Video Object Detection by Classification Using String Kernels

Wan-Hsuan Yu, Chi-Han Chuang
Department of Computer Science and Engineering
National Taiwan Ocean University
Keelung, Taiwan
e-mail: yuwanhsuan@gmail.com

Shyi-Chyi Cheng
Department of Computer Science and Engineering
National Taiwan Ocean University
Keelung, Taiwan
e-mail: csc@mail.ntou.edu.tw

***Abstract*—Video object detection is one of the most important research problems for video event detection, indexing, and retrieval. For a variety of applications such as video surveillance and event annotation, the spatial-temporal boundaries between video objects are required for annotating visual content with high-level semantics. In this paper, we define spatial-temporal sampling as a unified process of extracting video objects and computing their spatial-temporal boundaries using a learnt video object model. We first provide a learning approach to build a class-specific video object model from a set of training video clips. Then the learnt model is used to locate the video objects with precise spatial-temporal boundaries from a test video clip using graph kernels. A frame sorting process as a preprocessing is also proposed to transform the graph, modeling the shot configuration of a video clip, into a string of shots. Thus, the computation of graph kernels is simplified to be string kernels. The string kernels for support vector machine (SVM) classification are finally adopted to train the SVM classifiers from a set of training samples and detect the video objects in a test video clip by classification. A human action detection and recognition system is finally constructed to verify the performance of the proposed method. Experimental results show that the proposed method gives good performance on several publicly available datasets in terms of detection accuracy and recognition rate.**

Keywords-video objects; string kernels; dynamic programming; video object modeling; SVM classification.

I. INTRODUCTION

Video object detection (VOD) is the primary step to semantically annotate a video sequence in semantic video database indexing and retrieval, intelligent video surveillance, and advanced man-machine interfaces [1,2]. Early works in video object detection focused on detecting and recognizing the scene and objects shown in a representative key-frame of a video shot, thus the temporal information of video objects is lost [3,4]. Recently, semantic-based video analysis tended to model a video clip as a graph whose nodes are high-level video objects performing a specific action individually [5]. Techniques of graph matching are then applied to annotate the event type of the input video clip [6]. Detection and classification of video objects from video clips help bridge the semantic gap between high-level features and low-level features and the construction of modern semantic-based video analysis.

Conventional VOD algorithms, which characterize objects as spatially cohesive with locally smooth trajectories, use these techniques for tracking or body pose estimation to extract spatial-temporal tubes from the input video clip [7-9]. However, using a tracking or body pose estimation in real world videos is generally not reliable due to object occlusion, distortion and changes in lighting. Instead, we formulate the tracking process for VOD [7, 10] as a classification problem because objects are, in general, spatially and temporally cohesive. Also, by assuming relatively slow camera motions, the shape and location of these objects vary slowly from frame to frame. Thus, the size of the search space to track an object across many frames is reduced significantly by exploiting this coherence. By considering a parameter set in the feasible search space as a class, the object tracking for VOD casts into a classification framework [11].

A primary motivation for the work presented here is to question the benefits of tracking object boundaries across frames for video-based applications such as activity analysis. In practice, the accuracy of any boundary estimate is limited by a number of systemic factors such as image resolution, noise, motion skew, and the accuracy of the model. For example, formulating VOD as motion segmentation using optical flow rests on the assumption of brightness constancy, which is violated at moving boundaries, resulting in poor estimates of object contours [12]. For some applications, the object detection at each frame only needs to be known up to a limited precision, as long as good shape and trajectories are maintained.

In addition to segmentation, conventional VODs also try to detect and segment the observed motions into semantic meaningful instances of particular activities from videos [13,14]. To reach this goal, recent approaches consider the detection and recognition of the video object as an extension of 2D object detection [15,16] with higher dimensionality. Some well-known approaches include space-time interest-point detectors [17] and bag-of-words models [18]. These techniques aim at employing a combination of local space-time features and global 3D shape features to estimate the space-time boundaries of a given video object. Two issues which are therefore of particular importance are dealing with local patch sampling and exploring the rich relationships among spatial-temporal "words" inherited from objects [19].

Video object classification is the key step in high-level video-based applications. Conventional machine learning techniques are applied to train the state-of-the-art methods using a large, diverse set of manually annotated images. The typical level of annotation needed is a bounding-box for each object instance [15]. To ensure the performance of a detector, a large amount of annotated instances is generally needed [20]. Recently, object classification approaches borrowed from unlabeled or weakly annotated data have attracted much attention to reduce tedious manual annotation to a minimum [21]. However, training a detector without location annotation is very difficult and performance is still below fully supervised methods [22].

