Boosted String Representation and its Application to Video Surveillance

Yung-Tai Hsu and Jun-Wei Hsieh*
Department of Electrical Engineering, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li 320
Tel:886-3-463-8800 Ext. 2430 Fax:886-3-463-9355
*shieh@saturn.yzu.edu.tw

Abstract

This paper presents a new behavior classification system for analyzing human movements directly from video sequences. First of all, we propose a triangulation-based method to transform each action sequence into a set of symbols. Then, for analyzing the human behavior via those strings representation, we propose a boosted string representation method to extract important string features for accurately analyzing and recognizing different action sequences. The boosted method not only can solve the problem of time warping, but also can reduce the error effects when some postures are wrongly coded into symbols. Since the Adaboost algorithm is proposed for solving two-class problems, we use the error coding concept to modify the Adaboost algorithm such that multiple human action events can be well solved. Then, each action can be well recognized by its correspondence boosted classifier. Experiment results prove that the proposed method is a robust, accurate, and powerful tool for human movement analysis.

1. Introduction

The analysis of human actions [3]-[4] is important and can be applied in various application domains like human-computer interaction systems, video retrieval, video surveillance, and so on. There have been many approaches proposed for tackling problems in video-based human action analysis. For example, Aggarwal et al. [4] used multi-layer finite state automata (FSA) to model human interactions. Cucchiara et al. [7] used a probabilistic projection map to model postures and performed frame-by-frame posture classification to recognize human behavior. The advantage of FAS approach is that it doesn't need a large set of data for model training. However, the number of states and the transitions between states often needs manual efforts to be properly settled.

Context free grammar is another good tool to analyze semantic events from videos. For example, in [2], Ivanov et al. used a context-free grammar parsing scheme to analyze video targets like persons or cars. In [10], Ogale et. al used multi-view training videos to automatically create a view-independent probabilistic context-free grammar to recognize human actions.. The difficulty in the context-free grammar approach is how to transform video images into semantic descriptors.

Hidden Markov model (HMM) is another commonly-used stochastic method for human action analysis. In [1], Oliver et al. used HMMs for classifying the interactions between humans into different types. Nguyen et al. [5] used the abstract hidden Markov model and objects' trajectories to recognize human behaviors. In [6], Navaratnam et al. used HMM and a set of 2D templates created from a 3D model for 3-D human body pose recovering. A serious problem related to HMMs is how to specify or learn the HMM model structure. Usually, human actions have different spatial-temporal scaling changes. The change will make the construction of an accurate state transition graph and the estimate of model parameters become very difficult. In addition, human actions have many unexpected variations. If these unexpected variations are fed into HMM, wrong recognition results will be produced.

This paper presents a boosting method for modeling and recognizing actions directly from videos. First of all, we use a triangulation-based method [9] to convert a human action sequence to a set of symbols. Then, a novel hierarchical histogram representation method is proposed to generate a bank of string features for effectively analyzing human actions. Usually, a person cannot perform the same behavior with the same speed at different times. Our proposed string hypothesis has good ability to tackle the above time-warping problem. In addition, different initial statuses of action events will also affect the accuracy of event recognition. Since the representation does not create any state transition graph, our method can well avoid the errors if state conditions or state transitions are wrongly set. After that, we use an error correction concept to

Proceedings of the 2006 International Conference on Intelligent
Information Hiding and Multimedia Signal Processing (IIH-MSP'06)
0-7695-2745-0/06 $20.00 © 2006 IEEE

modify the original Adaboost algorithm so that a multi-class classier can be trained. The trained multi-class classifier can learn important scaling-invariant feature and thus can well classify any action sequence even if they have different temporal scaling changes. In addition, the classifier has higher tolerances to the coding errors of frames. Experiment results demonstrate the feasibility and superiority of the proposed approach for analyzing human behavior with string representation.

2. Deformable Triangulation Technique for Frame-to-Symbol Converting

To better convert an action sequence into a set of symbols, we use the constrained Delaunay triangulation technique [11] to make each posture which extracted from sequence into triangular meshes.

Fig. 1: Polar Transform of a human posture.

