Detection of Fence Climbing from Monocular Video

Elden Yu and J.K. Aggarwal
Computer and Vision Research Center, Department of ECE
The University of Texas at Austin, Austin TX 78712, USA
elden.yu@gmail.com, aggarwaljk@mail.utexas.edu

Abstract

This paper presents a system that detects humans climbing fences. After extracting a binary blob contour, the system models the human with an extended star-skeleton representation consisting of the highest contour point and the blob centroid as the two stars. Distances between stars and contour points are computed and smoothed to detect local maximum points. The system then finds certain predicates to form a feature vector for each frame. To analyze the resulting time series, a block based discrete Hidden Markov Model (HMM) is built with predefined action classes {walk, climb up, cross over, drop down} as the state blocks. Each block contains a subset of hidden states and is trained independently to improve the model estimation accuracy with a limited number of sequences. The detection is achieved by decoding the state sequence of the block based HMM. The experiments on image sequences of human climbing fences yield excellent results.

1. Introduction

Human motion analysis is a significant research topic in the computer vision community. General reviews [1, 2] provide a detailed documentation of research in this field. By using the relationship between humans and the involved environment, we may broadly divide human motion into three types: (a) single person motion, such as walking, bending and sitting, as described in [3, 4], where the action is performed by a single person and involves no interaction with the environment; (b) interactions between people, such as hugging and punching, as presented in [5, 6]; (c) activities involving inanimate objects, such as climbing and opening a file cabinet in [7].

Our research focuses on detecting a person climbing a fence. Fences can be of a variety of different sizes and types. Climbing motion can differ considerably as well as the type of fences. In our case, we consider only simple fences such as chain link and iron bar fences. We have chosen flat-topped fences about 5 feet high. Our algorithm comprises of two major parts: human body representation and time series analysis. We use an extended star-skeleton representation for extracting predicates to form a feature vector for every frame. The resulting time series is decoded in a trained HMM to obtain the state sequence. We have designed a block based HMM training method so that a limited number of sequences is sufficient to yield satisfactory testing accuracy.

The paper is organized as follows: we introduce the overall system design in section 2, describe the segmentation in section 2.1, explain our extended star-skeleton representation in section 2.2, describe the method of feature selection and extraction in section 2.3 and train a block based discrete HMM in section 2.4. We also review a few relevant works in each section. Experimental results are presented in section 3, and conclusions are given in section 4.

2. System overview and details

The overall system architecture is described in four consecutive modules as detailed below.

2.1. Foreground segmentation

Working with outdoor image sequences poses many challenges for successful segmentation. We adopt the common approach to build a statistical background model for background subtraction, where each pixel follows a normal distribution. This is followed by thresholding and binary morphological operations. For simplicity, we assume the sequence contains only one person. To ensure there is just one blob extracted, we

Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

extract only the largest blob and ignore all smaller ones after connected component analysis.

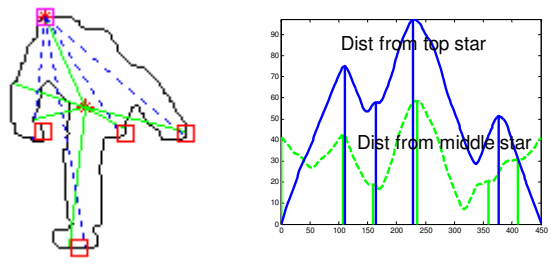
2.2. Body representation

After computing the blob contour with a contour following algorithm, we represent the human body with two stars and a small number of extreme points on the contour. Fujiyoshi [8] built a star-skeleton model to represent the human body, where the blob centroid is the star. The distances between contour points and the star are computed and smoothed to extract points where the distance is the largest in its contour neighborhood. He then computes features from such a star-skeleton representation to obtain a time series for differentiating walking from running. A similar model is applied in [9].

However, the above representation [8, 9] has a few drawbacks. First, noise smoothing involves a trade-off between noise reduction and preservation of details, which makes the detection accuracy highly dependent on the choice of the smoothing parameter. Second, detection errors occur with unwanted contour parts, such as the shoulders in our case, which are indeed the local maximums for certain poses, but not the extreme points that we wish to obtain in our application. Finally, for our climbing action, the blob centroid as the only star would detect the extreme points at an offset location of the desired position, especially for hands. This is because the blob centroid is in the middle of the torso, which doesn't view the hands sticking out as apparently as feet.

In order to alleviate these drawbacks of using the blob centroid as the only star, we use the highest contour point as a second star and then compute the local maximum points in the contour with respect to this second star as well. The resulting extreme points with respect to the two stars are then paired up by proximity to compute their middle contour points as hand/foot candidates. When the numbers of extreme points from the two stars are different, we simply ignore those that cannot be paired up.

In Figure 1 (a), the middle star is the blob centroid, the top star with a surrounding square is the highest contour point, the solid and dash lines represent connections from the middle or top star to candidate extreme points respectively, and the squares are the detected extreme points. In Figure 1 (b) two sets of distances are displayed for local maximum detection and grouping in pairs. One extreme point from the middle star is left unpaired, as it's not recognized by the top star as an extreme point.



