Fast Simplex-HMM for One-Shot Learning Activity Recognition

Mario Rodriguez, Carlos Orrite, Carlos Medrano
Zaragoza University
Zaragoza, Spain.
[mrodrigo, corrite, ctmedra]@unizar.es

Dimitrios Makris
Kingston University
London,UK.
D.Makris@kingston.ac.uk

Abstract

The work presented in this paper deals with the challenging task of learning an activity class representation using a single sequence for training. Recently, Simplex-HMM framework has been shown to be an efficient representation for activity classes, however, it presents high computational costs making it impractical in several situations. A dimensionality reduction of the features spaces based on a Maximum at Posteriori adaptation combined with a fast estimation of the optimal parameters in the Expectation Maximization algorithm are presented in this paper. As confirmed by the experimental results, these two modifications not only reduce the computational cost but also maintain the performance or even improve it. The process suitability is experimentally confirmed using the human activity datasets Weizmann, KTH and IXMAS and the gesture dataset ChaLearn.

1. Introduction

In recent years, the focus of activity recognition research has moved from constrained scenarios where videos were recorded under controlled settings to unconstrained scenarios such as videos shared on the Internet. Thanks to this shift, better generalization of recognition is obtained, allowing the use of trained systems in open domain. Nevertheless, many applications like surveillance, gesture recognition or ambient assisted living assume constrained scenarios, being a specific location or a specific subject, and therefore the learning of some specific features can improve the performance. The use of cross domain approaches and the use of even limited training data from the target domain may boost the system accuracy.

As mentioned before, the first datasets in activity recognition were constrained to controlled laboratory settings as the examples of Weizmann [5], KTH [19] or IXMAS [24]. Currently, many recognition systems obtain almost perfect performance on them, assuming training with substantial data. On the other hand, more recent datasets are composed by lots of training examples from unconstrained scenarios, obtained for instance from Youtube or movie extractions, such as HMDB51 [8], UCF101 [21] or OlympicSports [12].

A lot of work has been done in feature extraction design over the years. Recognition in constrained scenarios was properly solved with global descriptors such as Motion History Images, Motion Energy Images [2] [3], spatio-temporal shapes [27] or spatio-temporal volumes spanned by silhouette images [5]. The change of research focus from constrained to unconstrainted scenarios resulted to a shift from global descriptors to local descriptors, being local spatio-temporal descriptors more robust to variabilities [9] [4] [10] [7]. Recent state-of-the-art ad-hoc descriptors are based on hybrid models where spatio-temporal interest points are tracked during some frames obtaining a trajectory around which the descriptor is computed, being Dense Trajectories (DT) [22] and Improved Dense Trajectories (IDT) [23] the most successful method thus far. Finally, the improvement in computer computational capacities and encouraging results in many disciplines have inspired researches using Deep Neural Networks [25].

In spite of the system versatility obtained by the combination of the above training datasets and recognition algorithms, the achieved accuracy is not yet at the level required in many commercial applications and if any constrains are present in the target domain, other approaches can be explored. Fixing some of the settings such as the background, the viewpoint or the subject performing the activity has the advantage of suppressing in some degree the clutter introduced by them, leading to higher reliability. However, the acquisition of sequences of activities in a new scenario is expensive in time and may be tedious. While most methods obtaining state-of-the-art results use several examples of the same class for training, the use of a reliable one-shot learning method facilitates a correct modeling from the first recorded sequence and therefore accelerates the working of a customized system in a target scenario. Little research has been done in training human activity recognition systems with limited number of labeled examples although being an essential feature for such practical situations [20] [14] [26]

[17]. On the other hand, a wider study has been done in one-shot learning for gesture recognition, triggered by the ChaLearn gesture dataset and the 2011/2012 challenge [6].

