

© 2010 IEEE. Reprinted, with permission, from Massimo Piccardi, Histogram-Based Training Initialisation of Hidden Markov Models for Human Action Recognition . Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on, and September 2010. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Technology, Sydney's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it

Histogram-Based Training Initialisation of Hidden Markov Models for Human Action Recognition

Zia Moghaddam

University of Technology, Sydney
15 Broadway, Ultimo NSW 2007, Australia
ziam@it.uts.edu.au

Massimo Piccardi

University of Technology, Sydney
15 Broadway, Ultimo NSW 2007, Australia
massimo@it.uts.edu.au

Abstract

Human action recognition is often addressed by use of latent-state models such as the hidden Markov model and similar graphical models. As such models require Expectation-Maximisation training, arbitrary choices must be made for training initialisation, with major impact on the final recognition accuracy. In this paper, we propose a histogram-based deterministic initialisation and compare it with both random and a time-based deterministic initialisations. Experiments on a human action dataset show that the accuracy of the proposed method proved higher than that of the other tested methods.

1. Introduction

Human action recognition from live videos is one of the most important problems currently tackled by video surveillance research. Humans are the main actors in surveillance scenes and their actions and interaction with the environment provide useful information for safety, security and environment personalisation.

Recognising human actions in videos can be framed as a problem of time-series classification. Features about each single human subject are directly extracted from the video's frames and linked along the time dimension by tracking and data association algorithms. By calling o_t the feature set extracted for a human subject at the generic time t , the overall time series can be noted as $O = \{o_1, \dots, o_t, \dots, o_T\}$, with T the action length in frames. The feature set must be chosen to be action-discriminative and recognition can be provided by use of sequential classifiers. Two recent surveys in [1, 2] offer a broad overview of the many action recognition approaches proposed over the last two decades. In terms of classification approaches, two main lines of investigations have been followed: 1) recognising the action directly in the time domain; and 2) recognising the action by probabilistic graphical models. The former group has dynamic time warping (DTW) as its main representative;

the latter has the hidden Markov model (HMM). Despite the recent renaissance in interest in time-warping approaches, probabilistic models such as HMM have maintained widespread adoption for their recognised strength against the intrinsic variations of action instances. However, certain problems with the training of such probabilistic models are still partially unresolved. Since modelling of the dependencies across observations alone can prove complicated and ad-hoc, latent variables, or states, are often utilised in the models. This imposes the use of Expectation-Maximisation (EM) style algorithms for model training. However, it is well known that EM algorithms can only find local maxima of their target function (typically, the likelihood), and that such maxima strongly depend on the arbitrary initialisation of the model's parameters, λ . Moreover, the problem of the quality of the maxima and the generalization to unseen examples is often exacerbated by the scarcity of training samples.

The common approach to EM initialisation is that of running multiple training sessions with different random initialisations of λ and choosing the best session by cross-validation. In contrast to the random initialisation approach, we recently proposed a deterministic initialisation method based on partitioning the observations in the time domain [3]. However, we assumed linearity in the human actions' dynamic which we found a restrictive hypothesis in some cases. Hence, in this paper we present a deterministic initialisation method based on data histograms which does not rely on an equivalent linearity assumption. Given the high dimensionality of the observation space and the relative scarcity of samples, this paper proposes to use an approximate histogram based on an incremental, feature-by-feature approach. Experiments results seem to confirm the validity of this approach.

The rest of the paper is organized as follows. Section 2 offers a brief review of the related work. In Section 3, the training initialisation for hidden Markov model is described and then we propose our histogram-based initialisation. The feature set and experimental results are reported and discussed in Section 4. Finally, conclusions are presented.

