NHK STRL at TRECVID 2009: Surveillance Event Detection and High-Level Feature Extraction

Masaki Takahashi[†] [‡] Yoshihiko Kawai[†] [*] Mahito Fujii[†] Masahiro Shibata[†]

Noboru Babaguchi[*] Shin'ichi Satoh[‡] [§]

† NHK Science and Technology Research Laboratories, 1-10-11 Kinuta, Setagaya-ku, Tokyo, Japan

‡ The Graduate University for Advanced Studies, Shonan Village, Hayama, Kanagawa, Japan

* Osaka University, 2-1 Yamadaoka, Suita-shi, Osaka, Japan

[§]National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan

Abstract

NHK Science and Technology Research Laboratories participated in two tasks at TRECVID 2009: surveillance event detection task and high level feature extraction task. For surveillance event detection tasks, we targeted four events: "PersonRuns", "PeopleMeet", "ObjectPut", and "OpposingFlow". The proposed method detects human regions using HOG descriptor and SVM classifier and tracks human regions using 2D color histograms. It recognizes each event based on trajectories of people and optical flow. For the high-level feature extraction task, we calculate feature vectors based on local features and global features, and then classify key-frames using a ball vector machine (BVM). The proposed method uses both SIFT feature and SURF feature to get various kinds of local feature points and descriptors. Features such as color moments, wavelet texture, local binary patterns, and face appearance are used as global features. We adopt the BVM method to reduce the computational cost of training.

1 Introduction

1.1 Surveillance event detection

With the recent rapid spread of the surveillance camera, the demand for development of equipment which can not only track a human object but also detect abnormal action automatically has been increasing. Many researchers have been studying the recognition of human motion, action, and events through video content analysis [1]-[3]. However, most of these technologies have been developed assuming relatively simple videos in which it is comparatively easy to detect and track people. Technologies have not yet been developed that can robustly detect abnormal actions in crowded situations in real surveillance video in places such as stations or airports.

We targeted four events in TRECVID Surveillance event detection task: “PersonRuns”, “Opposing Flow”, “PeopleMeet”, and “ObjectPut”. Our proposed method detects person regions using a Histogram of Oriented Gradients (HOG) descriptor [4], [5] and a Support Vector Machine (SVM) classifier [6], [7], and creates trajectories by tracking regions that contain people. We then determine whether or not the trajectory represents specific events on the basis of feature values taken from the trajectory.

1.2 High level feature extraction

Recent increases in the speed of networks and the popularity of digital recording devices such as hard disk recorders have made it common for individuals to retain large quantities of video data. Because of this, there is a demand for a method for efficiently retrieving desired video footage from the stored video data. Broadcast stations also require new techniques to efficiently search for necessary video footage from the huge archives of past TV program videos they have created in order to effectively use their video resources. Analysis based on the semantic content of video footage is important for efficient retrieval. Our research focuses on the detection of generic objects or events (called high-level features) within video footage, to investigate a versatile method that can be adapted for various different high-level features by replacing the training data. Typical related studies include methods that calculated local features, such as scale invariant feature transform (SIFT) [10] and categorized images on the basis of frequency histograms of clustered local features (called visual words) [11], [12]. Methods have also been developed that use the latent Dirichlet allocation (LDA) [13] to categorize images [14] or consider the positional relationships of local keypoints [15], [16]. These methods based on local features are called “bag-of-keypoints” or “bag-of-visual-words” and have been verified in the field of generic object recognition and at previous TRECVID workshops [17], [18].

The proposed method also detects high-level features on the basis of bag-of-keypoints approach. We try to classify more accurately by combining two algorithms to extract local features. Global features are also used, such as color and texture [19], [20] and the results of face detection. We employ the ball vector machine (BVM) [21], which is a high-speed support vector machine (SVM) algorithm, in order to reduce computational cost for training a classifier.

This paper is organized as follows. Section 2 describes the method of detecting surveillance events and the experimental results for it. Section 3 gives a detailed description of the method of extracting high-level features and the experimental results for it. We conclude the paper in Section 4.

2. Surveillance event detection

2.1 Overview

Our system consists of the three steps shown in Fig.1. Step 1 is human detection. The system detects regions with a human in them from the input image using a HOG descriptor and an SVM classifier. Step 2 is human tracking. The system tracks the region by evaluating the distance from a 2-dimensional color histogram. It robustly tracks human regions using a Kalman Filter. Step 3 is event recognition. The system judges events based on a trajectory of the human region and the optical flow in that region. In the following subsections, we explain the processes in each step.

