# Informedia @ TRECVID 2011

Lei Bao<sup>2,3,4</sup>, Longfei Zhang<sup>1,2</sup>, Shoou-I Yu<sup>2</sup>, Zhen-zhong Lan<sup>2</sup>, Lu Jiang<sup>2</sup>, Arnold Overwijk<sup>2</sup>, Qin Jin<sup>2</sup>, Shohei Takahashi<sup>5</sup>, Brian Langner<sup>2</sup>, Yuanpeng Li<sup>2</sup>, Michael Garbus<sup>2</sup>, Susanne Burger<sup>2</sup>, Florian Metze<sup>2</sup>, and Alexander Hauptmann<sup>2</sup>

<sup>1</sup> School of Software, Beijing Institute of Technology, Beijing, 100081, P.R China
 <sup>2</sup> School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213, USA
 <sup>3</sup> Laboratory for Advanced Computing Technology Research, ICT, CAS, Beijing 100190, China
 <sup>4</sup> Graduate University of Chinese Academy of Sciences, Beijing 100049, China
 <sup>5</sup> Graduate School of Global Information and Telecommunication Studies, Waseda University, Tokyo, Japan

The Informedia group participated in three tasks this year, including Multimedia Event Detection (MED), Semantic Indexing (SIN) and Surveillance Event Detection (SED). The first half of the report describes our efforts on MED and SIN, while the second part discusses our approaches to SED.

For Multimedia Event Detection and Semantic Indexing of concepts, generally, both of these tasks consist of three main steps: extracting features, training detectors and fusion. In the feature extraction part, we extracted many low-level features, high-level features and text features. Specifically, we used the Spatial-Pyramid Matching technique to represent the low-level visual local features, such as SIFT and MoSIFT, which describe the location information of feature points. In the detector training part, besides the traditional SVM, we proposed a Sequential Boosting SVM classifier to deal with the large-scale unbalanced data classification problem. In the fusion part, to take the advantage of different features, we tried three different fusion methods: early fusion, late fusion and double fusion. Double fusion is a combination of early fusion and late fusion. The experimental results demonstrated that double fusion is consistently better than or at worst comparable to early fusion and late fusion.

The Surveillance Event Detection report in the second half of this paper presents a generic event detection system evaluated in the SED task of TRECVID 2011. We investigated a generic statistical approach with spatio-temporal features applied to seven events, which were defined by the SED task. This approach is based on local spatio-temporal descriptors, called MoSIFT, and generated from pair-wise video frames. Visual vocabularies are generated by cluster centers of MoSIFT features, which were sampled from the video clips. We also estimated the spatial distribution of actions by over-generated person detection and background subtraction. Different sliding window sizes and steps were adopted for different events based on the event duration priors. Several sets of one-against-all action classifiers were trained using cascade non-linear SVMs and Random Forests, which improved the classification performance on unbalanced data just like the SED datasets. Results of 9 runs were presented with variations in i) sliding window size ii) step size of BOW, iii) classifier threshold and iv) classifiers. The performance shows improvement over last year on the event detection task.

Acknowledgments: This work was supported in part by the National Science Foundation under Grant No. IIS-0205219 and Grant No. IIS-0705491. The work was also supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

# Informedia @ TRECVID 2011 Multimedia Event Detection, Semantic Indexing

Lei Bao<sup>1,2,3</sup>, Shoou-I Yu<sup>1</sup>, Zhen-zhong Lan<sup>1</sup>, Arnold Overwijk<sup>1</sup>, Qin Jin<sup>1</sup>, Brian Langner<sup>1</sup>, Michael Garbus<sup>1</sup>, Susanne Burger<sup>1</sup>, Florian Metze<sup>1</sup>, Alexander Hauptmann<sup>1</sup> <sup>1</sup>Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA <sup>2</sup>Laboratory for Advanced Computing Technology Research, ICT, CAS, Beijing 100190, China <sup>3</sup>Graduate University of Chinese Academy of Sciences, Beijing 100049, China

# Abstract

We report on our results in the TRECVID 2011 Multimedia Event Detection (MED) and Semantic Indexing (SIN) tasks. Generally, both of these tasks consist of three main steps: extracting features, training detectors and fusing. In the feature extraction part, we extracted many low-level features, high-level features and text features. We used the Spatial-Pyramid Matching technique to represent the low-level visual local features, such as SIFT and MoSIFT, which describe the location information of feature points. In the detector training part, besides the traditional SVM, we proposed a Sequential Boosting SVM classifier to deal with the large-scale unbalanced classification problem. In the fusion part, to take the advantages from different features, we tried three different fusion methods: early fusion, late fusion and double fusion. Double fusion is a combination of early fusion and late fusion. The experimental results demonstrated that double fusion.

# **1** Multimedia Event Detection (MED)

#### 1.1 Feature Extraction

In order to encompass all aspects of a video, we extracted a wide variety of visual and audio features as shown in figure 1.

	Visual Features	Audio Features
Low-level Features	<ul> <li>SIFT [19]</li> <li>Color SIFT [19]</li> <li>Transformed Color Histogram [19]</li> <li>Motion SIFT [3]</li> <li>STIP [9]</li> </ul>	Mel-Frequency Cepstral Coefficients
High-level Features	<ul><li>PittPatt Face Detection [12]</li><li>Semantic Indexing Concepts [15]</li></ul>	Acoustic Scene Analysis
Text Features	Optical Character Recognition	Automatic Speech Recognition

Table 1.	Features	used	for the	MED	task
rable r.	1 catures	uscu	101 th		tasn.

# 1.1.1 SIFT, Color SIFT (CSIFT), Transformed Color Histogram (TCH)

These three features describe the gradient and color information of a static image. We used the Harris-Laplace detector for corner detection. For more details, please see [19]. Instead of extracting features from all frames for all videos, we first run shot-break detection and only extract features from the keyframe of a corresponding shot. The shot-break detection algorithm detects large color histogram differences between adjacent frames and a shot-boundary is detected when the histogram difference is larger than a threshold. For the 16507 training videos, we extracted 572,881 keyframes. For the 32061 testing videos, we extracted 1,035,412 keyframes.

Once we have the keyframes, we extract the three features as in [19]. Given the raw feature files, a 4096 word codebook is acquired using the K-Means clustering algorithm. According to the codebook and given a region in an image, we can create a 4096 dimensional vector representing that region. Using the Spatial-Pyramid Matching [10] technique, we extract 8 regions from an keyframe image and calculate a bag-of-words vector for each region. At the end, we get a  $8 \times 4096 = 32768$  dimensional bag-of-words vector. The 8 regions are calculated as follows.

- The whole image as one region.
- Split the image into 4 quadrants and each quadrant is a region.
- Split the image horizontally into 3 equally sized rectangles and each rectangle is a region.

Since we only have feature vectors describing a keyframe, and a video is described by many keyframes, we compute a vector representing a whole video by averaging over the feature vectors from each keyframe. The features are then provided to a classifier for classification.

