DETECTING HUMAN ACTION IN ACTIVE VIDEO

Hao Jiang, Ze-Nian Li and Mark S. Drew

School of Computing Science, Simon Fraser University, Vancouver, BC, Canada V5A 1S6

ABSTRACT

We propose a novel scheme to detect human actions in *active* videos. Active videos such as movies or sports broadcasting are taken purposively by "clever" photographers. They are object and action oriented and usually involve complex camera motions. Detecting actions in active videos is both important and challenging. We study a three-step scheme to detect complex human actions in such videos. The proposed method first locates potential objects and removes clutter with a composite filter scheme. The detected object candidates in successive frames are then associated to form object trajectories based on a consistent labeling formulation, and solved with belief propagation. Finally, specific human actions are detected in video with a linear programming matching approach that can efficiently deal with matching problems having a large target point set. The proposed method has been successfully applied in action detection for general videos and TV hockey games.

1. INTRODUCTION

Detecting human actions has been one of the most interesting and challenging problems in multimedia applications. It has many applications such as human computer interaction, surveillance etc. Detecting human actions in controlled environments has been quite successful and many real-time systems have been built, such as the MIT Kidsroom [1] and systems using magnetic sensors [2]. On the other hand, detecting human actions in videos taken in an uncontrolled environment is still an unsolved problem. Most current methods detect actions in videos taken by static cameras, for which background subtraction can be applied to remove part of the background clutter, and silhouettes are used for action detection. Other schemes detect human actions by matching specific motion patterns [3]. Appearance based schemes have also been studied for detecting human body postures in static images [4, 5] and body part configurations in videos [6]. Although different schemes have been proposed, detecting human actions in complex environments involving multiple human subjects is still an unsolved problem.

In this paper, we study action detection in active videos [7]. Active video is shot purposively by a smart photographer and usually involves complex camera motions such as pan/tilt, zooming and position (or view) changes. One example of such videos is TV broadcast of a sports program. Active videos also appear more and more in surveillance videos generated by automatic pan-tilt cameras. Comparing to videos generated by still cameras, active videos are more similar to the world seen from people's eyes. They are object oriented and full of action. Active video is a good information source for action detection because of its unique characteristics. At the same time, it also presents many new challenges because of complex camera motions.

We present a novel three-step approach to detect human actions in active videos which may involve multiple subjects. Human action is defined as a sequence of body postures with a specific timing constraint. In the first step, we detect potential locations of subjects with a composite filtering method [8], which removes most of the clutter and therefore greatly speeds up the whole detection process. In the second step, we find the correspondence of the objects in successive video frames. We solve the object correspondence problem using a consistent labeling approach with pairwise spatial constraint and object disappearing inference. The labeling problem is solved with belief propagation [9]. Different from traditional tracking approaches, this method is computationally more efficient and can deal with cases of objects moving into or out of the scene robustly. After the first two steps, we establish a set of object trajectories through time. We propose a robust linear programming matching scheme to match each object sequence to specific template sequences based on edge features. The proposed linear programming approach has a unique property of using small sets of basis target points in the optimization process. For each template feature point, the basis set usually has a much smaller size than that of the whole target point set, which makes the scheme well suited for matching problems involving a large number of target points. The detail matching using linear programming is deformable and therefore enables the method to detect actions in which objects have large distortion from the template.

2. DETECTING ACTIONS IN ACTIVE VIDEO

2.1. Object Pre-localization with Composite Filtering

We propose a correlation based method to locate potential objects and remove most of the clutter. The naive scheme of matching each of the possible shapes to human targets is infeasible in real applications. So we make use of a composite filtering approach that matches targets using a single template.