The recent work presented in [18] shows an approach for modeling human activities using only one domain sequence in the called Simplex Hidden Markov Model (SHMM), which is presented in a framework that benefits from a transfer learning stage based on a Maximum at Posteriori (MAP) adaptation. The following division of one-shot learning problems was suggested. First, strict-one-shot-learning assumes only one training example available which is used to model a single class. After training several models (one per available example) of different classes separately, it is possible to combine these models in order to train a recognition system. *Seo and Milanfar* [20] follows this approach using a nearest-neighbor classifier. Second, relaxed-one-shot-learning process uses simultaneously multiple training examples available, assuming one per class. This relaxation allows sharing some information among the examples in order to model the classes or directly training a recognition system. Most approaches follow this description as *Yang et al.* [26] and *Orrite et al.* [14] do by creating a vocabulary of features using sequences of the different classes. The framework created in [18] is designed for the strict model although both models were tested. However, following the framework description a drawback in the SHMM arises. The computational cost of training an adaptation and an SHMM per training sequence is high. In this paper we propose to reduce this cost by modifying some of the framework stages and also we propose to extend the study to a different domain. So, the contributions proposed in this paper are summarized below:

1. Reduce computational cost by decreasing the number of Gaussians in a Universal Background Model (UBM) and by doing a fast estimation of the Expectation Maximization optimum. We call this novel approach Fast-SHMM.

2. Verify the versatility of the method in different domains by applying it to gesture recognition, in addition to human activity recognition.

The rest of the paper is divided in the following sections. Section 2 describes the MAP adapted SHMM. Section 3 introduces the two modifications that reduce the computational cost presenting the Fast-SHMM. Experiments conducted in Weizman, KTH, IXMAS and ChaLearn datasets using Strict One-shot Learning are presented in Section 4. Finally, conclusions are discussed in Section 5.

2. Simplex-HMM with MAP adaptation

This section briefly describes the SHMM framework complemented with an MAP adaptation of the features space.

In Figure 1 a flow diagram of the system proposed in [18] is depicted. From the wide range of features extractors available in the literature, IDT [23] has shown state-of-the-art performance in several challenging datasets and so it has been selected in the Features Extraction stage. The method extracts the IDTs from videos in public datasets of human activities, considered the source domain, and creates a UBM vocabulary modeled with a Gaussian Mixture Model (GMM) as in [16], representing general, person and scenario independent features. Once selected the target scenario, the first performance of each activity class is recorded and labeled. The corresponding IDTs are extracted from this initial training video and used in a twofold task. First, with the unordered IDTs, the UBM vocabulary is transferred to the target scenario using an MAP adaptation, and obtaining a sequence specific vocabulary. Second, the IDTs are grouped into temporal windows where they are soft-assigned to the adapted vocabulary, obtaining a Bag of Features (BoF) per window. Each BoF histogram is normalized so that it sums one, equivalent to say it belongs to a unit simplex. Finally, given an activity video, the proposed encoding represents the activity as a sequence of normalized BoF, $\mathcal{O} = \{O_1, \cdots, O_T\}$, each one belonging to the simplex $\Delta = \{\mathbf{v}_\lambda \in \mathbb{R}^K : v_{\lambda_i} \geq 0 : \sum_{k=1}^{K} v_{\lambda_k} = 1\}$. These observations are $\mathbb{R}^K$ vectors although the real dimensionality of the space is $(K-1)$. The sequence of normalized BoF can be used for training a classifier based on HMM, suitable for modeling time sequences. As HMM presents numerical problems when working with limited training data of high dimensionality [11], the use of an observation emission function in a simplex can be used to prevent these numerical instabilities.