#### 1.1.2 Motion SIFT (MoSIFT)

Motion SIFT [3] is a motion-based feature that combines information from SIFT and optical flow. The algorithm first extract SIFT points, and for each SIFT point, it checks whether there is a large enough optical flow near the point. If the optical flow value is larger than a threshold, a 256 dimensional feature is computed for that point. The first 128 dimensions of the feature vector is the SIFT descriptor, and the latter 128 dimensions describes the optical flow near the point. We extracted Motion SIFT by calculating the optical flow between neighboring frames, but due to speed issues, we only extract Motion SIFT for the every third frame. Once we have the raw features, a 4096 dimensional codebook is computed, and using the same process as SIFT, a 32768 dimensional vector is created for classification.

# 1.1.3 Space-Time Interest Points (STIP)

Space-Time Interest Points are computed like in [9]. Given the raw features, a 4096 dimensional code is computed, and using the same process as SIFT, a 32768 dimensional vector is created for classification.

#### 1.1.4 Semantic Indexing (SIN)

We predicted the 346 semantic concepts from Semantic Indexing 11 onto the MED keyframes. For details on how we created the models for the 346 concepts, please refer to section 2. Once we have the prediction scores of each concept on each keyframe, we compute a 346 dimensional feature that represents a video. The value of each dimension is the mean value of the concept prediction scores on all keyframes in a given video. We tried out different kinds of score merging techniques, including mean and max, and mean had the best performance. These features are then provided to a classifier for classification.

# 1.1.5 Face

We ran face detection over all videos using the PittPatt Face Detection software [12], and extracted information on the location of the face, the size of the face and whether the face is frontal or profile. In order to speed up the process, we sample 10 frames per second from each video and only perform face detection on the sampled frames. From the extracted face information, we create a 9 dimensional vector where the meaning of each dimension is as follows.

1. Number of faces in the video divided by the total number of frames

- 2. Maximum number of faces in a frame in the whole video.
- 3. Number of frames with more than (including) one face divided by the total number of frames.
- 4. Number of frames with more than (including) two faces divided by the total number of frames.
- 5. Number of frontal faces divided by the total number of faces.
- 6. The median of the ratio  $\frac{face width}{frame width}$  for all faces in the video.
- 7. The median of the ratio  $\frac{face height}{frame height}$  for all faces in the video.
- 8. Number of frames in the center of the frame divided by total number of faces. If w and h is the width and height of the video respectively, and let (x, y) be the location of the center of a face, then the face is in the center of the frame if  $\frac{w}{4} \le x \le \frac{3 \times w}{4}$  and  $\frac{h}{4} \le y \le \frac{3 \times h}{4}$ .
- 9. Median of the confidences of all faces in the video.

We did not perform face tracking or face identification.

#### 1.1.6 Optical Character Recognition (OCR)

We used the Informedia system [5] to extract the OCR. We extracted OCR at a sample rate of 10 frames per second. For details of the OCR process, please refer to [11]. Once we have the OCR output, we create TF-IDF [13] bag-of-words features for each video. Since OCR rarely gets a word completely correct, the vocabulary we use here is a trigram of characters. For example the word "rarely" will be split into "rar", "are", "rel" and "ely". In this way, if one of the characters was miss recognized, there are still some trigrams that are correct.

#### 1.1.7 Automatic Speech Recognition (ASR)

We run automatic speech recognition using the Janus [17] and the Microsoft ASR system. Once completed, for each video, we combine the output of each system into one file and view it as a document. We then perform stemming using the Porter Stemmer, and calculate the TF-IDF [13] bag-of-words vectors for each video.

#### **1.1.8** Mel-Frequency Cepstral Coefficients (MFCC)

We extracted Mel-frequency cepstral coefficients (MFCC) features using the Janus system. Given the raw features, we treat the raw features as a computer vision feature (e.g. SIFT) and run the MFCC features through the same computer vision pipeline. Therefore, we compute a 4096 word codebook and aggregate all MFCC features in one video to create a 4096 dimensional bag-of-words vector. Spatial Pyramid Matching is not reasonable here, so it is not applied.

#### **1.1.9** Acoustic Scene Analysis (ASA)

An expert manually annotated about 3 hours ( $\approx$  120 files) of videos with 42 semantic concepts, which can be derived from the audio: a small ontology links the annotated "small engine" sound concept to video concepts, and the words mentioned in the event kits. Using these labels, we trained 42 Gaussian Mixture Models, which we connected as an ergodic Hidden Markov Model, used to decode the test data with Viterbi. The symbol sequence generated by this step is treated as a bag-of-word, and fed into an SVM classifier.

#### **1.1.10** Performance of features

Table 2 and 3 show the performances of the above features when we use non-linear support vector machine as classifier. The mean minNDC score on 10 Events is used to measure their performances. The smaller mean minNDC score means the better performance. From Table 2, we can find that:

- Generally, comparing low-level visual features, high-level visual features and text visual features, low-level visual features work best.
- Comparing three kinds of image-based low-level features: SIFT, CSIFT and TCH, SIFT describes the gradient information, TCH describes the color information and CSIFT describes both gradient and color information. The performance of TCH is much worse than

SIFT. It means the gradient information is more discriminant than color information in MED task and also explains why the performance of CSIFT is slightly worse than SIFT.

- Comparing two kinds of motion-based feature: MoSIFT and STIP. MoSIFT works around 8% better than STIP, which indicates MoSIFT is a better motion-based feature for MED task.
- Comparing high-level based feature SIN with low-level features, the performance of SIN is comparable to CSIFT and MoSIFT features, better than TCH and STIP, and around 6% worse than SIFT. Generally, with only SIN feature, the system also can get a reasonable performance.

features	SIFT	CSIFT	TCH	MoSIFT	STIP	SIN	Face	OCR
mean MinNDC	0.689	0.717	0.778	0.724	0.782	0.730	0.985	0.90

From Table 3, we can find that:

- Generally, audio features work worse than visual based features. However, these two kind of features are very complementary. When we just simply combined audio and visual features by average late fusion, the mean minNDC can be improved around 12%. It decreased from 0.600 to 0.528.
- Comparing low-level audio feature(MFCC), high-level audio feature (ASA) and text feature (ASR), low-level audio feature works best.
- Comparing high-level audio feature ASA with high-level visual feature SIN, ASA is much worse than SIN. It is only slightly better than random. The reason could be that we only have 42 audio concepts and there are not enough to describe the 10 events, however, the SIN provides 346 visual concepts which are reasonable large enough to describe the 10 events.

Table 3.	The performan	nces of audio	features
Table 5.	The performan	ices of audio	icatures

features	MFCC	ASA	ASR
mean MinNDC	0.805	0.981	0.897

#### 1.2 Classifier Training and Fusion

A large variety of classifiers exist for mapping the feature space into score space. In our final submission, three classifiers are adopted, i.e. non-linear support vector machine (SVM) [2] and kernel regression (KR), and Sequential Boosting SVM in Section 2.2. SVM is one of the most commonly used classifier due to its simple implementation, low computational cost, relatively mature theory and high performance. In TRECVID MED 2010, most of the teams [8] [7] use SVM as their classifiers. Compared to SVM, KR is a simpler but less used algorithm. However, our experiment shows that the performance of KR is consistently better than the performance of SVM.

For combining features from multiple modalities and the outputs of different classifiers, we use three fusion methods, which are early fusion, late fusion and double fusion.