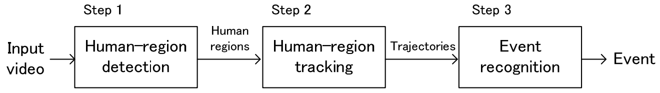

Fig. 1: Overview of event detection

2.2 Human-detection process

2.2.1 Processing flow

In the human detection process, the system detects regions with a human following the flow in Fig. 2.

The pre-processing and post-processing are done before and after the processing of human detection. The preprocessing (changed area detection) contributes to reducing the total processing cost by limiting the areas for human searches. The post-processing (clustering) contributes to stabilizing the region for human tracking.

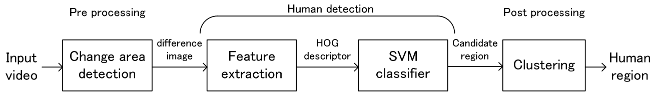

Fig. 2: Flow of the human-detection process

2.2.2 Changed area detection

When the system searches for human regions in all ranges of the input image, the processing cost is extremely high. Thus, it searches human regions only where there are frame differences or background subtractions. Fig. 3 shows a sample of a background-subtraction image. The system can detect even a standing person by referring to the image.

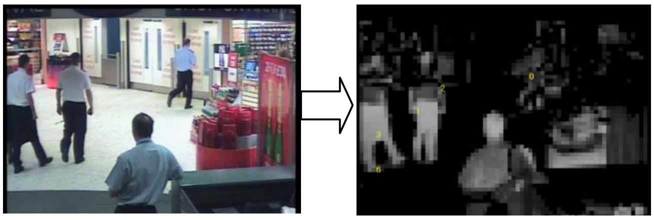

Fig. 3: Detection of changed area

2.2.2 Human detection

Our human detector searches regions with a human by calculating image features around the changed region. We adapted a human detector that combined the HOG feature descriptors and an SVM. Many studies have reported that the HOG is suitable for detecting regions with a human because it is robust to the wide range of variations of poses [8].

We trained the classifier with about 1,000 HOG descriptors in human regions (positive data) and about 2,000 HOG descriptors in other regions (negative data) for each camera. Some samples are shown in Figs. 4 and 5. All sample data were cut out manually from the development-video datasets.

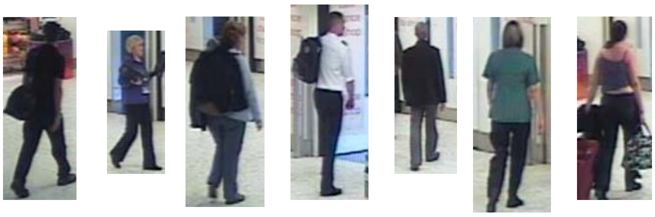

Fig. 4: Samples of positive data

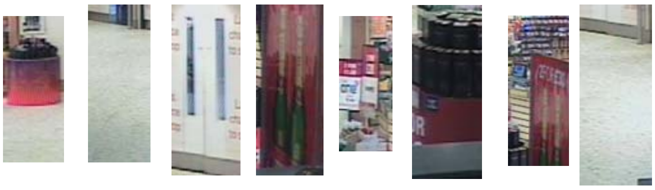

Fig. 5: Samples of negative data

2.2.3 Clustering human regions

More than one candidate-human region is usually detected around one person. Therefore, the system clusters the candidate-human regions, and determines a representative human region for one person from candidate-human regions in each cluster.

The similarity between nearby candidate-human regions is evaluated by calculating the distance in a color histogram. We used a 2-dimensional color histogram of HSV color space (H, S). Fig. 6 shows samples of the 2D color histogram. The similarity in the 2-dimensional color histogram is calculated by using the Bhattacharyya distance [9]. A candidate-human region, which is located in the center, is selected as a representative-human region from the cluster.

Fig. 6: Samples of 2-dimensional color histogram

2.3 Human tracking

2.3.1 Processing flow

The human region is tracked following the flow in Fig. 7.

The system tracks the regions with a human by searching a similar human region in the past image. The similarity is calculated based on their distance in the 2-dimensional color histogram same as the clustering process.

