論文 / 著書情報
Article / Book Information

Title	TokyoTech+Canon at TRECVID 2011
Authors	Nakamasa Inoue, Yusuke Kamishima, Toshiya Wada, Koichi Shinoda, Shunsuke Sato
Citation	Proc.TRECVID Workshop 2011, Vol. , No. , pp.
Pub. date	2011, 12

TokyoTech+Canon at TRECVID 2011

Nakamasa Inoue Yusuke Kamishima
Toshiya Wada Koichi Shinoda
Department of Computer Science,
Tokyo Institute of Technology
{inoue,kamishi,wada}@ks.cs.titech.ac.jp
{shinoda}@cs.titech.ac.jp

Shunsuke Sato
Canon Corporation
sato.shunsuke@canon.co.jp

1 Semantic Indexing

The aim of this section is to develop a high-performance semantic indexing system using Gaussian mixture model (GMM) supervectors and tree-structured GMMs [1, 2]. GMM spervectors corresponding to six types of audio and visual features are extracted from video shots by using tree-structured GMMs. The computational cost of maximum a posteriori (MAP) adaptation for estimating GMM parameters are reduced by tree-structured GMMs by keeping accuracy at high levels. Our best result was 17.3 % in terms of Mean InfAP, which was ranked 1st over all semantic indexing runs in the full task.

1.1 Feature extraction

The following six types of visual and audio features are extracted from video data:

1. SIFT features with Harris-Affine detector (SIFT-Har)

   The scale invariant feature transform (SIFT) proposed by Lowe [3] is a local feature extraction method that is widely used for object categorization since it is invariant to image scaling and changing illumination. The Harris-Affine detector [5], which is an extension of the Harris corner detector, improves robustness against affine transform of local regions. These features are extracted from every other frame, and principal component analysis (PCA) is applied to reduce their dimensions from 128 to 32.

2. SIFT features with Hessian-Affine detector (SIFT-Hes)

   SIFT features are extracted with the Hessian-Affine detector [5], which is complementary to the Harris-Affine detector. The combination of several different detectors can improve the robustness against noise. These features are extracted from every other frame, and PCA is applied to reduce their dimensions from 128 to 32.

3. SIFT and hue histogram with dense sampling (SIFTH-Dense)

   SIFT features and 36-dimensional hue histograms [4] are combined to capture color information. These features are extracted from key-frames by using dense sampling. PCA is applied to reduce dimensions from 164 to 32.

4. HOG with dense sampling (HOG-Dense)

   32-dimensional histogram of oriented gradients (HOG) are extracted from key-frames by using dense sampling. PCA is applied but dimensions of the HOG features are kept to 32.

5. HOG from temporal subtraction images (HOG-Sub)

   Temporal subtraction images which are obtained by subtracting the key frames from the sub-key frames are used to capture movement information. A frame image five frames after a key frame is used as a sub-key frame. HOG features are extracted from the temporal subtraction images in the same way as the HOG-Dense features.

6. MFCC audio features (MFCC)

   Mel-frequency cepstral coefficients (MFCCs), which describe the short-time spectral shape of audio frames, are extracted to capture audio information. MFCCs are widely used not only for speech recognition but also for generic audio classification. Δ MFCCs, $\Delta\Delta$ MFCCs, Δ log-power and $\Delta\Delta$ log-power are extracted in addition to the MFCCs. Here, "Δ" means the derivation of the feature. The dimension of the audio feature is 38, including 12-dimensional MFCCs.

1.2 Gaussian Mixture Models

Gaussian mixture models (GMMs), whose probability density function (pdf) is given by

$$p(x|\theta) = \sum_{k=1}^{K} w_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{1}$$

are used to model video shots. Here, x is a local feature, $\theta = \{w_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ is a set of GMM parameters, K is the number of Gaussian components (vocabulary size), w_k is a mixture coefficient, and $\mathcal{N}(x|\mu_k, \Sigma_k)$ is a Gaussian pdf with a mean vector μ_k and a covariance matrix Σ_k.

The GMM parameters are estimated for each shot under the maximum a posteriori (MAP) criterion. The MAP solution for GMM means, namely MAP adaptation, is given by

$$\hat{\mu}_k = \frac{\tau \hat{\mu}_k^{(\mathrm{U})} + \sum_{i=1}^{n} c_{ik} x_i}{\tau + C_k}, \quad c_{ik} = \frac{w_k g_k(x_i)}{\sum_{k=1}^{K} w_k g_k(x_i)}, \quad C_k = \sum_{i=1}^{n} c_{ik}, \quad g_k(x) = \mathcal{N}(x|\hat{\mu}_k^{(\mathrm{U})}, \hat{\Sigma}_k^{(\mathrm{U})}), \tag{2}$$

where $X_{\mathrm{F}} = \{x_i\}_{i=1}^{n}$ ($\mathrm{F} \in \{$SIFT-Har, SIFT-Hes, SIFTH-Dense, HOG-Dense, HOG-Sub, MFCC$\}$) is a set of feature vectors extracted from a shot, τ is a predefined hyper-parameter, and $\hat{\theta}^{(\mathrm{U})}$ is the parameter for a universal background model (UBM). The UBM presents how the features are distributed in the general case: therefore, the parameter $\hat{\theta}^{(\mathrm{U})}$ is estimated by using all features in the training set.

1.3 Fast MAP adaptation

The fast MAP adaptation technique [1] reduces computational costs for calculating posterior probabilities c_{ik} in Eq. (9) by constructing a tree-structed GMM. The basic idea of the tree-structured GMM is to cluster Gaussian components and approximate them with a single Gaussian. Each leaf node corresponds to a Gaussian component of the UBM, and each non-leaf node has a single Gaussian that approximates its descendant Gaussian components.