Early Fusion [4] is a combination scheme that runs before classification. Both feature fusion and kernel space fusion are example of early fusion. The main advantage of early fusion is that only one learning phase is required. Two early fusion strategies, i.e., rule-based combination and multiple kernel learning [16], have been tried to combine kernels from different features. For rule-based combination, we use the average of the kernel matrix. Multiple kernel learning [16] is a natural extension of average combination. It aims to automatically learn the weights for different kernel matrix. However,our experimental results show that the performance of multiple kernel learning is only slightly better than average combination. Considering that average combination is much less time consuming than multiple kernel learning, average combination is used as our early fusion method for final submission.

In contrast to early fusion, late fusion [4] happens after classification. While late fusion is easier to perform, in general, it needs more computational effort and has potential to lose the correlation in mixed feature space. Normally, another learning procedure is needed to combine these outputs, but in general, because of the overfitting problem, simply averaging the output scores together yields better or at least comparable results than training another classifier for fusion. Compared to early fusion, late fusion is more robust to features that have negative influence. In our final submission, we use both average combination and logistic regression to combine the outputs of different classifiers.

In our system, we also use a fusion method called double fusion, which combines early fusion and late fusion together. Specifically, for early fusion, we fuse multiple subsets of single features by using standard early fusion technologies; for late fusion, we combine output of classifiers trained from single and combined features. By using this scheme, we can freely combine different early fusion and late fusion techniques, and get benefits of both. Our results show that double fusion is consistently better, or at least comparable than early fusion and late fusion.

Classifier	Early Fusion	Late Fusion	Double Fusion
SVM	0.632	0.528	0.519
Sequential Boosting SVM	0.651	0.556	0.554
KR	0.585	0.516	0.506

Table 4: Comparison of classifiers, and Fusion Methods, MinNDC is used as evaluation criteria.

#### 1.3 Submissions

A submission for a MED 2011 event consists of a video list with scores and a threshold. The score for each video is computed by a classifier that is trained on a number of features in Section 1.1. We experimented with three different types of classifiers: SVM, Sequential Boosting and Kernel Regression in Section 1.2. For each classifier we explored early fusion, late fusion and a technique that we call double fusion in Section 1.2. Given the scores for each video per event, we have two methods to compute the actual threshold. The first method is a simple cutoff at 1800. This number guarantees us that the false alarm rate will be lower than 6%, because we return less than 6% of all videos. Moreover the number of videos that our system did not detect will be as low as possible within the 6% false alarm criteria. Notice that this method is to use the best threshold of our cross validation experiments in the training data. This method is obviously less conservative and turned out to be very unstable in weighted fusion techniques, but as we will see later it performs well for average fusion techniques.

#### 1.3.1 Primary run

Since the primary run is the most important, we prefer it to be our best run, however we don't want to risk the chance that we are over-fitting on the training data and therefore the results have to be stable across different splits in our training data. Moreover the actual threshold also plays an important role in the evaluation and should therefore be stable as well. We believed that both our early and late fusion approach would have a decent performance, but from our experiments none of them was consistently better even though there sometimes was a significant difference between the two approaches. Double fusion on the other hand showed very promising results, better than or comparable to both early and late fusion. Moreover the actual thresholds seemed to be stable, based on the number of retrieved videos for each event. Therefore we decided to submit a double fusion run, using SVM and average fusion in the late fusion stage of double fusion, as our primary run. We chose SVM, because the average performance of our Sequential Boosting SVM and Kernel Regression experiments were very similar. Furthermore we preferred average fusion over a weighting scheme learned by logistic regression, because we believed that the weighting scheme was likely over-fitting to the training data.

#### 1.3.2 DoubleKernelLG

In our experiments performed on the training data, we got better performance using a weighting scheme learned by logistic regression for the late fusion part of double fusion. Also kernel regression gave slightly better results than the other classifiers. However we believe that this behavior might

be explained by over-fitting and therefore not transform to the testing data. Nonetheless it is worth being a non-primary run in case that it actually would be better.

#### 1.3.3 Double3ClassifierLG

This is slightly more conservative run compared to the DoubleKernelLG run, because we use a weighted combination learned by logistic regression of the three different classifiers: SVM, Sequential Boosting SVM and Kernel Regression. For the average performance across events, this does not make a significant difference on average. However it does reduce the variance within events, because not all three classifiers perform equally well for all events. Therefore it is a more stable run when we take individual events into account.

#### 1.3.4 Late3ClassifierAverage

The previous three submissions are all dependent on the early fusion performance, while this run omits that part completely. For the threshold, we simply set the actual threshold to 1800 videos to make the false alarm rate will be lower than 6%. Similarly to the Double3ClassiferLG, we again performed an average fusion on the results from the three classifers to reduce the variance within events.

#### 1.4 Results

The results on the MED 2011 evaluation data are shown in Figure 1. We can see that all our runs have similar performance, because we use the same features across all runs. The slight differences in performance are therefore mainly due to overfitting, learning weights for different features in a logistic regression setting did harm our performance in the final evaluation data. On the other hand additional experiments showed that using kernel regression does perform better than SVM.

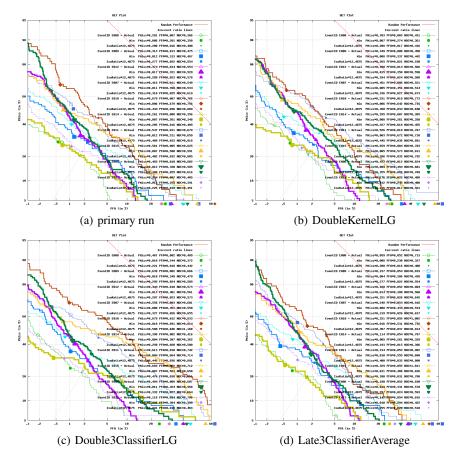


Figure 1: Preliminary results.

# 2 Semantic Indexing (SIN)

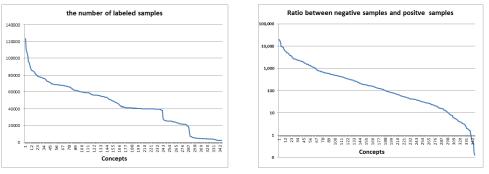
#### 2.1 Feature Extraction

As we know, MED task focus on the multimedia content analysis on video level. SIN task focus on the video clips (shot) level. We can expect that, most of the useful low-level features in MED task can also be useful for SIN task. However, considering the time-consuming problem, we only used three most representative features for SIN task: SIFT, Color SIFT (CSIFT) and Motion SIFT (MoSIFT). SIFT and CSIFT describe the gradient and color information of images. MoSIFT describes both the optical flow and gradient information of video clips. Since the Harris-Laplace detectors only can detect a few feature points for some simple scenes such as sky, we also used dense-sampling detector to sample feature points besides Harris-Laplace detector. The more details about these features please refer to Section 1.1. Generally, these three features provide most of the useful information for SIN task.