(a) (b)

Figure 1. Extended star-skeleton

2.3. Feature selection and extraction

After all extreme points are extracted, they can be associated with head, left foot, right foot, left hand or right hand. Fujiyoshi [8] regards lower extreme points as feet, and the highest as head. Petkovic *et al.* [9] consider only a subset of extreme points. It is not necessary for our model construction to configure the extreme points as human body parts, as available information provides strong enough cues.

We use three cues for our identification of human climbing fences. The first one is the coordinates of the blob centroid. A change in the y-coordinate indicates a possible climbing action. The second cue is the local maximum point configuration from section 2.2, which could be approximately thought of as a human limb shape. The third cue is the height of the fence, which could be obtained by doing a simple horizontal line extraction or be given directly as domain knowledge.

We compute five features as shown in Table 1.

Table 1. Features

Feature	Explanation
1	centroid x-coordinate changes?
2	centroid y-coordinate up, down, or not
3	centroid y-coordinate above fence?
4	2 or more extreme points above fence?
5	2 or less extreme points under fence?

2.4. HMM

After computing a feature vector for every frame, we obtain a time series by converting the feature vector into a discrete symbol by building a codebook, where each symbol represents a feature vector with unique value. Hidden Markov Model (HMM) is an effective technique to analyze such sequences. Petkovic *et al.* [9] train a discrete HMM for each type of stroke with various states and observation symbol sizes. Starner and Pentland [10] train a four state HMM for each word of the gesture vocabulary. All these works

require manual segmentation of the sequences to classify each action.

Each sequence contains a few basic actions. We consider only four in our experiments: *walking, climbing up, crossing over the top of the fence, and dropping down*. The generalization to include more actions is straightforward. So our desired HMM will have the four actions as the hidden states. We use the Viterbi algorithm of the HMM decoding problem to infer the action sequences.

The most intuitive implementation is to build a four state discrete HMM, where each state is a basic action. We denote this model as our baseline model A. Taking into consideration the fact that each basic action itself has some action phases, we represent each basic action with a set of hidden states, instead of only one state as in A. We refer to this model as our baseline model B. Since we have a limited number of sequences, we might not have sufficient data to train and test our model B. We reduce the model parameters by constructing a block based HMM, as shown in Figure 2, where each block represents one of the four actions in consideration and includes the associated subset of hidden states. We denote this model as C. The difference between model B and C lies only in their training method. The idea is to train the model at a finer level incorporating domain knowledge. We will prove by experiments that regular baseline models A and B are inferior to our proposed model C.

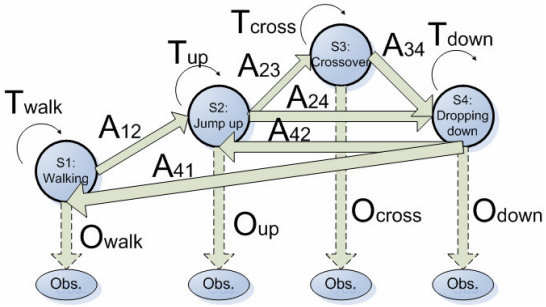


Figure 2. The block HMM

In Figure 2, each circle stands for either a hidden state in model A, a set of hidden states in model B, or a state block in model C. Again, we mention that a set of hidden states is the same as a state block. The ovals always represent a discrete observation symbol. The thin loop arrows represent the probability of staying in a current state in model A, or the local state transition matrix within that block of hidden states in models B and C. The solid arrows represent the transition probability between states or transition probability matrix between blocks. The dash arrows represent the observation probability matrix.

After training a discrete HMM for each of the four basic actions, we have four sets of HMM parameters $\{ p_i \; T_i \; O_i \}$ where p_i is prior state distribution vector, T_i is state transition matrix, and O_i is observation distribution matrix, $i \in$ {walk, up, cross, down}. We concatenate these parameters as proper parameter blocks in the model C, as shown in the following equations.

$$prior = [\, p_{walk} \quad 0 \quad 0 \quad 0 \,] \tag{1}$$

$$trans = \begin{bmatrix} a_1 T_{walk} & A_{12} & 0 & 0 \\ 0 & a_2 T_{up} & A_{23} & A_{24} \\ 0 & 0 & a_3 T_{cross} & A_{34} \\ A_{41} & A_{42} & 0 & a_4 T_{down} \end{bmatrix} \tag{2}$$

$$obs = [\, O_{walk}; O_{up}; O_{cross}; O_{down} \,] \tag{3}$$

Note that the zeros mean appropriate size matrices with all zero values. Comparing the equations (1), (2) and (3) against Figure 2, we have equation (1) as we always start the sequence by walking, equation (2) as we assume only some transitions between actions are possible, where *A* blocks are the random matrices with fixed weight and have the same meaning as in Figure 2, and *a* values mean weights to sum every row up to 1, equation (3) as every block state could observe all observation symbols, so we just concatenate by rows. The model C could be used as an initial estimate in training to get a refined version, referred as model D.

