Instantly Telling What Happens in a Video Sequence Using Simple Features

Liang Wang[1,2], Yizhou Wang[2,3], Tingting Jiang[2,3], Wen Gao[2,3]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2]Nat'l Engineering Lab for Video Technology, Peking University, Beijing, China
[3]Key Lab. of Machine Perception (MoE), School of EECS, Peking University, Beijing, China
wangliang@jdl.ac.cn, Yizhou.Wang@pku.edu.cn, ttjiang@pku.edu.cn, wgao@pku.edu.cn

Abstract

This paper presents an efficient method to tell what happens (e.g. recognize actions) in a video sequence from only a couple of frames in real time. For the sake of instantaneity, we employ two types of computationally efficient but perceptually important features, optical flow and edge, to capture motion and shape/structure information in video sequences. It is known that the two types of features are not sparse and can be unreliable or ambiguous at certain parts of a video. In order to endow them with strong discriminative power, we extend an efficient contrast set mining technique, the Emerging Pattern (EP) mining *method, to learn joint features from videos to differentiate action classes. Experimental results show that the combination of the two types of features achieves superior performance in differentiating actions than that of using each single type of features alone. The learned features are discriminative, statistically significant (reliable) and display semantically meaningful shape-motion structures of human actions. Besides the instant action recognition, we also extend the proposed approach to anomaly detection and sequential event detection. The experiments demonstrate encouraging results.*

1. Introduction

Action recognition has been extensively studied in the literature due to its wide spectrum of applications, e.g. video surveillance and human computer interaction. Although most of the state-of-the-art methods report impressive results, e.g. [7][5], the recognition engines usually require to extract a large amount of sophisticated features over certain period of time in order to obtain a reliable action label. Consequently, a judgment can only be made after a long delay of feature collection. The latency can be an entire period or even several repetitions of a whole action. However, human usually can instantly tell what happens in a scene only at a glance. In the literature, some researchers studied the capability of human perception, for example, Li *et al.*[8] reported that even with a glimpse of a single image, human can reliably tell what is in the image and what event happens in the scene. Whereas, this instantaneity property has been rarely considered as a key criterion of action recognition engines. As a result, even for the state-of-the-art action recognition methods, if supplied with only a couple of frames, the recognition accuracy is barely above the chance (see example results in Fig. 3-5).

In this paper, we present an efficient method which is able to instantly tell what happens in a video sequence based on only a few frames. Specifically, the proposed method is able to recognize actions using any two consecutive frames of an action video of resolution 160×120 with an average speed of 0.04s. (The experiment is conducted on an Intel Core2 Duo 3.20GHz CPU, 3.0G RAM PC.) In order to satisfy this instantaneity requirement, we employ two types of computationally efficient (simple) but perceptually important features, the optical flows [9] and Canny edges [11], to capture the motion and shape/structure information in video sequences.

Both features are fast to compute and have small memory demand. However, the optical flows are usually considered to be unreliable and ambiguous due to various reasons, e.g. aperture problem and low image quality. Similarly, shape features are also thought to be difficult to extract from motion sequences [12]. We admit that simple features do have their limits, e.g. sensitive to different kinds of variations and noise. But the situation is not always that bad. For instances, it is known that optical flow at junctions and corners usually can be accurately estimated, and edge information has been successfully utilized to describe object geometric and topological properties in many challenging applications [4]. In addition, although motion or shape features alone may have weak discrimination power, the combination of the two cues can exhibit distinctive semantic characteristics as shown in Fig. 2. Based on these observations, we strongly believe that there always exist some reliable and inexpensive features which can be *further exploited* to serve for certain challenging visual tasks.

