AN EFFICIENT BAYESIAN FRAMEWORK FOR ON-LINE ACTION RECOGNITION

R. Vezzani[1]*, M. Piccardi*[2]*, R. Cucchiara*[1]

[1] Univ. of Modena and Reggio Emilia - Italy, [2] Univ. of Technology, Sydney - Australia

ABSTRACT

On-line action recognition from a continuous stream of actions is still an open problem with fewer solutions proposed compared to time-segmented action recognition. The most challenging task is to classify the current action while finding its time boundaries at the same time. In this paper we propose an approach capable of performing on-line action segmentation and recognition by means of batteries of HMM taking into account all the possible time boundaries and action classes. A suitable Bayesian normalization is applied to make observation sequences of different length comparable and computational optimizations are introduce to achieve real-time performances. Results on a well known action dataset prove the efficacy of the proposed method.

Index Terms— HMM, on-line action recognition

1. INTRODUCTION

Real-time action recognition in videos is a challenging task due to many intrinsic difficulties such as the high variability of action instances across different scenarios and subjects, the high dimensionality of the feature sets, and the typically stringent real-time requirements. On the other hand, accurate and prompt action recognition can take important video-based applications such as video surveillance, ambient intelligence, and smart human-computer interaction to an all new level and is therefore catalysing much research world-wide.

Action recognition approaches mainly categorise into two groups: time-warping approaches and model-based approaches. In time-warping approaches, an action template and an action instance to be compared are both represented in the domain of time. Comparison is performed by "warping" the instance onto the template by point correspondence and measuring the distance required for performing the warping. Smaller distances indicate better matching. In model-based approaches, the action template is a probabilistic model and the comparison with the instance is typically performed by measuring the likelihood of the instance in the model. Many graphical models have been proposed for this purpose, with the hidden Markov model (HMM) being the most frequently recurring in the literature.

In most real situations, actions from a subject take place as a continuous stream. A simplifying assumption, often made, is that the start and end times of each action in the stream are provided by a prior, external time segmentation step. More realistically, action recognition has to be performed simultaneously with time segmentation (on-line action recognition). Joint action recognition and time segmentation is far more challenging and has only recently started receiving adequate attention in the literature. The typical approach consists of repeating the matching between the instance and the templates over sliding, overlapping temporal windows. This implies that at any given time there are a number of matches in progress of different age. However, this approach carries inherent issues: a) it is not immediate how to select the most likely match amongst matches of different age, and b) the computational load grows proportionally with the number of matches in progress, thus hindering real-time performance. This paper proposes a rigorous Bayesian treatment of this problem, formulates context assumptions and proposes an HMM-based solution (named *streaming HMM*) that achieves both accuracy and efficiency. Experimental results show that the proposed solution attains comparable accuracy with that of an HMM solution that is informed with the ground-truth time segmentation and an average speed up of 7x compared to a sliding window-based HMM approach.

2. RELATED WORKS

On-line action recognition from a continuous stream of actions has a relatively limited literature compared to time-segmented action recognition. Even a very recent, comprehensive survey on action recognition does not cover this topic [1]. In [2], Chen *et al.* propose an approach for on-line action recognition based on a sliding window approach. Calling C the number of action classes, they start the evaluation of a new battery of C HMMs, one per class, every w frames. In this way, at every time t there are several HMM batteries of different age, T, available to support the action's classification. However, the authors make the restrictive assumption that the "voting" battery is that of a given, fixed age. This is an oversimplifying assumption since actions have intrinsically variable durations. In [3], the authors address the issue of the different stage of evaluation across HMM batteries of different age. They propose to simply divide the likelihood reported by each battery by the battery's age and add up votes from all batteries. The reported accuracy for a

dataset with four classes is high. Unfortunately, such a simple compensation does not make the likelihoods of the various batteries correctly compensated in a full Bayesian sense. In [4], Mori *et al.* propose the use of a preliminary stage where the continuous stream is partitioned into time segments of no-detectable and detectable actions. In this way, they avoid having to recognise actions in frames where the action is still in early stages. This distinction can also help compare performance of on-line approaches with that of off-line approaches which recognise actions only after their conclusion. In [5], Ali and Aggarwal propose to approach time segmentation by detection of breakpoint frames. However, such frames are not easy to detect in general. In [6] and [7], the authors propose an HMM of very high semantic level, where each action coincides with a state. Therefore, state decoding provides the desired action recognition. However, the hidden nature of the HMM's states does not permit to enforce the desired state semantic in a general case. To overcome the aforementioned limitations, this paper propose a complete Bayesian treatment of on-line action recognition that is presented in Section 4.

