GRAPH BASED EVENT DETECTION FROM REALISTIC VIDEOS USING WEAK FEATURE CORRESPONDENCE

Lei Ding, Quanfu Fan, Jen-Hao Hsiao and Sharath Pankanti

IBM T. J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532

ABSTRACT

We study the problem of event detection from realistic videos with repetitive sequential human activities. Despite the large body of work on event detection and recognition, very few have addressed low-quality videos captured from realistic environments. Our framework is based on solving the shortest path on a temporal-event graph constructed from the video content. Graph vertices correspond to detected event primitives, and edge weights are set according to generic knowledge of the event patterns and the discrepancy between event primitives based on a greedy matching of their visual features. Experimental results on videos collected from a retail environment validate the usefulness of the proposed approach.

Index Terms— Video signal processing, image analysis, feature extraction, graph theory.

1. INTRODUCTION

We address the visual event detection problem using low-resolution videos captured in realistic environments. Our problem domain is videos of repetitive activities, which abound in industrial and commercial scenarios, where the actor performs activities consisting of repeated sequences of event primitives. For instance, a retail store cashier performs checkouts consisting of item-related primitives, and a factory worker performs assembling consisting of product-related primitives. In this way, several consecutive event primitives compose an event of interest. Since the accuracy of event primitive detection is typically not very high, grouping algorithms (e.g. [1]) exploiting temporal dependencies between primitives would yield many false positives due to the significant distractions in realistic environments.

Intuitively, visual feature correspondence between adjacent primitives offers useful cues for pruning out false detection. However, strong correspondence in terms of global one-to-one matching of image features is usually not available due to low video quality, rapid motion, occlusion and environmental distraction, etc. In this paper, we explore weak feature correspondence between event primitives that takes advantage of an efficient greedy matching. Besides, previous work [2, 3] use individual primitive models learned from labeled data, whereas in this paper, no labeled data is required, as the constraints are not put on individual primitives but on their consistency.

The underlying structure of our framework is a weighted temporal-event graph constructed from preliminary detection of event primitives at their respective spatial locations. The path of the smallest cost corresponds to the optimal parsing of the video into a sequence of pre-defined events. To make the main contribution clear, we adopt a simple event model that takes the form of a regular expression characterizing the repetitive patterns. We choose to use the checkout event detection in a store as a running example, although we emphasize that our framework is a general one applicable to other problems with template patterned iterative activities as well. The two main ingredients of the paper namely feature consistency and graph modeling are presented in Sections 2 and 3 respectively, followed by experimental results in Section 4.

A brief overview of recent related work goes as follows. A framework for gesture recognition and segmentation is proposed in [4], where dynamic programming is used on hypothesized locations of hands for spatial and temporal activity segmentation. A linear programming based approach in [5] also depends on initially hypothesized object locations to perform multiple object tracking. However, such object-wise initialization is nearly impossible to achieve in realistic videos where the categories of objects are many and usually unable to obtain a priori. Our problem domain is unconstrained real-world videos, which do not lend themselves to robust localization of individual objects and thus lead to the extra difficulty. Besides, graph based models such as hidden Markov models [6] and Bayesian networks [7] have been widely applied to model complex events by combining event primitives. In comparison, our graph model is simpler, less expensive and suited to the particular problem of repetitive activities.

2. FEATURES AND THEIR CONSISTENCY

We consider each event as a combination of event primitives. In this way, each primitive is represented as an unordered set of visual features including both appearance and texture information in terms of colors and gradients. Further, a score is computed between two primitives reflecting their dissimilarity, which in turn provides for useful clues for combining primitives into events of interest.
2.1. Feature detection and representation

It is a common practice that an image or video is transformed into a set of feature descriptors at salient locations. In this way, we represent a video clip as \( \{ v_1, v_2, \ldots, v_{N_v} \} \), where \( v_i \) is a description vector (e.g. color, gradient, optical flow) at the \( i^{th} \) spatial-temporal salient location in the video. There are various choices that we can use for interest point selection from videos. Among them, spatial temporal interest point (STIP) is a popular one. Despite its power demonstrated in [8, 9], the associated high computational cost motivates us to use a simpler detector, which makes our method feasible for real-time deployment.

Specifically, we take the locally maximum points of the following squared weighted gradient norm function as the interest points in a video represented by the function \( I(x, y, t) \):
\[
g(x, y, t) = \left( \frac{\partial I}{\partial x} \right)^2 + \left( \frac{\partial I}{\partial y} \right)^2 + \alpha \left( \frac{\partial I}{\partial t} \right)^2,
\]
where \( \alpha \) is larger than 1 to emphasize the fact that temporally salient points are more likely to be related to interesting events. We also exclude those \((x, y, t)\) points with relatively small \(g(\cdot)\) values below the \( D^{th} \) percentile over the whole video clip in order to achieve the truly spatial-temporal salient points.