We assume there are k possible appearances of an object, represented as images $I_1, I_2, ..., I_k$ with resolution $m \times n$. I_i appears with probability p_i. We would like to use a single $m \times n$ template P to detect objects with different appearances and minimize the mean square error: $\min_P \sum_{i=1}^{k} ||P - I_i||^2$, where $|| \cdot ||$ is the Frobenius norm. By calculating the derivatives of the objective function with respect to each element of P and setting them to zero, we have $P = \sum_{i=1}^{k} p_i I_i$, i.e., using mean square error, the optimal composite template is the expectation of the appearance of an object [8]. In real implementations, $\{I_i\}$ is a large set of exemplars. To detect an object in image I, we sweep P across the image and calculate the residue errors between the template and the image at each position. In a more efficient implementation, because the template is fixed, we simply need to calculate the correlation map $J * G_\sigma - 2I * Q_\sigma$, in which $*$ is the convolution operation; each element of J is the square of the corresponding element in image I; G_σ is a Gaussian filter kernel with standard deviation σ; Q_σ is obtained by flipping P in the x and y direction and filtering by a Gaussian with standard deviation σ; typically $\sigma = 5$. To reduce influence of clothing changes, edge maps of template images are first converted to distance transformed grayscale images and then averaged to generate the composite template. Target images are also converted to distance transformed images in composite filtering. The local valleys of the correlation map are selected as potential positions of objects. There are usually only a small number of these local minima in the correlation map, which therefore greatly reduces the searching space during further detail matching. The coarse object detection process usually also generates many false positives but this poses no problem for action detection since the final, detail matching, step will remove the false detections.

2.2. Matching Objects in Successive Frames

For action detection, we further need to set up correspondences of objects in successive video frames, which can be used to generate a trajectory for each potential object in a video sequence. Another purpose of matching objects in successive frames is to remove any non-consistent false detection from the first step. Fig. 1 shows a scenario of matching objects (hockey players) in two successive frames with object disappearing inference. At the same time, inconsistent false detections from step one are eliminated.

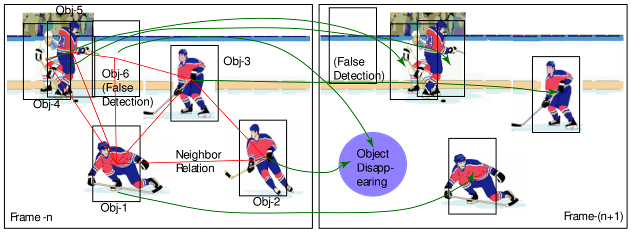


Fig. 1. Matching objects in successive frames.

Since the pre-localization step has high recall in detection, objects almost always appear in the detected positions in each of the frames. Instead of tracking, we formulate the inter-frame object correspondence problem as a *consistent labeling problem*. In two successive video frames, the objects detected in the first frame are treated as sites and the objects detected in the second image are labels. The label set also contains a label to indicate an object disappearing in the second image. Each site is enforced to receive a single label and each label assignment has a given cost. The labeling problem can thus be formulated as the following energy minimization problem:

$$\min_{l} \left\{ \sum_{s \in B} g(s, l_s) + \mu \sum_{s_a \text{ and } s_b \text{ are neighbors}} d(s_a, l_{s_a}, s_b, l_{s_b}) \right\}$$

where B contains the objects in the first image; $g(s, l_s)$ is the cost of labeling object s with label l_s; the function d introduces the pairwise constraint; μ is a smoothing coefficient. $g(s, l_s) = ||\mathbf{h}(s) - \mathbf{h}(l_s)||$ if l_s indicates an object in the second image, otherwise a constant C, where $\mathbf{h}(\mathbf{s})$ is the histogram vector for object s and $|| \cdot ||$ is the L_1 norm. $d(s_a, l_{s_a}, s_b, l_{s_b}) = ||\mathbf{p}(s_a) - \mathbf{p}(s_b) - \mathbf{p}(l_{s_a}) + \mathbf{p}(l_{s_b})||$ if l_{s_a} and $l_{s_b}$are both non-missing labels; otherwise d vanishes. Here $\mathbf{p}(\cdot)$ is the center coordinate of an object and $|| \cdot ||$ is the cityblock norm. We use belief propagation [9] to solve the labeling problem.