#### 2.2 Sequential Boosting SVM

#### 2.2.1 Problem Analysis

In feature extraction, we got the spatial bag-of-words feature representation for every shot. With these low-level feature representations, the most popular solution is to train a two-class non-linear kernel SVM classifier for every concept. However, as the increasing of training samples, this solution has some problems. For this year's SIN task, the development set includes around 11,000 videos and 26,000 shots. Obviously, we are facing a large-scale classification problem. As shown in the Figure 2(a), among 346 concepts, there are 152 concepts have over 50,000 labeled samples. Only 56 concepts have less than 10,000 labeled samples. The time cost becomes a big issue if we want to train a non-linear kernel SVM classifier on the over 50,000 training samples. Furthermore, the labeled samples for each concept are extreme unbalanced. In Figure 2(b), we analyzed the ratio between negative samples and positive samples. There are 65 concepts whose number of negative samples is over 1000 times than the number of positive samples. There are 189 concepts which the ratio between negative and positive samples is over 100. Only 46 concepts have reasonable balanced training data. Their ratios between negative and positive samples are between 10 and 0.1. As a result, the SVM's optimal hyperplane will be biased toward the negative samples due to the unbalance of training samples. Therefore, we proposed the Sequential Boosting SVM to deal with large-scale unbalance classification problem.



(a) the number of labeled samples



Figure 2: SIN Label Analysis

#### 2.2.2 Bagging and AdaBoost

The proposed Sequential Boosting SVM comes from the idea of Bagging and AdaBoost. The main ideas of Bagging and AdaBoost are:

- **Bagging** [1]: The basic idea of Bagging is to train multiple classifiers. The training samples for each classifier are generated by uniformly sampling with replacement. The final prediction is the combination by average the multiple classifiers.
- AdaBoost [6] [14]: The basic idea of AdaBoost is to train a sequence of weak classifiers by maintaining set of weights over training samples and adaptively updating these weights

after each Boosting iteration: the samples that are misclassified gain weight while the samples that are classified correctly lose weight. Therefore, the future weak classifier will be forced to focus on the hard samples. Finally, the combination of these weak classifiers will be a strong classifier.

#### 2.2.3 Sequential Boosting SVM

Intuitively, Bagging strategy can help us to solve the large-scale unbalance classification problem. Firstly, Bagging strategy divides the large-scale training problem to several smaller training problems. Each of them only contains a reasonable number of training examples. The training time cost will not a big issue. Meanwhile, to overcome the unbalanced problem, we can keep all of the positive examples and only execute the random sampling on negative examples. The number of sampled negative examples of each set is the same with the number of positive samples. Therefore, each classifier will be trained on a balanced number of positive and negative samples. This is the Asymmetric Bagging strategy proposed in [18]. However, since the training data for SIN task is extreme unbalanced and its size is large, in most of cases, the sampled examples for each bagging classifier cannot cover the whole training examples. This will hurt the final performance.

To improve the performance of bagging classifiers, an intuitive solution is to choose the most importance examples for each bagging classifier. Therefore, even the bagging classifiers only use a limited number of training examples, the sampled most importance examples already contain the most information of the whole training set. Inspired by the main idea of AdaBoost weighting [14], we proposed the Sequential Boosted Sampling strategy. The adaptively updating weights of training examples are used as a metric to measure the importance of training examples. Examples that can be easily misclassified get high possibility to be sampled. Examples that can be easily classified get low possibility. Therefore, the small classifier will focus on the hard examples, which will boost the performance even only a small part of training examples are used.

The algorithm of Sequential Boosting SVM is described in Algorithm 1.

#### Algorithm 1: Algorithm of Sequential Boosting SVM.

Input: positive example set  $\mathbf{S}^+ = (x_1^+, y_1^+), \dots, (x_{N^+}^+, y_{N^+}^+)$ , where  $y_i^+ = 1$ ; negative example set  $\mathbf{S}^- = (x_1^-, y_1^-), \dots, (x_{N^-}^-, y_{N^-}^-)$ , where  $y_i^- = 0$ ; SVM classifier I; number of generated classifiers: T; sample  $\mathbf{K}^+$  positive examples and  $\mathbf{K}^-$  negative examples in each iteration. **begin**   $D_1^+(i) = 1/N^+;$   $D_1^-(i) = 1/N^-;$ for  $t \leftarrow 1$  to T do Sample: • Sample positive example set  $\mathbf{S}_t^+$  from  $\mathbf{S}^+$  via distribution  $D_t^+, |\mathbf{S}_t^+| = \mathbf{K}^+;$ • Sample negative example set  $\mathbf{S}_t^-$  from  $\mathbf{S}^-$  via distribution  $D_t^-, |\mathbf{S}_t^-| = \mathbf{K}^-;$ Train SVM classifier:  $C_t = \mathbf{I}(\mathbf{S}_t^+, \mathbf{S}_t^-);$ Predict:  $C^*(x_i) = \frac{1}{t} \sum_{p=1}^t C_p(x_i);$ Update: •  $D_{t+1}^+(i) = \frac{D_t^+(i)}{Z_t^+} \times (1 - C^*(x_i^+))$ , where  $Z_t^+$  is a normalization factor (chosen so that  $D_{t+1}^+$  will be a distribution); •  $D_{t+1}^-(i) = \frac{D_t^-(i)}{Z_t^-} \times (1 - C^*(x_i^-))$ , where  $Z_t^-$  is a normalization factor (chosen so that  $D_{t+1}^+$  will be a distribution);

**Output**: classifier  $C^*(x_i) = \frac{1}{\mathbf{T}} \sum_{p=1}^{\mathbf{T}} C_p(x_i)$ 

#### 2.3 Fusion

Generally, there are two kinds of fusion methods. One is early fusion, which combines different features before training classifier. The other is late fusion, which fuses the prediction scores of

different features' classifiers. Considering the time cost to train classifier, we only took early fusion, which only need train a classifier. In order to explore multi-modal features, we also design a multi-modal Sequential Boosting SVM. In each layer, not only the training samples will be re-sampled by their weights, but also the using feature will be change Sequentially. Since we extracted five kinds of features for SIN task: MoSIFT spatial bag-of-word (MoSIFT), SIFT spatial bag-of-word by Harris-Laplace detector (SIFT-HL), SIFT spatial bag-of-word by dense sampling (SIFT-DS), Color SIFT spatial bag-of-word by Harris-Laplace detector (CSIFT-HL) and Color SIFT spatial bag-of-word by dense sampling (CSIFT-DS). We pre-computed the distance matrix between training data for all of these five features. In early fusion part, we just weighted fused their distance matrix. We tried several different kinds of fusion combination and got some combined features.

- SIFT-HL-DS: averagely fuse SIFT-HL and SIFT-DS;
- CSIFT-HL-DS: averagely fuse CSIFT-HL and CSIFT-DS;
- MoSIFT-SIFT-CSIFT: averagely fuse MoSIFT, SIFT-HL and CSIFT-HL;
- MoSIFT-SIFT2-CSIFT2: averagely fuse MoSIFT, SIFT-HL-DS and CSIFT-HL-DS.

#### 2.4 Submission

This year, we trained 4 different kinds of models and submitted 4 runs.

- **MoSIFT\_model**: we used MoSIFT spatial bag-of-word feature and trained 10-layers Sequential Boosting SVM classifier. We submitted this as CMU\_1 run.
- MoSIFT-SIFT-CSIFT\_model: we used MoSIFT-SIFT-CSIFT feature and trained 10layers Sequential Boosting SVM classifier. We submitted this as CMU\_2 run.
- **MoSIFT-SIFT2-CSIFT2\_model**: we used MoSIFT-SIFT2-CSIFT2 feature and trained 10-layers Sequential Boosting SVM classifier. We didn't submit this run.
- MoSIFT-SIFT2-CSIFT2\_multimodal: we used MoSIFT, SIFT-HL-DS and CSIFT-HL-DS to train 20-layers multi-modal Sequential Boosting SVM. The order of the features is MoSIFT, SIFT-HL-DS and CSIFT-HL-DS. We submitted this run as CMU\_3 run.
- MoSIFT-SIFT2-CSIFT2\_latefusion: we averagely fused the prediction scores from MoSIFT-SIFT2-CSIFT2\_model and MoSIFT-SIFT2-CSIFT2\_multimodal and submitted this as CMU\_4 run.