If a similar human region is detected, the same ID in the past frame is set to the current human region. Otherwise, a new ID number is set. By connecting the position of the same ID region, we can obtain a trajectory of the human region. Fig. 8 shows a sample of the trajectory of the human region.

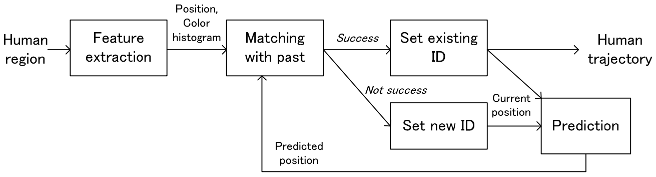

Fig. 7: Flow of the human-region tracking

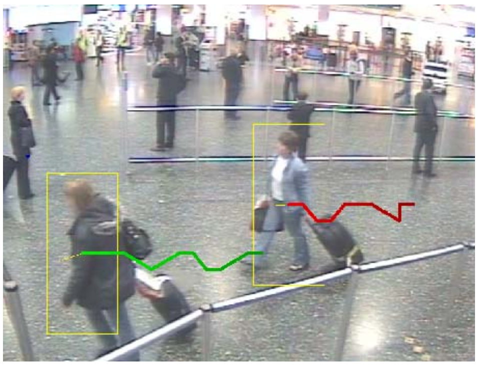

Fig. 8: Sample of human-region trajectory

2.3.2 Prediction processing

That approach, however, does not necessarily detect the position of the human region in every frame. Detection can fail when occlusion occurs in a crowded scene. For that reason, we supplemented the system with a prediction-based re-tracking function using a Kalman filter. The prediction continues even after a detection failure, so it is possible to select the human region again after it has passed through an area where detection is difficult.

The system tries to match the 2-dimensional color histogram information of the missing human region with any new human regions around the predicted position. If the distance of color histogram in two regions is low, the system connects the new detected position with the missed trajectory. This prediction processing contributes to the robustness of human-region tracking.

2.4 Event recognition

2.4.1 Features from trajectory

A trajectory of detected human regions contains a great deal of information on human motion, such as the walking speed and the travel distance. The event recognition process recognizes specific events based on the features which are extracted from a trajectory.

However, the length of the motion vector differs according to the detected position in the image coordinates. For example, the motion vectors of people who are detected in front of the image are large, and the motion vectors of people who are detected behind the image are small. Consequently, we normalized before calculating the features.

In addition, if the system extracts these features from the full length of a trajectory, the features fade away by averaging them. Therefore, we divided the trajectories every one second.

We extracted ten features from a trajectory:

- First detected position (x, y)
- Last detected position (x, y)
- Total vector (x, y)
- Travel distance
- Average velocity
- Acceleration
- Linearity

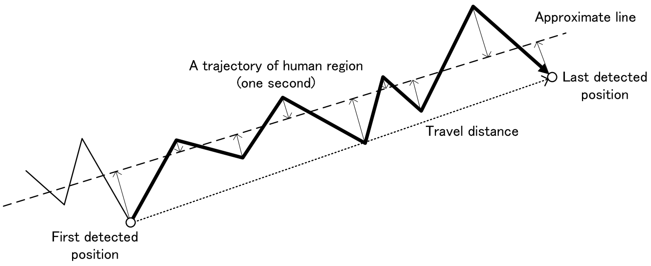

Fig. 9: Trajectory of human region

Fig.9 shows relations between the detected human trajectory and its features. The total vector is calculated by accumulating the motion vectors of each frame in a trajectory. The system calculates the vector horizontally (x) and vertically (y). The travel distance means the distance from the first detected position to the last detected position in a trajectory. We can reject a trajectory of a human who is walking around in the same position by referring to this travel distance. The average velocity is the average length of motion vectors in a trajectory. The acceleration is a shift in velocity. We can calculate this feature by differentiating velocities. We can detect a person who has stopped or suddenly started to move from this feature. The linearity is the average distance from each detected position to an approximate line. The approximate line is calculated by least-square method from the center positions of human regions. If a person has walked straight ahead, the linearity is close to zero. As a person who is almost running is moving straight ahead, this feature can effectively detect running people.

2.4.2 PersonRuns

When a PersonRuns event occurs, a human region moves rapidly compared to human regions with people walking normally. Our event detector detects the PersonRuns event based on the average velocity, the travel distance, and linearity in particular.

