LETTER

A Fast Sub-Volume Search Method for Human Action Detection

Ping GUO[†a)], **Zhenjiang MIAO**[†], **Xiao-Ping ZHANG**[††], ***Nonmembers***, ***and*** **Zhe WANG**[†], ***Member***

SUMMARY This paper discusses the task of human action detection. It requires not only classifying what type the action of interest is, but also finding actions' spatial-temporal locations in a video. The novelty of this paper lies on two significant aspects. One is to introduce a new graph based representation for the search space in a video. The other is to propose a novel sub-volume search method by Minimum Cycle detection. The proposed method has a low computation complexity while maintaining a high action detection accuracy. It is evaluated on two challenging datasets which are captured in cluttered backgrounds. The proposed approach outperforms other state-of-the-art methods in most situations in terms of both Precision-Recall values and running speeds.

key words: *action detection, graph representation, Minimum Cycle detection, sub-volume search*

1. Introduction

Analysis of human actions in a video is a crucial and challenging task in many video applications such as video surveillance, content based video retrieval, and human-computer interfaces. This paper focuses on the human action detection task, which requires identifying not only which type of action occurs (action classification), but also when and where it occurs (spatial-temporal localization) in a video. Action classification has attracted many researchers and has obtained outstanding evaluation results on public datasets such as KTH [1] and Weizmann [2]. However, spatial-temporal localization is still a challenging problem, especially in uncontrolled scenes (e.g., CMU [3] and MSR [4] datasets).

The task of human action detection is often regarded as searching the sub-volume that contains the action of interest. To locate the sub-volume, human detection and tracking methods are applied [5]. These approaches are usually fast. However, cluttered background, object occlusion and camera shaking have significant impacts on the performance of human detection and tracking, especially for on-line video data [6]. Therefore, a strict reliance on human detection and tracking is not a promising solution for human action detection in a video. Recent research works have shown that sliding window matching methods through spatial and temporal sub-volume space is simple and effective [3], [4], [7], [8]. However, the computation complexity is usually high. There are $W^2 \times H^2 \times T^2$ candidate sub-volumes in a video of $W \times H \times T$ size. This requires large computations. For example, searching a $50 \times 25 \times 20$ sub-volume in a $144 \times 180 \times 200$ video requires 30 minutes on a Pentium 4 3.0 GHz processor [8]. To save computation time, the search space is reduced by down sampling [9] and using coarse-to-fine strategies [8]. However, these treatments probably lose detection accuracies. A fast sub-volume search approach based on the branch-and-bound theory is proposed in [7]. Its computation complexity is significantly reduced compared to the sliding window methods. However, it is still very slow. For instance, it costs about 20 hours to evaluate the 1-hour MSR dataset II [10].

This paper proposes a novel sub-volume search method that runs much faster than previous approaches while maintaining similar action detection accuracies. A new representation for the search space in a video is presented based on graphs. The graph structure offers a means to efficiently search sub-volumes by Minimum Cycle detection. As far as we know, both the graph representation and the sub-volume search method have not been used for human action detection in previous papers. Evaluations of the proposed method are conducted on a set of challenging natural videos. Our method outperforms other state-of-the-art methods in most situations in terms of both Precision-Recalls values and running speeds.

The rest of this paper is organized as follows. The next section introduces concepts about graphs that are used in this paper. The proposed sub-volume search method is presented in Sect. 3 and evaluations are conducted in Sect. 4. Finally, this paper is concluded in Sect. 5.

2. Preliminaries

A graph consists of two elements namely nodes and edges. Each edge can be assigned with a weight that might represent costs, lengths or capacities, etc. In a graph, if the edge has a direction, the edge is called an arrow or directed edge, and the graph is called a directed graph or digraph. A closed walk in an undirected graph or a closed directed walk in a digraph is called a cycle. A cycle can also have a weight. In this paper, the weight of a cycle is defined as follows:

$$w_{cycle}(c) = \sum_{e \in c} w(e), \tag{1}$$

where e is an edge/arrow, c is a cycle that passes e, and $w(e)$ is the edge/arrow weight.