Recent approaches for video object detection follow the following steps: a target object is initialized by human annotation or with a preexisting detector in one frame, then a classifier is trained on-line to redetect the object in each frame [23]. A significant limitation to these approaches is the trained classifier is that a video-specific detector but not a generic class detector. In contrast, Ali et al. [24] proposed a semi-supervised boosting variant that exploits temporal consistency of video frames to learn a complex appearance model from a subset of fully annotated frames in each training video for video object detection. Testing is performed on videos of the same scene, but at different time instances.

An image object often consists of several parts arranged in a deformable configuration [15]. The use of visual patterns of local patches in shape modeling is related to several ideas including the approach of local appearance codebooks [16] and the generalized Hough transform (GHT) [25] for object detection. At training time, these methods learn a model of the spatial occurrence distributions of local patches with respect to object centers. At testing time, based on the trained object class classifiers, the appearances of interest points in images or video are matched in the visual codebooks to detect a specific object using the voting framework of GHT. The effectiveness of visual pattern grouping by Hough voting is thus well verified.

In this paper, we formulate a video object as a graph of postures (key-objects) to model the temporally relationship between key-objects. The graph edit distance (GED) can then be used to measure the spatial-temporal content difference between two video objects. A frame sorting process [26] as a preprocessing is also used to transform the graph, modeling the shot configuration of a video clip, into a string of shots. Thus, the computation of graph kernels is simplified to be string kernels. The string kernels for support vector machine (SVM) classification are finally adopted to train the SVM classifier from a set of training samples and detect the video objects in a test video clip by classification. We also create a template video object for each class to achieve the goal of speeding up the VOD process. A human action detection and recognition system is finally constructed to verify the performance of the proposed method. Experimental results show that the proposed method gives good performance on several publicly available datasets in terms of detection accuracy and recognition rate.

II. Problem Definition

The proposed video object detection by classification using string kernels is inspirited from the work of [4] but of very different implementation. Let $V = \{F_t\}_{t=1}^n$ and $\overline{O} = \{O_t\}_{t=1}^n$ be a video clip of *n* frames and the corresponding video object consisting of *n* 2D target objects, respectively. Suppose $\vec{x} = \{x_t(s_t)\}_{t=1}^n \in R^{D\times n}$ be the feature vectors for every location s_t to locate O_t in F_t, we want to build a classifier

$$\varphi : R^{D\times n} \to R \qquad (1)$$

such that the set of locations $\vec{s} = \{s_t\}_{t=1}^n$

$$\{\vec{s} : \varphi(\vec{x}(\vec{s})) \ge 0\} \qquad (2)$$

detects a visible target video object from video frames. Given a training data set that comprises *N* input vectors $\vec{x}_1$, ..., $\vec{x}_N$, with corresponding target values y_1, ..., y_N where $y_i \in \{-1,1\}, i = 1,...,N$. The support vector machines (SVMs) approach [27] finds the linear decision boundary $\varphi(\vec{x})$ as:

$$\varphi(\vec{x}) = w^T \phi(\vec{x}) + b \qquad (3)$$

where ϕ denotes a fixed feature-space transformation, *b* is a bias parameter, so that, if the training data set is linearly separable, $y_i\varphi(\vec{x}_i) > 0$ for all points. The maximum marginal solution of SVMs is found by solving for the optimal weight vector $\vec{\alpha} = (\alpha_1,...,\alpha_N)$ by maximizing

$$\widetilde{L}(\vec{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left\langle \phi(\vec{x}_i), \phi(\vec{x}_j) \right\rangle \qquad (4)$$

with respect to $\vec{\alpha}$, that is subject to the constraints:

$$\sum_{i=1}^{N} \alpha_i y_i = 0,\ \ \alpha_i \ge 0, \text{ for } i = 1,\dots, N \qquad (5)$$

$\left\langle \phi(\vec{x}_i), \phi(\vec{x}_j) \right\rangle$ is the inner product of $\vec{x}_i$ and $\vec{x}_j$ in the feature space. The parameters *w* and *b* are then derived from the optimal $\vec{\alpha}$.

The computational cost of the inner product could be much reduced by introducing kernel functions to avoid explicitly perform the transformation $\left\langle \phi(\vec{x}_i), \phi(\vec{x}_j) \right\rangle$. If the kernel function *k* satisfies the Merced condition, then there exists a feature space and a mapping function ϕ such that *k* acts as an inner product in the feature space [28]. In this work, we propose to use the string kernel, starting from a Gaussian kernel, which has been proved to be effective for event recognition [1]. The string kernel is defined as

$$k(\vec{x}, \vec{x}') = \exp(-d(\vec{x}, \vec{x}')) \qquad (6)$$

where $d(\vec{x}, \vec{x}')$ is the distance between $\vec{x}$ and $\vec{x}'$ using the dynamic programming process retaining the spatial-temporal consistency of the targets.