Formally, the parameters of the HMM are $\theta = \{N, A, B, \pi\}$, N is the number of states, i.e., $S = \{S_1, \ldots, S_N\}$. Each observation, O_t is the emission produced by the hidden state z_t. A set of hidden states forms a sequence, $Z = \{z_1, \ldots, z_T\}$ where $z_t \in S$. $A = \{a_{ij}\}$ is the state transition matrix where a_{ij} represents the transition probability from state i to state j, $a_{ij} = p(z_{t+1} = S_j | z_t = S_i)$. $\pi = \{\pi_i\}$ is the initial state probability distribution where $\pi_i = p(z_1 = S_i)$, $1 \leq i \leq N$ being S_i the state at the beginning of the time series. Finally, B represents the observation probability distribution in every state where $b_j(O_t) = p(O_t \mid z_t = S_j)$. As previously mentioned, the observation space is the simplex Δ, which is a continuous space where SHMM finds a stable solution simplifying the observation model by fulfilling the condition $b_j(O_t) \geq 0 \ \forall O_t, \ O_t \in \Delta$. Specifically, $b_j(O_t)$ should be designed as a decreasing function with a maximum equal to 1 in a point in Δ, and its value decreases while it separates from that point, being always non-negative.

A fair comparison between histograms is the exponential

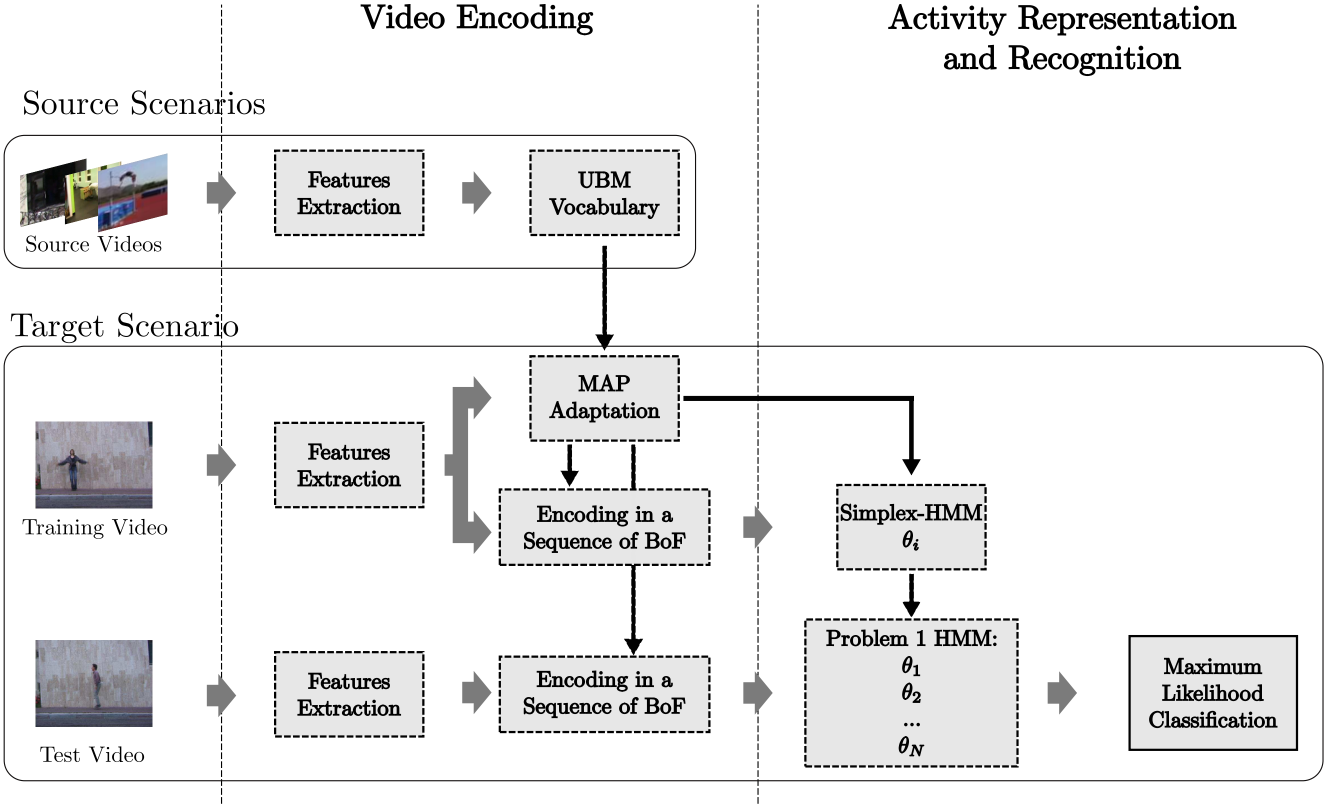

Figure 1. Flow diagram of the Simplex-HMM with MAP adaptation.