Manuscript received June 7, 2011.
Manuscript revised August 31, 2011.
[†]The authors are with the Beijing Jiaotong University, China.
[††]The author is with the Department of Electrical & Computer Engineering, Ryerson University, China.
a) E-mail: gp1224@163.com
DOI: 10.1587/transinf.E95.D.285

A cycle is called a negative cycle if its weight is negative, similarly for a zero cycle and a positive cycle. The problem of finding the cycle who has the minimum weight in a graph/digraph is called the problem of Minimum Cycle detection in this paper. More details about graphs can be found in [11].

3. A New Sub-Volume Search Method for Human Action Detection

The general problem of interest addressed in this paper can be briefly described as follows: given a query video of an action of interest, we aim to detect its spatial-temporal locations in a test video. This task can be regarded as searching the target sub-volume that contains a similar content to the query video. Note that the size and duration of the target sub-volume may not be the same as the query video, and their contents may not be exactly the same. For instance, the action can be performed by different people in different scenes. To deal with this task, a novel graph structure is going to be presented for video representation, and a fast sub-volume search method is going to be proposed based on the Minimum Cycle detection technique.

Important notations used in the following paper are defined here:

- v_0: the query video with a size of $W_0 \times H_0 \times T_0$.
- V: a test video with a size of $W \times H \times T$.
- v: a sub-volume in V
- v^*: the target sub-volume we want to detect.

3.1 A New Graph Based Video Representation

A new weighted digraph structure is built for the video representation. A sub-volume v is considered a cuboid that consists of 12 arris. The task of locating the target sub-volume v^* equals to the task of locating its arris. As a matter of fact, only half of these arris are necessary since there are 4 arris in each surface of a cuboid but a surface can actually be decided by just 2 intersected arris. That is to say, 6 arris that each two of them intersect at a vertex are enough to represent a sub-volume. These 6 arris happen to form a cycle (Fig. 1 (a)). For description convenience, the arris are named as $A1, A2, ...A6$, and their intersection points (vertices) are named as $V1, V2, ...V6$.

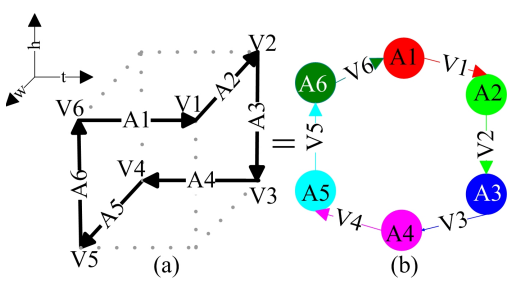

Fig. 1 A cuboid is represented by six of its arris and six of its vertexes. (a) A cuboid; (b) A cuboid can be represented by a cycle in the digraph.

To search v^* in V, its search space is represented by a digraph. Since a cuboid can be represented by 6 of its arris $A1, .A6$ and 6 of its vertexes $V1, ..., V6$, the search space of v^* is also represented by the locations of these arris and vertexes. Candidate locations of each $Ai(1 \leq i \leq 6)$ are represented by a set of nodes in the digraph (Fig. 2). Each node is assigned a location attribute of (w, h, t) in which w, h, and t are the coordinates of an arris in the width, height and time direction, respectively. For instance, the $A1$ arris has WH candidate locations, so WH nodes are developed for the $A1$ arris. Since the $A1$ arris is parallel to the t axes (Fig. 1), no coordinate values are assigned for $A1$ nodes in the t dimension (the coordinate value in the t dimension is represented by "-"), similarly for other types of nodes.

A vertex is an intersection point of two arris, which therefore is represented by an arrow (labeled by "e") that connects two nodes. Since there are six types of vertexes in Fig. 1, there are six types of arrows in the digraph that represent candidate locations of the six vertexes $V1, V2, ...V6$ respectively. Each arrow e in the digraph has a attribute of $e(x, y, z)$ which is actually the coordinate of the vertex that the arrow represents, and its coordinate can also be observed from the two nodes that it connects. Take the arrow that connects $A1(2, 1, -)$ and $A2(-, 1, 10)$ as an example, it shows that these two arris intersects at a vertex of $V1(2, 1, 10)$. The digraph is illustrated in Fig. 2 that different types of nodes and arrows are filled in different colors for a clear view.

