Schuurmans, editors, Advances in Large Margin Classifiers, pages 61–74. The MIT Press, 2000.

[5] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV, 73(2):213–238, 2007.

[6] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders, "Early versus Late Fusion in Semantic Video Analysis," in Proceedings of the ACM International Conference on Multimedia, Singapore, 2005, pp. 399-402.

[7] Ming-yu Chen and Alex Hauptmann, " MoSIFT: Reocgnizing Human Actions in Surveillance Videos ". CMU-CS-09-161, Carnegie Mellon University, 2009

[8] I. Laptev, and T. Lindeberg. Space-time interest points, In ICCV, p. 432-439, 2003

[9] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In ICPR, 32-36, 2004

[10] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In ICCV, 166-173, 2005

[11] P. Dollár, V. Rabaud, G. Gottrell, and S. Belongie. Behavior Recognition via Sparse Spatio-Temporal Features, In VS-PETS 2005, page 65-72

[12] J. C. Nibles, H. Wang, and L. F.-F. Li. Unsupervised learning of human action categories using spatial-temporal words. In BMVC, 2006.

[13] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In CVPR 2008

[14] K. Schindler, and L.V. Gool. Action Snippets: How many frames does human action recognition require? In CVPR 2008

[15] Schneiderman, H., Kanade, T. Object detection using the statistics of parts. Int'l J. of Comp. Vision, 56(3): 151-177, 2002.

[16] G. Tzanetakis and P. Cook, "A Framework for Audio Analysis," Organized Sound, vol.4, no. 3, 2000, pp. 169–175.