The performance of the above runs are in Table 5. As we can see:

- MoSIFT spatial bag-of-word feature is a good feature for SIN task. Only MoSIFT itself can help us get reasonable performance (mean infAP: 0.1064).
- SIFT-HL and CSIFT-HL are very complemental features for MoSIFT. After we combined SIFT-HL and CSIFT-HL feature with MoSIFT feature, the mean infAP is improved from 0.1064 to 0.1337. We got about 30% improvement from SIFT-HL and CSIFT-HL feature.
- SIFT-DS and CSIFT-DS can improve the performance of SIFT-HL and CSIFT-HL in SIN task. Based on CMU\_2, after we combined SIFT-DS and CSIFT-HL, we got 5% improvement from 0.1337 to 0.1407.
- Multi-modal Sequential Boosting SVM works slightly better than early fusion. The performance of MoSIFT-SIFT2-CSIFT2\_multimodal is 0.1464, which is 4% better than the performance of early fusion 0.1407.

Run_ID	model	mean infAP of 50 concepts
CMU_1	MoSIFT_model	0.1064
CMU_2	MoSIFT-SIFT-CSIFT_model	0.1337
	MoSIFT-SIFT2-CSIFT2_model	0.1407
CMU_3	MoSIFT-SIFT2-CSIFT2_multimodal	0.1458
CMU_4	MoSIFT-SIFT2-CSIFT2_latefusion	0.1464

Table 5: The performances of submissions

#### 2.5 Future work

In feature part, we only tried three most representative visual features. Obviously, for some concepts in SIN task, such as Speech, Singing and Talking, the audio feature can be very useful. In our current experiment of MED task, MFCC bag-of-word feature works well for MED task and is very complementary to visual features. We will try MFCC audio feature to improve the current SIN performance. In classification part, Sequential Boosting SVM works well for SIN task. However, there is an open issue that how to decide the number of classifier layers. That will be another future work for SIN task. In fusion part, Multi-modal Sequential Boosting SVM is a good solution to combine different modalities. Currently, we used a fixed feature order. However, it will be an interesting question that how to choose the most useful feature for next layer classifier.

#### **3** Acknowledgments

This work has been supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

#### References

- [1] L. Breiman and L. Breiman. Bagging predictors. In Machine Learning, pages 123–140, 1996.
- [2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011.
- [3] M.-Y. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos, 2009.
- [4] C. Cortes, M. Mohri, and A. Rostamizadeh. L 2 regularization for learning kernels. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pages 109–116. AUAI Press, 2009.
- [5] D. Das, D. Chen, and A. G. Hauptmann. Improving multimedia retrieval with a video OCR. In Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, volume 6820 of Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, January 2008.
- [6] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting, 1997.
- [7] G. Iyengar. Discriminative model fusion for semantic concept detection and annotation in video. *Proceedings of the eleventh ACM international Conference on Multimedia*, pages 255– 258, 2003.
- [8] Y. Jiang, X. Zeng, G. Ye, S. Bhattacharya, D. Ellis, M. Shah, and S. Chang. Columbia-UCF TRECVID2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching. In *NIST TRECVID Workshop*, 2010.
- [9] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.
- [10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In CVPR (2), pages 2169–2178, 2006.
- [11] H. Li, L. Bao, Z. Gao, A. Overwijk, W. Liu, L. Zhang, S. Yu, M. Chen, F. Metze, and A. Hauptmann. Informedia @ TRECVID 2010. TRECVID Video Retrieval Evaluation Workshop, NIST, 2010.
- [12] PittPatt. Pittpatt face detection. http://www.pittpatt.com/.
- [13] G. Salton and e. M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
- [14] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Machine Learning*, pages 80–91, 1999.

- [15] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVID. In MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pages 321–330, New York, NY, USA, 2006. ACM Press.
- [16] C. Snoek, M. Worring, and A. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM International Conference on Multimedia*, pages 399–402. ACM, 2005.
- [17] H. Soltau, F. Metze, C. Fügen, and A. Waibel. A One-pass Decoder based on Polymorphic Linguistic Context Assignment. In *Proc. Automatic Speech Recognition and Understanding* (ASRU), Madonna di Campiglio, Italy, Dec. 2001. IEEE.
- [18] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1088–1099, July 2006.
- [19] 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*, 32(9):1582–1596, 2010.

# Informedia@TRECVID 2011: Surveillance Event Detection

Longfei Zhang<sup>1</sup>, Lu Jiang<sup>2</sup>, Lei Bao<sup>3</sup>, Shohei Takahashi<sup>4</sup>, Yuanpeng Li<sup>2</sup>, Alexander Hauptmann<sup>2</sup>

<sup>1</sup> School of Software, Beijing Institute of Technology, Beijing, 100081, P.R China

<sup>2</sup> School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213, USA

<sup>3</sup> Laboratory for Advanced Computing Technology Research, ICT, CAS, Beijing 100190, China

<sup>4</sup> Graduate School of Global Information and Telecommunication Studies, Waseda University, Tokyo,

Japan

**Abstract:** We present a generic event detection system evaluated in the SED task of TRECVID 2011. We investigated a generic statistical approach with spatio-temporal features applied to seven events, which were defined by the SED task. This approach is based on local spatio-temporal descriptors, called MoSIFT, and generated by pair-wise video frames. Visual vocabularies are generated from cluster centers of MoSIFT features, which were sampled from the event video clips. We also estimated the spatial distribution of actions by over-generated person detection and background subtraction. Different sliding window sizes and steps were adopted for different events based on the event duration priors. Several sets of one-against-all action classifiers were trained using cascade non-linear SVMs and Random Forests, which improved the classification performance on unbalanced data just like the SED datasets. Results of 9 runs were presented with variations in i) sliding window size ii) step size of BOW, iii) classifier threshold and iv) classifiers. The performance shows improvement over last year on the event detection task.

### 1. Introduction

Surveillance video recording is becoming ubiquitous in daily life for public areas such as supermarkets, banks, and airports. This has attracted more and more research interests and experiences rapid advances in recent years. A lot of schemes have been proposed for the human action recognition, among them, local interest points algorithm have been widely adopted. Methods based on feature descriptors around local interest points are now widely used in object recognition. This part-based approach assumes that a collection of distinctive parts can effectively describe the whole object. Compared to global appearance descriptions, a part-based approach has better tolerance to posture, illumination, occlusion, deformation and cluttered background. Recently, spatio-temporal local features [1-6] have been used for motion recognition in video. The key to the success of part-based methods is that the interest points are distinctive and descriptive. Therefore, interest point detection algorithms play an important role in a part-based approach.