A challenge of the problem is it might require a large training set which results in tedious human-labeled effort in training the classifier *φ*. In this work, we tackle this problem by using an initial hand-labeled training video, and by going back-and-forth between the optimization of the labels of non-labeled videos. Many approaches train object detectors from images without location annotation [8]. Although the outputs of these operators are not precise, they can provide the initial training video object for learning the classifier *φ*. In this case, the proposed learning algorithm can be performed automatically without any human-labeled effort.

Another challenge is the performance of the string-kernel approach degrades greatly when the input video clip contains repetitive behaviors. In this case, we first represent a video clip as a set of shots and then lexicographically sort these shots to obtain a compact and normalized string of postures. The complexity of string-kernel computation is thus reduced by representing a video clip as a shot sequence.

III. The Proposed Apporach

Figure 1 shows the block diagram of the system. A preprocessing to lexicographically sort video frames is first applied to temporally normalize the video frames. Then, a key-frame detection procedure is applied to detect key-frames from a normalized video sequence to achieve the goal of eliminating redundant frames. The system is divided into the training and detection phases, where both of them are based on the proposed SVM classification with string kernels.

A. The Training

Many various image analysis tasks have verified the effectiveness of presenting video frames using bag-of-words (BoW) [1]. A common BoW approach to model video class is to extract features from all video patches in all training video clips of a video class to learn the appearance variability of the class, which is modeled as a local appearance codebook consisting of multiple codewords, where each of them is determined by the mean features of a video patch cluster. Based on this codebook, we could compute a histogram of codeword frequencies to represent a video frame by mapping every patch of the frame to a codeword. Thus, each frame is represented as a BoW histogram.

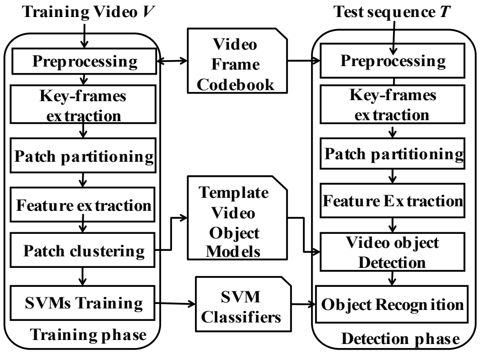
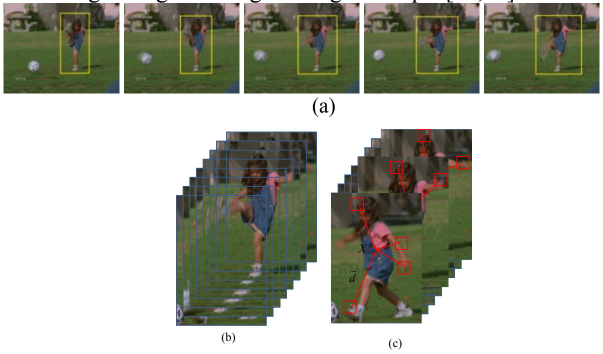


Figure 1. Block diagram of the proposed video object detection by classification using the Hough-voting approach.

In the preprocessing, the first step to temporally normalize the video frames in a video clip is to generate a video shot codebook through vector quantization of large sets of BoW histograms extracted from a collection of training video frames. The video frame codebook is generated by clustering the video frames in the feature space using *k*-means clustering algorithm and Euclidian distance as the clustering metric. The center of each resulting cluster is defined as a frame codeword. Let the video frame codebook *FC* have *m* cluster centers. Our approach uses *FC* to temporally normalize a video clip by grouping similar frames in which the temporal information is preserved. Given a video clip *V* of *n* BoW histograms, $h_i \in V, i = 1,...,n$, there is a collection of cluster assignment: $A = \{c_1, c_2, ..., c_n\}$ where c_i is the cluster label indicating that cluster center *i* is the nearest neighbor of h_i in *FC*. By sorting *A* in lexicographical order, we can obtain $\widetilde{A} = \{\widetilde{c}_1^{\pi(1)}, \widetilde{c}_2^{\pi(2)}, ..., \widetilde{c}_n^{\pi(n)}\}$ where $\widetilde{c}_i^{\pi(i)}$ is the cluster label of the *i*-th video shot in $\widetilde{A}$ and $\pi(i)$ returns the index of frame *i* in *V*. The pair $(\widetilde{c}_i^{\pi(i)}, \widetilde{c}_j^{\pi(j)}), i < j,$ belongs to $\widetilde{A}$ if and only if either $\widetilde{c}_i^{\pi(i)} < \widetilde{c}_j^{\pi(j)}$ or $(\widetilde{c}_i^{\pi(i)} = \widetilde{c}_j^{\pi(j)}) \wedge (\pi(i) < \pi(j))$. We finally permute the frames of *V* using $\widetilde{A}$. The preprocessing step brings the system two obvious advantages: (1) similar frames are clustered to transform the repetitive activities into a single activity implicitly performed by the corresponding video object; (2) all video objects in the same class are starting from a common posture when we represent an activity as a sequence of postures.