Two kinds of features are extracted from around and at the interest points. The first kind is averaged color values in a small window around the interest points, i.e., \([R, G, B] \). The second kind is image gradient vectors at the interest points, i.e., \([\partial I/\partial x, \partial I/\partial y, \partial I/\partial t] \). They together compose a description vector containing both appearance and texture information at each detected interest point in a video.

2.2. Weak feature correspondence

In our formulation, a visual event is composed of event primitives. For example, an item checkout event in a retail store can be decomposed into item pickup, passing over the scanner and item drop-off. Each event primitive \( P \) is treated as a bag of features; that is, \( P = \{ p_i \}_{i=1}^{|P|} \). The mean Hausdorff distance \( H^c(A, B) \) between primitives \( A \) and \( B \) consisting of feature vectors \( \{ a_i^c \}_{i=1}^{|A|} \) and \( \{ b_j^c \}_{j=1}^{|B|} \) has the following form, where superscript \( c \) refers to the aforementioned color features: \( \max \left\{ \frac{1}{|A|} \sum_{i=1}^{|A|} \min_j d(a_i^c, b_j^c), \frac{1}{|B|} \sum_{j=1}^{|B|} \min_i d(a_i^c, b_j^c) \right\} \). The larger \( H^c(A, B) \), the less consistent they are in terms of the color appearance. Similarly, we use the distance \( H^q(A, B) \) for the gradients. Compared to the standard Hausdorff distance that takes maxima over minima, the mean Hausdorff distance is more robust to outliers. Finally we use a linear function to combine them: \( H(A, B) = \gamma H^q(A, B) + (1-\gamma) H^c(A, B) \), where \( \gamma \) is a coefficient balancing the importance of two kinds of features as the dissimilarity measure between primitives. Note that the Hausdorff distance is based on a greedy matching of features in the two primitives, which captures our intuition that adjacent primitives that contain similar objects inside them should have a small dissimilarity measure. Also note that using the Hausdorff distance implicitly defines a weak correspondence between interest points in the two primitives.

Sample results generated this way are displayed in Section 4 demonstrating the power of weak feature correspondence.

3. A GRAPH BASED FORMULATION

Our method is based on a temporal-event graph constructed from the video content. We define a graph \( G = (V, E, W) \), where \( V \) is the set of vertices representing preliminarily detected frames or sets of consecutive frames that are hypothesized for event primitives, together with vertices representing idle states, and \( E \) and \( W \) are the edge set and the edge weights respectively. We use \( a_i \) to represent a vertex of primitive type \( i \) and \( a_i(t) \) to represent the corresponding idle state following it in time. The goal is to map the vertices to a regular expression \((a_1 a_2^0 a_2 a_2^0 \cdots a_M a_M^0)^*\) in a temporal order, where \( M \) is the number of pre-defined event primitives.

Our key observation is that an adjacent set of primitives share similar features. However, because of the presence of idle states between detected primitives, the computation of feature distance cannot always be done between the vertices representing detected primitives. To address this problem, we give a detected primitive a unique identifier, such that all the idle states following it are readily identifiable using the same number. Specifically, a vertex \( a_i \) becomes \( a_p^i \), where \( p \) is its identifier, and its corresponding idle states are denoted \( a_0^p \). Further, the comparison of visual features now involves vertex \( a_0^q \) and \( a_1^q + 1 \), where \( q \) is the identification number for a vertex representing the next primitive in the event of interest. In other words, our graph model has long-term memory through keeping track of the originating vertex of an idle state. An idle state share both the identifier and the visual features with its corresponding detected primitive.

With this augmented notation, we are now ready to define the edge set \( E \) and the associated weights \( W \). Note that the edges only exist between adjacent frames from one to the next, such that our temporal-event graph is a sparse one that strictly obeys the temporal monotonicity of the input video.

- \( a_0^p \rightarrow a_0^q \) and \( a_0^q \rightarrow a_1^q \): weight is any positive constant \( c \);
- \( a_0^q \rightarrow a_1^q + 1 \): weight is the feature distance transformed by a sigmoid function \( s(H(a_0^p, a_1^q + 1)) \), where \( s(x) = \frac{K}{1 + e^{-x \cdot (2(x - c))}} \) and \( K \gg c \);
- \( a_0^q \rightarrow a_1^q + 1 \): weight is the transformed feature distance \( s(H(a_0^p, a_1^q + 1)) \);
- \( a_0^q \rightarrow a_1^q \) and \( a_0^q \rightarrow a_1^q + 1 \): weight is \( c \).