2. Related work

Using HMM for human action recognition goes a long way back. The first paper that we are aware of, from Yamato *et al.*, dates 1992 [4]. The authors used HMM to recognise six different tennis actions. In their work, each frame is background-subtracted and the extracted foreground object is partitioned into a grid of blocks, centred on the centroid. The number of foreground pixels in each block is the feature set that is then mapped onto a symbol by vector quantization. Discrete-output HMMs with 36 states are used for recognition. This early work already epitomises two major problems of action recognition: a) the adoption of a discriminative and workable feature set and b) the choice of a suitable classification approach.

For the feature set, a variety of approaches have been exploited, including optical flow [5], body parts tracking [6, 7], silhouettes [8] space-time interest points [9] and local interest points [10, 11]. Researchers are left with the decision whether to use a rich feature vector, possibly invariant to the viewpoint (e.g. [12]), or a simple, fast-to-extract feature vector designed with opportunistic action discrimination in mind [3, 10, 13].

Various graphical models have been used to classify actions from observation sequences. While the main model has been the HMM, other models such as HMM variants (coupled, hierarchical, layered, entropic etc.), DBNs, CRFs have been used (e.g. [8]). However, they are all highly parametric models and the tuning of their parameters may prove unsatisfactory. In particular, EM learning is sensitive to the choice of the initial parameter assignment, and this problem was recognised and addressed by various authors. For instance, Ferrer *et al.* in [14] reviewed various HMM initialisation methods based on random techniques and introduced their own method based on choosing the random start that leads to nearly-equiprobable states. In [15], Toledano *et al.* have explored three different ways of initialising HMM training: 1) by a fixed template for all classes; 2) by historical averages; and 3) by oracle initialisation (this last only to establish offline upper bounds). While these methods remove undesired randomness, they are not adaptive in the training samples. In a recent work [3], we exploited partitioning of the training sequences in the time domain to select the HMM's initial means. We reported an accuracy that is comparable to the average accuracy obtainable by multiple random initialisations. However, some restrictive assumptions were made: a) that the action's states would occupy equal proportions of time (as if the actions' dynamics would develop linearly along the time dimension), and b) that the states would evolve in a "left-to-right" manner. Such assumptions do not seem to reflect the possibly complex dynamics of human actions and

justify exploring alternative deterministic initialisations more flexible in their assumptions.

3. Action classification using HMM

Using HMM for action recognition converts the recognition problem into classification of time series. A much-cited tutorial on HMM and its three main problems – evaluation, decoding and estimation – can be found in [16]. Let us call C the set of K action classes, $C = \{C_1, \dots, C_k, \dots, C_K\}$. Given an HMM for each class, noted by its set of parameters, λ_k , $k=1..K$, maximum-likelihood classification of a time series can be achieved as:

$$C_k^* = \arg \max_k \left(p(O | \lambda_k) \right) \quad (1)$$

where $p(O|\lambda_k)$ is a likelihood function that can be effectively computed based on the forward or backward algorithm [16]. If maximum-a-posteriori or minimum risk classification is sought, priors and costs can be easily added.

3.1. Hidden Markov model

HMM is a probabilistic graphical model in which the modelled system has observed outputs, or observations, but the states are hidden. The observation sequence is noted as $O=\{o_1, \dots, o_T\}$, where T is its length. An HMM with N states is represented by the following parameter set:

$$\lambda = \{A, B, \pi\} \quad (2)$$

where A is the $N \times N$ state transition probability matrix, B are the observation probabilities and π are the $N \times 1$ initial state probabilities. In our case, the observations are continuous, multivariate random variables and their distribution in each state is modelled by a mixture of M Gaussian components (Gaussian mixture model - GMM):

$$b_j(o) = \sum_{l=1}^M c_{jl} G(o | \mu_{jl}, \Sigma_{jl}), \quad j = 1 \dots N \quad (3)$$

In Eq. (3), μ_{jl} and Σ_{jl} are the mean and covariance of the l -th Gaussian and c_{jl} is its weight, or prior probability, in the mixture. Hence, the total size of B is $(N * M * \text{sizeof} \{\mu_{jl}, \Sigma_{jl}, c_{jl}\})$. Such a number is typically high and confirms that an HMM is a highly parametric model.