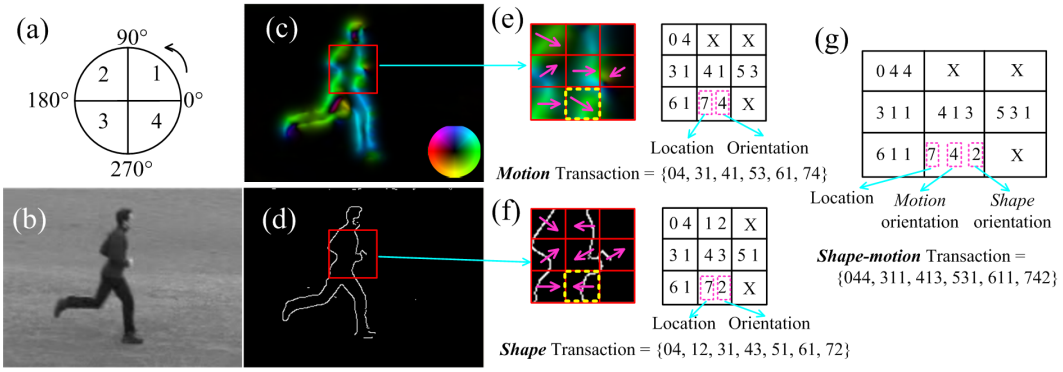


Figure 1. Encoding patch features as transactions. (a) Optical flows and Canny edges are quantized into four sections according to their mean orientation indexed by 1-4. (b) A frame of action 'Running'. (c) & (d) show its optical flows and Canny edges respectively. The intensity indicates the magnitudes of the two features. The color in (c) encodes the optical flow orientation. (e)-(g) illustrate the encoding method of the cells and the patch. See Section 2.2 for detailed explanation.

The key to the success is how to identify the 'right' ones.

The simple features are usually densely distributed in the data (video sequences in our context), the number of their combination is even larger. In order to quickly identify those discriminative ones from the large search space, we employ an efficient data mining technique, the *Emerging Pattern (EP) Mining* method [6]. The EP mining was originally proposed to analyze social statistics data and find differences among social groups. In this paper, we extend the method to learn discriminative features (including their combinations) in-between different actions. Experiment results show that the learned features are discriminative, statistically significant (reliable) and they can display semantically meaningful shape-motion structures of human actions. Moreover, by using this method, the training can be accomplished in a weakly-supervised way, i.e., we only need to assign action labels to the videos without annotating the bounding-box of actions, aligning video sequences, or labeling the start and end of actions, etc. This saves tremendous manual work; Furthermore, the proposed feature combination and recognition strategy enables two other challenging applications – anomaly detection and sequential event detection – to be handled in real time after reformulating the problems. This real-time/online feature is not possessed by other state-of-the-art methods to our best knowledge.

Because there is a lack of literature about action recognition using very few frames, we only identified three related ones and introduce them as follows. Li *et al.* [8] recognize actions in single images by integrating scene and object level image interpretations without leveraging motion cues. However, it is known that to obtain such high level semantic information from an image is not only computationally expensive, but also can be unreliably in general. Wang *et al.* [19] propose a hidden Conditional Random Field (hCRF) model, which combines global and local features of motion fields in a single frame to distinguish actions. Schindler *et al.* [15] also study the problem of recognizing actions from a small number of frames ('snippets'). However, both these methods require explicit annotation of the actors in training data by bounding-boxes. Whereas, the proposed training method only needs to specify action labels of training data. Moreover, our method focuses on exploiting an ingenious combination of simple features to achieve real-time performance in action recognition which is rarely addressed in the literature.

In the following, we first introduce the method of learning discriminative features using EP mining in Section 2. Applications based on the proposed method are presented in Section 3 & 4. We conclude the paper in Section 5.

2. Learning Discriminative Feature via Contrast Set Mining

In this section, we first introduce the EP mining method, then we show how to extend it to learn discriminative visual features from videos of different classes.

2.1. Emerging Pattern Mining

Emerging pattern mining is proposed to analyze the difference between two datasets/classes [6]. We follow the notation in [6] to introduce the mathematical definition of the emerging pattern. Let $I = \{i_1, i_2, ..., i_N\}$ be a set of N items. A *transaction* refers to a subset T of I. A *dataset* D is composed of a set of transactions. A subset X of I is also called an *itemset*, and if $X \subseteq T$, we say the transaction T contains the itemset X. The *support* of an itemset X in a dataset D is defined to be $\varrho_D(X) = \text{count}_D(X)/\|D\|$ where $\text{count}_D(X)$ is the number of transactions in D containing X. Given an itemset X and a pair of datasets D_1