The tree-structured GMM $\mathcal{T}$ is defined as

$$\mathcal{T} = (V, E, G_{\mathrm{TREE}}), \tag{3}$$

where V is a set of nodes, E is a set of edges, and G_{TREE} is a set of Gaussian components for nodes in V. Here, $g^{(v)} \in G_{\mathrm{TREE}}$ denotes a Gaussian component for $v \in V$. The Gaussian components for the UBM, $G_{\mathrm{UBM}} = \{g_k\}_{k=1}^{K}$, are assigned to leaf nodes, i.e., for each leaf $\ell \in V$, there exists $g_k \in G_{\mathrm{UBM}}$ that satisfies $g^{(\ell)} = g_k$. Gaussian components for non-leaf nodes and their children are determined by applying k-means clustering to G_{UBM} (see [1] for details).

With the tree-structured GMM, posterior probabilities c_{ik} in Eq. (9) are calculated only for *active* Gaussian components. The following algorithm finds active leafs by expanding active nodes V_A from the root node to output c_{ik} quickly.

1. Set $V_A \leftarrow \{r\}$, where r is the root node.
2. Expand active nodes by making child nodes of the active nodes active:

$$V_A \leftarrow \bigcup_{v \in V_A} C(v), \tag{4}$$

   where $C(v)$ is a set of child nodes of the node v. Here, $C(\ell) = \{\ell\}$ is used for leaf nodes ℓ to keep the leaf nodes active.
3. Calculate posterior probabilities $c_i^{(v)}$ for an active GMM given by

$$p(x|V_A) = \sum_{v \in V_A} \tilde{w}^{(v)} g^{(v)}(x), \quad \tilde{w}^{(v)} = \frac{w^{(v)}}{\sum_{v \in V_A} w^{(v)}}, \tag{5}$$

   i.e., calculate

$$c_i^{(v)} = \frac{\tilde{w}^{(v)} g^{(v)}(x_i)}{\sum_{v \in V_A} \tilde{w}^{(v)} g^{(v)}(x_i)} = \frac{w^{(v)} g^{(v)}(x_i)}{\sum_{v \in V_A} w^{(v)} g^{(v)}(x_i)}. \tag{6}$$

4. Keep a node v active if $c_i^{(v)}$ is larger than the predetermined threshold c_{TH}, i.e.

$$V_A \leftarrow \{v \in V_A \mid c_i^{(v)} > c_{\text{TH}}\}. \tag{7}$$

5. If all nodes in V_A are leaf nodes, output

$$\hat{c}_{ik} = \begin{cases} c_i^{(\ell)} & (\ell \in V_A,\ g^{(\ell)} = g_k) \\ 0 & (\text{otherwise}) \end{cases}. \tag{8}$$

Otherwise, return to Step 2.

Finally, MAP adaptation is given by

$$\hat{\mu}_k = \frac{\tau \hat{\mu}_k^{(\text{U})} + \sum_{\hat{c}_{ik} \neq 0} \hat{c}_{ik} x_i}{\tau + \hat{C}_k}, \quad \hat{C}_k = \sum_{\hat{c}_{ik} \neq 0} \hat{c}_{ik}. \tag{9}$$

1.4 GMM supervector SVM

After video shots are represented by GMMs, GMM supervectors are extracted by combining normalized mean vectors as

$$\phi(X_{\text{F}}) = \begin{pmatrix} \tilde{\mu}_1 \\ \tilde{\mu}_2 \\ \vdots \\ \tilde{\mu}_K \end{pmatrix}, \quad \tilde{\mu}_k = \sqrt{w_k^{(U)}} (\Sigma_k^{(U)})^{-\frac{1}{2}} \hat{\mu}_k. \tag{10}$$

Support vector machines (SVMs) with the following RBF-kernel are used to train discriminative models for each semantic concepts.

$$k(X_{\text{F}}, X'_{\text{F}}) = \exp\left(-\gamma \|\phi(X_{\text{F}}) - \phi(X'_{\text{F}})\|_2^2\right), \quad \gamma = \frac{1}{\tilde{d}}, \tag{11}$$

where $\tilde{d}$ is the average distance between two GMM supervectors. Finally, trained discriminative functions are linearly combined as

$$f(X) = \sum_{\text{F} \in \mathcal{F}} \alpha_{\text{F}} f_{\text{F}}(X_{\text{F}}), \quad 0 \leq \alpha_{\text{F}} \leq 1, \quad \sum_{\text{F}} \alpha_{\text{F}} = 1. \tag{12}$$

where $\mathcal{F} = \{\text{SIFT-Har}, \text{SIFT-Hes}, \text{SIFTH-Dense}, \text{HOG-Dense}, \text{HOG-Sub}, \text{MFCC}\}$. Combination coefficients α_{F} are optimized on a validation set.

1.5 Experimental conditions

Features were extracted by using Mikolajczyk's implementation [5] for SIFT-Har and SIFT-Hes, SIFT++ [6] for SIFTH-Dense, and HTK [7] for MFCC. The sum of calculation time (for PCA projection, MAP adaptation, and SVM prediction) was reduced from 2.47 sec to 1.00 sec (59.5%) by using the fast MAP adaptation technique. The calculation time was measured by using a Intel Xeon 2.93 GHz CPU.

The following four runs are submitted to the TRECVID 2011 semantic indexing.

F_A_TokyoTech_Canon_1

This run used GMM supervector SVMs with the six type of features described in Section 1.1. The number of Gaussian components was $K = 512$ for the visual features, and $K = 256$ for the audio feature. The UBMs were trained using 1,000,000 samples. Optimal tree structures for the UBMs were selected as to minimize computational costs for MAP adaptation (see [1]). The hyper-parameter τ for MAP adaptation was set to 20.0.