After triangulation, we project a posture sample onto a log-polar coordinate and label each mesh. Then, we can define a centroid context to finely represent this posture. Assume all postures are normalized to a unit size. We use *m* to represent the number of shells used to quantize the radial axis and *n* to represent the number of sectors that we want to quantize in each shell. Fig. 1 shows an example of polar transform with 3 shells and 8 sectors. For the centroid r of the triangular mesh of a posture, we construct a vector histogram $h_r = (h_r(1), \ ..., h_r(k), \ldots, \ h_r(mn))$, in which $h_r(k)$ is the number of triangular mesh centroids in the kth bin when r is considered as the origin, i.e.,

$$h_r(k) = \# \ \{q \mid q \neq r, \ (q - r) \ \in \ bin^k\}, \qquad (1)$$

where bin^k is the *k*th bin of the log-polar coordinate. Then, given two histograms, $h_{r_i}(k)$ and $h_{r_j}(k)$, the distance between them can be measured by

$$C(r_i, r_j) = 1 - \frac{1}{N_{mesh}} \sum_{k=1}^{K_{bin}} \min\{h_{r_i}(k), h_{r_j}(k)\}, \qquad (2)$$

where K_{bin} is the number of bins and N_{mesh} denotes the number of meshes calculated from a posture. Using Eqs. (1) and (2), we can define a centroid context to describe the characteristics of an arbitrary posture *P*.

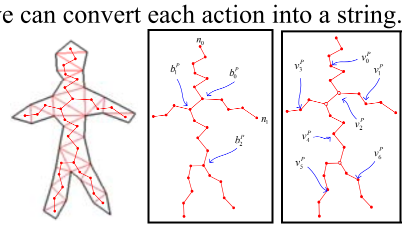
To define the centroid context of *P*, we need to derive a skeleton of *P* using a graph search. The skeleton extraction method can be found from our previous work[9]. Then, from *P*, we can get its skeleton T^P_{dfs}. As shown in Fig. 2, (b) is the skeleton feature of *P*. Here, we call a node a branch node if it has more than one child. By this definition, there are three branch nodes in Fig. 2(b), i.e., b^P_0, b^P_1, and b^P_2. The branch nodes are the key points used to decompose P into different body parts, such as the hands, feet, or torso. If we remove all the branch nodes from T^P_{dfs}, it will be decomposed into different branch paths $path^P_i$. For example, in Fig. 2(b), if we remove b^P_0 from T^P_{dfs}, two branch paths will be formed, i.e., one from node n_0 to b^P_0 and one from b^P_0 to node n_1. Given a path $path^P_i$, we collect a set of triangular meshes V^P_i along it. Let c^P_i be the centroid of the triangular mesh closest to the center of the set of meshes. Given a centroid c^P_i, we can obtain its corresponding histogram $h_{c^P_i}(k)$ using Eq.(1). Assume that the set of these path centroids is V^P. Based on V^P, the centroid context of P is defined by:

$$P = \{h_{c^P_i}\}_{i=0,\ldots,|V^P|-1},$$

where $|V^P|$ is the number of elements in V^P. Given two postures, *P* and *Q*, the distance between their centroid contexts is measured by

$$d_{cc}(P,Q) = \frac{1}{2|V^P|} \sum_{i=0}^{|V^P|-1} w^P_i \min_{0 \le j < |V^P|} C(c^P_i, c^Q_j) + \frac{1}{2|V^Q|} \sum_{j=0}^{|V^Q|-1} w^Q_j \min_{0 \le i < |V^Q|} C(c^P_i, c^Q_j), \qquad (3)$$

where w^P_i and w^Q_j are the area ratios of the ith and jth body parts residing in *P* and *Q*, respectively. Based on Eq.(3), an arbitrary pair of postures can be compared. Then, a clustering technique can be used to extract a set of key postures from different action sequences. Then, we can convert each action into a string.



(a) (b) (c)

Fig. 2: Body component extraction: (a) triangulation resut; (b) Skeleton of (a); and (c) centroids of different parts (determined by removing all the branch nodes).

3. Novel String Hypothesis Generator

Assume that S(p) is a string generated from an action sequence. In order to deal with different spatial-temporal scaling changes, coding errors, beginning symbols, and noise, we present a novel method to generate a bank of hypotheses for string classification.

Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06)
0-7695-2745-0/06 $20.00 © 2006 IEEE

Assume that I is the set of key postures. The number of key postures in I is M. A hypothesis is a string histogram, which accumulates symbol-to-symbol patterns appearing in S(p), generated under different orders, sampling rates, and quantization levels. The first-order hypothesis is generated by counting the repeated pattern of symbol-to-itself with the sampling rate i_1 and quantized by d, i.e.,

$$f_{i_1}^{d} S(x_1) = \#\{p \mid S(p) = x_1 \ \& \ S(p+i_1) = x_1\} / d$$

for all symbols $x_1 \in I$,where i_1 and $d \in Z^+$. The second-order hypothesis is created for capturing the relations between pairs of symbols:

$$f_{i_1}^{d} S(x_1, x_2) = \#\{p \mid s(p) = x_1 \ \& \ s(p+i_1) = x_2\} / d \ , \quad (5)$$