All candidate locations of the six arris in Fig. 1 are represented by six types of nodes in the digraph, and all candidate locations of the six vertexes in Fig. 1 are represented by six types of arrows in the digraph. Therefore, all candidate locations of v^* are reflected in the digraph in Fig. 2 in terms of candidate cycles. A cycle that contains six types of arris and six types of vertexes stands for a possible location of v^*, and the task of finding v^* now equals to the task of locating the optimal "Ai"s and "Vi"s (Fig. 1 (b)). Thus, the sub-volume search task is actually to find the optimal cycle in the digraph.

In the digraph, each e is assigned a weight $n(e)$ that measures the difference between v_0 and a sub-volume v_e

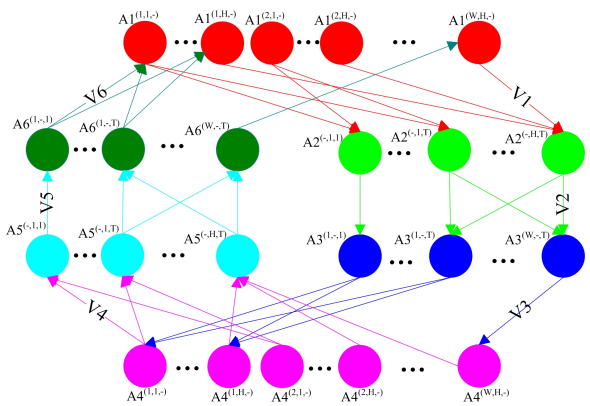

Fig. 2 A new digraph for video representation.

one of whose vertex locates at $e(x, y, z)$. The type of e in v^* should be the same as it in v_e. For instance, if e is a $V1$ type(front-top-right vertex in Fig. 1) with a coordinate of $e(x, y, z)$, then v_e's front-top-right vertex should locate at $e(x, y, z)$. The size of v_e does not stand for the size of v^*. In this paper, we simply give v_e the same size as v_0, and the size of v^* is decided by the path of c^* after Minimum Cycle detection. The value of $n(e)$ is computed by:

$$n(e) = \frac{\sum_{0 \le i < B} F_e(i) \cdot F_0(i)}{|F_e| \cdot |F_0|}, \quad (2)$$

where F_0 and F_e are B-dimensional orientation distribution vectors (the orientations=$2n\pi/B, n = 0, 1, ...B-1$) of optical flows on Harris points in v_0 and v_e, respectively.

For a cycle c, the lower $n(e)$ is, it is more likely to be the optimal cycle. Since one cycle in the digraph stands for one candidate solution of v^*, the problem of finding v^* is equivalent to finding the optimal cycle c^* that:

$$c^* = \arg \eta^* = \arg \min_{c \subseteq \Delta} \sum_{e \in c} n(e), \quad (3)$$

where Δ is the set of all cycles in the digraph and η^* is the weight of c^*.

3.2 A New Sub-Volume Search Method by Minimum Cycle Detection

The Minimum Cycle detection algorithm proposed in [12] is adopted here. Given a cycle weight $\eta = \eta^*$ in (3), give each e in the digraph a new weight $w(e)$ using the following equation, the optimal cycle c^* turns a zero cycle.

$$w(e) = n(e) - \eta/L_c, \quad (4)$$

where L_c is the length of c that equals to the number of nodes in c. Similarly, for a cycle weight $\eta > \eta^*$, a negative cycle must be processed by (4).

Therefore, the minimum weight cycle is found by negative cycle detection iteratively. Set the initial value of η to be its upperbound, and adjust it to a lower value at every iteration. The algorithm will not stop until there are no negative cycles in the graph. In our experiments, only 2-3 iterations are necessary by setting a suitable η. The algorithm is given in Fig. 3. The details of this algorithm and the Bellman-Ford algorithm in Fig. 3 can be found in [12]–[14].