of the negative Hellinger distance. As vectors in the simplex may be considered normalized histograms then equation 1 is a valid observation model.

$$b_j(O_t) = e^{-\varphi\sqrt{\sum_{k=1}^{K}(\sqrt{v^t_{\lambda_k}}-\sqrt{m_{jk}})^2}} \tag{1}$$

Given one, or several training observations, the HMM parameters can be estimated using the Maximum Likelihood through a Baum-Welch algorithm. This iterative estimation is obtained by maximizing the Baum's auxiliary function $Q(\hat{\theta},\theta)$ [15] [1].

$$Q(\hat{\theta},\theta) = \sum_{Z} p(Z|\mathcal{O},\theta)\ln p(\mathcal{O},Z|\hat{\theta}) \tag{2}$$

Defining $\gamma_t(i) = p(z_t = S_i|\mathcal{O},\theta)$ and $\xi_t(i,j) = p(z_t = S_i, z_{t+1} = S_j|\mathcal{O},\theta)$, the function Q can be expressed as:

$$Q(\hat{\theta},\theta) = \sum_{j=1}^{N}\gamma_1(j)\ln\pi_j + \sum_{t=1}^{T-1}\sum_{i=1}^{N}\sum_{j=1}^{N}\xi_t(i,j)\ln a_{ij} + \sum_{t=1}^{T}\sum_{j=1}^{N}\gamma_t(j)\ln(b_j(O_t)) \tag{3}$$

The proposed $b_j(O_t)$ requires a special modification on the EM algorithm. First, a change in notation is done in order to simplify the equations. The terms $\sqrt{v^t_{\lambda_k}}$ are named without the square root so $\sqrt{v^t_{\lambda_k}} \rightarrow \tilde{v}^t_{\lambda_k}$. This change implies a notational change in the constraints so $\sum_{k=1}^{K} v_{\lambda_k} = 1 \rightarrow \sum_{k=1}^{K}\tilde{v}^2_{\lambda_k} = 1$. On the other hand, the same change can be done for the terms $\sqrt{m_{jk}}$, naming them $\tilde{m}_{jk}$ directly. Now, the observation model is defined by equation 4.

$$b_j(O_t) = e^{-\varphi\sqrt{\sum_{k=1}^{K}(\tilde{v}^t_{\lambda_k}-\tilde{m}_{jk})^2}} \tag{4}$$

With this observation model the EM algorithm is processed modifying the M-step in order to maximize the third term in equation 3 with respect to $\tilde{m}_{jk}$.

$$\sum_{t}\sum_{j}\gamma_t(j)\ln(b_j(O_t)) = \sum_{t}\sum_{j}\gamma_t(j)\left(-\varphi\sqrt{\sum_{k=1}^{K}(\tilde{v}^t_{\lambda_k}-\tilde{m}_{jk})^2}\right) \tag{5}$$

By setting $\frac{\partial}{\partial\tilde{m}_{jk}} = 0$, the following equation is obtained:

$$\varphi\sum_{t=1}^{T}\gamma_t(j)\frac{(\tilde{v}^t_{\lambda_k}-\tilde{m}_{jk})}{\sqrt{\sum_{k'=1}^{K}(\tilde{v}^t_{\lambda_{k'}}-\tilde{m}_{jk'})^2}} = 0 \tag{6}$$

Since $\tilde{m}_{jk}$ does not depend on t and $\gamma_t(j)$ are treated as constants in the M-step once computed in the E-step, parameters maximization is performed by using the fixed point iteration explained in Algorithm 1.