3. OFF-LINE HMM ACTION CLASSIFICATION

As a term of comparison, we need to provide a probabilistic solution for the classification problem of a pre-segmented clip, containing a single atomic action. Given a set of C action classes $\Lambda = \lambda^1 \dots \lambda^C$, our aim is to find the class λ^* which maximise the probability $P(\lambda|O)$, where $O = \{o_1 \dots o_T\}$ is the entire sequence of frame-wise observations (features). We will refer to this problem as Off-line Action Classification. In his famous tutorial [8], Rabiner proposed to use hidden Markov models to solve this kind of classification problems. An HMM should be learned for each action; the classification of an observation sequence O is then carried out selecting the model whose likelihood is highest, $\lambda^* = \arg\max_{1 \le l \le L} \left[P\left(O|\lambda^l\right)\right]$. If the classes are equally likely, this solution is optimal also in a Bayesian sense. If the decoding of state sequence is not required, the recursive forward algorithm with the three well known initialization, induction and termination equations can be applied.

$$\begin{aligned} \alpha_1(j) &= \pi_i b_j(o_1), 1 \le i \le N \\ \alpha_{t+1}(j) &= \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j\left(o_{t+1}\right) \\ P\left(O|\lambda\right) &= \sum_{j=1}^{N} \alpha_T\left(j\right) \end{aligned} \tag{1}$$

The term $b_j(o)$ depends on the type of the observations. We adopted the K-dimensional feature set described in the following, which requires to model the observation probabilities by means of density functions. As usual, we adopt a Gaussian Mixture Model, which simplifies the learning phase allowing a simultaneous estimation of both the HMM and the Mixtures parameters, given the numbers N and M of hidden states and Gaussians per state respectively [9]. In this case, the term $b_j(o_t)$ of Eq. 1 can be approximated as:

$$b_j(o_t) = \sum_{l=1}^{M} c_{jl} \mathcal{N}\left(o_t|\mu_{jl}, \Sigma_{jl}\right) \tag{2}$$

3.1. Feature set

In this work we exploited a simple feature set, discriminative enough to obtain reasonable classification rates, but not too complex to permit fast processing. Each frame t is processed to extract the foreground mask by means of a background subtraction step since the videos were acquired by a fixed camera. Then, the extracted silhouettes are divided into five slices $S^1 \dots S^5$ using a radial partitioning centered in the gravity center $\{x_c(t), y_c(t)\}$. Calling A_t and $\{A_t^i\}_{i=1\dots5}$ the areas of the whole silhouette and of each slice $\{S^i\}$ respectively, the 17-dimensional feature set is obtained as reported in Fig. 1. The features contain both motion (o^1 and o^2) and shape information ($o^3 \dots o^{17}$).

$$o_t = \left\{o_t^1 \dots o_t^{17}\right\},$$
$$\begin{cases} o_t^1 = x_c(t) - x_c(t-1) \\ o_t^2 = y_c(t) - y_c(t-1) \\ o_t^{3\dots7} = A_t^i / A_t \\ o_t^{8\dots12} = \max\limits_{S_i}\left((x - x_c)/\sqrt{A_t^i}\right) \\ o_t^{13\dots17} = \max\limits_{S_i}\left((y - y_c)/\sqrt{A_t^i}\right) \end{cases}$$

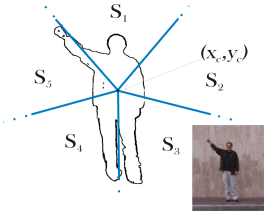

Fig. 1. 17-dimensional Feature set

4. ON-LINE HMM ACTION RECOGNITION

Differently from off-line action classification, on-line action recognition requires to estimate the most likely action currently performed by the monitored subject, given only the observations until now. The observed sequence may contain more subsequent actions and the current action may be in progress. The current action at a generic time t, λ_t^*, could be determined from the entire sequence of observations, $O_t = \{o_1, \dots o_t\}$, by estimating and maximizing probability $P(\lambda_t|O)$. This probability does not have an obvious form in general, yet it could be reasonably approximated by the posterior probability of an HMM given its sequence of observations, $P_{HMM}(\lambda|O_T)$, where T is the length of the frame sequence since the inception of the current action, $O_T = \{o_{t-T+1}, \dots o_t\}$. However, T is unknown. We could solve this problem by finding an expression for the joint probability $P(\lambda_t, T|O)$ and marginalising T:

$$P\left(\lambda_t|O\right) = \sum_{T} P\left(\lambda_t, T|O\right). \tag{3}$$

We apply the Bayes theorem to rewrite (3) as:

$$\sum_{T} P\left(\lambda_t, T|O\right) = \sum_{T} P\left(T|\lambda_t, O\right) \cdot P\left(\lambda_t|O\right). \tag{4}$$