The parameter γ for RBF-kernels is set to $\gamma = h\tilde{d}^{-1}$ $(h = 0.5, 1.0, 2.0)$ to train three SVMs for each type of features. Final scores for ranking video shots were calculated as

$$f(X) = \sum_{\substack{h \in \{0.5, 1.0, 2.0\} \\ \text{F} \in \mathcal{F}}} \alpha_{\text{F}}^{(h)} f_{\text{F}}^{(h)}(X_{\text{F}}), \quad 0 \leq \alpha_{\text{F}}^{(h)} \leq 1, \quad \sum_{\text{F},h} \alpha_{\text{F}}^{(h)} = 1, \tag{13}$$

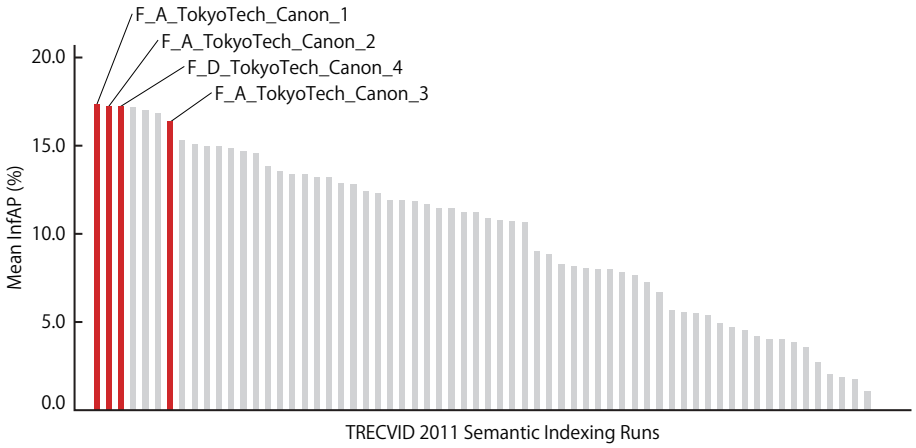

Figure 1: Overview of results of the semantic indexing task in TRECIVD 2010. Mean Inf. AP of our best run using MFCC and SIFT GMM supervector kernels was 7.36%.

where $f_{\mathrm{F}}^{(h)}$ is a discriminative function for the feature F and the parameter h. The data sets *IACC_1_tv10_training* and *IACC_1_A* are used for training and validating respectively, to optimize combination coefficients $\alpha_{\mathrm{F}}^{(h)}$. The gradient ascent method was applied to maximize average pricision (AP) on the validation set. Initial values for $\alpha_{\mathrm{F}}^{(h)}$ was 1/18.

F_A_TokyoTech_Canon_2

This run used the same features and parameters as F_A_TokyoTech_Canon_1 but γ for RBF-kernels was fix to $\tilde{d}^{-1}$ and initial values of α_{F} was 1/6.

F_A_TokyoTech_Canon_3

This run combined scores for all semantic concepts. The final score for semantic concept S was calculated as

$$g_{\mathrm{S}}(X) = \sum_{\mathrm{S'}} \beta_{\mathrm{S'}} f_{\mathrm{S'}}(X), \quad \sum_{\mathrm{S'}} \beta_{\mathrm{S'}} = 1, \tag{14}$$

where $f_{\mathrm{S'}}(X)$ is a score for semantic concept S' for the run of F_A_TokyoTech_Canon_2. Combination coefficients $\beta_{\mathrm{S'}}$ are optimized on the validation set by using the gradient ascent method. Initial values of $\beta_{\mathrm{S'}}$ were 1 for S' = S, 0 for otherwise.

F_D_TokyoTech_Canon_4

This run used additional training data from the ImageNET dataset. The names of semantic concepts were separated by "or" and transformed into *valid* forms (e.g. cats $\mapsto$ cat) to find synsets whose images can be used for training data. A new classifier $f_{\text{HOG-Image}}$ was trained with positive samples form TRECVID and ImageNET and negative samples from TRECVID. GMM supervectors with the HOG-Dense features were used for the classifier. The final score was calculated as

$$f(X) = \sum_{\mathrm{F} \in \mathcal{F} \cup \{\text{HOG-Image}\}} \alpha_{\mathrm{F}} f_{\mathrm{F}}(X_{\mathrm{F}}), \quad 0 \leq \alpha_{\mathrm{F}} \leq 1, \quad \sum_{\mathrm{F}} \alpha_{\mathrm{F}} = 1. \tag{15}$$

1.6 Results

Our results in the semantic indexing task are illustrated in Figure 1 and Figure 2. The best result (F_A_TokyoTech_Canon_1) was 17.3% in terms of Mean InfAP, which is ranked first in the TRECVID 2011 semantic indexing task.

The second run F_A_TokyoTech_Canon_2 shows the appropriateness of the selection of $\gamma (= \tilde{d}^{-1})$ in Eq. 11, since this run shows almost the same results as the first run in Figure 2.

