AUTOMATIC ANNOTATION OF WEB VIDEOS

Shih-Wei Sun[a], Yu-Chiang Frank Wang[a,b], Yao-Ling Hung[a], Chia-Ling Chang[b]
Kuan-Chieh Chen[b], Shih-Sian Cheng[a], Hsin-Min Wang[a,b], and Hong-Yuan Mark Liao[a,b]

[a]Institute of Information Science and [b]Research Center for Information Technology Innovation
Academia Sinica, Taipei, Taiwan

ABSTRACT

Most Web videos are captured in uncontrolled environments (e.g. videos captured by freely-moving cameras with low resolution); this makes automatic video annotation very difficult. To address this problem, we present a robust moving foreground object detection method followed by the integration of features collected from heterogeneous domains. We advance SIFT feature matching and present a probabilistic framework to construct consensus foreground object templates (CFOT). The CFOT can detect moving foreground objects of interest across video frames, and this allows us to extract visual features from foreground regions of interest. Together with the use of audio features, we are able to improve resulting annotation accuracy. We conduct experiments and achieve promising results on a Web video dataset collected from YouTube.

Index Terms— Video annotation, object detection

1. INTRODUCTION

Web video annotation receives increasing attention due to a large amount of Web-based applications such as online video sharing and search. Typically, these applications rely on the associated tag information, but noisy or incorrect tags annotated by the users will degrade the performance of annotation, retrieval, or higher-level tasks such as activity and behavior analysis. Therefore, the development of an automated annotation technique for Web video data becomes necessary, and its success would benefit the above applications.

Another challenge for applications of Web videos is that not all the multimedia data on the Web can be used as a satisfactory training data resource. For example, some Web videos are captured by cell phone cameras with significant camera motion, cluttered background, and noisy sound present. Many of the existing Web videos are still with low image resolution, low bit rate, or with blocking effects. Another problem is that, the recorded audio track of a video sequence might not be relevant to the subject of interest due to background noise captured by the microphone, e.g. crowd chatting in the background. In other words, most Web-based video data are captured under uncontrolled conditions, and this would prohibit the direct use of this type of data for real-world applications.

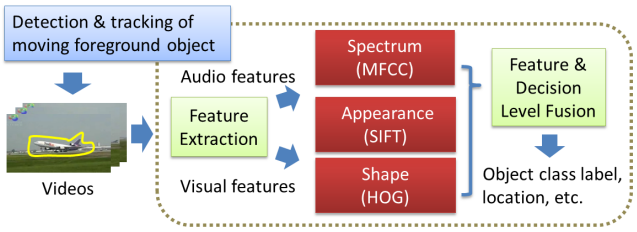

Fig. 1. The system diagram of our approach.

In this paper, we propose an automated video annotation method which is able to detect and label the foreground region of interest using appearance and motion information. More specifically, we focus on annotating rigid moving objects such as cars, airplanes, etc. in Web videos, and we consider videos with only one foreground object present. We propose to construct consensus foreground object templates (CFOT) to address moving object detection. Our method is robust to significant camera motions (e.g. panning, tilting, zooming, etc.) or low contrast environments. We also consider the integration of features collected in different domains, which further improves the annotation accuracy. Figure 1 depicts the our proposed framework, and its output will be the video annotation results with both object location and class label information.

2. RELATED WORK

Detecting moving foreground objects from videos taken by non-stationary cameras (or cameras with low resolution) has been a challenging task. In [1], Sheikh and Shah proposed to build foreground and background models by using a joint representation of pixel color and spatial structures. Patwardhan et al. decomposed a scene into layers and used maximum likelihood techniques to assign pixels into different layers for foreground estimation [2]. Although attractive results were reported, only videos which are captured by a camera with nominal to mild motion can be handled.