2.3. Video Sequence Matching and Action Detection

From the first two steps, we detect potential objects and their corresponding trajectories in time. We would like to detect actions from these hypothesized object image sequences. The basic idea of action recognition is to match a sequence of corresponding objects, detected in the first two steps, with a template sequence. The template sequence is swept *across time* and compared with all the object sequences at each instant. An action is found if the matching error is sufficiently small.

We use a deformable template matching scheme based on linear programming to match each template and object image pair in template sequence to object sequence matching. The linear programming approach can solve large scale matching problems for which other schemes such as BP become too slow. Each posture template in the action template sequence is a graph structure in which nodes are randomly selected image edge points and graph edges indicate neighbor relations. We choose image edge features to make the method insensitive to human clothing. Edge maps of the template and target images are first turned to grayscale images by fully-rectified distance transform. We then apply log-polar transforms centered on selected points in the template and target images to generate features for matching. The log-polar transform blurs the peripheral view to make features insensitive to small shifts.

As outlined in [10] by Jiang et al., a matching problem with pairwise constraint using L_1 norm can be formulated as the following mixed integer linear programming problem:

$$\min \left\{ \sum_{\mathbf{s} \in S} \sum_{\mathbf{j} \in \mathcal{T}} c(\mathbf{s}, \mathbf{j}) \cdot \xi_{\mathbf{s},\mathbf{j}} + \lambda \sum_{\{\mathbf{p},\mathbf{q}\} \in \mathcal{N}} (x^+_{\mathbf{p},\mathbf{q}} + x^-_{\mathbf{p},\mathbf{q}} + y^+_{\mathbf{p},\mathbf{q}} + y^-_{\mathbf{p},\mathbf{q}}) \right\}$$

$$\text{with constraints: } \sum_{\mathbf{j} \in \mathcal{T}} \xi_{\mathbf{s},\mathbf{j}} = 1, \forall \mathbf{s} \in S,$$

$$\sum_{\mathbf{j} \in \mathcal{T}} \xi_{\mathbf{s},\mathbf{j}} \cdot \phi_x(\mathbf{j}) = x_\mathbf{s}, \ \sum_{\mathbf{j} \in \mathcal{T}} \xi_{\mathbf{s},\mathbf{j}} \cdot \phi_y(\mathbf{j}) = y_\mathbf{s}, \forall \mathbf{s} \in S,$$

$$x_\mathbf{p} - x_\mathbf{q} - \phi_x(\mathbf{p}) + \phi_x(\mathbf{q}) = x^+_{\mathbf{p},\mathbf{q}} - x^-_{\mathbf{p},\mathbf{q}},$$

$$y_\mathbf{p} - y_\mathbf{q} - \phi_y(\mathbf{p}) + \phi_y(\mathbf{q}) = y^+_{\mathbf{p},\mathbf{q}} - y^-_{\mathbf{p},\mathbf{q}}, \forall \ \{\mathbf{p},\mathbf{q}\} \in \mathcal{N},$$

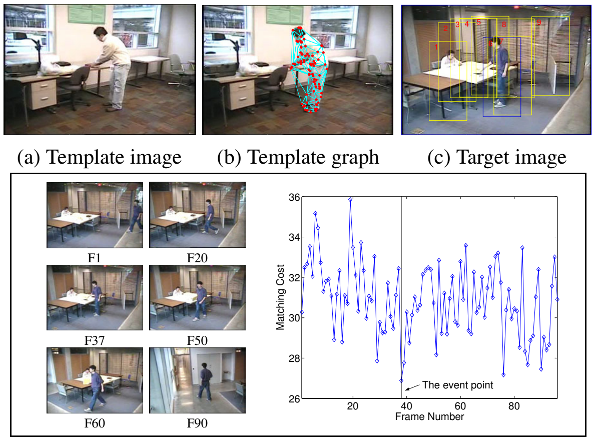
$$\xi_{\mathbf{s},\mathbf{j}} = 0, 1, x^+_{\mathbf{p},\mathbf{q}}, x^-_{\mathbf{p},\mathbf{q}}, y^+_{\mathbf{p},\mathbf{q}}, y^-_{\mathbf{p},\mathbf{q}} \geq 0$$