Weights of all other edges are set to \( \infty \). It can be verified that a path on the graph \( G \) corresponds to a temporal parsing of the video, where the video is mapped into a sequence \((a_1 a_2^0 a_2 a_2^0 \cdots a_M a_M^0)^*\), since all paths in conflict with it are simply disconnected. Among them, we pick the one with the smallest cost, i.e., \( \sum_{t=1}^{T-1} w(v_t, v_{t+1}) \) as the solution.
v is the target pattern, and shaded vertices refer to detected primitives regions and randomly selected pairs of frames that are 1 to 40 frames apart. Both consider only features inside the regions of interest.

where \( v_i \) are the vertices indexed according to time, and \( T \) is the overall time. This is achieved by invoking the Dijkstra algorithm \([10]\) with asymmetric costs. Displayed in Figure 1 is an example temporal-event graph with shorthand notations that do not include the primitive identifier for simplicity. There are only two kinds of event primitives, and the shaded vertices refer to the detected primitives. The optimal path picked up there consists of \( a_1 a_{10} a_2 a_{20} a_{220} a_{20} a_2 a_1 \), as compared to more detected events if feature distance is not used.

The only parameter that is to be learned is the soft threshold \( \kappa \) for the transformed feature distance. The estimation algorithm is straightforward: we take a set of matched primitives and let their distance \( H \) be \( h_i \) for the \( i^{th} \) pair. We empirically compute the mean \( \mu_h \) and the standard deviation \( \sigma_h \), and set \( \kappa = \mu_h + \gamma \sigma_h \). With a subset of the video data we use in experiments, we plot the distance histograms for matched primitives and random pairs in Figure 2, which verifies that the design of edge weights in our model is reasonable.

4. EXPERIMENTAL RESULTS

We test out our proposed framework using a one-hour video collected from a grocery store with 346 checkout events, which we consider in this paper as a sequence of two primitives, namely pickup and scan \((M = 2)\). Although drop-off also occurs at the end of most checkout events, the large variation in its position leads us to use only two primitives. Two sample frames together with the regions of interest for pickup and scan primitives are displayed in Figure 3, where the difficulty of event detection is made appreciable: low-resolution images, a large of number of objects without distinctive features, among other complications. Regions of interest for the two primitives are shown in rectangles.

The primitives detected for constructing the temporal-event graph are derived from frame differencing the video by taking \( D_t = |I_t - I_{t-1}| \), where \( I_t \) is a grayscale image at time \( t \). A primitive is hypothesized to be present inside a frame if \( D_t \) has more than \( n \) pixels with significant values (larger than \( 20 \)) inside its corresponding region of interest. Other parameters we used: \( \alpha = 5, P = 80, \gamma = 0.4 \). To demonstrate the power of feature consistency on real videos, we visualize in Figure 4 some of the matched pixels in the two frames as follows: a pixel in one image is linked to the pixel in the other image that it is the closest to in visual features with a yellow line.

Quantitatively, we compare the proposed method with two other baselines. The first \((B1)\) is the graph shortest path algorithm without feature correspondence, which is a reduced version of the proposed method. The second \((B2)\) is an implementation of Viterbi algorithm \([2]\) to combine primitives using the same frame differencing with optimized regions of interest. The evaluation metrics are precision and recall according to the ground truth labeling. Specifically, a detected checkout event is matched with a real checkout in the ground truth when they have reasonable overlapping, and no reuse of either is allowed. We define a correctly detected event to be one that exists in both ground truth and detection results as reported by an algorithm. Let the number of correctly detected events be \( r_1 \), the number of all detected events be \( r_2 \), and the number of all ground truth events be \( r_3 \). Then precision is \( r_1/r_2 \) and recall is \( r_1/r_3 \), which measures the quality of results from two aspects. The commonly used \( F_1 \) measure is defined as \( 2pr/(p + r) \) reflecting the overall quality of the results. We report the results in terms of these quantitative
Fig. 4. Similar interest points are matched in the two primitives of the same event. The three columns correspond to three cases. The first row is matching from the first primitive (pickup) to the second (scan), and the second row is the reverse.

Fig. 5. $F_1$ measure with respect to the threshold $n$ of the proposed method and B1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>73.5%</td>
<td>81.8%</td>
<td>77.4%</td>
</tr>
<tr>
<td>B1</td>
<td>72.4%</td>
<td>77.5%</td>
<td>74.9%</td>
</tr>
<tr>
<td>B2</td>
<td>77.9%</td>
<td>69.4%</td>
<td>73.4%</td>
</tr>
</tbody>
</table>

Table 1. Accuracy measures of different methods at the parametrization for their best $F_1$.

measures in Figure 5 and Table 1. In most cases, our proposed method shows reasonable improvements in accuracy over the baselines. Only at the end of the curves in Figure 5 when event primitives are under-detected does B1 perform slightly better because it works more conservatively.

5. CONCLUSIONS

We have approached the challenging problem of event detection in realistic low-quality videos with a graph-based method. The main point we have argued for in this paper is that using the discrepancy between visual features in hypothesized event primitives can enhance the quality of event detection results. Although experiments were carried out on data from a retail checkout environment, we believe that our framework is generally applicable to problems where certain repetitive activities are present. In the future, we are going to address the possible concurrency among events by leveraging more sophisticated spatial-temporal models.

6. REFERENCES