3.2. HMM training

During training, the HMM parameters are typically estimated to fit the training observation sequences with maximum likelihood [16]. The most popular HMM training algorithm is the Baum-Welch re-estimation algorithm [16], which is of EM style. Like all EM

algorithms, it guarantees convergence to a local optimum (or a saddle point) of the data likelihood, and the position and quality of such a maximum depend in turn on the initialisation parameters. Moreover, the set of HMM parameters, λ , contains two hyperparameters: the number of states, N , and the number of Gaussian components in each $b_j(o)$, M . For these two parameters, we simply adopt exhaustive search over a plausible range, $N, M \in \{1..6\}$ and a selection based on cross-validation accuracy. As software, we have used and extended Kevin Murphy's HMM toolbox for Matlab [17].

3.3. HMM training initialisation

The Baum-Welch re-estimation algorithm requires an initial assignment of the HMM parameters to initiate training. Whereas all the parameters influence the outcome of training, in the following we focus only on B because of their typically overwhelming size. For instance, in an HMM with $N = M = 5$, $F = 10$ -dimensional observations (a conservative figure) and full covariance matrices, the size of (B) is equivalent to 1,645 scalar parameters. The problem with initialisation stems from the fact that only a set of training observation sequences is given, without knowledge of the states generating the observations. The training data permit an easy estimate of $p(o)$, the observation probability marginalised over the states; yet, our estimation targets are the conditional observation densities, $b_j(o) := p(o|q=j)$.

Before we start describing our initialisation approach, we illustrate a conventional method taking Murphy's toolbox as the reference. Parameter B requires to be initialized with $(N*M)$ sets of weighted Gaussian components, $\{\mu_{ji}, \Sigma_{ji}, c_{ji}\}$. Murphy's toolbox obtains such values in the following two steps:

1. **Cluster creation:** In the first step, $N * M$ initial clusters' centres are chosen and supplied to a k -means algorithm to cluster the observation data into $N * M$ clusters. In fact, the k -means algorithm produces the initial mean for each Gaussian of a single GMM with $N * M$ components. The Gaussian's covariance and its weight in the mixture are calculated based on the resulting clusters.

2. **Component dispatching:** The $N * M$ resulting components are dispatched over the N states (M components to each state) in "appearance" order.

In Murphy's toolbox, the $N * M$ initial centroids for k -means algorithm are chosen randomly from all the training data instances coalesced into a single *super vector* (Figure 1.a). While this method (called *random centres* hereafter) can produce effective initialisation, the training and verification phases might have to be applied several times before satisfactory parameters (in terms of classification accuracy) can be found.

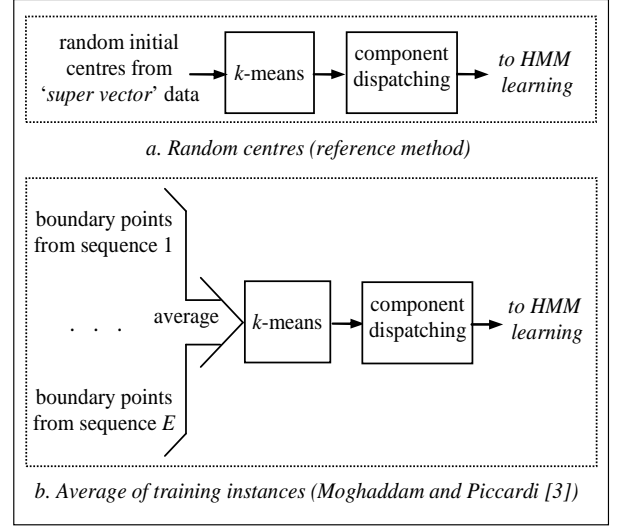


Figure 1: Different cluster creation methods for initialisation.

Conversely, deterministic initialisation methods permit one-off training of the HMM. Their main advantage is that of saving substantial training time which is significant as generally training has to be iterated many times over various experimental setups (datasets etc) and can prove very time consuming.