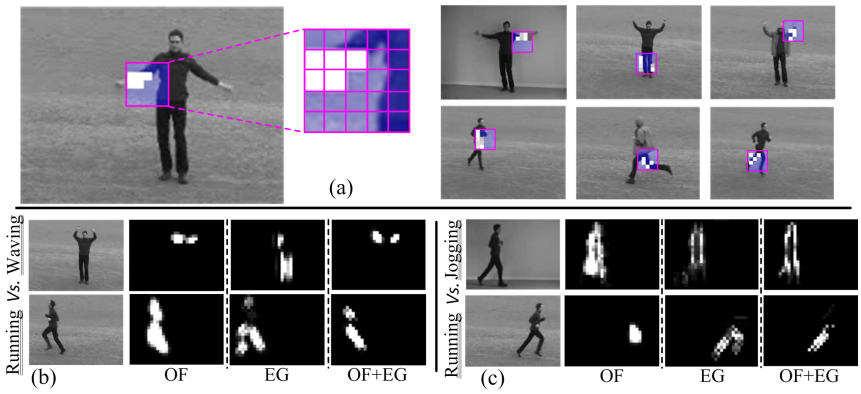


Figure 2. Mined discriminative features and their density maps for two action frame pairs from KTH dataset [16], (a) Examples of discriminative features. A discriminative feature corresponds to a mined itemset (highlighted white cells) distributed in a frame patch (a transaction) within the purple window. (b) Three feature density maps of a 'Handwaving' frame vs. a 'Running' frame, i.e., optical flow (OF), Canny edge (EG) and combined shape-motion features (OF+EG). (c) Three feature density maps of a 'Jogging' frame vs. a 'Running' frame.

and D_2, the *growth ratio* of an itemset X from D_1 to D_2 is computed as

$$\upsilon_{D_1/D_2}(X) = \begin{cases} 0, & \text{if } \varrho_{D_1}(X) = 0 \& \varrho_{D_2}(X) = 0 \\ \infty, & \text{if } \varrho_{D_1}(X) = 0 \& \varrho_{D_2}(X) \neq 0 \\ \frac{\varrho_{D_2}(X)}{\varrho_{D_1}(X)}, & \text{otherwise} \end{cases} \quad (1)$$

An itemset is said to be an ε-*emerging pattern* ($\varepsilon > 1$) from D_1 to D_2, if $\upsilon_{D_1/D_2}(X) > \varepsilon$. Thus, intuitively, EP mining finds the itemsets whose support ratios vary a lot from one dataset to another. It captures the contrasts between pairs of datasets that can be employed to design classifiers. EP mining has been successfully applied to various fields such as medical research and social science.

In this paper, we extend the EP mining method to study visual data, in particular, to learn discriminative features between different classes of videos. A major reason we employ this method as a component of our framework is that the method is so efficient that it is able to find EPs from tens of thousands transactions in seconds. In the following, we introduce how to learn discriminative visual features between different video classes with EP mining.

2.2. Representing Video Patches as Transactions

Because EP mining method was originally proposed to study transaction data, in order to extend the method to video analysis, the first step is to convert videos from a visual feature representation into a transaction representation.

The feature quantization is illustrated in Fig. 1. For a video sequence, we first compute its optical flow and Canny edges at each frame. Then, in a patch of size $M \times N$ pixels, we encode its visual features into a transaction in the following steps. (i) The patch is equally divided into a grid, of which each cell is $m \times n$ pixels (Fig. 1(e)&(f)). In our implementation, $M = N = 40$ and $m = n = 8$. The optical flow and Canny edges in a cell are quantized into two integers respectively, each ranges from 1 to 4 according to their mean orientation (see Fig. 1(a)). (ii) The features of a cell are treated as an item of the transaction, and is encoded by concatenating its location in the patch and the quantized feature values. For example, the cell highlighted by a yellow dotted rectangle in Fig. 1(e) is located at the 7-th cell of the patch, and its optical flow feature is quantized to be 4. Hence, the digitized optical flow feature item of the cell is 74. Similarly, its shape item are 72 (Fig. 1(f)). If considering both shape and motion, the encoded item is 742. (iii) The transaction of a patch is composed of the items of all its cells. Noted that if the mean magnitude of the features in a cell is smaller than a threshold, the cell is ignored in the transaction and denoted as 'X' in Fig. 1(e)-(g).

2.3. Learning Simple Discriminative Features

To learn discriminative features between different video classes, frame patches are sampled from both the positive video set and the negative one. On each frame, we sequentially extract overlapping image patches whose centers are 5 pixels away either vertically or horizontally. Then we convert these patches into transactions using the method described above. From the obtained positive transaction dataset and the negative one, discriminative features are mined as emerging patterns using the method described in Section 2.1. The two parameters used in EP mining are the *support* in positive dataset and *growth ratio* as defined in Section 2.1. The *support* parameter measures how statistically significant (or descriptive) a mined pattern is in the positive dataset. Whereas, the *growth ratio* reflects the discriminative power of a mined pattern.

Some learned discriminative features are illustrated in Fig. 2 (a). A discriminative feature corresponds to an itemset (the highlighted white cells) residing in a patch

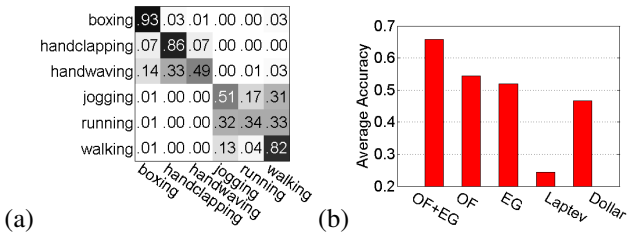


Figure 3. (a) Confusion matrix of the frame-based action recognition method on KTH dataset using shape-motion features. (b) Comparison of recognition accuracies using different feature combinations (OF: optical flow, EG: Canny edge, OF+EG: shape-motion), also with two other methods in [13] and [5]).