A video shot detection procedure is then followed to separate a normalized video clip into multiple video shots, where each of them is represented as a key-frame. Finally, a video clip is represented as a sequence of key-frames. Let $\vec{O}_1 = \{\overline{o}_{1,i}\}_{i=1}^m$ denote the initial video object of *m* key-objects detected from corresponding key-frames of a training video clip in a class. For each key-object $\overline{o}_{1,i} \in \vec{O}_1$, we partition it into a set S_i of patches $P_j = (f, \vec{d}, s_P)$ where *f* is the feature vector characterized by a histogram of orientations (HOG) [29]; $\vec{d}$ is the displacement vector defined from the patch center to key-object center; s_P is the size of the patch. As shown in Figure 2, the patch set S_i forms a GHT model and implicitly describes the structure of $\overline{o}_{1,i}$ which can be used to detect similar objects from another image using the Hough-voting technique [15,16].


(a)

(b)

(c)

Figure 2. Representing a video clip by a sequence of key-frames: (a) detecting the key-frames and key-objects from a video clip; (b) piling up the normalized key-objects to form a 3D video object; (c) modeling (b) using a sequence of GHT models.

To achieve the goal of detecting the target object from an image *I* using the patch set *S* of key-object $\overline{O}$, we look after similar patches $P' \in S$ for each patch $P \in I$ located at (x_P, y_P) using the following distance function:

$$d(P, P') = 1 - \sum_i \sqrt{h_P(i) h_{P'}(i)} \tag{7}$$

where $h_P(i)$ and $h_{P'}(i)$ are the factions of the *i*-th bin of the HOGs of *P* and P', respectively. The local distance measurement for (P,P') should be added to the entry of the Hough-voting volume $H_I(x, y, s)$ at the image *I*:

$$H_I(x, y, s)^{(new)} = H_I(x, y, s)^{(old)} + (1 - d(P, P')) \tag{8}$$

where $s = s_P / s_{P'}$ is the ratio of sizes of *P* to P' and $(x, y) = (x_P, y_P) - s \times \vec{d}_{P'}$. Furthermore, a match pair of its similarity value less than a pre-defined threshold, *i.e*, 0.8, is excluded from casting a vote on the Hough-voting volume to avoid generating spurious peaks. Obviously, the peaks in H_I group patches in *I* into meaningful objects. The member patches to constitute a key-object can be found through performing the inverse Hough transform on the corresponding peak. Also, multiple peaks can be detected from H_I to locate multiple similar objects for the target object $\overline{O}$.

We also propose a parameter verification process to fine tune the location $\Lambda = (x, y)$ of the detected object in *I*. For each object, including the target and detected objects, we also construct a global HOG to characterize the shape of the objet [15]. The distance between the detected and target objects can then be obtained by (7). The object O^* located at Λ^* is thus defined as

$$\Lambda^* = \arg \max_{\Lambda' \in N(\Lambda)} [1 - d(o_{\Lambda'}, \overline{o})] \tag{9}$$

where $N(\Lambda)$ returns all significant peaks from the neighborhood of Λ in H_I. Based on Λ^*, the system fine tunes the location of the detected object o_{Λ^*} in *I*. Moreover, the similarity between the detected and target objects is obtained. Although this process results in additional time for fine tuning the geometric transformation parameters, our experimental results show that it significantly improves the accuracy of object locations.