In the *average of training instances* deterministic initialisation method of [3], the clusters' centres are initialised by using time segmentation: the frame sequence of each training instance is divided into $(N * M + 1)$ consecutive segments and the boundary points between consecutive segments are collected and then averaged over all the instances to compose the initial centres (Figure 1.b). Such an initialisation is equivalent to assuming that each boundary point is in correspondence with a cluster's centre, and that the various clusters occur in a "left-to-right" sequence with equal duration.

An alternative to identifying centres by time segmentation is that of resorting to the marginal distribution of the observations, $p(o)$, and identify centres therein. Whereas mode interference may hide certain modes of the state-conditional distributions, a reasonable expectancy on modes' separation supports this approach. Considering this issue, we propose an approach based on an approximated histogram in the following subsection.

3.4. Histogram-based HMM initialisation

The method we propose to initialise the $(N*M)$ clusters' centres is based on the histogram of the "super vector". Our aim is to locate the positions of the main modes to then use them as initial clusters' centres. However, mode seeking in multi-dimensional data can prove inaccurate,

especially when the training data are not sufficiently dense. To mollify this issue, we compute 1-D histograms of each individual feature separately, and then the initial centre values are incrementally constructed feature-by-feature. For simplicity, we assume convex clusters and independence of the features.

Here, for each feature f_i , $i=1..F$, four consecutive steps are performed (Figure 2):

- i. $(N*M)$ equally-sized bins are formed from the super vector based on feature f_i . Then, we consider the samples falling in each bin and we compute the means of their first i co-ordinates.
- ii. The bin with the highest density percentage is chosen as one of the modes for feature f_i . Then, the density's percentage of the chosen bin is decreased by $(1/(N*M)) * 100$.
- iii. The procedure at the previous step is repeated $(N*M)$ times. The mean of the resulting modes are noted as $\{C_p\}$, $p=1..(N*M)$, each being an i -dimensional vector. A bin may end up being chosen more than once if its density remains greater than that of the other bins in successive iterations. By proceeding in this way, the selected bins account for the largest part of the measurements and equate to a quantisation step.
- iv. For the first feature, the value of the first co-ordinate of C_p , $p=1..(N*M)$, is assigned to be the first co-ordinate of the cluster's centre K_p , $p=1..(N*M)$. For the following features f_i , $i=2..F$, the $(i-1)$ co-ordinates of each K_p , $p=1..(N*M)$, are already calculated from the previous cycles. In the current cycle, for each K_p , the mode with minimum Euclidean distance based on the first $(i-1)$ co-ordinates is selected and its i -dimensional mean recorded as C_l , $l=1..(N*M)$. Then, the value of the i -th co-ordinate of C_l is assigned to be the i -th co-ordinate of K_p .

The underlying assumption of this procedure is that the modes of the multi-dimensional histogram can be approximately located based on the histogram's 1-D projections. Its main advantage is that a mode along one feature can be in correspondence with multiple modes along any other feature, therefore overcoming the main limitations normally associated with mode analysis from 1-D histogram projections.

4. Experiments

We have evaluated our cluster initialisation method with the recent MuHAVi (Multicamera Human Action Video) video dataset from Kingston University London [18]. Comparative experiments between the proposed method, the deterministic initialisation from [3] and random initialisations are presented in this section.