The system occasionally mistracks, especially when it is tracking a running person because of his/her large motion vectors. Thus, the system verifies the trajectory by searching backward after the PersonRuns event has occurred forward. The system tracks a person in the same way as forward tracking, and if the PersonRun event is detected backward as well, our detector gives the event a high decision score.

2.4.3 OpposingFlow

When an OpposingFlow event occurs, a person is walking in an unusual direction. Therefore, it is necessary to calculate the direction of a trajectory to detect the OpposingFlow event.

Our event detector detects the OpposingFlow event based on the total vector and the travel distance in particular.

2.4.4 PeopleMeet

To detect a PeopleMeet event, we need to consider the trajectories of more than one person. However, it is difficult to accurately track human regions simultaneously in a crowded scene. Therefore, we observed and determined the features of a PeopleMeet action that could be extracted from a trajectory.

When a PeopleMeet event occurs, a person is moving over a long distance, slows his/her walking speed, and finally stops. Therefore, our event detector detects the PeopleMeet event based on the acceleration and the travel distance in particular.

2.4.5 ObjectPut

An ObjectPut event is a small action compared to the other three events. As it is difficult to detect small actions only from a trajectory, we took into consideration the movement of a person's body by calculating the optical flows in the detected human region.

After a person is detected in the same position, the system calculates optical flows in the human region. An event is detected if numerous flows in the downward direction are detected. Fig.10 shows samples of detected optical flows.

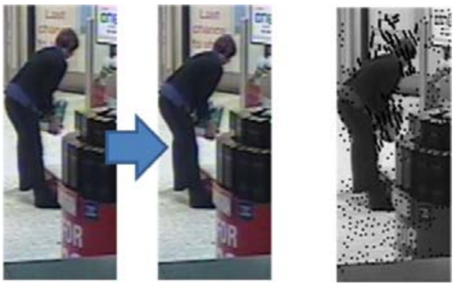

Fig. 10: Optical flows in ObjectPut scene

2.5 Results

The actual and minimum DCRs for each of the events can be seen in Table 1. We can see from this table that our system is not necessarily sufficiently accurate. There is still room to improve the detecting and tracking accuracy especially in crowded scenes. In addition, we should use not only local features but also global features for event recognition.

We recently focused on accurately detecting human regions in an image, so that the detecting accuracy was comparatively reliable. We hope to further improve tracking and recognition processing for TRECVID 2010.

Table 1: Results for detecting events

Event	Act.DCR	Min.DCR
PersonRuns	0.971	0.965
OpposingFlow	1.027	1.029
PeopleMeet	1.174	0.999
ObjectPut	1.123	1.022

3. **High level feature detection**

3.1 Method overview

An overview of the proposed method is shown in Fig. 11. We first detect shot boundaries within the video footage and extract a frame at the temporally-central position of each shot as a key frame. The proposed method uses these key frames as representative images of shots. The method then obtains local features and global features from each key frame, classifies the frame images on the basis of each of these features, and finally integrates the results to extract high-level features. For local features, the method uses the two algorithms SIFT [10] and speeded up robust features (SURF) [22] to obtain keypoints and feature descriptors. It then creates a visual vocabulary by clustering the extracted features, and calculates feature vectors on the basis of how frequently the visual words appear. We use a method for applying weighting on the basis of the distance from each visual word [23] to create feature vectors. For global features, we combine the results of color moment, Haar wavelet, local binary pattern (LBP) [24], and face detection [25] as feature vectors.