In order to detect foreground objects in uncontrolled videos, a typical technique is to first estimate the global motion of the camera, and the motion induced by foreground objects is thus considered as outlier. Meng and Chang [3] utilized the motion vector field to generate the global motion of

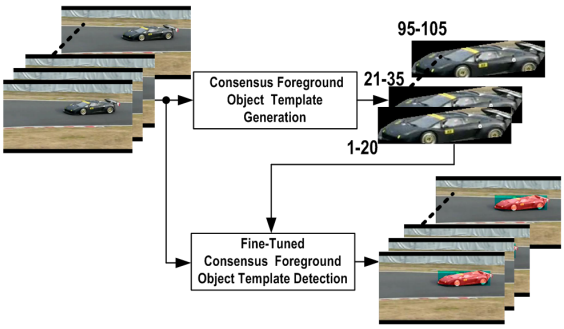

Fig. 2. Our framework for foreground object detection.

each image frame. Irani and Anandan [4] proposed to detect moving objects using 2D and 3D parameters. Wang et al. [5] used motion vectors in MPEG videos to estimate affine parameters for zooming and translation. Nevertheless, estimation of camera motion is still a challenging task due to background clutter present in a scene, or the appearance and scale variations of the foreground objects, etc. Another category of object detection algorithms is to model a reference background image. Felip et al. [6] estimated the dominant motion from the sampled motion vectors, and the alignment based on inter-image homography can be achieved by the dominant motion. Zhao et al. [7] proposed to detect objects from videos captured by a non-static camera in an indoor scene with an incrementally learned scene model. Their method is based on a matching scheme using SIFT features and homography calculation. However, this setting may not be practical for videos captured in outdoor scenes or with complex backgrounds.

Due to limited quality of Web video data, recent work on Web video classification typically considers the use of multiple types of features. Most prior methods utilized various static features such as appearance, color, etc. of each frame, while some also considered space-time features to extract motion information [8, 9]. Methods which combine features from different domains (e.g. text in [8, 10] and audio in [10]) also exist. However, as pointed out in [11], most existing approaches on video classification cannot be easily generalized to applications of web video clips. Due to low quality and diversity of Web videos, the direct use of web-based data could dramatically degrade the performance of the designed algorithm. Moreover, they cannot be easily extended to address video annotation problems. As a result, it usually requires some preprocessing techniques to obtain refined data/features to improve the performance of Web video classification/annotation. For example, a hierarchical taxonomy structure was proposed in [10] to alleviate noisy data, and in [8], the authors pruned the motion features using spatial and temporal statistics. In our work, in order to reduce the effect of camera motion and cluttered background information, we present a unique way to extract the region of interest for the foreground object (Section 2), followed by the integration of features from multiple domains (Section 3).

3. FOREGROUND OBJECT DETECTION

Our method for foreground object detection consists of two major steps: construction of the consensus foreground object template (CFOT) and its use for object detection. Figure 2 illustrates the framework, and Figure 3 shows a detailed flow chart of the construction of CFOT.

3.1. Consensus Foreground Object Template (CFOT)

3.1.1. Foreground region estimation

Scale-invariant feature transform (SIFT) [12] is a popular computer vision algorithm, which is used to detect local interest points in an image, and to describe the associated appearance information. As an initial stage of our foreground feature point extraction, we apply the SIFT feature detector in each frame of a video sequence.

For each pair of corresponding SIFT feature points in adjacent frames, a motion vector can be calculated. Assuming that the motion vectors extracted from moving foreground objects are significantly different from those from background clutter, we apply a vector clustering algorithm to perform initial foreground region extraction. A quantization process applied on the magnitude of the motion vector is used to categorize the motion vectors into a small number of classes. Each motion vector is assigned with a corresponding quantization index, and a histogram of these indices is calculated. For this histogram, the bin with the maximum value is identified as foreground, and the associated SIFT interest points are considered as initial foreground points. More specifically, let $\hat{q}_t$ denote the quantization index of the histogram $hist_t$ at frame t with the maximum value, and thus $\hat{q}_t$ satisfies:

$$\hat{q}_t = \arg \max_{q \in \{1,2,\cdots,Q\}} (hist_t(q)), \quad (1)$$

where Q represents the number of quantization categories (we chose $Q = 6$, and did not observe the results will be sensitive to this choice). Finally, a set of foreground interest points at frame t , denoted by F_t, can be defined as follows:

$$\mathrm{F}_t = \{f_i : g(v_i) = \hat{q}_i \text{ and } i \in \{1, 2, \ldots, N_t\}\}, \quad (2)$$

where v_i is the motion vector of feature point f_i, $g(\cdot)$ is the quantization procedure for v_i, and N_t is the number of motion vectors obtained from two adjacent frames.