The straightforward way to detect a spatio-temporal interest point is to extend a 2D interest point detection algorithm. Laptev et al. [2] extended 2D Harris corner detectors to a 3D Harris corner detector, which detects points with high intensity variations in both spatial and temporal dimensions. In other words, a 3D Harris detector finds spatial corners with velocity change, which can produce compact and distinctive interest points. However, since the assumption of change in all 3 dimensions is quite restrictive, very few point results and many motion types may not be well distinguished. Dollar et al. [7] discarded spatial constraints and focused only on the temporal domain. Since they relaxed the spatial constraints, their detector detects more interest points than a 3D Harris detector by applying Gabor filters on the temporal dimension to detect periodic frequency components. Although they state that regions with strong periodic responses normally contain distinguishing characteristics, it is not clear that periodic movements are sufficient to describe complex actions. Since recognizing human motion is more complicated than object recognition, motion recognition is likely to require with enhanced local features that provide both shape and motion information. Thus, MoSIFT features [8] are proposed. MoSIFT detects spatially distinctive interest points with substantial motion by pair-wise frames. They first apply the well-know 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.

However, in the local interest point algorithms, most of them [1-7] did not care where the interest points located, as their experiment scenes are relative simple and clear, and most of conditions, just one or two peoples have some actions. However, these conditions seldom hold in real-world surveillance videos. Even the same type of actions may exhibit enormous variations due to cluttered background, different viewpoints

and many other factors in unconstrained real-world environment, such as TREC Video Retrieval Evaluation (TRECVID) [9]. To our knowledge, TRECVID has made the largest effort to bridge the research efforts and the challenges in real-world conditions by providing an extensive 144-hour surveillance video dataset recorded in London Gatwick Airport. In this dataset, the cameras are fixed, but the scenes are very complex, and there are a lot of people walking through on the scenes. Thus, if we just adopt the local interest points to detect the events on the scene, there are a lot of noise interest points for some events. In TRECVID 2011 Evaluation, there are 7 required events such as "CellToEar", "Embrace", "ObjectPut", "Pointing", "PeopleMeet", "PeopleSplitUp" and "PersonRuns". All of them are relative to the human. Therefore, we will use detection methods, such as human detection and foreground object subtraction, and tracking approaches to locate these interest points, and filter the noise interest points. Finally, we also adopt the results of human detection to estimate the correctness of detection.

In the following section, we will describe our system overview, and then the MoSIFT algorithm, the human detection and tracking will be introduced. After that, the experiments and discussions will be given. Finally, we will conclude the paper.

#### 2. System Framework

For the tasks in TRECVID 2011 Event Detection Evaluation, we focus on human-related events. We mainly follow the framework we employed in TRECVID 2009 and 2010 Evaluation, which incorporates interesting point extraction, clustering and classification modules. In TRECVID 2009 Evaluation, the MoSIFT interesting points are extracted for each video firstly, and then Bag-of-Words (BoW) are adopted. After that, the cascade SVM will be trained. The details can be viewed in [16].

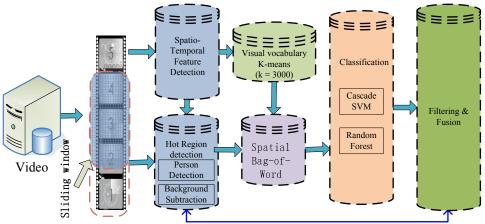


Figure 1: The framework of our surveillance event detection system

In contrast, we extend our framework by three kinds of processing. Firstly, visual vocabularies were sampled from the event part video features. 3000 cluster centers were generated by K-means algorithm. In general, instance sampling from positive data might cause over fitting. But in surveillance video event detection, since the background did not change much, more positive samples might cause better video representation than random samples in visual vocabulary selection procedure. Secondly, each event had different durations. Laptev [18] tried several kinds of slide window size in Bag-of-Word procedure. We followed this idea and used three kind of slide window size for different event. Thirdly, we also used Spatial Bag-of-Word for a better video representation. Zhu [19] used a 3 layer spatial pyramid, 1x1 to 4x4, to represent the video. Laptev [18] used localization map to construct the BoW. Their experiment shows that localization map was better than grid spatial Bag-of-Word. However, localization map was based their additional annotation. Following the same idea, we used person detection and background subtraction analysis to find hot regions of actions automatically. Based on that region, we defined a different grid for video representation which will be introduced in subsection 3.4. Fourthly, for each frame, the MoSIFT points were extracted unsupervised. But these feature points might generate not by action but by noise, and we cannot discriminate them. Thus, the over generated human detection algorithm was adopted. We kept these MoSIFT points located in the region of human. Fifthly, both cascade SVM and Random Forest classifiers were trained for solving the unbalanced training and testing data. After got the probabilities of each event, we fused these results. The system framework is illustrated in the Figure.1.

#### 3. MoSIFT Feature Based Action Recognition

For action recognition, there are three major steps: detecting interest points, constructing a feature descriptor, and building a classifier. Detecting interest points reduces the video from a volume of pixels to compact but descriptive interest points.

This section outlines our algorithm [8] to detect and describe spatio-temporal interest points. It was shown [8] to outperform the similar Laptev's method [2]. The approach first applies the SIFT algorithm to find visually distinctive components in the spatial domain and detects spatio-temporal interest points through (temporal) motion constraints. The motion constraint consists of a 'sufficient' amount of optical flow around the distinctive points.

#### 3.1. Motion Interest Point Detection and Feature Description

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 [10] and optical flow computation matching the scale of the SIFT points.

SIFT was designed to detect distinctive interest points in still images. The candidate points are distinctive in appearance, but they are independent of the motions in the video. For example, a cluttered background produces interest points unrelated to human actions. Clearly, only interest points with sufficient motion provide the necessary information for action recognition.

Multiple-scale optical flows are calculated according to the SIFT scales. Then, as long as the amount of movement is suitable, the candidate interest point contains are retained as a motion interest point.

The advantage of using optical flow, rather than video cuboids or volumes, is that it explicitly captures the magnitude and direction of a motion, instead of implicitly modeling motion through appearance change over time.

Motion interest points are scale invariant in the spatial domain. However, we do not make them scale invariant in the temporal domain. Temporal scale invariance could be achieved by calculating optical flow on multiple scales in time.

After getting the MoSIFT interest points, we need describe these points. Appearance and motion information together are the essential components for an action classifier. Since an action is only represented by a set of spatio-temporal point descriptors, the descriptor features critically determine the information available for recognition.

The motion descriptor adapts the idea of grid aggregation in SIFT to describe motions. Optical flow detects the magnitude and direction of a movement. Since, 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. For human activity recognition, rotation invariance of appearance remains important due to varying view angles and deformations. Since our videos are captured by stationary cameras, the direction of movement is an important (non-invariant) vector to help recognize an action. Therefore, our method omits adjusting for orientation invariance in the motion descriptors.

Finally, the two aggregated histograms (appearance and optical flow) are combined into the descriptor, which now has 256 dimensions.

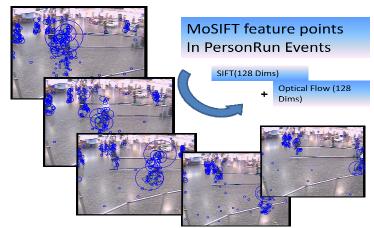


Figure 2: MoSIFT points in PersonRun events. The center of blue circle is the location of the points. The radius of the blue circle is the scale of the movements

#### 3.2. Hot Region Detection

MoSIFT feature does a great job in human behavior representation for human action recognition. However, Are the MoSIFT interesting points caused by human? The MoSIFT points might be caused by moving, light shaking, or shadow. If we could sample the MoSIFT points from human body or action region, we might reduce more noise interesting points. Thus, in this section, we use person detection and background subtraction method to create the hot region. These hot region could be used in feature selection and building spatial BoW.