where x_1 and $x_2 \in I$, and i_1 and $d \in Z^+$. The general form of string hypothesis can be extended to the Kth order using the form:

$$f_{i_1, i_2, ..., i_{K-1}}^{d} S(x_1, x_2, ..., x_K) = \#\{p \mid s(p) = x_1 \ \& \ s(p+i_1) = x_2 \ \& ..... \& \ s(p+i_1+i_2+...+i_{K-1}) = x_K\} / d, \quad (6)$$

where x_1, x_2, ..., and $x_K \in I$, and i_1 , i_2 , …, i_{K-1} , and $d \in Z^+$. Then, using Eqs.(4)-(6), a bank of string hypotheses can be generated. Each hypothesis is a weak classifier for classifying action events into different events. In Section 4, we will use the Adaboost algorithm to learn a stronger classifier from the set of weak classifiers.

4. Event Classification with Adaboost

In this section, the original Adaboost algorithm and its modification are introduced.

4.1 Single Event Classification Using Adaboost

Assume that there is a string S which denotes an action event. Then, we can generate a set Ω of string features to represent this action event using Eq.(6). For convenience, we use $f_i S$ to denote the ith feature in Ω . For each $f_i S$, we can use a indexing technique to convert it into a feature vector with the length l_i . Similarly, given an unknown string x, we can also generate different string features where the ith feature in x is $f_i x$. Then, the dissimilarity between $f_i S$ and $f_i x$ is calculated by

$$d(f_i S, f_i x) = \sum_{j=1}^{l_i} f_i S(j) \exp(| f_i S(j) - f_i x(j) |) + \sum_{j=1}^{l_i} f_i x(j) \exp(| f_i S(j) - f_i x(j) |) \ .(7)$$

Then, given N_p training sequences of the same action event, there are totally $|\Omega| N_p$ string features generated for event classification. These features form a bank of weak classifiers. The Adaboost algorithm uses an iterative scheme to gradually improve the ability of the learned stronger classifier to classify action events.

At each iteration t, a "good" weak classifier is selected and added in turn to form a strong classifier which is a weighted sum of individual selected weak classifiers. The selected weak classifier $h_t(x)$ is the one which has the minimum classification error ε_t when the feature f_t is selected among the bank of features. At the tth step, the Adaboost algorithm combines the weak classifiers h_1, ..., h_t to form the strong classifier H_t using the weight α_t by the form (4) α_t = 0.5 ln $((1-\varepsilon_t)/\varepsilon_t)$. Thus, the tth strong classifier H_t has the form:

$$H_t = \sum_{i=1}^{t} \alpha_i h_i = H_{t-1} + \alpha_t h_t \ .$$

Assume that there are N training samples. Details of the Adaboosing algorithm is described as follows.

AdaBoosting Algorithm

Initially assign uniform weights $w_i^0 = 1/N$ for all x.

At each iteration t:

1. Find the best weak classifier with $h_t(x)$ with the error ε_t and W^t , and get $\alpha_t = 0.5 \ln((1-\varepsilon_t)/\varepsilon_t)$;
2. Get the hypothesis $H_t = H_{t-1} + \alpha_t h_t$;
3. $w_i^{t+1} = \exp(-H_t(x_i) y_i)$ and normalize it.

Output the final hypothesis $H(x_i) = sign\,[\sum_{t=1}^{T} \alpha_t \ h_t(x_i)\,]$.

4.2 Multiple Event Classification

For tackling the multiple-class problem, we modify the error correction concept [12] to simplify the multi-class problems into a series of binary classification problems. Assume that there are N event categories. Then, for each event, we can train its stronger classifier $H_i(x)$ for i=1, …, N. Then, we can create a weight W_i to code the ith action event. Assume that there are c action events e_k^i for training W_i to represent the ith event category. Then, the jth bit W_{ij} of W_i can be decided by

$$W_{ij} = \frac{1}{c} \sum_{k=1}^{c} sign(H_j(e_k^i)) \ . \quad (8)$$

Then, W_i becomes a histogram recording different contributions of H_j to the event category E_i. Then, the similarity between x and E_i can be measured by

$$similarity(x, \ E_i) = \sum_{j=1}^{N} W_{ij} sign(H_j(x)) \ . \quad (9)$$

Then, the correct type of x can be decided by

$$Type(x) = \arg\max_{E_i} similarity(x, \ E_i) \ .$$

5. Experiment Results

Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06)
0-7695-2745-0/06 $20.00 © 2006 IEEE

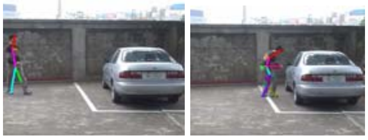
To analyze the efficiency and effectiveness of the proposed approach, we created a large database which includes ten types of action events. For each type, thirty action sequences are included. For the first experiment, the performances of our proposed method against different temporal-scaling changes, beginning states, and coding error rates were examined. Our proposed string hypothesis generator has good ability to tackle the above time-warping problem. In addition, different initial statuses of action events will also affect the accuracy of event recognition. The symbol errors will also affect the accuracy of action event recognition. Table 6 shows the accuracy analyses when different environmental variations happen. Clearly, no matter how the symbol error rate is, our method still performs well to recognize all the input action sequences.