Once the foreground interest points are obtained, we define a candidate foreground region R_t according to the spatial distribution of F_t with a Gaussian distribution assumption. More specifically, let $(\overline{x}, \overline{y})$ denote the centroid of the foreground SIFT points, and σ_x and σ_y represent the corresponding standard deviation, we thus use the upper left corner $(\overline{x} - 2\sigma_x, \overline{y} - 2\sigma_y)$ and the bottom-right corner $(\overline{x} + 2\sigma_x, \overline{y} + 2\sigma_y)$ to set the boundary of R_t. We note that the recently proposed SIFT flow [13] also advocates SIFT matching for determining

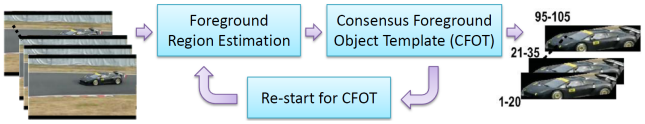

Fig. 3. Construction of consensus foreground object template.

corresponding feature points. However, they focus on applications of image alignment and registration, not moving foreground object detection in videos (as we do in this paper).

3.1.2. *Consensus Foreground Object Template*

With the candidate foreground region R_t, we further define the foreground object probability, which indicates how likely a pixel at location (x, y) within R_t belongs to foreground. This probability is calculated as follows:

$$P_t(x,y) = \begin{cases} P_{t-1}(x - \overline{\triangle x}_t, y - \overline{\triangle y}_t) \cdot \lambda + (1-\lambda), & \text{if } (x,y) \in \mathrm{R}_t; \\ P_{t-1}(x,y) \cdot \lambda, & \text{otherwise}, \end{cases} \tag{3}$$

where $\lambda = 0.95$ is an update factor, as suggested in [14]. The notations $\overline{\triangle x}_t$ and $\overline{\triangle y}_t$ are x and y components of the average foreground motion vector at frame t, respectively. The final R_t is thus calibrated according to $\overline{\triangle x}_t$ and $\overline{\triangle y}_t$, and we use the calibrated R_t to construct an object image template up to the T-th frame, in which each pixel value is calculated as:

$$\overline{I}_T(x,y) = \left\{ \sum_{t=1}^{T} I_t(x,y) \right\} / c_{map,T}(x,y), \text{ if } (x,y) \in \mathrm{R}_t. \tag{4}$$

$I_t(x, y)$ returns the pixel value of (x, y) at frame I_t. To normalize this template, a counter map $c_{map,T}(x, y)$ in the denominator records the number of frames (out of T) that a pixel belongs to the foreground region. As a result, each foreground object pixel contributes $\frac{1}{c_{map,T}(x,y)}$ of its value to the final average foreground image. Thus, the final CFOT is produced by using the averaged foreground pixel model $\overline{I}_T(x, y)$, followed by an adaptive thresholding of the associated foreground object probability $P_t(x, y)$.

3.1.3. *Re-start mechanism for CFOT updates*

For Web videos, it is not surprising that the appearance, scale, illumination, etc. of a moving object can vary significantly throughout the video. Under these severe variations, the aforementioned foreground probability model will not be sufficient to provide effective information when constructing the CFOT (as shown in Figure 4). In other words, use of a single CFOT will not be able to produce satisfying object detection results, since such a CFOT will contain noisy information.