Person detection is the most direct method to detect the area of human. Histogram of Oriented Gradient (HOG) feature [12] and Haar like feature [13] are the most popular features used in person detection. Locally normalized HOG descriptors are computed on a dense grid of uniformly spaced cells and use overlapping local contrast normalizations for improved performance. Haar like feature person detection used in VJ (Viola and Jones) works is using AdaBoost to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences. It consists of a filter that takes image windows from n consecutive frames as input, a threshold and a positive and negative vote. Since there are too many people in Gatwick surveillance video(especially camera 2, 3 and 5), full body person detection is very limited in detecting the person blinded by some background objects, such as showed in figure 3. In our experiments, both HOG person upper/full body detectors and Haar person upper/full body detectors are trained on the development videos in Dev08 and INRIA dataset, and over generated for more reliable action region coverage.



Figure 3: Mosift points with over-generated person detections results in sequence. The red rectangles are bounding boxes of objects detected by Person detecton, and the red points with green arrows are MoSIFT points. The green arrows are the direction of movements

We also use the background subtraction method to build the spatial priority maps. SED video data are captured from static cameras. People who don't act any movement have no relation with events. Therefore, background subtraction is effective to extract the area where people move and each event occurs.

To reduce the noise we use median filter, close operation and open operation for foreground. Foreground is expanded by open operation because surroundings area of foreground may be related with movements.

We built a spatial priority map by adding p = 1/n to each pixel with the foreground in *n* frames. Each spatial priority map is built for each camera. Fig4. shows spatial priority map from camera 2,3,4 and 5.



Figure 4: Priority map of cameras

#### 3.3. Spatial Priority Maps based Spatial Bag of Features

From the prior knowledge of hot region, which were generated by person detection and spatial priority maps for each camera, the grid parsing strategy could be estimated. Based on this thinking, each frame is divided into a set of rectangular tiles or grids. The resulting Bow features are derived by concatenating the BoW features captured in each grid. Spatial bag of features allows for that the spatial distribution of interest points could be taken into account by the classifier. Generally, it facilitate the classifier distinguish events happen in different part of the frame. In our experiment, 8 rectangular tiles are adopted, as illustrated Fig.5. We choose to partition the frame in this way because according to our empirical observation in training set, the partition may reasonably capture the hot region of actions.

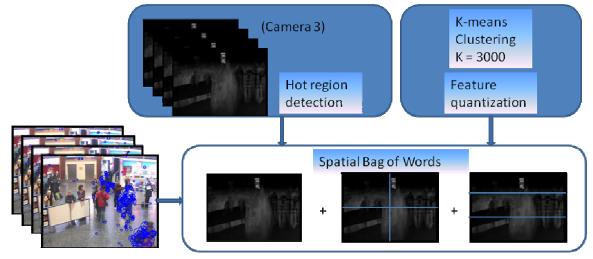


Figure 5: Spatial bag-of-words.

#### 4. Experiments and Discussion

In TRECVID 2011 Event Detection Evaluation [9], 99 hours videos are provided as the development set and about 44 hours videos as the evaluation set, where the videos were captured using 5 different cameras with image resolution 720×576 at 25 fps.

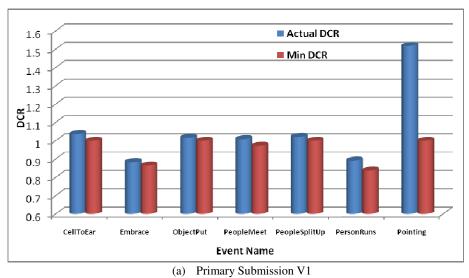
Our experiment start by extracting MoSIFT interest points in each sliding window. MoSIFT is a scale invariant local feature which is less affected by global appearance, posture, illumination and occlusion. Figure 2 illustrates an example of the extracted MoSIFT features from Person Run Events. It can be seen

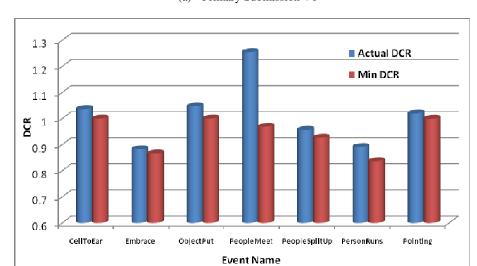
that MoSIFT feature reasonably captures the areas with human activity. As different sliding window size can affect the final performance, in the experiment we manually set different window sizes for different events to ensure the window can capture the whole event. E.g. in our primary submission, for the event "ObjectPut", "PeopleMeet" and "PeopleSplitUp", the window size is set to 60 frames and repeats every 10 frames as these event cover a relatively long time-span; whereas for the event "PersonRuns", "CelltoEar", Embrace and Pointing , the window size is set to 60 and repeats every 15 frames.

We assume that an event can be described by a combination of different types of small motions. Therefore we use BoW to quantify MoSIFT 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 interest 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 BoW features. In our experiments, the vocabulary size is 3,000, and a soft boundary to form our bag-of-word features is applied. The spatial bag-of-word is also incorporated while constructing the resulting vocabulary. Each frame is divided into 8 tiles i.e.  $1 \times 1$ ,  $2 \times 2$  and  $1 \times 3$  rectangular tiles as illustrated in Figure 5. Consequently the dimension of the resulting BoW features is  $8 \times 3,000 = 24,000$ . Once the BoW features are obtained, a binary SVM [11] classifier with a  $\chi^2$  kernel is trained for each event. Finally we apply one-against-all strategy to construct action models.

The sliding window results in a highly unbalanced dataset (positive windows are much less frequent than negative windows). Two approached is conducted to tackle the unbalanced data. The first is cascade SVM. We build a one, five and ten layers cascade classifier to overcome this imbalance in the data and reduce false alarms. For each layer, we choose an equal ratio of (positive vs. negative) training data to build a classifier to favors to positive examples. This leads the classifier with high detection rates. In the training process, the cross-validation is adopted. By cascading five or ten layers of these high detections. We also aggregate consecutive positive predictions to achieve multi-resolution. The second approach is under sampling the majority class. Each sub-dataset is constructed in a way that all positive instances are preserved and negative instances are randomly supplemented. We choose negative to positive ratio 2.5 to reflect the natural imbalance in data. For each sub-dataset, we train a model using the random forest algorithm, in which the number of trees in forest is set to 100 and the maximum number of features is set to 2000. Finally all models are trained and aggregated together by averaging their votes. The empirical analysis suggests both solutions can improve the performance on unbalanced dataset.

Figure 6 summarizes the DCR(Detection Cost Rate) analysis for some of our this year's final submissions. Fig.6 (a) illustrates our primary submission, in which the window size is set to 60 frames and repeats every 10 frames for the event "ObjectPut", "PeopleMeet" and "PeopleSplitUp". The window size is set to 60 and repeats every 15 frames for other events. Fig.6 (b) illustrates our submission version 6, in which the window size is set to 60 frames and repeats every 10 frames for the same sliding window setting with our primary submission. The difference lies in this version is that it adopts the random forest rather than the cascade SVM used in other submissions. It can be seen that in our primary submission (SVM) the Actual DCR (ADCR) and Minimum DCR (MinDCR) are quite similar, which is mainly because we search the best threshold in the training set. Generally, both cascade SVM classifiers and random forest classifiers are robust. Although, generally, cascade SVM classifiers outperform random forest classifiers, random forest classifiers are much faster and give relatively good predictions.