Computation complexity analysis: The total computation cost of our approach includes two aspects: 1) weight setting that needs $O(E)$ computation where E is the number of edges in the digraph, and 2) Minimal cycle detection that has a computation of $O(uE)$ ($E = 6 \sim W \sim H \sim T$ and u is a linear factor with a typical value of around 15 in this paper). The total computation cost of our approach is $O(uWHT)$. This computation is much lower than pure sliding window based searches $O(W^2H^2T^2)$ and lower than the branch-and-bound based method in [4], [7], [10] ($O(W^2H^2T)$).

Input: a digraph
Initialization: η=upperbound(η^*). c^*=null
Repeat:
1: For each e, $\omega(e)$= $n(e)$- η/L_c
2: Bellman-Ford algorithm: Compute the shortest distance from the source node to any node and record its path, and returns a flag that flag=1 when there is a negative cycle.
3: **If** (flag==1)
Find any negative cycle c and update the value of η
c^*=c and $\eta = \sum_{e \in c} n(e)$
Else
Return c^*
Output: c^*.

Fig. 3 Minimum Cycle Detection in a digraph.

4. Experimental Results

Our approach is evaluated on the CMU dataset [3] and the MSR action dataset II [4]. The experimental results show that our approach outperforms other state-of-the-art papers for action detection in terms of both Precision-Recall values and running speeds in most cases.

Videos in the CMU dataset are captured in crowded backgrounds using a hand held camera. Five actions are included namely "one-handed wave", "two-handed wave", "push elevator button", "pick up" and "jumping jacks". The total dataset has approximately 20 minutes of video containing 114 actions of interest. Each action has about 14-30 testing instances. Videos in this dataset are downscaled to 160×120 in resolution. Each category has a binary template video as its query video. The duration of templates ranges from 9 frames to 22 frames and with a typical size of 60×80.

The action detection results are compared with ground truth that manually labeled by ourselves. If the time interval of two detected instances is small (less than 20 frames here), they are connected to be one instance. If a detection result has more than 50% overlap with a labeled ground truth, it is considered to be a correct decision. The average Precision-Recall (P-R) values are given in Table 1 with a comparison to previous papers. Precision=$TP/(TP + FP)$ and Recall=TP/NP where TP is the number of true positives, FP is the number of false positives and NP is the total number of positives of ground truth.

Our computational cost is the lowest. The computation complexity of paper [3] and [5] is $W^2H^2T^2$. The computation complexity of our approach is $O(uHWT)$. Paper [15] reports that each 100 frame clip needs about 2-3 minutes on an Intel 2.5 GHz PC. This searching time is about 30 times of the video length. For the CMU dataset, our computation time is less than 6 times of the video length on an Intel 2.83 GHz processor (less than 7 times of the video length on an Intel 2.5 GHz PC as in [15]). Paper [16] reports that searching a $50 \times 25 \times 20$ template in a $144 \times 180 \times 200$ video requires 36 seconds on a 2.3 GHz processor. Using

Table 1 Average precision-recalls on the CMU dataset.

P-R	R=0.2	R=0.4	R=0.6	R=0.8	Computations
This paper	0.9	0.7	0.52	0.5	$O(uWHT)$
[3]	0.9	0.5	0.25	0.15	$O(W^2H^2T^2)$
[5]	1.0	0.8	0.52	0.4	$O(W^2H^2T^2)$
[15]	0.9	0.7	0.5	0.3	-

Table 2 Average precision-recalls on the MSR action dataset II.

P-R	R=0.2	R=0.4	R=0.6	R=0.8	Computations
This paper	0.7	0.43	0.43	0.3	8 h
[4]	0.54	0.3	0.2	-	20 h [10]

our approach, our computation time is around 25 seconds on an Intel 2.83 GHz processor (around 31 seconds on an Intel 2.3 GHz PC as in [16]).

In addition,we can see that our P-R values are better than other papers in most situations. The main reason is that we search for all candidate locations of v^* but these papers are not. Due to the computational complexity of exhaustive search, paper [3] and [15] search for the best location of v^* in only one scale, and paper [5] locates v^* with the help of a human head detection technique which is not reliable in complex environments. Compared to these works, our approach provides a fast sub-volume search way and also improves the P-R values in most situations.