(highlighted in blue). An itemset encodes discriminative shape/motion information as well as its spatial distribution in an image patch. Fig. 2 (b) & (c) show density maps of three types of learned discriminative features distributed in two action image pairs, including optical flow (OF), Canny edge (EG) and combined shape-motion features (OF+EG). As can be observed, the mined discriminative features of each type capture the semantic structures of actions from different aspects. For instance, in Fig. 2 (b), the optical flow features differentiate the 'Waving' frame from the 'Running' frame by the arm motion, whereas they distinguish the 'Running' from 'Waving' by the motion of leg and torso (which generally move horizontally). The Canny edge features discriminate the two action frames by the poses of leg contours. The combined features capture the two action frames' characteristic differences of motion and shape/pose on both the arm and leg simultaneously.

For the confusing action frame pair 'Jogging' vs. 'Running' (Fig. 2 (c)), the motion feature alone cannot distinguish them well. However, the edge features identify the 'Jogging' frame using the vertical lines along the torso, and pick up the slant lines along the leg as discriminative features of the 'Running' frame. This may due to the motion magnitude difference between the two actions. Compared to the shape feature, the shape-motion features further include some new bits around the arms for the 'Running' frame. These observations confirm the enhancement of the discriminative power brought by combining motion and shape features.

The detection of a learned discriminative feature (emerging pattern) in a frame patch is also very efficient. It is realized by just checking whether the emerging pattern (an itemset) is contained in the transaction related to the patch.

3. Action Recognition

In this section, we present two efficient action recognition methods using the learned discriminative features.

3.1. Frame-based Action Recognition

We adopt a boosting framework [17] to recognize actions using any frame of an action video. A strong classifier $H_c(\mathbf{I})$ of action class $c \in C$ is trained by integrating the learned discriminative features as weak classifiers $h_{f_c}(\mathbf{I})$, where $\mathbf{I}$ denotes an action video frame, and f_c is a learned discriminative feature of c. The discriminative features are learned in two directions, i.e., the EPs are mined either from the negative class to the positive one or vice versa.

Each learned feature f_c has a discriminative score $s_{f_c}(\mathbf{I})$. It is defined as a function of its support $\varrho_{D_*}(f_c)$, growth ratio $\upsilon_{D_./D_*}(f_c)$ (defined in Section 2.1 and (1)), and the number of instances of f_c detected in $\mathbf{I}$ is denoted as $n(\mathbf{I}, f_c)$.

$$s_{f_c}(\mathbf{I}) = \varrho_{D_*}(f_c) \cdot \frac{\upsilon_{D_./D_*}(f_c)}{\upsilon_{D_./D_*}(f_c) + 1} \cdot n(\mathbf{I}, f_c), \quad (2)$$

where D_+ is the positive training set, which contains video frames of a target action. D_- is the negative one, which is composed of all action frames of other types. $D_./D_*(f_c)$ can be $D_-/D_+(f_c)$ or D_+/D_- depending on which direction f_c is mined from.

The output of a weak classifier $h_{f_c}(\mathbf{I}) = 1$, if $s_{f_c}(\mathbf{I}) > \tau$; otherwise, 0. τ is learned by minimizing the training error using $s_{f_c}(\mathbf{I})$.

After training as in [17], the action label for an action video frame can be determined by $\arg\max_c \left(\sum_{f_c} \alpha_{f_c} h_{f_c}(\mathbf{I})\right)$, where α_{f_c} is the learned weight of h_{f_c}.

3.2. Video-based Action Recognition

We further apply the frame-based method to recognize actions using more frames or even whole videos. The assumption is, if we can get good predictions using some of the frames in a video, by accumulation, the prediction accuracy can be improved over the whole video sequence.