(b) Submission V6

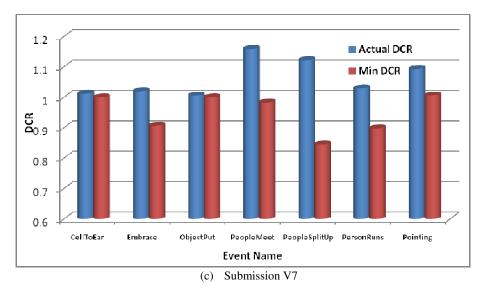


Figure 6: DCR (Detection Cost Rate) Analysis results

Table 1 summarizes the comparison between this year's result and last year's result in terms of MinDCR, in which the first and the third row represents the best score of our 2010 and 2011 submission for each event, respectively; the second row indicates the best score for each event reported in TRECVID 2010 document[17]. It can be seen that, compared with our last year's result, we improve the performance for the event Embrace, "PeopleSplitUp", "PersonRuns" and "Pointing", in which "Embrace", "PeopleSplitUp" and "PersonRuns" even beats the last year's best results. The most significant improvement we achieve this year regards to the event "PeopleSplitUp", in which the MinDCR is reduced by 20.7%. The improvement is probably credited to the larger vocabulary and the introduction of spatial BoW. However, this strategy results in a considerable high dimension space, e.g. this year's 24,000 dimension versus last year's 2,000. Therefore efficient algorithms are recommended to be applied in this considerable high dimension space, which explains the reason of adopting cascade SVM and random forest algorithm in our experiment.

Table 1: Comparison between the best result of this year and last year in MinDC
---

	CellToEar	Embrace	ObjectPut	PeopleMeet	PeopleSplitUp	PersonRuns	Pointing
2010 CMU	1.0003	0.9838	1.0003	0.9793	0.9889	0.9477	1.0003
2010 Best	$1^{\dagger}$	0.9663 <sup>†</sup>	0.9971 <sup>•</sup>	$0.9787^{\ddagger}$	0.9889	0.6818*	<b>0.996</b> <sup>†</sup>
2011 CMU	1.0003	0.8658	1.0003	0.9684	0.7838	0.837	0.9996

<sup>†</sup>the result attributes to IPG-BJTU <sup>‡</sup> the result attributes to TJU  $\star$  the result attributes to QMUL-ACTIVA • the result attributes to CRIM. The best score for each event is in bold

Table 2 presents the comparison between CMU results with TRECVID 2011 best results. The comparison is conducted on each team's primary submission. The "Best TRECVid Sys. MinDCR" column denotes for the best MinDCR reported in TRECVID 2011 Formal Evaluation Comparative Results. The ranking column represents our results' ranking among all groups in terms of the MinDCR.

Actions	Ranking	Best TRECVid Sys.	CMU sys. Primary Run				
Actions	Kanking	MinDCR	MinDCR	ADCR	#CorDet	#FA	#Miss
CellToEar	1	1.0003	1.0003	1.0365	1	127	193
Embrace	1	0.8658	0.8658	0.8840	58	657	117
ObjectPut	4	0.9983	1.0003	1.0171	0	57	620
PeopleMeet	1	0.9724	0.9724	1.0100	45	336	404
PeopleSplitUp	5	0.8809	1.0003	1.0217	3	115	184
PersonRuns	1	0.8370	0.8370	0.8924	26	413	81
Pointing	3	0.9730	1.0001	1.5186	132	1960	931

Table 2: Primary run results comparison between CMU and TRECVID 2011 best results

#### 5. Conclusion

In this paper we have described our implementation to SED TRECVID2011. A real surveillance dataset from London Gatwick airport have been analyzed, using spatio-temporal interest points descriptor, MoSIFT, and spatial feature selection and representation. The obtained performances show good scores using this generic scheme, in particular for three actions: "PersonRuns", "PeopleMeet" and "Embrace". In the future work we plan to extend the current framework with better spatio-temporal models of actions as well as person-focused analysis of video.

#### 6. Acknowledgments

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

#### References

- [1]. Schuldt, C. Laptev, and B. I. Caputo. Recognizing human actions: a local SVM approach. ICPR(17), pp 32-36, 2004.
- [2]. I. Laptev and T. Lindeberg. Space-time interest points. ICCV, pages 432–439, 2003.
- [3]. S.-F. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. ICCV, pp 1-8, 2007.
- [4]. A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. BMVC, 2008.
- [5]. G. Willems, T. Tuytelaars, and L. V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. ECCV, pp 650-663, 2008.
- [6]. A. Oikonomopoulos, L. Patras, and M. Pantic. Spatiotemporal saliency for human action recognition. ICME, pp 1-4, 2005.
- [7]. P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp 65-72, 2005.
- [8]. M.-y. Chen and A.Hauptmann. MoSIFT: Reocgnizing Human Actions in Surveillance Videos . CMU-CS-09-161, Carnegie Mellon University, 2009.
- [9] National Institute of Standards and Technology (NIST): TRECVID 2009 Evaluation for Surveillance Event trecvidDetection.http://www.nist.gov/speech/tests/trecvid/2009/ trecvidhttp://www.itl.nist.gov/iad/mig/tests/trecvid/2009/doc/eventdet09-evalplan-v03.htm, 2009. 1, 7
- [10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.
- [11] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at <u>http://www.csie.ntu.edu.tw/~cjlin/libsvm/</u>.
- [12] N. Dalal and B. Triggs. Histogram of oriented gradient for human detection. In CVPR, 2005.
- [13] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In CVPR, 2003
- [14] Medeiros, H., Park, J.; Kak, A.A parallel color-based particle filter for object tracking, In CVPR2008 Workshops, p1-8, June 2008
- [15] Babenko, B. Ming-Hsuan Yang; Belongie, S. Visual tracking with online Multiple Instance Learning. In CVPR2009 Workshops), p 983-90, 2009.
- [16] Ming-yu Chen, Huan Li, and Alexander Hauptmann. Informedia @ TRECVID 2009: Analyzing Video Motions.
- [17] Over, Paul and Awad, George and Fiscus, Jon and Antonishek, Brian and Michel, Martial and Smeaton, Alan F. and Kraaij, Wessel and Quénot, Georges (2011) TRECVID 2010 An overview of the goals, tasks, data, evaluation mechanisms, and metrics. In: TRECVID 2010, 15-17 November 2010, Gaithersburg, Md., USA.

[18] R. Benmokhtar and I. Laptev, "INRIA-WILLOW at TRECVid2010: Surveillance Event Detection", http://www-nlpir.nist.gov/projects/tvpubs/tv10.papers/inria-willow.pdf

[19] G. Zhu, M. Yang, K. Yu, W. Xu and Y. Gong, "Detecting Video Events Based on Action Recognition in Complex Scenes Using Spatio-Temporal Descriptor" In proc. ACM Multimedia 2009, Beijing.