The core idea of our approach is to automatically compute labels for non-labeled samples belonging to the same class by minimizing the video object detection errors using dynamic

ISBN: 978-1-61208-265-3

programming. The dynamic programming process (DPP) optimally aligns the initial (seed) video object $\vec{O}_1 = \{\overline{o}_{1,i}\}_{i=1}^{m}$ with the frames of the input video clip $V = \{F_t\}_{t=1}^{n}$ with the shortest distance. Let $A[i,j]$ denote the distance of the optimal alignment of $\vec{O}_1^{(i)} = (\overline{o}_{1,1}, \overline{o}_{1,2}, \dots, \overline{o}_{1,i})$ and $V_j = (F_1, F_2, \dots, F_j)$. The recurrent equation used to align $\vec{O}_1^{(i)}$ and V_j with the shortest distance in a bottom-up fashion by dynamic programming is

$$A[i,j] = \min(A[i-1,j-1], A[i,j-1], A[i-1,j]) + d(\overline{o}_{1,i}, o_{s_j}) \quad (10)$$

where o_{s_j} is the object detected from F_j at location s_j with the distance measurement $d(\overline{o}_{1,i}, o_{s_j})$ using (7). The goal of the recurrent equation is to find out the value of $A[m][n]$ which denotes the error to detect the video object from the input video clip using the seed video object. The initial condition for $A[i,j]$ is

$$A[i,j] = \begin{cases} 0 & if \quad i = j = 0 \\ \infty & if \quad i \neq 0 \wedge j = 0. \\ \infty & if \quad i = 0 \wedge j \neq 0 \end{cases} \quad (11)$$

Given the set of detected video objects of a class, the SVM classifier φ with the string kernel defined in (6) is then trained to generate a new seed video object for further improving the detection and classification accuracy by an optimization loop. Let $C = \{V_i\}_{i=1}^{N}$ be set of training video clips and K be the maximal number of iterative loop. Given that initial video object $\vec{O}_1 = \{\overline{o}_{1,i}\}_{i=1}^{m}$, we define the proposed class-specific training (CST) algorithm as follows.

```
CST($C$, $\vec{O}_1$,$K$){
  $O_t \leftarrow \overline{O}_1$
  for $k$ = 1 to $K$ do{
    for $i$ = 1 to $|C|$ do $\widetilde{O}_i \leftarrow DPP(O_s, V_i)$;
    $\varphi \leftarrow SVM\_Training(\{\widetilde{O}_i\}_{i=1}^{|C|})$;
    $O_t \leftarrow \arg\max_{\widetilde{O}_i, i=1,..,|C|} [\varphi(\widetilde{O}_i)]$;
  }
  return ($O_t, \varphi$);
}
```

Finally, each class is represented as a template video object O_t and a SVM classifier φ. The former is used to detect a candidate video object from an input video clip using the proposed dynamic programming process. The classifier is then used to verify the correctness of the detected object.

B. The Detection

Given a test video clip, we first perform the same preprocessing procedure to temporally normalize the input video. The normalized video is also represented as a set of key-frames using the same key-frames detection in the training phase to reduce the time complexity of the successive video object detection using the Hough voting and dynamic programming. The detected video objects are then verified by the classifier φ.

The video object detection actually consists of two major steps: (1) detect the target video object O_V from the input video clip V based on the template video object of a class obtained in the training phase using the Hough voting and dynamic programming; (2) the class label of O_V is then defined to be

$$c(O_V) = \arg\max_{c \in C} \varphi_c(O_V) \quad (12)$$

where $c(O_V)$ is the class label of the video object O_V ; C is the set of classes; φ_c is the SVM classifier of the class c.

Let $\mathrm{T} = \{1,\dots,T\}$ be the set of time steps, and $\Omega = \{1,\dots,W\} \times \{1,\dots,H\}$ the set of locations, where W and H are the width and heights of the video frames. Given a classifier, the complexity of the video object detection by classification would be $O(W^T H^T)$ if we check all candidate video objects in a brute-force fashion. The time complexity of the proposed video object detection by dynamic programming is $O(T^2 W^2 H^2)$ which is much faster than the brute-forth approach.

IV. Experimental Results

A series of experiments was conducted on an Intel PENTIUM Dual Processor 3.0GHz PC and three video datasets, the KTH dataset [30], the Weizmann dataset [31], and the UCF sports [32] are constructed to evaluate the performance of the human action detection and recognition system. The KTH video sequences have been used in many human action recognition studies. It contains six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping. Each action is performed several times by 25 actors in four different scenarios: outdoor, outdoor with camera zooming, outdoor wearing different clothes, indoor. In total, there are 599 videos. The Weizmann dataset provides 90 video sequences of 9 actors performing 10 different actions. The UCF sports dataset is a collection of 150 broadcast sports sequences from network news videos and features ten different events: diving, golfing, kicking, weight-lifting, horseback-riding, running, skateboarding, swinging 1 (gymnastics, on the pommel horse and floor), swinging 2 (gymnastics, on the high and uneven bars), and walking. It is a very challenging dataset due to the camera motion and background clutter. These datasets have been used in many human action recognition studies.