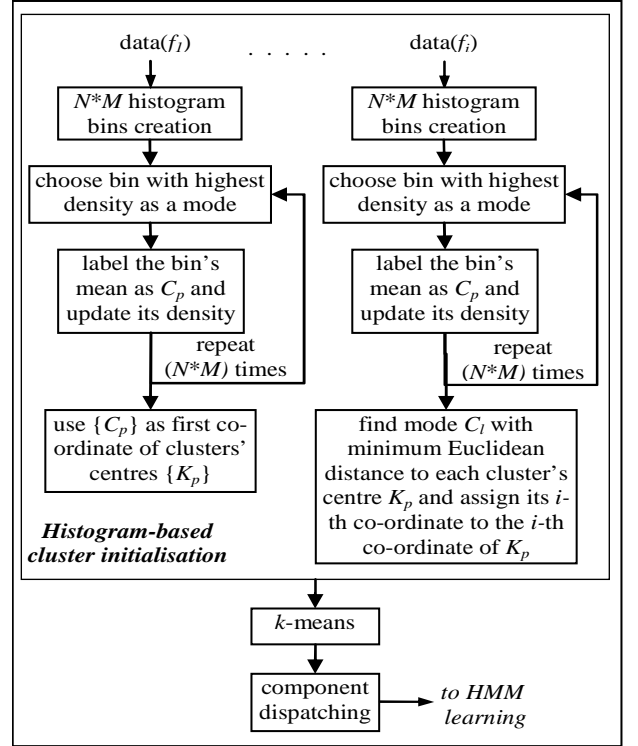


Figure 2: Histogram-based initialisation of clusters' centres.

4.1. Segmentation of the MuHAVi dataset

The MuHAVi dataset includes videos from 17 action classes performed several times by 7 different subjects and captured by 8 cameras simultaneously. To the best of our knowledge, it is the most comprehensive public action dataset to date in terms of combined number of action classes, subjects and cameras. In order to keep recognition and segmentation issues separate, the dataset aims to make manually annotated silhouettes available (MuHAVi-MAS). However, manual annotation is time consuming and the current number of manually-segmented sequences is rather limited (a total of 136 sequences for only two cameras). Furthermore, using manually generated masks for training or validation would provide an optimistic estimate of a method's accuracy compared to a real embedding where segmentation must be automated. Hence, we decided to use the main dataset and extract the masks automatically.

We chose the frame sequences of one camera (*camera4*) and applied a simple, but effective background subtraction technique. Assuming a fixed camera, a fixed scene and no major illumination changes, we fixed the background frame as the initial frame for the whole action time and provided foreground extraction by a thresholded difference. The inevitable errors were corrected by morphological operations, to join unconnected regions and remove smaller objects from foreground regions.

Table 1: Number of action samples with automated masks from the MuHAVi dataset (camera 4).

Action	Number of samples
ClimbLadder	7
CrawlOnKnees	29
DrawGraffiti	7
DrunkWalk	26
JumpOverFence	28
JumpOverGap	14
Kick	32
LookInCar	29
PickupThrowObject	26
PullHeavyObject	28
Punch	32
RunStop	28
ShotGunCollapse	29
SmashObject	20
WalkFall	28
WalkTurnBack	28
WaveArms	7
<i>Total</i>	398

Whereas in real-life application additional segmentation issues may occur (multiple targets, severe illumination changes etc), our procedure is far more realistic than manual segmentation. In this way, we automatically segmented 398 sequences from the *camera4* viewpoint as summarised in Table 1.

4.2. Projection histograms as feature set

Given the typical tight real-time constraints of video surveillance, we chose to extract a minimal set of projection histograms as the feature set for the HMM. Such descriptors project an object’s pixels onto the image coordinate axes and compute an horizontal and vertical histograms [19]. As an action takes place, the two projection histograms reflect the changes in the object’s shape and promise to be action-discriminative. They are invariant to scale, but sensitive to rotation.

To compute the histograms, we first calculate the coordinates of the bounding box surrounding the human silhouette in the foreground frame. Then, a window is centred on the bounding box’s centre and histogram computed. We use histograms with 10 bins each, leading to a total feature vector of length $F = 20$.

4.3. Experiments on cluster initialisation methods

As validation test, we have used “leave-one-subject-out” cross validation i.e. in each run we leave one subject out during training and we use it for testing. This validation procedure is sensible as in real applications subjects would not have been seen during training. The final accuracy result is the average over the various subjects (7 folds).

Table 2: Classification accuracy (%) with the different cluster initialisation methods.