For training, we use the same method as in Section 3.1. Action recognition on a given video clip is accomplished by the following steps. (i) Key-frames are sampled from the video clip every four frames to reduce computational cost. (ii) Because the key-frames are sampled without any preference – some snapshots of actions can be ambiguous, while some are very distinguishable – different frames provide diverse confidence in judging its action label. Taking this into consideration, we propose to further select a subset of *confident key-frames* to participate in recognizing actions as follows. We compute a confidence score of a frame $\mathbf{I}_t$ as $\text{conf}(\mathbf{I}_t) = \max_c \sum_{f_c} \alpha_{f_c}(\mathbf{I}_t)$, where $\alpha_{f_c}(\mathbf{I}_t)$ is the learned weak classifier weight of a detected feature in $\mathbf{I}_t$. If $\text{conf}(\mathbf{I}_t) > 0.7$, the key-frame is a *confident key-frame*. The selected confident key-frame set is denoted as $\mathbf{K}$. (iii) We recognize the action in these confident key-frames using the

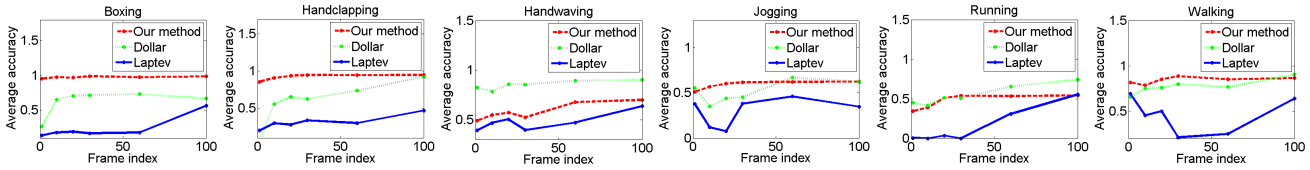


Figure 4. Recognition accuracy comparison between the proposed video-based action recognition method and the methods in [13] and [5]) on the KTH dataset. The proposed method uses shape-motion features.

method in Section 3.1. The recognition result is denoted as $H_c(\mathbf{I}_t) \in \{0, 1\}$. (iv) The action label of the video is determined by majority voting

$$c = \arg\max_{c'} \sum_{\mathbf{I}_t \in \mathbf{K}} H_{c'}(\mathbf{I}_t). \quad (3)$$

3.3. Evaluation

We use two public action datasets to evaluate the proposed method, the KTH dataset [16] and the YouTube dataset [14]. The KTH dataset contains 25 people performing 6 types of actions, such as boxing, and hand waving. Videos of the YouTube dataset are captured under uncontrolled environment with camera motions, cluttered background and variations in view-points. This dataset contains 11 types of sports actions, such as jumping and diving.

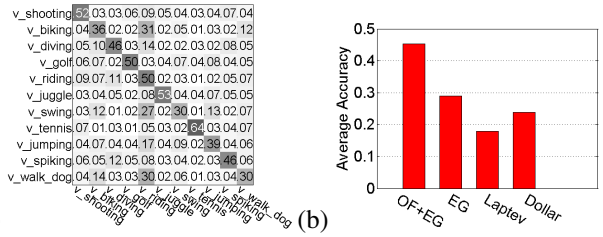
In the experiments, we set the growth ratio threshold as 2; the support threshold of each action is chosen such that there are at least 500 discriminative features mined. (These thresholds remain the same in the anomaly and sequential event detection applications in Section 4.)

Evaluation on KTH Dataset Fig. 3 shows the comparison results of the *frame-based action recognition task* between the proposed method and two popular action recognition methods that use spatiotemporal interest point (STIP) detectors proposed in [13] and [5]. To test the performance of the STIP based method, we use the same training/testing frames as that for our method. The STIPs are clustered into 400 clusters by the K-means algorithm as in [13] and [5]. As can be seen, (i) the combined shape-motion feature is superior to either of the single type features in differentiating actions; (ii) the proposed method performs better than the STIP based methods even when the comparison conditions are not equal. (Extracting STIPs requires usually more than 6 frames, whereas our method just use the information from single frames.)

Tab. 1 shows another comparison result to the method introduced in [19], which also recognizes actions in a frame based manner and obtains good results. The result show that although the proposed method uses much less training data, it still achieves a comparable performance. Moreover, our method is much more computationally efficient (making prediction in 0.04s on average), and it does not require explicitly tracking people in the videos as [19] does.

Method	Ours	[19]
Average accuracy	65.4%	66.9%
Train/test data ratio	20%/80%	50%/50%
Inference method	Linear aggregation	hCRF

Table 1. Comparison of the proposed frame-based method to a method in [19] on the KTH dataset.



(a) (b)

Figure 5. (a) Confusion matrix of the proposed frame-based action recognition method on YouTube dataset [14] using shape-motion features. (b) Comparison of recognition accuracies using different feature combinations (OF: optical flow, EG: Canny edge, OF+EG: shape-motion), also with two other methods in [13] and [5]).