The class labels, as the ground truth, for video sequences in the test datasets are used to determine the relevant matches in the test dataset to the query templates. Evaluations were done with a leave-one-out cross-validation. Classification results are shown in Table I and compared with state-of-the-art recognition systems [13, 18, 26, 43, 48-52]. The classification results provided in [13] include three variations of training and testing data: (A) training and testing on tracks generated from ground-truth annotations; (B) training on tracks from ground truth and testing on automatically extracted tracks; and (C) training and testing on automatically extracted tracks. The data variation C is used to construct the system. Table I shows that the classification accuracy of the method has better performance using the detected video objects as the input to class-specific SVM classifiers.

We follow the same localization evaluation rules in [13]: a detection is considered correct if, (1) the action object was correctly classified, and (2) the intersection-union ratio of the detection and ground truth bounding box is greater than 0.5. For the KTH and UCF datasets, selected frames were hand-annotated with bounding boxes, and the bounding boxes for the frames in between were generated by linear interpolation. For the UCF dataset, bounding boxes were provided as part of the ground truth annotation released with the data. Tables II and III show the performance comparison in localization accuracy using datasets KTH and UCF, respectively. All the compared methods perform well in action object detection and the proposed approach has the best performance in average detection accuracy. This illustrates the effectiveness of the GHT-based method in video action object detection.

TABLE I. CLASSIFICATION COMPARISON OF KTH, WEIZMANN, AND UCF WITH OTHER METHODS. '-' MEANS THE DATA IS NOT PROVIDED IN THE ORIGINAL PAPERS.

Method	Weizmann	KTH	UCF
Proposed	**100 %**	**95.2 %**	83.4 %
Hough forest (A) [13]	97.8 %	93.5 %	**86.6 %**
Hough forest (B) [13]	95.6 %	92.0 %	81.6 %
Hough forest (C) [13]	92.2 %	93.0 %	79.0 %
Rodriguez et al. [32]	-	85.66%	69.2%
Wang et al. [33]	-	90.1%	81.6%
Yeffet & Wolf [34]	**100%**	90.1%	79.2%
Niebles et al. [18]	90 %	83.3 %	-
Schindler et al. [35]	90 %	92.7 %	-
Laptev et al. [36]	**100%**	91.8 %	-
Ommer et al. [21]	97.2 %	87.9 %	-

TABLE II. KTH LOCALIZATION RESULTS.

Method		Propsoed	Hough Forest [13]	voc. Forest [37]
Precision	Boxing	0.97	0.88	**0.98**
	Hand Clapping	**0.98**	0.96	0.97
	Jogging	**0.90**	0.84	0.79
	Running	**0.80**	0.72	0.78
	Walking	**0.95**	**0.95**	0.86
	Hand Waving	**0.98**	**0.98**	0.96
	Average	**0.93**	0.89	0.89

TABLE III. UCF LOCALIZATION RESULTS.

Classes	Precision	
	Proposed	Hough Forest [13]
Diving	**0.62**	0.52
Weight Lifting	**1**	**1**
Walking	**0.70**	0.67
Golfing	**0.79**	0.77
Skateboarding	**0.41**	0.39
Kicking	**0.41**	0.28
Running	**0.43**	0.37
Horseback Riding	**0.78**	0.66
Swing 1	**0.46**	0.44
Swing 2	**0.32**	0.26
Average	**0.59**	0.48

Figure 3 shows a result of human action detection and recognition using the proposed method. The system correctly detects and classifies the video object in a test video clip belonging to the class "Hand Waving" using the template video object and classifier of "Hand Waving". On the contrary, the voting results of matching the sampled patches of the test video clips to other template video objects on the Hough voting volume *H* will generate low responses. The peaks of *H* are obvious and easy to detect using a simple thresholding technique. As compared with conventional VOD methods, the system detects video objects belonging to a specific class. Non-meaningful video objects are discarded by the system.