While (4) is an obvious identity (the second term does not depend on the sum index, T, and the first adds up to 1 for the axiom of total probability), it provides us with an expression suitable to perform approximations. $P(T|\lambda, O)$, is the probability that the current action has started T frames ago given the action, λ, and all the observations, O. Such a probability is hard to model in general, yet we could approximate it by a probability $P(T|\lambda_t)$ that does not depend on the observations and has a relation with the mean duration $\bar{D}_\lambda$ of each action. In particular, $P(T|\lambda_t)$ is the probability of action λ to be started T frames ago and to be still in execution at time t. Considering a Gaussian distribution of the action length $D_\lambda \propto \mathcal{N}(\bar{D}_\lambda, \sigma_\lambda)$, we choose to estimate $P(T|\lambda_t)$ through the complement of the Gaussian cumulative density function:

$$P(T|\lambda_t) = 1 - \int_{-\infty}^{T} \mathcal{N}(T|\bar{D}_\lambda, \sigma_\lambda) \tag{5}$$

where $\bar{D}_\lambda$ and σ_λ are estimated during the learning phase.

On the other hand, the current action λ_t does not depend on the observations related to the previous actions; thus the last term in (4) is equivalent to $P(\lambda|O_T)$. At its turn, using the Bayes' theorem, $P(\lambda|O_T)$ is proportional to:

$$P\left(\lambda|O_T\right) \propto P_{HMM}\left(O_T|\lambda\right)\ P\left(\lambda\right) \tag{6}$$

where $P_{HMM}(O_T|\lambda)$ could be obtained by means of the usual HMM forward formula (Eq. 1, with priors $P(\lambda)$ assumed equal). Unfortunately, the estimation of the current action by means of $P_{HMM}(O_T|\lambda)$ is not reliable due to a strong dependence of the forward algorithm with respect to the observation length T. First of all, from a scale point of view, the α terms have a dependence on T, since they are somehow related to the product of the probability of each observation. Furthermore, a dependence on T is introduced by the approximation of (2) in which point-wise values of Gaussian pdfs are used instead of integral values over suitable ranges. Actually, $b_j(o_t)$ should be computed taking into account at least the quantization step of the observation space:

$$P(o_t|j) \approx \tilde{b}_j(o_t) \approx \int_{o \in N(o_t,\delta)} b_j(o) \approx \delta^K b_j(o_t) \tag{7}$$

where $N(o_t, \delta)$ is a K-dimensional neighborhood of o_t with radius δ. The recursive forward formula of Eq. 1 can be better rewritten using corrected $\tilde{\alpha}$-values which are related to the usual ones:

$$\tilde{\alpha}_{t+1}(j) = \left[\sum_{i=1}^{N} \tilde{\alpha}_t(i) a_{ij}\right] \tilde{b}_j(o_{t+1}) = \delta^{K(t+1)} \alpha_{t+1}(i). \tag{8}$$

The coefficient $\delta^{K(t+1)}$ is usually discarded, since the comparisons are made among sequences with the same length T, but cannot be avoided in our case. To solve these numerical issues we propose to replace $P_{HMM}\left(O_T|\lambda\right)$ of Eq. 6 with the geometrical mean of a single observation probability o_t given the model:

$$\tilde{P}_{HMM}\left(O_T|\lambda\right) = \sqrt[T]{\sum_j \tilde{\alpha}_T(j)} = \delta^K \sqrt[T]{\sum_j \alpha_T(j)} \tag{9}$$

$\tilde{P}_{HMM}$ does not suffer from the T and δ dependences mentioned above and different-length sequences can now be compared.

4.1. Computational Optimizations

From a computational point of view the proposed technique is too heavy for real time implementations. Actually, at each frame the most likely action should be chosen among the response of many HMMs, which should consider all the action classes and all the possible action starting times. Furthermore this number grows over time. Using the forward algorithm, at time t the number of α values to be update is $C \times t \times N$, and each of them requires the estimation of an observation value from a Gaussian Mixture Model. To this aim we propose three improvements to allow an implementation with a reasonable computational weight and real time performances.

1. The maximum action length is set to T_{max}; this hypothesis stops the otherwise endless growth of the number of HMMs to be estimated;

2. The action starting times are sub-sampled by considering a possible starting time every w frames. Given that these videos have a frame rate of 25 fps, we chose $w = 10$ so as to start a new battery every 400 ms. Such a granularity is sufficient to not miss any significant part of an action;

3. We exploit the factorization of the α terms of Eq. 1 to reduce the computational complexity; since the $b_j(o)$ term depends on the current observation and the hidden state only, all the HMMs related to the same action and differing from the action starting point share the same values. Thus, we compute them only once.