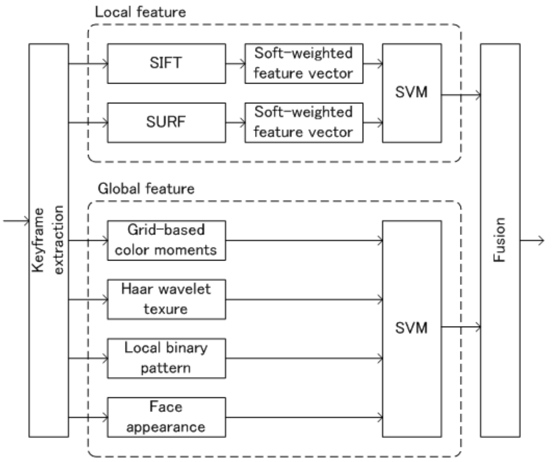

Fig.11: Overview of proposed method

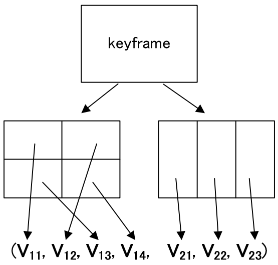

Fig.12: Calculation of local feature vector

3.2 Local Feature

The proposed method creates visual vocabularies for each SIFT and SURF, and it calculates feature vectors on the basis of them. The two vectors are linked to form a feature vector for the key frame. We expect that combining two different algorithms, SIFT and SURF, ensures that important interest points can be detected without being lost, enabling the features of an object to be captured accurately. The proposed method also calculates feature vectors for each grid region, as shown in Fig. 12, to enable positional information within the frame to be considered. For the number of divisions of the grid, we decided on 2x2 and 1x3 after considering the experimental results of previous work [18].

To create feature vectors, we use a method [23] that considers the distance between a feature descriptor at a detected keypoint and a visual word. This differs from the approach of the conventional method, which allocates a detected keypoint to one visual word, and is based on the concept that one keypoint can belong to a number of visual words. If there are K visual words, we calculate a K-dimensional feature vector $T = (t_1, \ldots, t_k, \ldots, t_K)$. We calculate each vector element t_k from the following equation:

$$t_k = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \frac{1}{2^{i-1}} sim(k_j, w_k), \qquad (1)$$

where M_i denotes the total number of keypoints at which the ith closest visual word is w_k, and $sim(k_j,\ w_k)$ denotes the degree of similarity between keypoint k_j and the visual word w_k. N is a constant that denotes the depth of distance considered, where N is set to 4 for the proposed method, in a similar manner to that of Jiang et al. [23].

3.3 Global Feature

3.3.1 Grid-based Color Moments

A color moment feature represents the color distribution in an image. The proposed method converts the input image into the HSV color space and Lab color space, then calculate the average pixel value μ, the standard deviation σ, and the cube root s of skew s. The equations for calculating these values are as follows:

$$\mu_c = \frac{1}{HW} \sum_{x} \sum_{\mathrm{y}} f_c(\mathrm{x}, y) \qquad (2)$$

$$\sigma_c = \left\{ \frac{1}{HW} \sum_{x} \sum_{y} \{f_c(x, y) - \mu_c\}^2 \right\}^{1/2} \qquad (3)$$

$$s_c = \left\{ \frac{1}{HW} \sum_x \sum_y \{f_c(x,y) - \mu_c\}^2 \right\}^{\frac{1}{3}} \qquad (4)$$

Here, $f_c(x, y)$ denotes the value of the component c ($c \in \{h, s, v, l, a, b\}$) at coordinates (x, y), and H and W denote the height and width of the image. The values of μ_c, σ_c, and s_c calculated for each component are combined to create a feature vector.

3.3.2 Haar Wavelet Texture

We divide each frame image into a 3 × 3 grid and applied Haar wavelet transforms in three stages to each region. We calculate the standard deviation of pixel values for each subband region and then link them together to obtain a feature vector.

3.3.3 Local Binary Pattern

A local binary pattern [24] denotes the density magnitude pattern of pixels surrounding the target pixel. The following is the equation for calculating the local binary pattern $L_{P,R}$ that is calculated from P pixels around the circumference of a circle of radius R:

$$L_{P,R}(x,y) = \begin{cases} \sum_{P=0}^{P-1} \delta_{P,R}(x_p, y_p), & if \;\; U_{P,R}(x,y) \\ \qquad P+1, & otherwise \end{cases} \qquad (5)$$

Here, $\delta_{P,R}$ denotes the magnitude relationship of the luminance values of the target pixel (x, y) and the surrounding pixels $(x+x_p, y+y_p)$, which is calculated from the following equation:

$$\delta_{P,R}(x_p, y_p) = \begin{cases} 1, & f(x+x_p, y+y_p) - f(x,y) \geqq 0 \\ 0, & otherwise \end{cases} \qquad (6)$$

The values of x_p and y_p are expressed as follows:

$$\begin{cases} x_p = Rcos\frac{2\pi p}{P} \\ y_p = Rsin\frac{2\pi p}{P} \end{cases} (0 \leqq p \leqq P-1) \qquad (7)$$