We also compare the performance of the proposed method on the *video-based action recognition task* to the STIP based methods. The interesting results shown in Fig. 4 worth further discussion: (i) For the other methods, the recognition accuracies gradually improve along with the increasing number of involved frames; whereas, the performance of the proposed method stabilizes quickly (usually within 20 frames). The reason might be that the actions in the KTH dataset are short cycled actions, the proposed method quickly captures the discrepancies among different actions. Even fed with more frames, the simple repetition does not provide extra information to improve the performance. (ii) The proposed method outperforms the other two methods in four action classes except 'Handwaving' and 'Running'. This may reveal one limitation of the proposed, i.e. some actions can only be reliably recognized by considering the feature correlation over certain period. For instance, to differentiate 'Running' and 'Jogging', the temporal frequency (can be obtained from more frames) may play a more important role than body poses (can be obtained from a single frame).

Evaluation on YouTube Dataset For the YouTube

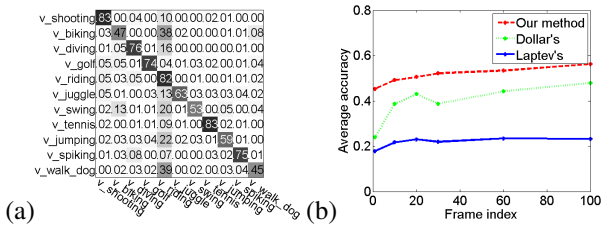


Figure 6. (a) Confusion matrix of the proposed video-based action recognition methods on YouTube dataset [14] using shape-motion features. (b) Recognition accuracy comparison between the proposed video-based action recognition method and the methods in [13] and [5]) on the YouTube dataset. The proposed method uses shape-motion features.

dataset [14], we randomly select 8 groups out of 25 as training data and use the rest for testing. (The grouping are provided by [14].) Fig. 5 shows the *frame-based action recognition* results. From the comparison result, we see that the proposed method especially with the combined shape-motion features still out performs the other two methods.

The performance of the proposed *video-based action recognition* method is shown in Fig. 6. It can be seen that, different from the performance on the KTH dataset, the recognition accuracy of the proposed method increases when more video frames are provided. This is because that the actions in the YouTube dataset are more complicated and natural than KTH. More information of different aspects of the actions (due to variation) is provided along the time, which helps to improve the recognition accuracy.

Tab. 2 shows the comparison result between our method and a method proposed in [14], which achieves the best performance on the YouTube dataset to our knowledge. It can be seen that even using much less training data and fewer number of features, the proposed method still reaches a comparable performance.

4. Two Other Applications

The proposed discriminative feature learning method can have more applications besides action recognition. In this section, we extend the method and apply it to detecting anomaly and sequential events in videos. Since there are few published datasets exclusive to the two visual tasks, we built a dataset for each task by collecting videos from some existing video datasets, such as the ones used in [3][12][10] and Google Video on the web. The anomaly detection dataset contains 9 videos of about 3 minutes, it includes abnormal human actions and traffic accidents, etc. The sequential event dataset contains 8 videos of about 37 minutes, it consists of events happening at airports and traffic junctions, etc. Some examples are shown in Fig. 7.

The proposed algorithms aim to locate the starting frames of anomaly events and novel events in the videos.

Figure 7. Sample frames of (a) the anomaly detection dataset and (b) the sequential event dataset.

Method	Ours	Method in [14]		
		Motion	Static	Hybrid
Accuracy	**67.2%**	65.1%	63.0%	71.2%
Train data ratio	8/25	24/25	24/25	24/25
Feature number	3000	400	8000	8400

Table 2. Comparison results on video-based action recognition task. The method [14] use three types of features: motion, static and hybrid. The feature number is the average number of detected features in each training video.

To quantitatively evaluate the performance of the methods, we recruit 5 undergraduate students to annotate the starting and ending time stamps of anomaly events and the sequential events in the videos, and use the average as the ground truth. Given a detected starting point t, we find its nearest ground-truth starting point t_s and the corresponding ending point t_d of the event. t is considered as a correct detection if

$$|t - t_s| < \min\left(\alpha * (t_d - t_s), \Gamma\right) \tag{4}$$

where $\alpha \in (0, 1)$ is a scaling factor for the event and Γ is a time interval threshold. In our implementation, we set $\alpha = 0.3$ and $\Gamma = 4s$.

4.1. Anomaly Detection

The proposed action recognition method in Section 3 belongs to *weakly-supervised* learning. Here we extend it to detecting abnormal actions/events in videos in an *unsupervised* manner. We define an action/event to be abnormal in a video, if (i) it happens locally in time, (ii) it occurs rarely in the video, and (iii) it "looks" different from its neighbors. To detect anomalous actions, a given video sequence is equally divided into a number of 40-frame-long subsections. We mine emerging shape-motion features from each pair of adjacent sub-clips using the method proposed in Section 2.1. Then the anomaly detection problem becomes a new problem, i.e., detecting "changes" between two consecutive chunks of the video. If the "change" is significant, an anomaly happens.