5. EXPERIMENTAL RESULTS

We experimented the proposed technique using the Weizmann dataset [10] which contains 90 videos of 10 main actions performed by 9 different people. Some actions are performed in different ways, thus in the on-line recognition we used all the 16 specific classes. We evaluated our off-line recognition approach based on a leave-one-out experiment using the training data. Goal of this evaluation task is to evaluate whether the feature set is adequate for the recognition task at hand. First, we empirically tuned the HMM parameters. In particular the number N of hidden states and the number M of Gaussians of the mixture model of Eq. (2 have been set to 5 and 3 respectively to maximize the recognition rates.The recognition rate of our approach is 87%, which is comparable with other HMM approaches proposed.

Table 1. Mean recognition rates

Method	Frame-wise Accuracy	Last Frame Accuracy
Traditional HMM	54.1%	78%
Sliding window	58.2%	86%
Our approach*	88.9%	99%

*online fast version with $Tmax$ defined and w set to 10.

For the on-line testing we generated sequences of actions by merging together multiple videos. The training of the HMMs was performed by using all the individual sample videos. We compared the results of our proposal with those of a sliding window approach, in which a fixed length (set to 60, i.e., the mean action length) observation sequence is used for the classification. Furthermore we tried to classify the current action by means of batteries of HMMs of variable age T without any compensation for the age. In this case, we choose λ according to the battery which maximizes the probability in Eq 6, as with a basic HMM approach. Outputs on a sample sequence of 20 actions are reported in Figure 2. The most likely action selected at each frame and the ground truth are plotted on the bottom part of each graph, while the corresponding probability values are reported on the top part. Mean accuracies considering several sequences of 50 actions are reported in Table 1. Since the performance evaluation on the initial frames of each action can be misleading, we also reported the mean accuracy of each method using only the last frame of every action in the sequence. Our proposal outperforms both the other approaches.

The proposed fast implementation (as described in section 4.1) is able to meet real-time processing constraints. The average execution time per frame for segmentation, feature extraction, and action recognition is 47ms (of a C++ implementation on a Pentium 4 Dual Core), of which only 3ms are devoted to the HMM update and action classification). The average speed up is of 7x compared to a sliding window-based HMM approach without the across-battery sharing of the observation probabilities.

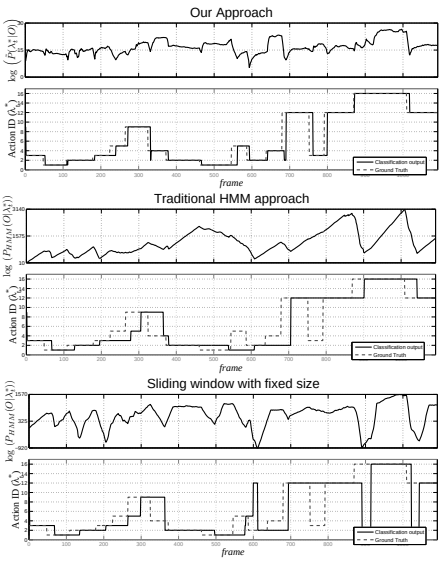

Fig. 2. Action recognition accuracy across different methods

6. CONCLUSIONS

We presented a Bayesian framework for on-line action recognition. We adopt an HMM approach based on a simple but effective feature set. Differently from methods recurring in the literature, we specifically considered the action length and we proposed a method to compare the HMM responses to different-length sequences. The high accuracy obtained on a well-known dataset confirms the validity of the method which outperforms the base HMM strategy as well as the the sliding window approach.

7. REFERENCES

[1] P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[2] Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen, and Suh-Yin Lee, "Human action recognition using star skeleton," in *Proc. of the 4th ACM Int'l Workshop on Video surveillance and sensor networks*, New York, NY, USA, 2006, pp. 171–178.

[3] F. Niu and M. Abdel-Mottaleb, "Hmm-based segmentation and recognition of human activities from video sequences," *IEEE Int'l Conf. on Multimedia and Expo*, pp. 804–807, July 2005.

[4] T. Mori, Y. Nejigane, M. Shimosaka, Y. Segawa, T. Harada, and T. Sato, "Online recognition and segmentation for time-series motion with hmm and conceptual relation of actions," *Int'l Conf. on Intelligent Robots and Systems*, pp. 3864–3870, Aug. 2005.

[5] A. Ali and J.K. Aggarwal, "Segmentation and recognition of continuous human activity," *Proc. of IEEE Workshop on Detection and Recognition of Events in Video*, pp. 28–35, 2001.

[6] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 844–851, Aug 2000.

[7] Y.A. Ivanov and A.F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, Aug 2000.

[8] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proc. of the IEEE*, 1989, vol. 77, pp. 257–286.

[9] Jeff A. Bilmes, "A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," Tech. Rep., 1997.

[10] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri, "Actions as space-time shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, Dec 2007.