In order to alleviate these problems, a re-calculation for CFOTs becomes necessary. We use the local maximum values observed in σ_x, σ_y, and sum of absolute difference (SAD) to determine whether we should re-start the CFOT generation process, including the reset of all probability models in previous stages. At a frame t, we compute the following indicating vector $r(t) = [r^{[1]}(t), r^{[2]}(t), r^{[3]}(t)]^T = [\sigma_x(t), \sigma_y(t), SAD(t)]^T$. If we observe a significant change in σ_x, σ_y, or SAD, we will recalculate the CFOT. Thus, the time instant m for this re-start mechanism is determined by the local maximums of σ_x, σ_y, or SAD (using MATLAB function `FINDPEAKS`), i.e.

$$m^{[j]} = arg \underset{t}{\text{local-max}}\{r^{[j]}(t)\}, \; j = 1, 2, 3. \tag{5}$$

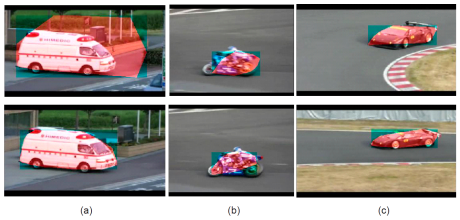

Fig. 4. Examples when the recalculation of CFOT is required. (a) Too much background, (b) blurring effects, and (c) significant scale and appearance variations observed in CFOT.

3.2. Foreground Object Detection Using CFOTs

To detect moving foreground objects, we use the CFOT as a query image over a number of video frames, and this CFOT will look for similar image patterns in each frame within this period of time (before an updated CFOT needs to be calculated). A similarity test based on SAD is performed to exhaustively search for a region in each frame which best matches the CFOT. Once this foreground region is determined, as the completion of the foreground detection process, we use a rectangle to mark the foreground region, as shown in Figure 2.

4. AUDIO AND VISUAL CLASSIFICATION

4.1. Audio Classification

To extract audio features for video annotation and for feature and classifier-level fusion, we convert audio signals of each video sequence into a stream of 19-dimensional Mel-frequency cepstrum coefficients (MFCCs) using a 32-ms Hamming-windowed frame with 10-ms shifts. Since Gaussian mixture model (GMM) has been widely used in audio classification [15, 16], we train a GMM for each class, and the output of a test video is determined by the maximum log-likelihood of the GMMs. To determine the number of components for each GMM, we apply the SGML algorithm [17] based on Bayesian information criterion.

4.2. Visual Classification

4.2.1. *Visual features considered*

In our work, we advance dense SIFT [12] and histogram of oriented gradients (HOG) descriptors [18] to capture appearance and shape information from video frames, respectively.

We do not use the salient SIFT descriptors (with local interest points detected at different image scales), since they cannot sufficiently describe the objects which are relatively small or with low contrast in an image. We also found that such a small percentage of local descriptors cannot provide adequate descriptive information for multi-class object categorization problems. Therefore, we choose to extract dense SIFT descriptors. Although the color rgSIFT descriptors which add appearance features from R and G color channels have been shown to work well in several applications, we did not find them useful in recognizing artificial objects with large variations of color (as the objects in our dataset do). Therefore, we do not use color salient descriptors. As for the HOG descriptors, we consider a dense grid of uniformly spaced cells and extract gradient histograms. We did not consider space-time features, since the foreground objects in our dataset are all moving objects, and thus motion information does not provide any additional discriminating ability.

4.2.2. Learning sparse feature representation

Sparse coding (SC) has been shown to be an effective technique in many vision tasks [19]. To produce sparse representation for both SIFT and HOG descriptors, we use the software package developed by Mairal *et al.* [20] to learn the dictionaries (one for each feature), and to encode the associated feature descriptor. The parameter λ, which controls the sparsity of the encoded coefficient vector, is set to 0.2 in our experiments. The average and the maximum number of non-zero elements in the encoded coefficient vectors are 4.87 and 18 for SIFT, and 5.43 and 16 for HOG (both out of $K = 225$, which is the size of the dictionary). After obtaining the encoded sparse coefficient vectors for both features, we use the max pooling technique to covert the SIFT (HOG) descriptors from each frame into a K-dimensional feature vector.

5. EXPERIMENTAL RESULTS

5.1. Web Video Dataset

We collect a complex, uncontrolled, and challenging Web video dataset from YouTube for our experiments. The video data are all captured by moving or shaky cameras, and the moving object of interest are present in cluttered background. Significant scale and viewpoint variations of the objects can be observed, and the resolution of a large portion of videos in our dataset is low. We consider six different moving object categories: *Airplane*, *Ambulance*, *Car*, *Fire Engine*, *Helicopter*, and *Motorbike*. Each object category has 25 to 30 video sequences, and each sequence has one moving foreground object present in it. We randomly select 10 from each class for training, and the remaining for testing. Figure 5 shows examples video frames of each object category in our dataset. We note that, for audio classification, in order to achieve comparable audio classification results as prior

Fig. 5. Example videos in our dataset.