The term $U_{P,R}$ in Equation (5) denotes the total number of times that 0 and 1 change in the $\delta_{P,R}$ sequence, which is calculated from the following:

$$U_{P,R}(\mathrm{x,y}) = |\delta_{P,R}(x_p, y_p)| + \sum_{p=1}^{P-1} |\delta_{P,R}(x_p, y_p) - \delta_{P,R}(x_{p-1}, y_{p-1})| \qquad (8)$$

We calculate $L_{P,R}$ (where $0 \leqq L_{P,R} \leqq P+1$) of Equation (5) for all pixels within the image, and take the resultant frequency histogram as a feature vector. In practice, to ensure resistance to changes in resolution, we calculate the frequency histogram for each $L_{P,R}$ for three combinations of (P,R) = (8,1), (16,2), then link them together to obtain the final feature vector.

3.3.4 Face Appearance

Since many of the high-level features that were set in TRECVID 2009 focus on people as subjects, we added a feature relating to faces to the proposed method. We use the total number, average size, maximum size, and minimum sizes of faces that appear within the frame. The method of Viola and Jones [25] is used for face detection.

3.4 Experiment

We performed experiments using the proposed method. We used only video data provided from the Netherlands Institute for Sound and Vision for training data. The video contents were mainly documentaries and children's educational programs, and included both color and monochrome video. The frame was 352 pixels wide by 288 pixels high, with a frame rate of 25 frames per second, and the video was compressed into a MPEG-1 format. We did not use any other images, such as those acquired over the Internet.

We will now describe the settings for each run. The runs differed in the following two points:

- Method of creating vocabulary items
- Method of mixing local features and global features

We used two methods to create the vocabulary: a method for creating vocabulary for each high-level feature and a method for creating common vocabulary from the entire training data. We also used two methods of mixing local and global features: one in which SVM results are combined linearly and one in which those results are classified once again by SVM. The settings for each run are shown in Table 2. In addition, we also used a method that uses only global features and a simple method based on the average degree of similarity with positive examples, for comparison. We used grid-based differences of a color histogram to calculate the degree of similarity between frames.

Table 2: Settings of each run

Run	Vocabulary type	Mixing method	Memo
1	Common	SVM	
2	Common	Linear sum	
3	Each HLF	SVM	
4	Each HLF	Linear sum	
5	-	-	Global feature only
6	-	-	Similaritry with positive examples

Table3: Evaluation results

Run	Inf AP
1	0.175%
2	0.135%
3	0.100%
4	0.065%
5	0.245%
6	0.285%

3.4.1 Experimental results

The evaluation results of each run are shown in Table 3. The results of these experiments showed that it is better to create a common vocabulary from all the training data than create vocabularies for each high-level feature. The results also showed that the identification by SVM was higher with the method mixing local and global features than with that using linear combinations. On the other hand, Run 5, in which only global features were used, exhibited higher results than runs in which local features were also used. Much previous research has demonstrated that local features usually can capture the characteristics of high-level features robustly, without any effects due to differences in size and position of objects, and also increase the accuracy over runs in which only global features are used. The results of these experiments are completely different, raising fears of mistakes in implementation or errors in the presentation format, so we consider that further investigation is necessary. Run 6, which was based on degrees of similarity with positive examples, exhibited even better results than Run 5. The accuracy was particularly high in Run 6 for “004 Traffic intersection”, “007 Person-playing-a-musical-instrument”, and “014 Demonstration Or Protest”.

4. Conclusion

This paper proposed a method of surveillance event detection and a high-level feature extraction.

For surveillance event detection task, we devised a method of automatically detecting specific events (“PersonRuns,” “Opposing Flow,” “PeopleMeet”, and “ObjectPut”) within video from fixed cameras installed in an airport to detect surveillance events. We obtained a person’s trajectory by detecting and tracking a region with humans in the camera image. We then determined whether the trajectory represented specific events on the basis of feature values taken from the trajectory. We plan to expand the range of events that can be detected in the future.

For high-level feature extraction task, we developed a method for extracting high-level features by using a bag-of-keypoints and global features. The proposed method tried to classify features more accurately by combining two algorithms to extract local features. It also used global features, such as color or texture and the results of face detection, to capture features of the entire image. Furthermore, it used a ball vector machine, which is a high speed support vector machine algorithm, in order to reduce computational cost for training processing. The results of experiments show that a method that creates vocabulary for the entire training data then classifies local and global features by SVM exhibits better results that other combinations.