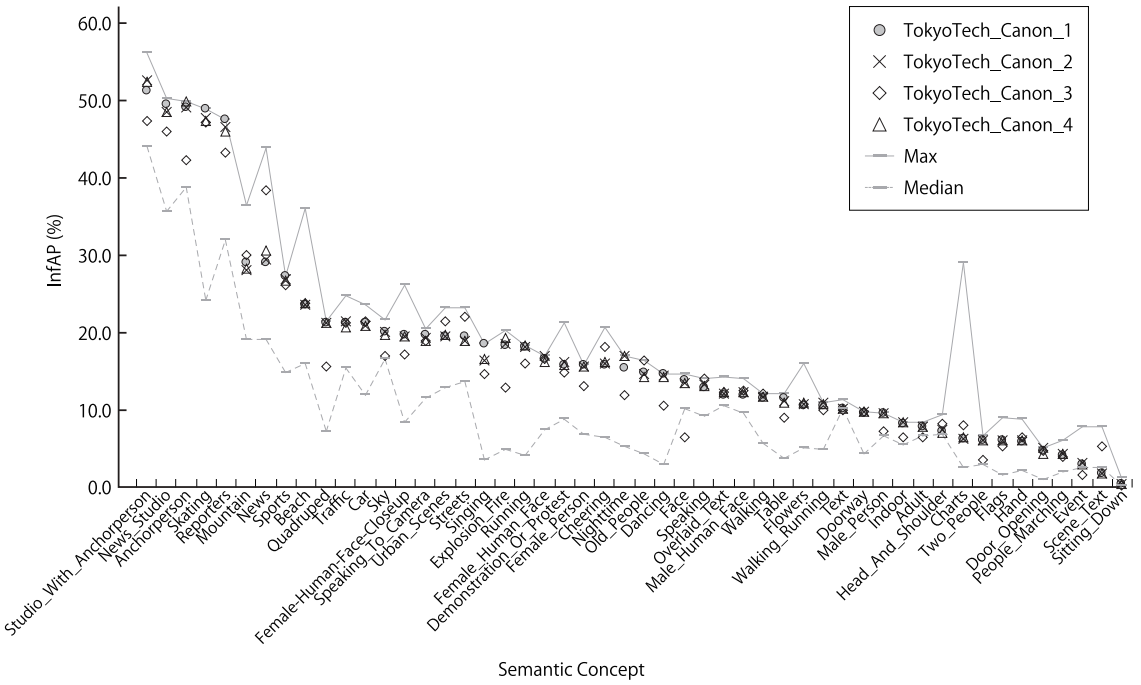

Figure 2: InfAP for each semantic concept.

The result of the run F_D_TokyoTech_Canon_4, which uses additional training data from ImageNET dataset, was 17.2%. Images from ImageNET were added to improve accuracy even if there were only few positive samples in TRECVID dataset, however, semantic concepts evaluated in the testing phase had at least 500 positive samples in the training data. This is because the Mean InfAP was not improved compared with the first and the second runs.

The run F_A_TokyoTech_Canon_3 achieved Mean InfAP 16.4%. The basic idea of this run was to combine all scores from all semantic concepts, however, the overfitting problem occurred in optimizing $\beta_{s'}$ in Eq. 14.

1.7 Conclusion

We proposed a high-performance semantic indexing system using Gaussian mixture model (GMM) supervectors with the six audio and visual features. Our best result was 17.3 % in terms of Mean InfAP, which was ranked 1st over all semantic indexing runs in the full task. Our future work will focus on the localization of semantic concepts and the temporal analysis for video indexing.

2 Multimedia Event Detection

In this section, we will present the details of our system in Multimedia Event Detection (MED) task.

2.1 Introduction

Our MED system consists of feature extraction (2.2), event detection using GMM-Supervectors and SVMs (2.3), and audio event detection using MFCC-HMM (2.4). We applied GMM-Supervector technique with multiple features used in our Semantic Indexing method and confirmed its effectiveness. We also tried to detect sounds which characterize the events using HMMs, to improve the event detection performance. We submitted four runs, all of which were applied to all testing events. In terms of Mean Minimum NDC, the best result of our runs ranked 7th of all 58 full-submission and 3rd among all the teams in TRECVID 2011 MED task.

2.2 Feature Extraction

In this year's MED task, we use five local features, each of which represents different characteristics.

2.2.1 SIFT features with Harris Affine regions and Hessian Affine regions

SIFT (Scale-Invariant Feature Transform) [3] has been effective in many researches of image analysis and video analysis including our semantic indexing method [1, 2]. We extract SIFT features, 128 dimensional vectors from two key point regions; Harris Affine region and Hessian Affine region [5]. These regions are robust for affine transformation. Since extracting SIFT features from all of the frames of all the clips is computationally too expensive, we extract from one frame in every two seconds. Next, we apply PCA (Principal Components Analysis) to reduce the dimension. It saves computational costs in training and detection step. As a result, we compute 32 dimensional vectors.

2.2.2 MFCC features

In MED task, audio is one of the important cues to analyze the video contents of video. We used MFCC (Mel Frequency Cepstral Coefficient) features, which is often used in speech recognition. In addition, we use ΔMFCC,Δpower,$\Delta\Delta$MFCC,$\Delta\Delta$power. The dimension of a feature vector is 38.

2.2.3 HOG and HOF features with STIP

STIP (Space-Time Interest Points) [8] is expected to be more effective to MED than SIFT because SIFT is extracted from still images, but STIP is extracted from spatio-temporal regions. We extract 72 dimensional HOG (Histograms of Oriented Gradient) features and 90 dimensional HOF (Histograms of Optical Flow) features from STIP, and then combine these two vectors. Finally, we convert 162 dimensional vectors into 64 dimensional vectors by applying PCA.

2.2.4 Dense HOG features

We also use 34 dimensional HOG [9] features sampled densely from image space. Differently from features based on keypoints like SIFT or STIP, dense sampling can give us fixed number of features although they may include some noise. In the same way as SIFT, we sample the features from one frame every two seconds.

2.3 Event detection using GMM-Supervectors and SVMs (GS-SVM)

We apply the GS-SVM used in our Semantic Indexing method. We construct a GMM-Supervector per one feature for each clip. GMM-Supervector represents the distribution of the features by one long vector consisted of GMM parameters [1, 2]. We set the number of Gaussian in GMM to 512. We train event models using support vector machines (SVMs) with RBF-kernel. The training data is the set of all of the clips in Development data set and Event kit.

Finally, we fuse the SVM scores of each feature by weighted average and convert them into $[0, 1]$ domain. The weights are determined by 2-fold cross validation using Minimum Normalized Detection Cost (Run 1,2), Average Presicion (Run 3), or no weighted average (Run 4).

2.4 Audio HMMs

Video clips positive for some events often include sounds which characterize the corresponding event. For example, in Flash_mob_gathering, you can often here BGM when people dance.

The ratio of such a period with sounds (Event Period) in duration is 10% in Flash_mob_gathering. Their features may be different from the global features of the whole video clip. Therefore it is expected that we can improve the event detection performance by identifying Event Periods and modeling them.