Table 1. Results of audio classification. MAP = 59.40%

	Airplane	Ambulance	Car	FireEngine	Helicopter	Motorbike
Airplane	50.00%	0%	22.22%	5.56%	16.67%	5.56%
Ambulance	0%	80.00%	10.00%	10.00%	0%	0%
Car	18.75%	0%	43.75%	6.25%	18.75%	12.50%
FireEngine	0%	5.56%	11.11%	83.33%	0%	0%
Helicopter	10.53%	0%	21.05%	0%	52.63%	15.79%
Motorbike	0%	0%	26.67%	6.67%	20.00%	46.67%

work using MFCC features, we further consider the use of auxiliary training audio data, which are also collected from YouTube. This additional video training data for audio classification does not necessarily have objects of interest visible, and we do not use this set of data for visual classification either. Nevertheless, none of the above training data is present in our test set, and we only use extracted audio and visual features to train the associated classifiers and to perform feature/classifier-level fusion.

For our visual features, SIFT descriptors are extracted from 16×16 pixel patches of an image, and the spacing between adjacent patches is 6 pixels (horizontally and vertically). HOG descriptors are extracted from each 8×8 pixel grid, and only one scale in an octave of the pyramid is used. We note that we resize the longer side of the image to 300 pixels if its width or height exceeds 300 pixels for both descriptors; this is to preserve the aspect ratio of each image.

5.2. Results of audio and visual classification

Table 1 shows the confusion matrix and the mean average precision (MAP)for audio classification. The extraction of MFCC audio features and the use of GMM classifiers were detailed in Section 4.1. As shown in Table 1, we achieved MAP $= 59.4\%$, while objects *Ambulance* and *Fire Engine* were with better classification results ($> 80\%$).

To classify video data using either SIFT or HOG features, we sub-sample 20 frames from each of the test video sequence

Table 2. Visual classification with or without CFOT.

	SIFT	HOG
without CFOT	39.68%	55.57%
with CFOT	**63.68**%	**63.45**%

Table 3. Results of visual classification.

SIFT — MAP = **63.68**%

	Airplane	Ambulance	Car	FireEngine	Helicopter	Motorbike
Airplane	50.00%	0%	11.11%	0%	38.89%	0%
Ambulance	0%	75.00%	0%	25.00%	0.00%	0%
Car	12.50%	0.00%	50.00%	12.50%	0%	25.00%
FireEngine	0%	0%	0%	94.44%	5.56%	0%
Helicopter	10.53%	5.26%	10.53%	5.26%	52.63%	15.79%
Motorbike	0%	0.00%	20.00%	6.67%	13.33%	60.00%

HOG — MAP = **63.45**%

	Airplane	Ambulance	Car	FireEngine	Helicopter	Motorbike
Airplane	50.00%	5.56%	11.11%	0%	27.78%	5.56%
Ambulance	0%	90.00%	5.00%	5.00%	0%	0%
Car	12.5%	6.25%	37.50%	12.50%	0%	31.25%
FireEngine	0%	0%	0%	94.44%	5.56%	0%
Helicopter	26.32%	5.26%	10.53%	0%	42.11%	15.79%
Motorbike	0%	0%	20.00%	13.33%	0%	66.67%

and extract the associated features from them. Since we need to first verify whether the use of our proposed CFOT not only provides the foreground candidate region, but also refines visual features by suppressing background clutter for improved performance, we present the averaged performance with or without using CFOTs in Table 2. For the results with CFOT, we multiply the CFOT masks on the corresponding training and test video data, and only the features extracted from the training data (within the CFOTs) were used to design the classifiers. In this paper, we consider one-vs-all linear SVM classifiers, and the parameter C in all SVMs in our work is selected via a 5-fold cross validation.