Random centres (average and standard deviation of 5 runs)

	M=1	M=2	M=3	M=4	M=5	M=6
N=1	64.1	70.4±0.4	73.5±0.5	74.1±0.5	74.3±0.3	74.5±0.1
N=2	67.8±1.2	71.7±0.9	72.5±1.9	72.2±1.0	72.0±1.1	72.6±1.4
N=3	73.1±1.3	75.0±2.9	74.0±1.3	75.2±1.3	75.1±1.2	74.1±0.7
N=4	76.3±1.1	76.0±1.4	76.4±0.9	76.8±1.0	77.5±1.0	78.5±1.0
N=5	76.2±1.5	78.1±1.5	78.2±1.6	77.3±1.9	77.5±1.2	77.8±1.0
N=6	76.8±0.5	78.9±0.8	78.9±0.9	77.6±1.3	78.0±0.9	78.7±1.1

Random centres (best of 5 runs)

	M=1	M=2	M=3	M=4	M=5	M=6
N=1	64.1	71.1	74.4	74.6	74.6	74.6
N=2	69.1	73.1	74.1	73.9	73.9	74.4
N=3	74.4	79.6	75.9	76.6	76.4	74.9
N=4	77.6	77.4	77.4	78.1	78.6	79.6
N=5	77.9	79.1	80.2	79.4	79.1	79.1
N=6	77.6	79.9	80.2	78.9	79.4	79.9

Average of training instances [3]

	M=1	M=2	M=3	M=4	M=5	M=6
N=1	64.1	70.6	73.4	74.4	74.4	74.4
N=2	67.6	73.6	70.9	72.4	71.9	72.1
N=3	72.1	72.9	75.4	74.9	73.4	72.1
N=4	75.4	75.6	76.4	73.6	73.9	76.4
N=5	77.6	76.6	77.6	78.6	77.9	76.6
N=6	78.1	78.6	78.4	77.4	78.4	77.9

Histogram based cluster initialisation (Our method)

	M=1	M=2	M=3	M=4	M=5	M=6
N=1	64.1	70.6	73.1	74.1	74.6	74.6
N=2	67.6	72.4	73.4	69.6	71.6	73.4
N=3	72.4	74.9	77.9	74.9	75.1	73.4
N=4	74.6	75.9	77.6	77.9	79.9	77.4
N=5	76.6	76.9	78.1	77.9	77.6	77.6
N=6	76.9	77.1	79.4	80.4	77.4	78.9

We compared the proposed initialisation method with the *random centres* and the *average of training instances* proposed in [3]; yet, we tried to provide an improvement: as the time segments’ boundary points could be occasionally heavily affected by noise, we used the average of all the points in the respective segments to increase robustness. Table 2 reports the comparative classification accuracy.

For *random centres*, we report the average accuracy over 5 different random starts alongside the standard deviation, and also the best accuracy out of the 5 starts. Since the number of HMM states, N , and the number of components per state, M , are hyperparameters in the Baum-Welch algorithm, we decided to experiment over

range $N, M = \{1..6\}$ and choose the best combination based on cross-validation accuracy.

In the tests, our method obtained a best accuracy of 80.4% ($N = 6, M = 4$), higher than even the best of the five random runs (80.2%; $N = 5, M = 3$). The *average of training instances* proved the worst, with a best accuracy of only 78.6% ($N = 5, M = 4$), possibly due to its restrictive assumptions. If applied an unbound number of times, it is likely that the *random centres* method would eventually provide the highest accuracy for any combination of N and M . However, for practical reasons the number of starts must always be capped and would limit the trade off between accuracy and training time.

5. Conclusions

In this paper, we have proposed a deterministic method for learning initialisation of HMMs based on an approximate data histogram. The proposed approach aims to deliver an accurate model from a single initialisation, thus saving substantial learning time compared to multiple-starts methods. Experiment results showed that the proposed initialisation was capable of achieving better accuracy than the best of five random initialisations and the method previously proposed in [3]. We argue that the proposed approach can be usefully extended to other discrete latent-state models popular for human action recognition such as dynamic Bayesian networks and conditional random fields where the probability of observable random variables must be modelled conditional to discrete latent states.