At first, we annotated Event Period to each positive clip manually. We define Event Period for each event as Table 1. we provide an Event Hidden Markov Model (Event HMM) for each event by using audio features (MFCC, Section 2.2). Because lengths of Event Periods are various, from several seconds to several minutes, we equip Event HMMs with many states. And then we reduce their free parameters by sharing state parameters, in order to avoid deterioration caused by the data sparseness. An Event HMM is left-to-right and has $N \times M$-state, $S_{11}, ..., S_{1M}, ..., S_{i1}, ..., S_{iM}, ..., S_{N1}, ..., S_{NM}$, where $S_{i1}, ..., S_{iM}$ with the same i share the same parameters (Figure 3). Here we set $N = 5$, $M = 60$.

We apply the word-spotting technology in speech recognition to determine the score of each clip for each event. We prepare a garbage model with one state whose parameters are estimated from all video clips. Then we place it before and after each Event HMM and calculate the log likelihood of the Event Period data. Let L_E be the log likelihood obtained by this Event HMM with the garbage models and

Event Name	Event Period
Birthday Party	During birthday party on video
Changing a vehicle tire	From removing a tire, to setting up all tires
Flash mob gathering	During people gathering
Getting a vehicle unstack	From applying force on a vehicle to getting the vehicle unstuck
Grooming an animal	During human touching and grooming an animal
Making a sandwich	During making a sandwich
Parade	During parade
Parkour	From human jumping to the end of the performance
Repairing an appliance	From human touching and repairing an appliance to the end of repairing
Working on a sewing project	During processing clothes

Table 1: Definition of event period

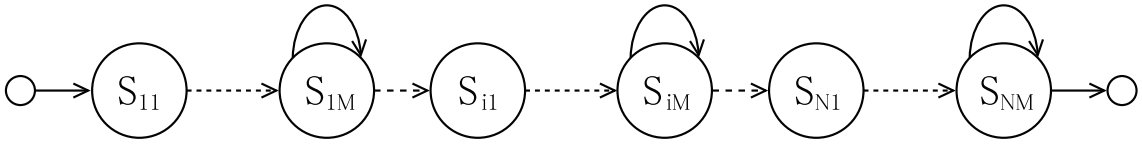

Figure 3: Event HMM topology

L_G be the log likelihood of the same period obtained by using the garbage model. Then the detection score S_{HMM} is defined as the log of the likelihood ratio between L_E and L_G:

$$S_{\text{HMM}} = \frac{1}{f}(L_E - L_G), \tag{16}$$

where f is the number of frames of the Event Period. We fuse S_{HMM} and GS-SVM scores by weighted average and make it as the event score.

In our preliminary experiment, this method was better than GS-SVM alone in Flash_mob_gathering, Parade, and Repairing_an_appliance. We applied it to these three events in Run1.

2.5 Runs and Results

We submitted four runs and Table 2 shows their brief description. Run 2-4 use GS-SVM systems. Their difference is the measure to decide the weights of each feature's SVM score. Run 2 uses Minimum Normalized Detection Cost. Run 3 uses Average Precision, which is often used in the validation of searching system. In Run 4, we use no weight, the SVM score of each feature is simply averaged. Run 1 is our primary run, which is the combination of Run 2 and audio event detection.

Figure 4 shows the overview of TRECVID 2011 MED task. In terms of Mean Actual NDC, our best run is ranked at 5th of all 58 full-submission and 3rd among all the teams. In terms of Mean Minimum NDC, our best run is ranked at 7th of 58 full-submission and 3rd among all the teams.

Table 3 shows the Actual NDC and Mimimum NDC of our four runs. The difference between Minimum NDC and Actual NDC is 0.032 on average. It means that our detection thresholds decided by 2-fold cross validation are about appropriate. However, it seems that we failed to set the optimized weights. We assumed that Run 2 gave better performance in the term of Mimimum NDC than Run 3 and Run 4. The cause may be that the population of the test data is different from that of the training data. In future, we try to develop a method to optimize the weight of each feature by fitting to the test data automatically.

From comparison of Run 1 and Run 2, it is shown that the gain of Audio HMMs doesn't appear as we expected even though we can see the affect in preliminary result. We speculate that nature of positive clips in training data are so different from test collection, or test data contains many negative clips which are similar to positive clips in training data.

Figure 5 shows the Minimum NDC of each features and combination of multiple features. STIP and Dense HOG features are shown to be effective in almost the all events. STIP gave, in particular, good performance to the event including dynamic moving like jumping of "Parkour", walking of "Parade" or dancing of "Flash mob gathering". Dense HOG features may be effective to represent static objects

Run ID	System ID	Subsystem(s)	The measure to decide the weight of each feature
Run 1	p-GSSVMndcHMM-r1	GS-SVM + Audio HMMs	Minimum NDC
Run 2	c-GSSVMndc-r2	GS-SVM	Minimum NDC
Run 3	c-GSSVMap-r3	GS-SVM	Average Precision
Run 4	c-GSSVMave-r4	GS-SVM	No weight

Table 2: Description of our four runs

Event Name	Actual NDC				Minimum NDC			
	Run1	Run2	Run3	Run4	Run1	Run2	Run3	Run4
Birthday party	.658	.658	.654	**.588**	.636	.636	.636	**.585**
Changing a vehicle tire	**.558**	**.558**	.567	.563	.554	.554	.556	**.541**
Flash mob gathering	.339	.347	**.326**	.353	.325	.326	**.324**	.334
Getting a vehicle unstuck	.429	.429	.444	**.388**	.406	.406	.421	**.388**
Grooming an animal	.754	.754	.713	**.669**	.693	.693	**.639**	.644
Making a sandwich	.755	.755	**.697**	.736	.735	.735	**.687**	.714
Parade	.596	**.511**	.514	.534	.582	.504	**.500**	.510
Parkour	.479	.479	**.418**	.506	.444	.444	**.400**	.441
Repairing an appliance	.557	.557	.559	**.512**	.454	.424	**.420**	.479
Working on a sewing project	.716	.716	**.688**	.733	.682	.682	**.664**	.679
Mean	.584	.576	**.556**	.558	.551	.540	**.525**	.531

Table 3: The Actual NDC and Minimum NDC of our runs. Bold is the best of the four runs.

like stopping car of "Changing a vehicle tire". As to audio MFCC features, it did not give so high performance by itself, but by combining with STIP and HOG features, the performance become much better as shown in "Repairing an appliance" or "Birthday party". Another fact we can see from the figure is that SIFT is not so effective, while it was effective in our experiment using training data. we confirmed the effectiveness of SIFT if the weight of SIFT is optimized. This may be because the average of the number of feature point of SIFT (Harris:∼42,000, Hessian:∼38,000) is less than that of STIP (∼78,000) and HOG (∼350,000). The use of more frames when extracting SIFT or the use of dense SIFT may improve the performance.