Algorithm 1

```
Randomly initialize $\tilde{m}_{jk} \ni \sum_{k=1}^{K} \tilde{m}_{jk}^2 = 1$
$\epsilon_m = \infty$
while $\epsilon_m > \epsilon$ do
    $\tilde{m}'_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j) \frac{\tilde{v}^t_{\lambda_k}}{\sqrt{\sum_{k'=1}^{K} (\tilde{v}^t_{\lambda_{k'}} - \tilde{m}_{jk'})^2}}}{\sum_{t=1}^{T} \gamma_t(j) \frac{1}{\sqrt{\sum_{k'=1}^{K} (\tilde{v}^t_{\lambda_{k'}} - \tilde{m}_{jk'})^2}}}$
    $\epsilon_m = \max_k |\tilde{m}_{jk} - \tilde{m}'_{jk}|$
    $\tilde{m}_{jk} = \tilde{m}'_{jk}, k = 1...K$
end while
```

It should be noted that the optimization process can be performed separately for each j.

3. Fast Simplex Hidden Markov Model

An important drawback of the SHMM algorithm proposed in [18] is its high computational cost. In this section we identify two stages that negatively affect the computational cost and we propose two methods to reduce their influence. First, the soft-assignment of the local features in a high dimensional space is performed for every extracted feature in both training and testing stages. Such a high dimensionality not only slows down the soft-assignment but also any computation in the method. We propose to reduce this dimensionality by taking advantage of the MAP-adaptation performed in the training process. Second, the iterative method previously described in Algorithm 1 increases the needed time for training. We propose the acceleration of the process by making a fast estimation of the optimum avoiding the iteration.

3.1. Reduced MAP Adaptation

One of the pillars that supports SHMM framework is the MAP adaptation of the features space. In the Strict One-shot learning process the MAP adaptation is processed for every training sequence in the target scenario, obtaining a GMM per sequence $\widehat{\lambda}_n(\widehat{\mu}_n, \Sigma), 1 \leq n \leq N$. The proposed framework assumes shared covariance matrices among models and no weights for the Gaussians, restricting the modification per model to the centroids only. For a BoF encoding the number of Gaussians, K, is several thousands and if we have N training sequences we need to store KN centroids. But the computational cost is even higher, because the evaluation of every new descriptor should be made against all these KN centroids. Given the set of descriptors $\mathbf{Q} = \{\mathbf{q}_m\}$, $\mathbf{q}_m \in \mathbb{R}^D, 1 \leq m \leq M$, for every Gaussian $\widehat{\lambda}_{ni}$ in the MAP-adapted UBMs, the probabilistic alignment of the feature vectors is computed by (7). If $\mathbf{Q}$ represents the test set of descriptors, this equation should be processed KNM times in test time.

$$p(\widehat{\lambda}_{ni}|\mathbf{q}_m) = \frac{\mathcal{N}(\mathbf{q}_m|\widehat{\mu}_{ni}, \Sigma_i)}{\sum_{k=1}^{K} \mathcal{N}(\mathbf{q}_m|\widehat{\mu}_{nk}, \Sigma_k)} \tag{7}$$

On the other hand, if $\mathbf{Q}$ represents the set of training descriptors in one sequence then, the MAP Adaptation uses this same probabilistic alignment for the calculus of the GMM transformation. In this case the UBM is represented only by one GMM, $\lambda(\mu, \Sigma)$. These probabilistic alignments and the feature vectors are used to compute the sufficient statistics where the initial step is the estimation of the global probabilistic alignment using (8).

$$n_i = \sum_{m=1}^{M} p(\lambda_i|\mathbf{q}_m) \tag{8}$$

For each Gaussian λ_i in the UBM the global probabilistic alignment gives the information on how related are the training data to the specific Gaussian. A large n_i means that there are some $\mathbf{q}_m$ samples close to the Gaussian and then this Gaussian is representative for the data, a small n_i means that the data is separated from the Gaussian and this Gaussian has low relevance to the samples in $\mathbf{Q}$. Therefore, we can use n_i not only for the MAP-adaptation but also as an activation parameter per Gaussian. Thus, we use a threshold over n_i. An $n_i \geq \xi$ means that the Gaussian is active and therefore MAP adapted, otherwise the Gaussian is discarded, as depicted in Figure 2. Finally, the MAP-Adapted GMM is composed by the adapted Gaussians that fulfil the threshold restriction as described in (9).