The MSR action dataset II contains three actions: "boxing", "handwaving" and "handclapping". This dataset is also taken in realistic scenarios. It includes 54 video sequences, has a total duration of around 45 minutes and contains 203 action instances in all. Each video has a size of 320×240.

The MSR action dataset II dose not provide template videos. For each action type, a video clip is segmented from the KTH [1] dataset. The experimental settings are the same as the above test on the CMU dataset, and the P-R values are shown in Table 2. Our P-R values are better than [4]. The reason we think is that paper [4] use GMM to model the distribution of spatial-temporal interest points based on KTH videos, but due to the camera shaking and zooming, noises are detected as interest points in the training stage. In this paper, no training stage is required. Thus, we only need to select one clean video as the template video and thus gain a better result. In addition, the total testing time of the whole MSR action dataset II is about 8 hours, which is much faster than [4] (20 hours reported in [10]).

5. Conclusions

In this paper, a novel sub-volume search method is presented based on the Minimum Cycle detection technique. A new weighted digraph is built to represent the search space in a video. Based on the digraph structure, the proposed algorithm has a computation complexity of $O(uHWT)$ which is significantly lower than traditional sliding window based methods ($O(H^2W^2T^2)$) [8] and other state-of-the-art methods such as the brand-and-bound based approach ($O(H^2W^2T)$) in [4], [7]. Experiments are conducted on the CMU dataset and the MSR action dataset II which are two challenging datasets captured in crowded scenes. Experimental results show that not only the computational complexity of the proposed method is the lowest, but also the precision-recall values of our method are as good as or even better than other methods in most cases.

References

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," Proc. International Conference on Pattern Recognition, vol.3, pp.32–36, 2004.
[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," Proc. International Conference on Computer Vision, vol.2, pp.1395–1402, 2005.
[3] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," Proc. International Conference on Computer Vision, pp.1–8, 2007.
[4] L. Cao, Z. Liu, and T.S. Huang, "Cross-dataset action detection," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1998–2005, 2010.
[5] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T.S. Huang, "Action detection in complex scenes with spatial and temporal ambiguities," Proc. International Conference on Computer Vision, pp.128–135, 2009.
[6] S. Paisitkriangkrai, C. Shen, and J. Zhang, "Fast pedestrian detection using a cascade of boosted covariance features," IEEE Trans. Circuits Syst. Video Technol., vol.18, no.8, pp.1140–1151, 2008.
[7] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.2442–2449, 2009.
[8] E. Shechtman and M. Irani, "Space-time behavior based correlation-OR -How to tell if two underlying motion fields are similar without computing them?," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.11, pp.2045–2056, 2007.
[9] H.J. Seo and P. Milanfar, "Action recognition from one example," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.5, pp.867–882, 2011.
[10] N.A. Goussies, Z. Liu, and J. Yuan, "Efficient search of Top-K video subvolumes for multi-instance action detection," Proc. International Conference on Multimedia and Expo, pp.328–333, 2010.
[11] D. B£West, Introduction to Graph Theory, Second ed., Published by Prentice Hall, 2001.
[12] I.H. Jermyn and H. Ishikawa, "Globally optimal regions and boundaries as minimum ratio weight cycles," IEEE Trans. Pattern Anal. Mach. Intell., vol.23, no.10, pp.1075–1088, 2001.
[13] R.E. Bellman, "On a routing problem," Quarterly of Applied Mathematics, vol.16, no.1, pp.87–90, 1958.
[14] L.R. Ford and D.R. Fulkerson, Flows in Networks, Princeton University Press, 1962.
[15] B. Yao and S.-C. Zhu, "Learning deformable action templates from cluttered videos," Proc. International Conference on Computer Vision, pp.1507–1514, 2009.
[16] K.G. Derpanis, M. Sizintsev, K. Cannons, and R.P. Wildes, "Efficient action spotting based on a spacetime oriented structure representation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1990–1997, 2010.