where $(x_\mathbf{s}, y_\mathbf{s})$ in the target image is matched to $\mathbf{s}$ in the template image; $c(\mathbf{s}, \mathbf{j})$ is the cost of matching point $\mathbf{s}$ with target point $\mathbf{j}$; $\xi_{\mathbf{s},\mathbf{j}}$ is 1 if $\mathbf{s}$ matches $\mathbf{j}$ and is zero otherwise; S and $\mathcal{T}$ are point sets in template and target images respectively; $\mathcal{N}$ contains all the neighboring point pairs in the template image; λ is a smoothing coefficient; functions $\phi_x(\cdot)$ and $\phi_y(\cdot)$ extract x and y components respectively. Instead of directly solving the mixed integer programming problem, we drop the binary constraint and solve a linear programming relaxation. For each template point $\mathbf{s}$, the linear programming relaxation replaces the cost function $c(\mathbf{s}, \mathbf{j})$ with the its lower convex hull. Therefore, we can choose the most compact set of target points for each site $\mathbf{s}$, corresponding to the coordinates of lower convex hull vertices of the cost surface $c(\mathbf{s}, \mathbf{j})$, to form a basis so that each candidate target point can be represented as a linear combination of these basis target points. Because the number of these basis points is much smaller than the original candidate target points, the efficiency of searching increases considerably. Another property of the method is that by using schemes such as the simplex method, the searching for each template point involves only at most three non-zero-weight basis target points at each stage, in a fast energy descent manner. Thus the scheme converges rapidly. We use a successive convexification [10] method to solve the matching problem, in which we construct linear programs recursively based on the previous searching result and gradually shrink the trust regions for each site systematically.

We use the following quantities to measure the difference between the template and the target object in video sequence matching. The first measure D is defined as the average of pairwise length changes from the template to the target. To compensate for the global deformation, a global affine transform is first estimated based on the matching and then applied to the template points. The length changes are further normalized with respect to the edge lengths of the template. The second measure M is the average matching cost based on the log-polar features. The total matching cost is simply defined as $M + \alpha D$, where α has a typical value 10 if image pixels are in [0,255]. Experiments show that only about 100 randomly selected feature points are needed in calculating D and M. The matching score for a video sequence is the average matching score for all the template to video frame pairs.

3. EXPERIMENTAL RESULTS

We first applied the proposed three-step method to a staged surveillance video sequence with camera motion and complex background. We detect the event of picking up a book. We use 70 training images of a different walking subject taken in a single color background. Edge maps are first extracted and distance transformed to generate grayscale counterparts. These images are averaged to generate the composite filter template. We detect potential objects and further generate correspondences in successive frames based on the first two steps of the proposed scheme. Fig. 2 (a) is the key posture template image from a different subject. The graph template as shown in Fig. 2 (b) is used in this experiment for linear programming detail matching and the minimum object matching scores in each video frame are shown in Fig. 2 (d). The instant of the event corresponds to the lowest matching score and is located correctly based on the proposed approach, as shown in Fig.2 (c).


(a) Template image (b) Template graph (c) Target image

(d) Matching costs

Fig. 2. Finding an action in staged surveillance video.