$$\widehat{\lambda} = \{\widehat{\lambda}_i\}, \ \forall i \ni n_i \geq \xi \tag{9}$$

Threshold ξ controls the reduction of the computational cost and storage. An ideal value of ξ should lead to an appropriate trade-off between computational cost and recognition accuracy. After the reduction each model has a different number K of Gaussians, and if the average number is $\overline{K} < K$ then, the final number of times the probabilistic alignment equation should be processed is $\overline{K}NM < KNM$.

3.2. Fast Estimation of the Optimal Parameters

It is worth noting that in strict-one-shot-learning paradigm the activity model is optimized using only one sequence and an optimization of the model to this single example might risk an over-fitting. Then, we propose to avoid the optimization method and substitute it with a direct estimation of the maximum of (5).

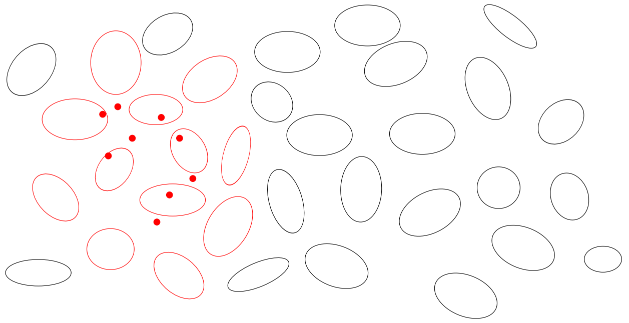

Figure 2. Gaussian model reduction. Red dots represent **Q** and red ellipses represent Gaussians fulfilling $n_i \geq \xi$.

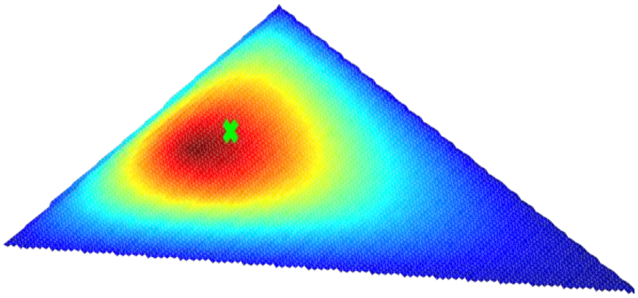

Figure 3. Difference between the position of the maximum in (5) and the fast estimation provided by (10) in a toy example of a 3-dimensional simplex. The green cross is the estimation and the real values of (5) are evaluated all over the simplex, with red colors representing the highest values.

The maximum value of (5) might be estimated using a trade-off among all the training observations, when higher values of $\ln b_j(O_t)$ are weighted by higher values of $\gamma_t(j)$. Although the trade-off can be different depending on the emission function, we propose the simple direct weight of the observations with the $\gamma_t(j)$ values through (10).

$$\hat{m}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j) v_{\lambda_k}^t}{\sum_{t=1}^{T} \gamma_t(j)} \tag{10}$$

Figure 3 shows the graphical representation of this approach in a toy example with a 3D simplex. $\gamma_t(j)$ values and O_t observations are randomly obtained. The triangle represents the $(\ln b_j(O_t))$ values along the whole simplex with colors, being red the maximum and blue the minimum. Additionally, the estimated value is represented by a green cross. The figure shows that the direct estimation provides a satisfactory approximation. Since only one sequence is available for training, the optimum of (5) is specific for the sequence, which may not be the optimum for the activity class, and the estimation is close enough to provide a good approach. Since this estimation is not optimal it may lead to a decrease in the log-likelihood. Therefore, we have included a condition in the EM algorithm so that the iterative process will be terminated if the log-likelihood decreases. Thanks to this approach we avoid possible convergence problems in EM and as shown in some experiments the time reduction is significant in training.