To classify a test video input, we first predict the label of each of the 20 sub-sampled frames, and we use a majority vote to determine the final object label for this input video. From Table 2, it is obvious that both type of visual features produced improved recognition performance using CFOTs. Thus, we confirm the effectiveness of our CFOTs. Table 3 presents the confusion matrices and MAPs for visual classification using SIFT and HOG features, both with CFOTs. Although the MAP results of both cases were slightly below that of audio classification, the difference is marginal.

5.3. Feature and Decision-Level Fusion

To combine different audio and visual features for improved performance, we first consider *feature-level fusion*. For simplicity, we concatenate audio and visual features and obtain a new feature representation for training and testing. In order to produce the same number (20) of each type of features, we uniformly divide a video clip into 20 segments and average the MFCCs in each to result in 20 audio features. To normalize the features before concatenation, we zero-mean each feature with a unit variance. After the features are concatenated, we train one-vs-all SVM classifiers for each object category

Table 4. Feature and decision-level fusion results. The best performance for each feature combination is shown in bold.

	Audio +SIFT+HOG	SIFT + HOG	Audio + SIFT	Audio + HOG
Feature fusion	64.31%	64.26%	65.35%	67.39%
Sum-rule	67.41%	**65.46**%	**70.78**%	66.39%
Max-rule	66.18%	65.26%	67.01%	64.36%
Weighted-sum	66.37%	63.55%	69.21%	63.27%
Late SVM fusion	**67.66**%	64.56%	70.09%	**70.09**%

with CFOT applied at each frame. To annotate a test video, we first determine and apply the associated CFOT on it, and we extract the features considered for classification. The results using feature-level fusion are shown in the first row of Table 4, in which we consider different feature combinations for fusion, i.e. Audio+SIFT+HOG, SIFT+HOG, Audio-SIFT, and Audio+HOG. We see that all feature combinations improved recognition performance. To remark on the improvements between audio classification (MAP = 59.4%) with other audio-visual fusion results (e.g. Audio+HOG at 67.39%), recall that there are much more than 20 MFCC features extracted for each video clip when designing GMM classifiers for audio classification; however, only 20 averaged MFCCs are used for feature-level fusion. Therefore, the improvements reported in Table 4 are quite remarkable.

Next, we consider *decision-level fusion* to combine the results from classifiers using audio and visual features using *sum*, *max*, and *weighted-sum rules*. Note that sum rule treats each classifier equally important, and it averages the classifier outputs (typically in terms of posterior probability) and assigns the test input to the class with the highest final probability score. On the other hand, max rule simply utilizes the largest posterior probability of each classifier for fusion. Besides, we also perform a weighted sum-rule, in which the weight for each classifier and each object class is determined by the confusion matrix of the training data (via cross-validation). From Table 4, we observe performance improvements with all feature combinations (e.g. Audio+SIFT at 70.78% with sum rule).

Another decision-level fusion strategy we considered is *late SVM fusion*. For each feature, we first calculate the posterior probability of an input video using each one-vs-all SVM classifier, then we concatenate the six probability scores (one for each object class) as a 6-dimensional vector. If all three types of features are used, the dimension of the final feature representation will be $6 \times 3 = 18$. The results using late SVM fusion is shown in the last row in Table 4. Among all fusion strategies and feature combinations, the use of Audio-SIFT for decision-level fusion with sum-rule resulted in the highest recognition rate (70.78%), and Audio+HOG for late SVM fusion also achieved a comparable MAP at 70.09%; both are about 7-13% improvements compared to those using a single type of audio or visual feature.

It is worth noting that, the results using all features (i.e.

Fig. 6. Example annotation results. The convex hulls shown in different colors are determined by our CFOT, and each color corresponds to the associated object class label.

Audio+SIFT+HOG) were not the best in Table 4, while the fusion of two visual features generally produced the smallest improvements. This is expected, since rather than adding features from the same domain or increasing the number of features for fusion, *integration of heterogeneous features from multiple domains is expected to provide complementary information for improved performance*, as supported by our empirical results (e.g. Audio+HOG or Audio+SIFT).