To mine the emerging features on the sub-clip pairs, the frames in the preceding sub-clip are designated as positive

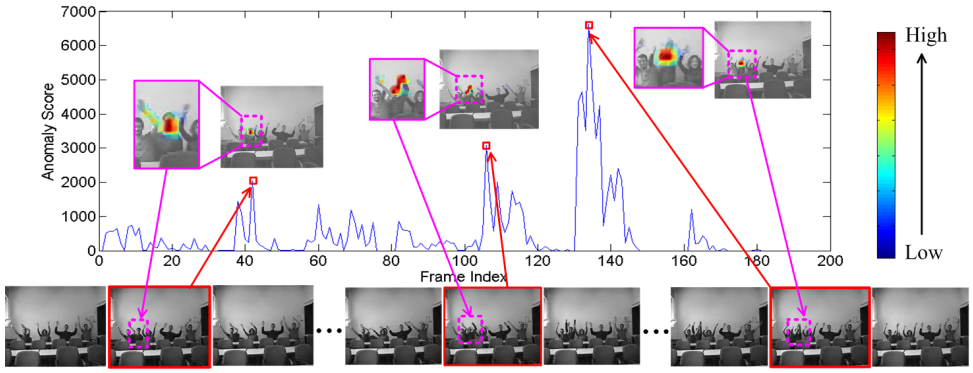


Figure 8. Anomaly scores of frames (shown in blue curve) in a video from Weizmann dataset [3]. Some abnormal frames are highlighted by red rectangles, and the identified abnormal feature density maps are displayed in heat maps.

Video	Precision	Recall	F-score
Weizmann part[3]	9/13	9/12	71.9%
Crowd part[12]	5/8	5/7	66.7%
Traffic part (Web)	4/5	4/4	88.9%

Table 3. Performance statistics of anomaly detection.

Video	Precision	Recall	F-score
Cross road	15/20	15/20	75.0%
Traffic1[10]	17/23	17/20	79.1%
Traffic2[18]	9/13	9/11	74.9%
Airport[1]	8/14	8/11	63.9%
Train Station[2]	22/36	22/25	72.1%

Table 4. Performance statistics of sequential event detection.

samples D_+; the frames in the posterior chunk are negative samples D_-. We define an *anomaly score* of a mined feature f as

$$\zeta(f) = \frac{1}{\varrho_{D_+}(f)} \cdot \frac{\upsilon_{D_-/D_+}(f)}{\upsilon_{D_-/D_+}(f)+1} \tag{5}$$

where $\varrho_{D_+}(f)$ and $\upsilon_{D_-/D_+}(f)$ has been defined in Section 2.1. Different from the discriminative score in Eqn. (2), the support $\varrho_{D_+}(f)$ is in the denominator. Thus, the mined features that occur rarely are preferred. The anomaly score of a video frame $\mathbf{I}_t$ is defined as the summation over the anomaly scores of all the mined features appearing in the frame, $\zeta(\mathbf{I}_t) = \sum_{f \in \mathbf{I}_t} \zeta(f)$. A higher anomaly score indicates that the frame contains many differences from its context, i.e. it is abnormal. Fig. 8 shows the anomaly scores of frames in a video sequence.

In our implementation, we set an adaptive threshold to be the 90% largest anomaly scores of all the frames in a video. The frames with anomaly scores higher than the threshold are detected as an anomalous frame, and only when more than 5 continuous anomalous frames are detected, the algorithm issues an anomaly alert. The anomaly detection results are listed in Tab. 3 using the combined dataset.

4.2. Sequential Event Detection

In a complex dynamic scene, dominant events change overtime. For instance, at a traffic junction, when traffic lights alter, traffic flows will also change correspondingly. To obtain an abstract description of a video, we propose to identify representative sequential events in the video. The technique may facilitate many applications including video summary, rule mining, dependency discovery, etc. Moreover, by promptly identifying a change, it helps a right model switching (e.g. different object tracking models) so as to correctly respond to new coming data.