References

[1] J. C. Niebles and Li Fei-Fei. “A hierarchical model of shape and appearance for human action classification,” In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8, Jun. 2007.

[2] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. “Recognizing action at a distance,” In Proc. of IEEE int. Conf. on Computer Vision, pp. 726–733, vol. 2, Oct. 2003.

[3] K. Mikolajczyk and H Uemura. “Action Recognition with Motion-Apperance Vocabulary Forest,” In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8, Jun. 2008.

[4] N.Dalal and B.Triggs, ”Histograms of Oriented Gradients for Human Detection,” IEEE Conputer Vision and Pattern Recognition, pp.886-893, 2005.

[5] F. Han, Y. Shan, R. Cekander: “A Two-Stage Approach to People and Vehicle Detection with HOG-Based SVM,” PerMIS, pp.133-140, 2006.

[6] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” Neural Computation, 13, pp.1443-1471, 2001

[7] P. H. Chen, C. J. Lin, and B. Schölkopf, “A tutorial on ν-support vector machines,” Applied Stochastic Models in Business and Industry, Vol.21, pp.111-136, 2005

[8] F. Han, Y. Shan, R. Cekander: “A Two-Stage Approach to People and Vehicle Detection with HOG-Based SVM,” PerMIS, pp.133-140, 2006.

[9] G. Xuan, P. Chai, M. Wu, “Bhattacharyya Distance Feature Selection,” In Proc. of the International Conference on Pattern Recognition, Vol. 2, pp. 195-199, 1996

[10] D.G. Lowe, “Object recognition from local scale-invariant features,” In Proc. ICCV’99. vol.2. pp.1150–1157, 1999.

[11] J. Sivic and A. Zisserman, “Video google: a text retrieval approach to object matching in videos,” In Proc. ICCV’03, 2003.

[12] G. Csurka, C. Bray, C. Dance and L. Fan, “Visual categorization with bags of keypoints,” in Proc. ECCV Workshop on Statistical Learning in Computer Vision, pp.59–74, 2004.

[13] D. Blei, A. Ng and M. Jordan, “Latent dirichlet allocation,” Journal of Machine Learning Research, vol.3, pp.993–1022, 2003.

[14] L. Fei-Fei and P. Perona, “A Bayesian hierarchical model for learning natural scene categories,” In Proc. IEEE Computer Vision and Pattern Recognition, pp.524–531, 2005.

[15] S. Lazebnik, C. Schmid and J. Ponce, “Beyond bags features: spatial pyramid matching for recognizing natural scene categories,” In Proc. IEEE CVPR’06, pp.2169–2178, 2006.

[16] R. Fergus, P. Perona and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” In Proc. IEEE Computer Vision and Pattern Recognition, pp.264–271, 2003.

[17] “TREC Video Retrieval Evaluation Notebook Papers and Slides,” http://www-nlpir.nist.gov/projects/tvpubs/ tv.pubs.org.html

[18] S.-F. Chang, J. He, Y.-G. Jiang, E.E. Khoury, C.-W. Ngo, A. Yanagawa and E. Zavesky, “Columbia University/VIREO-City/IRIT TRECVID2008 High-level feature extraction and interactive video search,” In Proc. TRECVID 2008 Workshop, 2008.

[19] D-D. Le, S. Satoh and T. Matsui, “NII-ISM, Japan at TRECVID 2007: high level feature extraction,” In Proc. TRECVID Workshop, 2007.

[20] A. Yanagawa, S.F. Chang, L. Kennedy and W. Hsu, “Columbia University’s baseline detector for 374 LSCOM semantic visual concepts,” Technical Report, Columbia University, 2007

[21] I.W. Tsang, A. Kocsor and J.T. Kwok, “Simpler core vector machines with enclosing balls,” In Proc. ICML’07, 2007.

[22] H. Bay, A. Ess, T. Tuytelaars and L.V. Gool, “SURF: Speeded Up Robust Features,” Computer Vision and Image Understanding, vol.110, no.3, pp.346–359, 2008.

[23] Y.-G. Jiang, C.-W. Hgo and J. Yang, “Towords optimal bag-of-features for object categorization and semantic video retrieval,” In Proc. ACM CIVR’07, 2007.

[24] T. Ojala M. Pietikaninen and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol.24, no.7, pp.971–987, 2002.

[25] P. Viola and M. Jones, “Robust Real-time Object Detection,” International Journal of Computer Vision, 2001.