2.6 Conclusion

In this year's MED task, we introduced GMM supervector technique used in our Semantic Indexing method and confirmed its significant effectiveness. We used five features to represent different characteristics. Since our results of preliminary experiments showed that the best features are different between

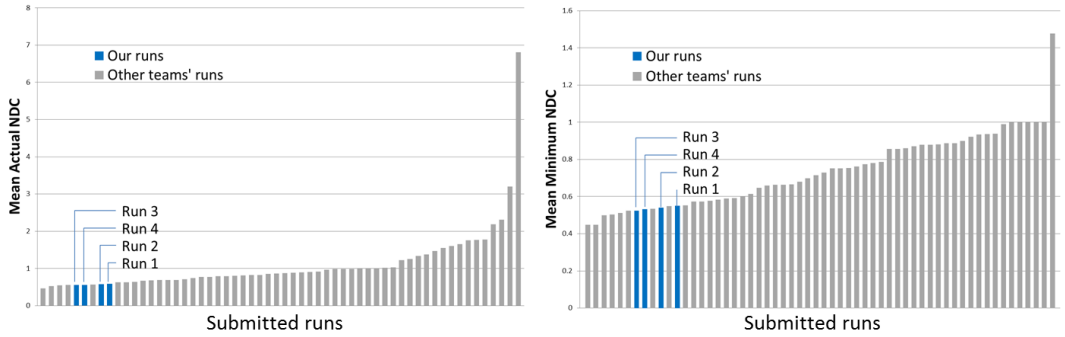

Figure 4: Overview of Mean Actual NDC and Mean Minimum NDC in TRECVID 2011 Multimedia Event Detection task.

events, we try to optimize the weight of each feature by cross-validation.This optimization, however, did not always improve the detection performance. This will be our future work.