To detect a climbing action, we first decode the observation sequences into hidden state sequences, and then generalize them into block sequences. Climbing is determined if there is a consecutive triple {up, cross, down} where each lasts a long enough frame period.

3. Experiment results

We collected climbing sequences from 4 men and 1 woman in two scenarios, shown in Figure 3. One is with an iron bar fence near a pool and the other is with a chain link fence dividing playgrounds. Each frame is a 24 bit RGB bitmap file of size 360 by 240 pixels. Each sequence contains both walking and climbing. The camera view is mainly a front or back view with respect to the fences. Figure 4 illustrate intermediate results of body representation (resized for display).

Since there are only 2*3*2*2*2 = 48 possible combinations of feature values, we have a codebook of size 48. We have 29 sequences, of which we use 19 for training and 10 for testing. Each sequence has a length

between 300 to 600 frames with a frame rate of 30 FPS. Frames in sequences are manually labeled as one of the basic action class.

Figure 3. Iron bar and chain link fences

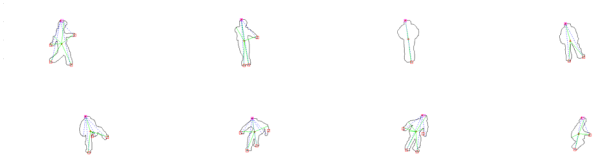


Figure 4. Extended star-skeleton

Table 2. HMMs (A,B,C,D) decoding accuracy

Models (# of hidden states)	Frame accuracy		Sequence accuracy
	mean	std	
Baseline A (4)	24%	11%	0/10
Baseline B (4*3)	11%	8%	0/10
Block model C (12)	81%	14%	8/10
Refined D (12)	85%	16%	8/10

As mentioned in the section 2.4, we train up to four discrete HMM models: A, B, C and D. The testing accuracy is reported in Table 2. Here frame accuracy is defined as the percentage of frames whose hidden action classes are correctly decoded, the sequence accuracy implies the percentage of sequences correctly detected as climbing. We observe that the two baseline models don't perform well; model A might have too less hidden states while model B probably needs more training data. The proposed block model C and its refined version D both achieve satisfactory results, considering we only have 19 training sequences.

Table 3. HMM sub-models accuracy

Models (# of hidden states)	Classification accuracy	
	With 2 stars	With 1 star
C1 walking (3)	18/18	18/18
C2 up (3)	7/10	0/10
C3 crossover (3)	10/10	7/10
C4 down (3)	10/10	8/10

We verify that the proposed extended star-skeleton representation is better than the original star-skeleton representation by testing the accuracy of the four sub-models denoted as C1, C2, C3 and C4. The results are shown in Table 3. The original star-skeleton representation fails to produce correct classification of the *up* actions, as it erroneously regards shoulders as extreme points, while the extended star-skeleton representation is proven to be more robust.

4. Conclusions

We present results on detection and recognition of fence climbing, a non-cyclic human activity. We also extend the star-skeleton model to incorporate the second star for robustness. Finally, we propose a block based HMM training method for analyzing sequences containing multiple actions.

5. References

[1] J.K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review", *CVIU Journal*, vol.73, no.3, pp.428-440, 1999

[2] J.J. Wang and S. Singh, "Video Analysis of Human Dynamics – A Survey", Real-Time Imaging, vol.9, no.5, pp.321-346, 2003

[3] A. Ali and J.K. Aggarwal, "Segmentation and Recognition of Continuous Human Activity", *Workshop on Detection and Recognition of Events in Video*, pp.28-35, 2001

[4] J.W. Davis and A.F. Bobick, "The Representation and Recognition of Human Movement Using Temporal Templates", *Proc. of CVPR*, pp.928-934, 1997

[5] S. Park and J.K. Aggarwal, "A Hierarchical Bayesian Network for Event Recognition of Human Actions and Interactions", *Multimedia Systems.*, 10(2), pp.164-179, 2004

[6] K. Sato and J.K. Aggarwal, "Temporal Spatio-Velocity Transform and Its Application to Tracking and Interaction", *CVIU Journal*, vol.96, no.2, pp.100-128, Nov. 2004

[7] C. Rao, A. Yilmaz, and M. Shah, "View-invariant Representation and Recognition of Actions", *Intl. J. of Computer Vision*, vol.50, no.2, pp.203–226, 2002

[8] H. Fujiyoshi, "Real-Time Human Motion Analysis by Image Skeletonization", *IEICE Trans. Inf. & Syst.*, vol.E87-D, pp.113-120, Jan. 2004

[9] M. Petkovic, W. Jonker, and Z. Zivkovic, "Recognizing Strokes in Tennis Videos Using Hidden Markov Models", *Proceedings of Intl. Conf. on Visualization, Imaging and Image Processing*, Marbella, Spain, 2001

[10] T.E. Starner and A.P. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models", Technical Report TR-306, Media Lab, MIT, 1995