Finally, we show some of our video annotation examples in Figure 6. Figure 6(a) illustrates an excellent annotation result, which predicted the correct class label with a perfect CFOT determined. We note that the helicopter in Figure 6(b) cannot be successfully recognized by either visual feature; this is probably because we do not have a large number of front-view helicopter videos in our dataset). However, with both feature and decision-level strategies, a correct annotation result was obtained. Figure 6(c) is a challenging video, since both the foreground object (aircraft) and background clutter (e.g. sky, cloud, etc.) are present, and the contrast between them is nominal. Similarly, its class label was successfully predicted using the combination of both visual and audio features, while the use of either feature did not produce a correct output. These examples again verify the effectiveness of our approach, which utilizes the significance of integrating heterogeneous features for improved video annotation.

6. CONCLUSION

We proposed a robust video annotation method which automatically determines the region of the foreground object and predicts its class label. The former was done using our consensus foreground object template (CFOT) for moving object detection, and the later was achieved by the integration of heterogeneous features from different domains. In this work, we especially focused on the challenging task of Web video annotation, in which most existing Web videos are captured under uncontrolled environments, with insufficient quality or limited tag information available. Unlike prior sliding window or object detector based methods, we do not require pixel-level ground truth data for training; instead, only the label of each video is utilized, which is especially practical for Web video applications. In our experiments, we collected a Web video dataset with only label information as ground truth. We verified that our CFOT is able to identify the foreground region of interest, while our proposed framework provides tag information (class label) using feature and decision-level fusion techniques. We observed a significant improvement in recognition (annotation) accuracy using our method. In future work, we expect to extend our CFOT framework for further high-level vision tasks such as activity or event recognition using Web videos.

Acknowledgements This work is supported in part by National Science Council of Taiwan via NSC 99-2221-E-001-020 and NSC 100-2631-H-001-013.

7. REFERENCES

[1] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," in *IEEE PAMI*, 2005.

[2] K. A. Patwardhan et al., "Robust foreground detection in video using pixel layers," in *IEEE PAMI*, 2008.

[3] J. Meng and S. Chang, "CVEPS - a compressed video editing and parsing system," in *ACM Multimedia*, 1996.

[4] M. Irani and P. Anandan, "A unified approach to moving object detection in 2D and 3D scenes," in *IEEE PAMI*, 1998.

[5] R. Wang, H.-J. Zhang, and Y. Zhang, "A confidence measure based moving object extraction system built for compressed domain," in *IEEE ISCAS*, 2000.

[6] R. L. Felip, L. Barcelo, X. Binefa, and J. R. Kender, "Robust dominant motion estimation using mpeg information in sport sequences," in *IEEE CSVT*, 2008.

[7] Y. Zhao, M. Casares, and S. Velipasalar, "Continuous background update and object detection with non-static cameras," in *IEEE AVSS*, 2008.

[8] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *CVPR*, 2009.

[9] L. Duan, D. Xu, I. W. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," in *CVPR*, 2010.

[10] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li, "Youtubecat: Learning to categorize wild web videos," in *CVPR*, 2010.

[11] S. Zanetti, L. Zelnik-Manor, and P. Perona, "A walk through the web's video clips," in *CVPR Workshop*, 2008.

[12] D. Lowe, "Object recognition from local scale-invariant features," in *PETS*, 1999.

[13] C. Liu et al., "SIFT flow: dense correspondence across different scenes and its applications," in *IEEE PAMI* , 2011.

[14] A. Senior, "Tracking People with Probabilistic Appearance Models," in *PETS*, 2002.

[15] J. Shirazi et al., "Improvements in audio classification based on sinusoidal modeling," in *IEEE ICME*, 2008.

[16] P. Dhanalakshmi et al., "Classification of audio signals using AANN and GMM," in *Applied Soft Computing*, 2010.

[17] S.-S. Cheng, H.-M. Wang, and H.-C. Fu, "A model-selection-based self-splitting Gaussian mixture learning with application to speaker identification," in *EURASIP JASP*, 2004.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[19] M. Elad et al., "Image denoising via sparse and redundant representations over learned dictionaries," in *IEEE Trans. Image Processing*, 2006.

[20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *ICML*, 2009.