6. References

- [1] T.B. Moeslund, A. Hilton and V. Krüger, "A Survey of Advances in Vision-Based Human Motion Capture and Analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, 2006, pp. 90-126.
- [2] P. Turaga, R. Chellappa, V.S. Subrahmanian and O. Udrea, "Machine Recognition of Human Activities: A Survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, 2008, pp. 1473-1488.
- [3] Z. Moghaddam and M. Piccardi, "Deterministic Initialisation of Hidden Markov Models for Human Action Recognition," in *Proc. of the Digital Image Computing: Techniques and Applications (DICTA 2009)*, 2009, pp. 188-195.
- [4] J. Yamato, J. Ohya and K. Ishii, "Recognizing Human Action in Time-Sequential Images using Hidden Markov Model," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 1992)*, 1992, pp. 379-385.
- [5] A.A. Efros, A.C. Berg, G. Mori and J. Malik, "Recognizing action at a distance," in *Proc. of the 9th IEEE Intl. Conf. on Computer Vision (ICCV 2003)*, 2003, pp. 726-733.
- [6] Z.L. Husz, A.M. Wallace and P.R. Green, "Human Activity Recognition with Action Primitives," in *Proc. of the IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance (AVSS 2007)*, 2007, pp. 330-335.
- [7] Y. Song, L. Goncalves and P. Perona, "Unsupervised learning of human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* vol. 25, no. 7, 2003, pp. 814-827.
- [8] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, "Actions as Space-Time Shapes," in *Proc. of the 10th IEEE Intl. Conf. on Computer Vision (ICCV 2005)*, 2005, pp. 1395-1402 Vol. 1392.
- [9] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. of the 2nd Joint IEEE Intl. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65-72.
- [10] H.-S. Chen, H.-T. Chen, Y.-W. Chen and S.-Y. Lee, "Human action recognition using star skeleton," in *Proc. of the 4th ACM Intl. Workshop on Video Surveillance and Sensor Networks*, ACM, 2006.
- [11] N. Li and D. Xu, "Action recognition using weighted three-state Hidden Markov Model," in *Proc. of the 9th Intl. Conf. on Signal Processing (ICSP 2008)*, 2008, pp. 1428-1431.
- [12] Y. Shen and H. Foroosh, "View-invariant action recognition using fundamental ratios," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008, pp. 1-6.
- [13] H. Fujiyoshi and A.J. Lipton, "Real-time human motion analysis by image skeletonization," in *Proc. of the 4th IEEE Workshop on Applications of Computer Vision (WACV 1998)*, 1998, pp. 15-21.
- [14] M.A. Ferrer, I.G. Alonso and C.M. Travieso, "Influence of initialisation and stop criteria on HMM based recognisers," *Electronics Letters*, vol. 36, no. 13, 2000, pp. 1165-1166.
- [15] D.T. Toledano, J.G. Villardebo and L.H. Gomez, "Initialisation, training, and context-dependency in HMM-based formant tracking," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, 2006, pp. 511-523.
- [16] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, 1989, pp. 257-286.
- [17] K. Murphy, "Hidden Markov Model Toolbox for Matlab," <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>, 1998 (last updated in 2005).
- [18] D.I.R.C. Kingston-University, "The muhavi-mas database," <http://dipersec.king.ac.uk/MuHAVi-MAS/>, 2008 (last updated in 2009).
- [19] L. Goldmann, M. Karaman and T. Sikora, "Human Body Posture Recognition Using MPEG-7 Descriptors," in *Visual Communications and Image Processing*, vol. 5308 of *Proceedings of SPIE*, pp. 177-188, San Jose, Calif, USA, January 2004.