V. CONCLUSION

In this paper we have presented a method for video object detection and recognition based on the fusion of template video object modeling and dynamic programming. The proposed template video object modeling encodes each class-specific template video object as a Hough model sequence. The dynamic programming framework is then used to optimally align the frames of an input test video sequence with the model sequences. The alignment results determine the positions of the corresponding video object in the test video sequence. The trained SVM classifiers are then used to annotate the type of the detected video object. An application to human action detection and recognition is also constructed to verify the performance of the system. As compared with related GHT-based human action detection and recognition methods, the proposed method has the following contributions. First of all, this paper models the process of video object detection by the fusion of Hough voting and dynamic programming which is optimally retain the spatial-temporal information of a video object. Secondly, taking the detected video objects as the input, a training procedure with effort of human-made labeling to learn SVM classifiers with string kernels is discussed. The SVM classifiers estimate the possibility of a specific video object which performs a certain activity. In the test phase, the system detects and recognizes video objects from the input video clip automatically. Finally, the key-object representation is robust to temporal scaling in video object detection and recognition. Experimental results show that the proposed method gives good performance on several publicly

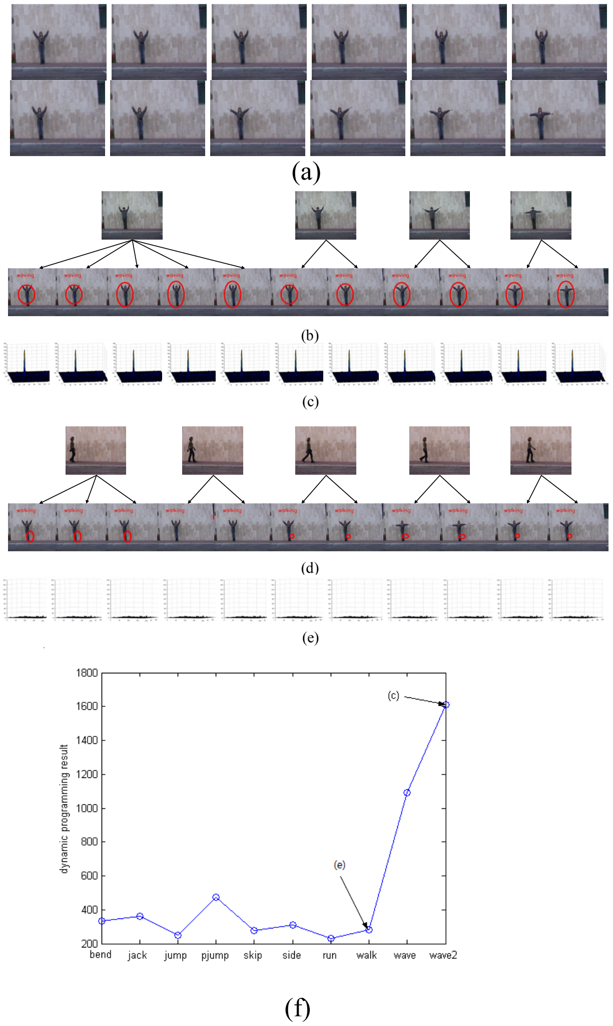


Figure 3. An example of human action detection and recognition using the proposed method on the dataset "Weizmann": (a) partial frames of a test video sequence belonging to the class "hand waving"; (b) detection results of (a) using the template video object of "hand waving"; (c) Hough voting results on the each frame of the test video sequence in (b); (d) detection results of (a) using the template video object of the class "walking"; (e) Hough voting results on the each frame of the test video sequence in (d); (f) Hough voting results of (a) for classification.

available datasets in terms of detection accuracy and recognition rate.

The proposed method suffers from the following limitations. The computational complexity of the approach using class-specific model matching by dynamic programming and GHT is essentially high. To implement the system on a parallel architecture, e.g., a GPU machine can solve the problem. Basically, GHT-based approaches can detect multiple objects from images or videos. However, the system based on its current implementation does not deal with the problem. Future work will deal with adding the detection of multiple video objects in a scene to the system, and increasing the database size.

ACKNOWLEDGMENT

This work was supported in part by National Science Council, Taiwan under Grant Numbers NSC 100-2221-E-019-054-MY3 and NSC 101-2918-1-019-003.

REFERENCES

[1] L. Ballan, M. Bertini, A. D. Bimbo, L. Seidenari, and G. Serra, "Event detection and recognition for semantic annotation of video," Multimedia Tools and Applications, Vol. 51, No. 1, 2011, pp. 279-302.

[2] G. Lavee, E. Rivlin, and M. Rudzsky, "Understanding video evenets: A survey of methods for automatic interpretation of semantic occurrences in video," IEEE Trans. Systems, Man, and Cybernetics Vol. 39, No. 5, 2009, pp. 485-504.

[3] R. Poppe, "A survey on vision-based human action recognition," Image and Vision Computing, vol. 28, no. 6, 2010, pp. 976-990.

[4] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 32, No. 2, 2010. 288-302.