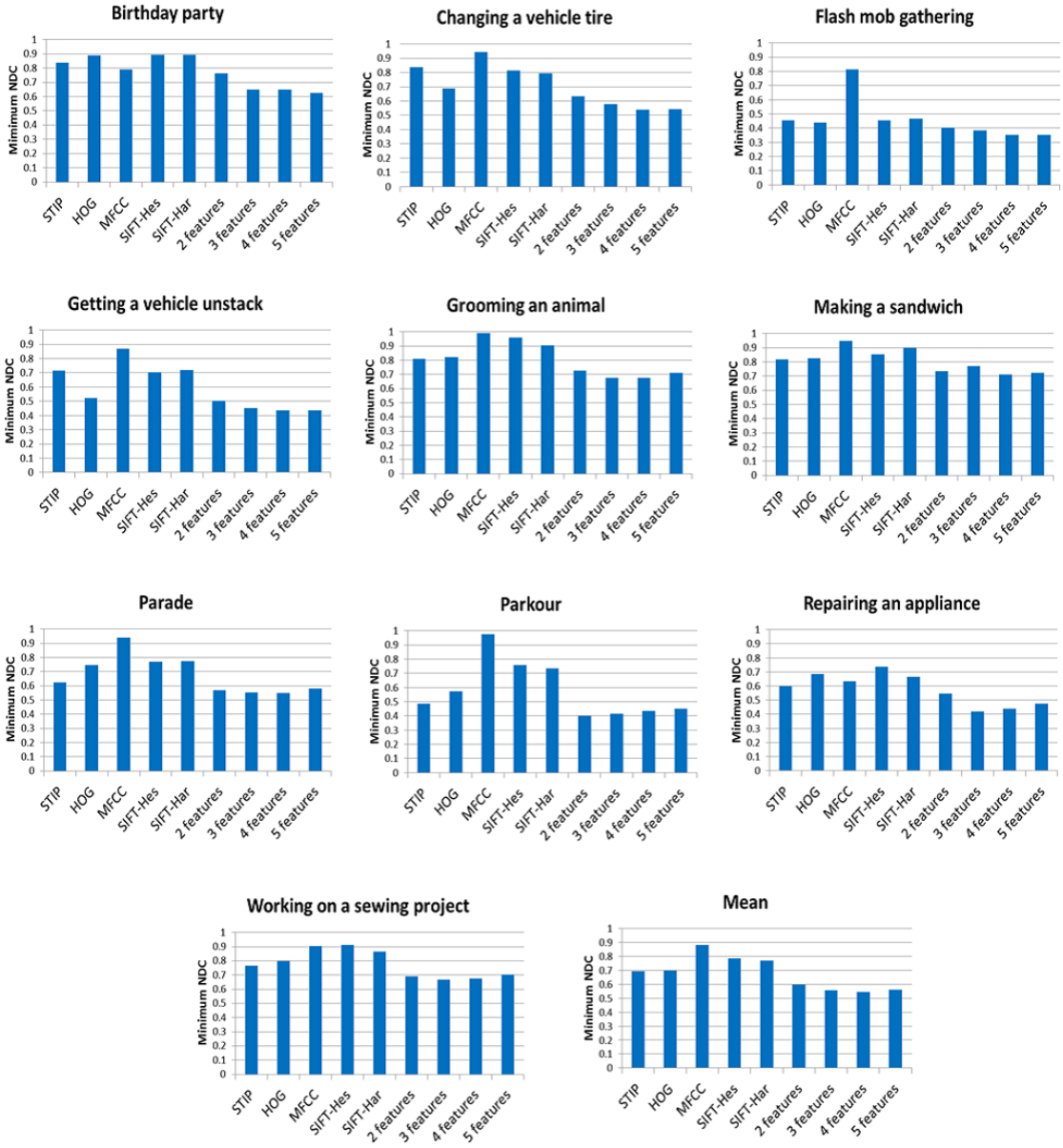

Figure 5: The Minimum NDC of each feature and multi features. Note that 2 features are STIP and HOG, 3 features are STIP, HOG and MFCC, 4 features are STIP, HOG, MFCC and SIFT-Hes, 5 features are STIP, HOG, MFCC, SIFT-Hes and SIFT-Har.The features are fused by score average, then 5 features means our Run 4. These results are evaluated using the preliminary version of the reference files.

3 Surveillance Event Detection

3.1 Introduction

As in the previous year, we combine local and global features and use them for detecting events. Local features obtained from a person detector represent movements of individuals. Global features based on optical flow represent the movement of crowds. This year, we improved local features. We selected three events out of the seven provided events: PersonRuns, PeopleMeet and PeopleSplitUp.

3.2 Local features

3.2.1 Outline

First, we detect persons using a particle filter. Then, we associate persons between successive frames and obtain his/her trajectory. Finally, we detect each of event using our event-trajectory model. We improved each these three steps this year. We use HOG features extracted from a person region for calculating a person existence probability. In addition, we refine background subtraction, and normalize the existence probability using the window size for a person. To obtain a person trajectory, we associate persons using the similarity of LBP histograms [10]. Our new event-trajectory models use a normalized velocity and the distance between two persons. Each step is detailed below.

3.2.2 Calculation of the person existence probability

In this year, we prepare two new ways for calculating the existence probability of a person. We submit one run for each.

HOG-Cascade
For each pixel in an image frame, the person existence probability is calculated using a cascade of person detectors [11]. Its input is an image with a size of 48×80 pixels, whose center is located at the pixel. This cascade sequentially connects 20 strong classifiers, each of which consists of 50 weak classifiers based on HOG features. We calculate the person existence probability as a function of the number of strong classifiers that the input passes through. The function $P(n)$ is defined as the ratio of the number of positive samples in the training samples which passed through exactly n classifiers, to the number of the total training samples which passed through exactly n classifiers.

HOG-SVM
Here the existence probability is calculated using a SVM whose input is a HOG feature vector extracted from an image with a size of 48×128 pixels. The vector's length is 2,700. Since the SVM's output score S_{SVM} is not a probability, it is transformed into a probability P_{person} by using the following sigmoid function.

$$P_{\mathrm{person}} = \frac{1}{1 + \exp(a(-S_{\mathrm{SVM}} + b))}. \tag{17}$$

The parameters a and b are optimized by using training data.

3.2.3 Person detection using a particle filter and clustering

A particle filter [12] is an estimator which estimates the current condition by considering past information, also known as sequential Monte Carlo method. We use 800 particles. The particles tend to concentrate to the regions where persons are likely to exist. We cluster the particles by using the furthest neighbor method, where we set a threshold on the distance between clusters. We identify each resulting cluster as a person. In order to improve the detection performance, we have changed the background subtraction and added a probability normalization process.

Background subtraction
Last year, the background image included static persons. But static persons also participate in PeopleMeet and PeopleSplitUp events. Therefore, this year we use a background image which does not include persons from training data. The way to apply the background subtraction is the same as that of last year. That is, we multiply the existence probability of a person by a factor β:

$$\beta = 1 - \alpha \exp\left(-\frac{d^2}{2\sigma^2}\right), \tag{18}$$

where

$$d = \sqrt{(r_a - r_b)^2 + (g_a - g_b)^2 + (b_a - b_b)^2} \tag{19}$$

is the difference between the color (r_a, g_a, b_a) of the pixel and the background color (r_b, g_b, b_b) of the same pixel. And α is the subtraction factor which controls the effect of the background subtraction.

Table 4: Values of α, ω, and T_C

	α	ω	T_C				
			Camera1	Camera2	Camera3	Camera4	Camera5
PersonRuns	50	0.30	44	57	58	100	131
PeopleMeet	150	0.77	187	140	201	176	173
PeopleSplitUp	150	0.80	48	55	142	45	86

Probability correction based on sub-window size
Because the target videos are recorded by fixed cameras, it is easy to estimate the size of a person in a fixed position of a frame image. We use this information to normalize the existence probability. Its purpose is to reject too small or too large persons compared with the estimated person size from its location. The process is as follows. First, we obtain the relational expression $h(y_p) = ay_p + b$ between the y-coordinate of the center of a person y_p and the estimated height of the person $h(y_p)$ by using linear regression estimated from training data. Then, we multiply the person existence probability of a particle, whose y-coordinate is y_p and sub-window height is h_p, by a factor γ:

$$\gamma = \exp\left(-\frac{(h_p - h(y_p))^2}{2\sigma^2}\right) \tag{20}$$

3.2.4 Human tracking

In order to obtain the trajectory of a person, each person in the i-th frame is associated with its corresponding person in the $(i+1)$-th frame. First, we estimate the position of person A at the $(i+1)$-th frame by using A's position and velocity at the i-th frame. The candidates to be associated with A are those detected at the $(i+1)$-th frame whose Euclidean distances from A are less than a predetermined threshold d_{th}. Then, we calculate the LBP histograms of A and the candidates. Among the candidates, we select the one which has the minimum Bhattacharyya distance from A.