4. Experiments

So far, we have explained the Fast-SHMM framework where a MAP adaptation of the UBM of features space is computed per training example. This novel framework includes two methods to reduce the high computational cost of the original SHMM described in the previous section. However, the speed and the accuracy after the modifications should be tested. The following experiments show the performance of the Fast-SHMM proposed in this paper compared with the original method described in [18]. The comparison is performed using the same three datasets tested in the cited paper. In addition, a novel experiment is conducted in order to evaluate the framework adaptability to a different domain. Other one-shot learning algorithms have been previously proposed for gesture recognition and therefore the Fast-SHMM is tested in that domain.

4.1. Datasets

Fast-SHMM is focused on human activity recognition applied on constrained scenarios, where videos are obtained by fixed viewpoint cameras. The proposed algorithm is trained using human motion information from external video sources using MAP adaptation, as described in Section 2. The method is evaluated using several datasets that accomplish the source and target domain constraints. We have selected three source domain datasets that include a high variability in unconstrained video clips that simulate the easily obtainable ones from the Internet. On the other hand, we have selected three popular datasets in the human activity recognition field as target domain where the videos are recoded from fixed cameras.

The configuration proposed in [18] is followed. IDTs [23] features are extracted and used for creating a 5000 Gaussians UBM. On the other hand, the parameter φ of the emission function is set to 1.57.

Source Domain Datasets Three public and extensive datasets, HMDB51 [8], OlympicSprots [12] and Virat Release 2.0 [13], are used as source domain. They include a high variability of movements in several locations. The three datasets combined have 79 different activity classes extracted from Youtube, movies or surveillance cameras in 7878 video clips. Randomly selecting 100000 IDTs from the three datasets, a UBM of 5000 Gaussians is obtained and used as base for all the experiments, including the gesture recognition.

Target Domain Datasets The Weizmann dataset [5] is composed by 93 low-resolution (180 x 144, 50 fps) video sequences showing nine different people, each performing

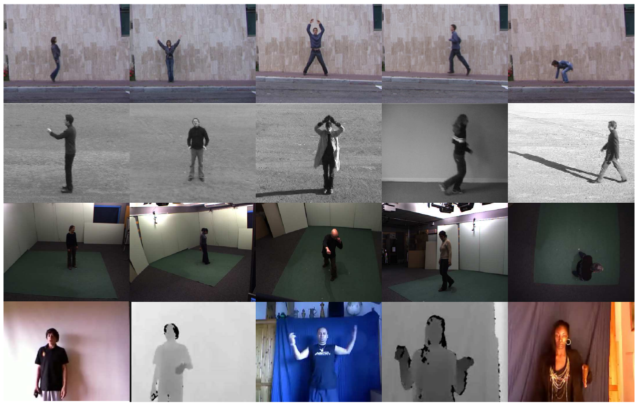

Figure 4. Target Domain Datasets: Weizmann (1st row), KTH (2nd row), IXMAS (3rd row) and ChaLearn (4rd row).

10 natural activities: *bend, jumping-jack, jump-forward-on-two-legs, jump-in-place-on-two-legs, run, gallop-sideways, skip, walk, wave-one-hand* and *wave-two-hands*. The IXMAS dataset [24] is composed by 5 camera viewpoints (390 x 291, 23 fps) of 11 actors performing 3 times each of the 13 activities included: *check-watch, cross-arms, scratch-head, sit-down, get-up, turn-around, walk, wave, punch, kick, point, pick-up* and *throw*. The KTH dataset [19] has been captured in 4 different scenarios where static cameras have recorded, at low-resolution (160 x 120, 25 fps), 25 subjects performing several times six types of activities: *walking, jogging, running, boxing, hand-waving* and *hand-clapping*.