Hockey is a fast pacing multi-player sports game. TV broadcast of a hockey game is a standard example of active video. Detecting action in hockey videos is a challenging and useful application. We use two hundred randomly selected hockey player images to construct the composite filter. Composite filtering is again based on distance transformed images to reduce the influence of clothing. The proposed correlation based method is then used to pre-locate each hockey player in the video. The detected regions are then correlated to generate trajectories based on belief propagation in the second stage of the proposed scheme. Fig. 3 shows sample results of hockey player detection and tracking results based on the first two steps of the proposed scheme. The correspondence scheme works very well even for cases involving partial occlusions. Fig. 4 shows the trajectories generated based on the two-step detection method from a longer video sequence. The points in Fig. 4 are object center locations and corresponding objects' center points are connected. Apart from some short trajectories generated by false positive detections, the proposed scheme reliably tracks multiple hockey players in fast video sequences. Based on the detection and tracking result, we use linear programming detail matching to find specific actions of hockey players. Fig. 5 shows the action detection result for a shooting action with a single posture template in a 900-frame video. The template from a different section of the game is shown overlapped at the upper-left corner of Fig. 5 (a). There is one shooting action in the sequence and the linear programming scheme locates the instance of the action at the top of the shortlist ranked by matching scores. The matching score for each video frame corresponds to the lowest object matching score in the frame. The black rectangle with thicker boundary shows the matching target with minimum score in each frame. The shortlist of hockey player image patches based on their matching scores is also shown in Fig. 6. Another experiment used a two-frame posture template as shown in Fig. 7(a) to detect similar actions in a 1000-frame video. The shortlist of frames at the starting points of the similar actions as the template sequence is shown in Fig. 7, ranked by matching scores for each target sequence. The starting and ending postures of of the hockey player sequences, based on their matching scores, are also shortlisted in Fig. 8. The proposed method successfully locates actions similar to the template sequence in the top ranks of the shortlist.

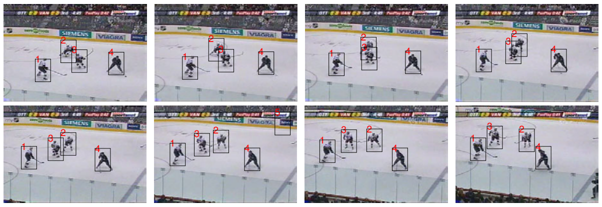


Fig. 3. Finding and tracking hockey players by steps I and II.

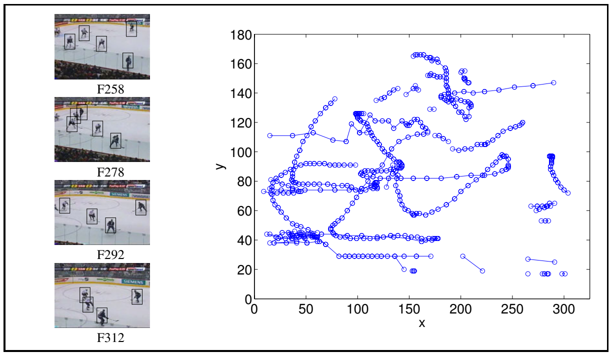
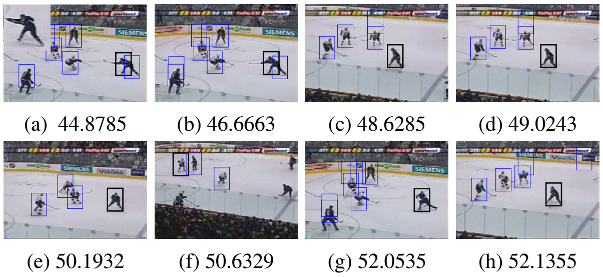


Fig. 4. Trajectories of objects in 55 frames.



(a) 44.8785 (b) 46.6663 (c) 48.6285 (d) 49.0243

(e) 50.1932 (f) 50.6329 (g) 52.0535 (h) 52.1355

Fig. 5. Shortlist of shooting action detection with linear programming. Numbers show matching scores.