3.2.5 Event detection using the event-trajectory model

We describe how to detect an event using the trajectory information. The input features of each trajectory are the positions and velocities of the associated persons for five frames. Let $\mathbf{v}_A(i)$ be the velocity of person A normalized by its size at time frame i, and $d_{AB}(i)$ be a distance between person A and person B. Then, we define the score for each of the three events S_{local} as follows:

- PersonRuns: $S_{\text{local}} = \max_{0 \le i \le 4} |\mathbf{v}_A(i)|$
- PeopleMeet: $S_{\text{local}} = \begin{cases} \sum_{i=0}^{3}(d_{AB}(i) - d_{AB}(i+1)) & (\forall i\ d_{AB}(i) > d_{AB}(i+1)) \\ 0 & (\text{otherwise}) \end{cases}$
- PeopleSplitUp: $S_{\text{local}} = \begin{cases} \sum_{i=0}^{3}(d_{AB}(i+1) - d_{AB}(i)) & (\forall i\ d_{AB}(i+1) > d_{AB}(i)) \\ 0 & (\text{otherwise}) \end{cases}$

We set the score for PeopleMeet and PeopleSplitUp to zero when there is at most one person in the frame image.

3.3 Global features

We use optical flow vectors [13] as global features. Using these vectors as input, the output score S_{global} from SVMs shows the degree of likelihood of an event occurring.

3.4 Combining local features and global features

The score S_{local} is obtained by using the event-trajectory models that use local features. And the score S_{global} is obtained by using the SVMs based on global features. We define a score S_C by combining these two scores with a weight ω as follows:

$$S_C = (1 - \omega)S_{\text{local}} + \alpha\omega S_{\text{global}}. \tag{21}$$

Table 5: Our SED results of TRECVID 2010

	Ref	Sys	CorDet	FA	Miss	Act.DCR	Min.DCR
PersonRuns	107	7	0	7	107	1.0023	1.0023
PeopleMeet	449	8	0	8	449	1.0026	1.0026
PeopleSplitUp	187	43	1	42	186	1.0084	1.0084

Table 6: Our SED results of TRECVID 2011

HOG-Cascade	Ref	Sys	CorDet	FA	Miss	Act.DCR	Min.DCR
PersonRuns	107	174	3	171	104	1.0280	1.0003
PeopleMeet	449	151	8	143	441	1.0291	1.0003
PeopleSplitUp	187	598	51	547	136	0.9066	0.8976

HOG-SVM	Ref	Sys	CorDet	FA	Miss	Act.DCR	Min.DCR
PersonRuns	107	205	0	205	107	1.0672	1.0003
PeopleMeet	449	148	8	140	441	1.0281	1.0003
PeopleSplitUp	187	608	51	557	136	0.9099	0.9066

The frames in which S_Cs are larger than the predetermined threshold T_C are detected as the event frames. The values of α, ω and T_C are shown in Table 4.

3.5 Results

Table 5 shows our last year's detection results for the three events, and Table 6 shows our results for this year. This year, our NDCR scores for PersonRuns and PeopleMeet are still larger than 1. On the other hand, our NDCR scores for PeopleSplitUp improved by 10.1% and are less than 1. In our preliminary experiments using dry-run data set, the new person detection process improved the person detection accuracy by 6.6% (the NDCR score by 0.0264) and the new person tracking process improved the NDCR score by 0.0052, and the new event detection process improved the NDCR score by 0.0123. We have also fixed bugs related to the calculation of S_{global} in our formal-evaluation runs. These might be the reasons for the improvement in PeopleSplitUp.

The PeopleMeet event is the reversed event of PeopleSplitUp, but its result was not significantly improved from last year. Since the end timing of PeopleMeet and the start timing of PeopleSplitUp share the similar video features, these timings can be detected equally. Also, it is easy to detect the end time for PeopleSplitUp because a person usually leaves the image frame. On the other hand, it is harder to detect the start timing for PeopleMeet, the timing when two persons start to communicate. We think this is the reason why we did not achieved improvement for PeopleMeet. We did not achieved improvement for PersonRuns either. We assume persons are all adults, so children tend to fail to be detected compared with adults. But there exist, particularly in this event, many children. We think this is the reason why we did not improved.

As explained in section 3.2.2, we prepared two new ways for calculating the person existence probability and submit one run for each. But there is no significant difference between the two runs. For detecting the three events, we use the same way to detect and track persons, but we use different event-trajectory models for each event. In the future, in order to increase the detection performance for PersonRuns and PeopleMeet, we will improve the event-trajectory models of these events. Also, we will improve the process to use global features.

References

[1] N. Inoue, and K. Shinoda. A Fast MAP Adaptation Technique for GMM-supervector-based Video Semantic Indexing Systems. In Proc. of *ACM Multimedia* (short paper), 2011.

[2] N. Inoue, and et al. High-Level Feature Extraction using SIFT GMMs and Audio Models. In Proc. of *ICPR*, 2010.

[3] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *In IJCV*, 2004.

[4] J. van de Weijer and C. Schmid. Coloring local feature extraction. In Proc. of *ECCV*, vol.2, pages 334–348, 2006.

[5] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. In *IJCV*, 60(1):63–86, 2004.

[6] A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms, http://www.vlfeat.org/, 2008,

[7] S. J. Young, G. Evermann, M. J. F. Gales, D. Kershaw, G. Moore, J. J. Odell, D. G. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. The htk book, version 3.4, 2006.

[8] I. Laptev. On space-time interest points. *In IJCV*, vol.64(2), pp.107-127, 2005.

[9] N. Dalal and B. Triggs.B. Histograms of oriented gradients for human detection. In Proc. of *CVPR*, 2005.

[10] T.Ojala, M.Pietikainen and D.Harwood. Acomparative study of texture measures with classification based on feature distribution. *Pattern Recognition*, vol. 29, no. 1, pp. 51-59 1996.

[11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR*, vol. 1, 2001.

[12] M. Isard and A. Blake. Condensation: Unifying low-level and high-level tracking in stochastic framework. *Proc. ECCV*, 1998

[13] B. K. P. Horn and B. G. Schunck, Determining optical flow. *Artifical Intelligence*, vol. 17, no. 1, pp. 185-203, 1981.