Different from the abnormal event definition in Section 4.1, a sequential event should occur frequently in a sub-clip, e.g. many cars drive in the same direction during green light, because it is dominant in certain period of a video. It also should be distinguished from the preceding event within a local window. Similarly, each input video is equally divided into 40-frame-long sub-clips. Between each consecutive sub-clip pairs (D_-/D_+), we mine emerging shape-motion features using the method in Section 2.1. The *novelty* score of a mined feature f is defined as

$$\xi(f) = \varrho_{D_+}(f) \cdot \frac{\upsilon_{D_-/D_+}(f)}{\upsilon_{D_-/D_+}(f)+1} \tag{6}$$

It favors the features that are "novel" to the preceding sub-clip and frequently occur in the current one. We define a *frame novelty score* as $\xi(\mathbf{I}_t) = \sum_{f \in \mathbf{I}_t} \xi(f)$. Fig. 9 shows the novelty scores of frames in a traffic video. It can be seen that the frames with local maximum novelty scores correspond to the changes of traffic events. For example, in Fig. 9(a), the mined novel features are strongly related to 'car stops' in the east side of the crossroad. Similarly, Fig. 9(b)(c)(d) show several other identified meaning-

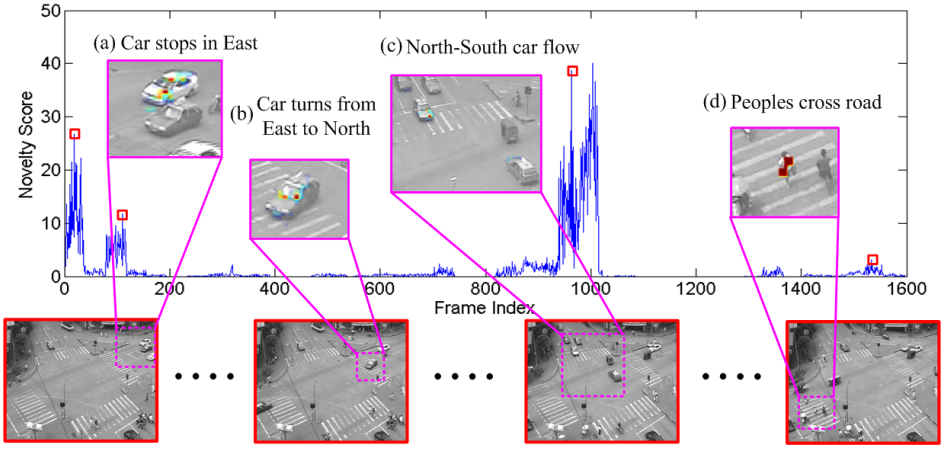


Figure 9. Novelty scores of frames in a traffic video. Some 'novel frames' are shown, and the zoomed-in areas are places where sequential events happen. Densities of the novel feature in the frames is represented by heat maps.

ful traffic events incurred by the change of traffic lights. We use the same method to detect the starting points of new events as the one to detect anomaly. The sequential event detection results are listed in Tab. 4.

5. Conclusion

In this paper, we present a method to instantly recognize actions in a video using only a couple of frames in real time. The instantaneity is achieved by employing the simple and discriminative visual features, which combine the perceptual important motion and shape information extracted from video frames. We further extend the frame-based action recognition method to detect actions on the whole videos, and apply it to anomaly detection and sequential event detection. Experimental results are encouraging.

Acknowledgment

This work was supported in part by NSFC-60872077 and Major State Basic Research Development Program of China (973 Program 2009CB320904).

References

[1] http://www.itl.nist.gov/iad/mig/tests/trecvid/2009/.

[2] http://www.cvg.rdg.ac.uk/PETS2007/data.html.

[3] O. Boiman and M. Irani. Detecting irregularities in images and in video. *IJCV*, 74(1):17–31, 2007.

[4] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. CVRP*, pages 886–893, 2005.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. IEEE Int'l Workshop on PETS*, pages 65–72, 2005.

[6] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. ACM SIGKDD*, pages 43–52, 2004.

[7] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. ICCV*, pages 726–733, 2003.

[8] L. Fei-Fei, A. Iyer, C. Kock, and P. Perona. What do we see in a glance of a scene? *J. Vis.*, 7(10):1–29, 2007.

[9] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[10] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in videos. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1165–1172, 2009.

[11] C. John. A computational approach to edge detection. *IEEE Trans. PAMI*, 8(6):679–698, 1986.

[12] Y. Ke, R. Sukthankar, and M. Hebert. Volumetric features for video event detection. *International Journal of Computer Vision*, 88(3):339–362, 2010.

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozeneld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

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

[15] K. Schindler and L. Gool. Action snippets: How many frames does human action recognition require? In *Proc. CVPR*, 2008.

[16] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proc. Int'l Conf. Pattern Recognition*, pages 32–36, 2004.

[17] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[18] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception by hierarchical bayesian models. In *Proc. IEEE Conf. CVPR*, pages 1–8, 2007.

[19] Y. Wang and G. Mori. Learning a discriminative hidden part model for human action recognition. In *NIPS*, 2008.