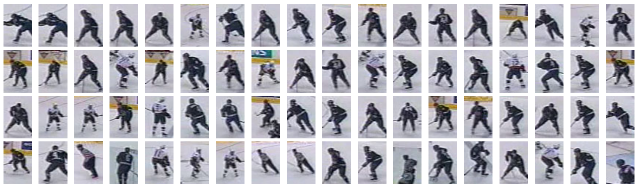
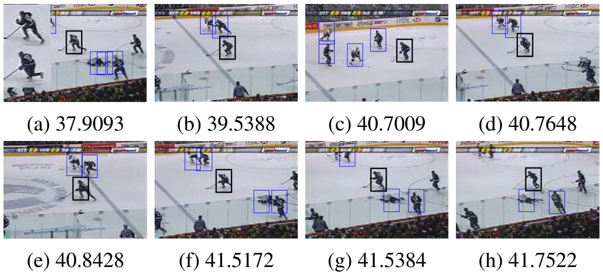


Fig. 6. Shortlist of the first 72 matched hockey player action key postures.



(a) 37.9093 (b) 39.5388 (c) 40.7009 (d) 40.7648

(e) 40.8428 (f) 41.5172 (g) 41.5384 (h) 41.7522

Fig. 7. Shortlist of action detection with linear programming. Numbers show matching scores.

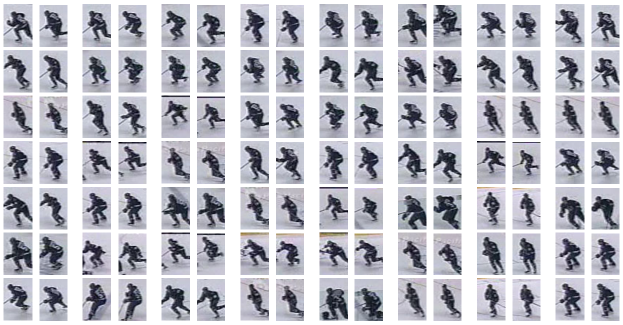


Fig. 8. Shortlist of starting and ending postures for the first 56 matched hockey player action sequences.

4. CONCLUSION

In this paper, we propose a novel three-step human action detection scheme for active videos. The proposed scheme first detects potential human targets and removes most of the background clutter, using an efficient composite filtering approach. The correspondences of objects in sequence of frames are found in the second step, using belief propagation. This method is fast and much more robust then traditional tracking schemes. We detect specific human actions by matching templates to the video sequences using a linear programming method. The linear programming method is robust and more efficient for large target set matching problem than previous methods. We successfully applied the proposed scheme in general videos and TV hockey games. In future work, we will study fusing other clues such as camera motions for action event detection.

References

[1] The Kidsroom. http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html

[2] L. Emering and B. Herbelin, “Body Gesture Recognition and Action Response”, Handbook of Virtual Humans, Wiley 2004, pp.287-302.

[3] E. Shechtman and M. Irani, “Space-Time Behavior Based Correlation”, CVPR 2005.

[4] S. Carlsson and J. Sullivan, “Action recognition by shape matching to key frames”, IEEE Computer Society Workshop on Models versus Exemplars in Computer Vision, 2001.

[5] G. Mori and J. Malik, “Estimating human body configurations using shape context matching”, ECCV, LNCS 2352, pp. 666–680, 2002.

[6] D. Ramanan, D. A. Forsyth, and A. Zisserman. “Strike a Pose: Tracking People by Finding Stylized Poses”, CVPR 2005.

[7] Y. Lu and Z.N. Li, “Active video object extraction”, IEEE Conf. on Multimedia and Expo (ICME 2004), 2004.

[8] A. Mahaloanobis, B.V.K. Vijaya Kumar et cl., “Unconstrained Correlation Filters”, Appl. Opt., 33:3751-3759, 1994.

[9] Y. Weiss and W.T. Freeman. “On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs”, IEEE Trans. on Information Theory, 47(2):723-735, 2001.

[10] H. Jiang, Z.N. Li, and M.S. Drew, “Linear Programming for Matching in Human Body Gesture Recognition”, AMFG 2005.