[5] C. Xu, J. Cheng, Y. Zhang, Y. Zhang, H. Lu, "Sports video analysis: Semantics extraction, editorial content creation and adaptation," Journal of Multimedia, Vol. 4, No. 2, APRIL 2009, pp. 69-79.

[6] T. Zhang, C. Xu, G. Zhu, S.Liu, H. Lu, "A generic framework for event detection in various video domains," in Proc. ACM Multimedia, 2010, pp. 103-112.

[7] W. Brendel and S. Todorovic, "Video object segmentation by tracking regions," in Proc. Int'l Conf. Computer Vision (ICCV), 2009.

[8] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear)

[9] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in Proc. ECCV, 2010.

[10] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," IEEE Trans. Pattern Analysis & Machine Intelligence, vol. 29, no. 12, 2007, pp. 2247–2253.

[11] L. Shang, P. Jasiobedzki, and M. Greenspan, "Model-based tracking by classification in a tiny discrete pose space," IEEE Trans. Pattern Analysis & Machine Intelligence, vol. 29, no. 6, 2007, pp. 976-989.

[12] M. Nicolescu and G. Medioni, "A voting-based computational framework for visual motion analysis and interpretation," IEEE TPAMI, vol. 27, no. 5, 2005, pp.739–752.

[13] A. Yao, J. Gall, and L. V. Gool, "A Hough transform-based voting framework for action recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2010.

[14] A. Oikonomopoulos, I. Patras, and M. Pantic, "An implicit spatiotemporal shape model for human activity localization and recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009.

[15] P. Felzenszwalb, R. Girshick, and . McAllester, "Cascade object detection with deformable part models," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2010.

[16] B. Leibe, A. Lenardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," International Journal of Computer Vision, vol. 77,no. 1-3, 2008, pp. 259-289.

[17] P. Scovanner, A. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in Proc. ACM Int'l Conf. Multimedia, 2007.

[18] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," in Proc. Int'l Computer Vision, vol. 79, no. 3, 2008, pp. 299-318.

[19] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," International Journal of Machine Learning and Cybernetics, 1, 1, 2010, pp. 43-52.

[20] S. Vijayanarasimhan and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds," in Proc. CVPR, 2011.

[21] B. Ommer, T. Mader, and J. M. Buhmann, "Seeing the objects behind the dots: Recognition in videos from a moving camera," International Journal of Computer Vision, 83, 1, 2009, pp. 57–71, 2009.

[22] A. Prest, C. Leistner, J. Civera, C. Schimd, V. Ferrari, "Learning object class detectors from weakly annotated video," in Proc. CVPR 2012.

[23] Y. Kalal, J. Matas, and K. Mikolajcyzk, "P-N learning: Bootstrapping binary classifiers from unlabeled data by structural constraints," in Proc. CVPR, 2010.

[24] K. Ali, D. Hasler, and F. Fleuret, "Flowboost -appearance learning from sparsly labeled video," in Proc. CVPR, 2011.

[25] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognition, vol. 13, 1981, pp. 111-122.

[26] Y.-L. Chen, S.-C. Cheng, and Y.-P. Phoebe Chen, "Reordering Video Shots for Event Classification Using Bag-of-Words Models and String Kernels," in Proc. Intl. Conf. Image and Vision Computing (*IVCNZ '12*) 2012.

[27] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," In Proc. of ACM Int'l Workshop on Computational Learning Theory, 1992.

[28] J. Shawe-Taylor and N. Cristianini, Kernel methods for pattern analysis. Cambridge University Press, New York, 2004.

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

[30] J. Liu, J. Luo, and M. "Shah, Recognizing realistic actions from videos 'in the wild'," in Proc. Int'l Conf. IEEE Computer Vision and Pattern Recognition, 2009.

[31] J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," in Int'l Conf. Computer Vision and Pattern Recognition, 2009, 1022-1029.

[32] M. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in IEEE Conf. Computer Vision and Pattern Recognition, 2008.

[33] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in Proc. 20th British Machine Vision Conference, 2009.

[34] L. Yeffet and L. Wolf, "Local Trinary Patterns for human action recognition," in Proceedings of International Conf. Computer Vision, 492-497, 2009.

[35] K. Schindler and L. J. V. Gool, "Action snippets: How many framesdoes human action recognition require?," in Proc. ICCVPR, 2008.

[36] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in Proc. CVPR, 2008.

[37] K. Mikolajczyk and H. Uemura, "Action recognition with motion-appearance vocabulary forest," In Proceedings of IEEE Conf. Computer Vision and Pattern Recognition, 2008.