Table 4: Accuracy analyses when actions have temporal-scaling changes, frames shift, or errors.

Sampling	Accur.	Shift	Accur.(%)	Error Rate	Accur.
rate=0.8	95.73	5	95.7	10%	96. 83
rate=1.2	93.28	10	94.38	20%	95.41
rate=1.5	92.81	20	94.21	30%	93.12
rate=2	90.73	30	94.17	40%	94.32

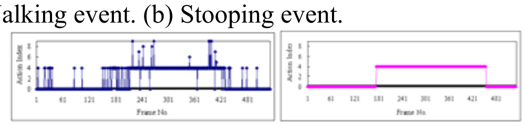
Table 5: Accuracy comparison of event recognition between our method and HHM.

Actions	Gym.	Walk	Squat	Stoop	Sitting
Boosting	95.9	97.2	95.3	98.4	99.6
HMM	85.79	88.5	89.7	90.62	91.57



(a) Frame #0 (b) Frame #152

Fig. 3: A video including multiple action events. (a) Walking event. (b) Stooping event.



(a) (b)

Fig. 4: Result of multiple event recognition. (a) Result of posture recognition. (b) Result of action event recognition.

In the second set of experiments, we used the real action video sequences to test our proposed method. In addition, the HMM method were implemented for comparisons. Table 5 lists the accuracy comparison between our method and HMM. Clearly, our method performs much better than HMM. In addition to recognize single action event, our method can also recognize multiple action events in the same video sequence. Shown as Fig. 3, there is a video sequence beginning with a walking event (see (a)), then a stooping event (see (b)), and ending with a walking one. The video sequence can be decoded into a series of postures. Then, ten key postured were extracted for recognizing these postures. Fig. 4(a) shows the set of key postures. Then, we can convert this video sequence into different symbols. Clearly, if only postures are used to recognize the behaviors, many errors will happen like Fig. 4(b). The errors can be avoided if our proposed method is adopted. Clearly, our method can well recognize any video sequences even though they include multiple action events. All the above results have proved that the proposed boosting method is a robust, accurate, and powerful tool for action event analysis.

6. References

[1] N. M. Oliver, *et al.*, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on PAMI.*, vol. 22(8), pp. 831-843, 2000.

[2] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on PAMI.*, vol. 22, no. 8, pp. 852-872, 2000.

[3] W. Hu, et al., "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Transactions on SMC-Part C.* vol. 34, no. 3, pp. 334-352, Aug. 2004.

[4] J. Aggarwal and S. Park, "Human motion: Modeling and recognition of actions and interactions," *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 640–647, 2004.

[5] N. T. Nguyen, et al., "Recognition and monitoring high-level behaviors in complex spatial environments," *CVPR*, vol. 2, pp. 620-625, June 2003.

[6] R. Navaratnam, *et al.*, "Hierarchical part-based human body pose estimation," *In* Proc. *British Machine Vision Conference*, vol. 1, pp. 479-488, UK, Sep. 2005.

[7] R. Cucchiara, *et al.*, "Probabilities posture classification for human-behavior analysis," *IEEE Transactions on SMC-Part A,* vol. 35, no. 1, pp. 42-54, Jan. 2005.

[8] N. Jojic, *et al.*, "Transformed hidden Markov models: Estimating mixture models and inferring spatial transformations in video sequences," *CVPR*, vol. 2, pp. 26-33, Hilton Head, SC, June 2000.

[9] Y. T. Hsu, J. W. Hsieh, and H. Y. Liao, "Human behavior analysis using deformable triangulations," *IEEE International Workshop on Multimedia Signal Processing*, Shanhai, China, Nov. 2005.

[10] A. S. Ogale, *et al.*, "View-invariant modeling and recognition of human actions using grammars," *In Workshop on Dynamical. Vision at ICCV'05*, October 2005.

[11] L. P. Chew, "Constrained delaunay triangulations," *Algorithmica*, vol. 4, no.1, pp. 97-108, 1989.

[12] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263-286, 1995.

Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06)
0-7695-2745-0/06 $20.00 © 2006 IEEE