Additionally, a dataset designed for one-shot-learning gesture recognition has been selected for validating the adaptability of the method to a different domain. The ChaLearn gesture dataset [6] includes more than 50000 gestures recorded with the Kinect^{TM} sensor, providing both RGB and depth videos. The gestures are grouped into more than 500 batches of 100 gestures, each batch including one example per class for training. The gestures come from over 30 different vocabularies and were performed by 20 different users. For the sake of comparison with the literature we only use the final 40 batches used in two challenges (*final1* and *final2*) composed by 20 batches each and the 20 batches provided for validation (*valid*). Frame examples of these datasets are shown in Figure 4.

4.2. Computational cost improvement

In order to validate the usefulness of the reduced MAP adaptation, the computational cost of three stages of the algorithms are evaluated: training of SHMM (TR), activity encoding (AE) and log-likelihood computation in SHMM (LgC). Experiments are run in Matlab in an Intel i7-4790 CPU at 3600 GHz using the Weizmann dataset. As done in [18] we select a UBM composed by $K = 5000$ Gaussians. For the reduction stage we select $\xi = 10^{-10}$, which is a small activation value despite producing a significant reduction, for Weizmann dataset an average of $\overline{K} = 373$. Table 1 shows the spent time in each of the stages evaluated using the original UBM or the reduced model.

Table 1. Comparison of spent time (s) in the original UBM and the reduced adaptation evaluating three stages of the process: training (TR), activity encoding (AE) and log-likelihood computation (LgC).

	$K = 5000$	$\overline{K} = 373$
TR	0.7035	0.1324
AE	0.8653	0.042
LgC	0.0054	0.0012

Table 2. Comparison of spent time (s) training the Simplex-HMM using the optimization of Algorithm 1 and the fast estimation of (10).

	Alg. 1	Fast
$K = 5000$	0.7035	0.1631
$\overline{K} = 373$	0.1324	0.0313

Results support the advantage of using the reduced model as the computational cost is significantly lower. In the SHMM training only a 18.8% of time is required with the proposed method. Even greater is the benefit obtained in the activity encoding (discarding the IDT computation) where only a 4.8% of the original time is required for computation. Moreover, this is the slower stage of the process and is used in both, training and testing stages. Finally, the less restrictive stage is the log-likelihood computation because it was fast even before the reduction. However, the reduction to a 22.2% of the original time is worth noting as it is computed in testing and after the improvement in the activity encoding its cost is representative.

On the other hand, the computational cost benefits experimented with the fast estimation are tested using Weizmann dataset again and the previously described computer. Table 2 shows the time spent in the HMM training comparing the EM using Algorithm 1 (Alg. 1) and the fast estimation (Fast). Thanks to the fast estimation the training time is reduced to a 23% of the original time in both cases: using $K = 5000$ Gaussians or using the reduced model with only $\overline{K} = 373$ Gaussians.

4.3. Recognition accuracy

In this section the validity of the Fast-SHMM is firstly tested using two experiments conducted in the three constrained datasets previously mentioned: Weizmann, KTH and IXMAS. Table 3 represents the accuracy of the methods evaluated. The performance of the *Seo and Milan-*

Table 3. Accuracy in Strict One-shot learning using the proposed improvements and comparison with other researches.

	Weizmann	KTH	IXMAS
Seo and Milanfar [20]	75%	65%	-
SHMM [18]	**81.9%**	70.4%	44.6%
SHMM + $\overline{K}$ Reduction	81.1%	71.8%	46.6%
FSHMM	81.5%	**74%**	**47.4%**

far method, as reported in [20] and the original SHMM, as shown in [18] are shown in the first two rows. Accuracy obtained with the reduced MAP adaptation is shown in third row with the name ($\overline{K}$ Reduction). After reduction, the average number of Gaussians is different for each dataset. Considering an original UBM of $K = 5000$ Gaussians and using a threshold of $\xi = 10^{-10}$, $\overline{K} = 373$ in Weizmann, $\overline{K} = 712$ in KTH and $\overline{K} = 1155$ in IXMAS. Two justification may be given for the variety of the $\overline{K}$ values: first, activities with varied movements have more Gaussians activated; second, longer sequences have more extracted features increasing the activated Gaussians as the activation equation is a direct addition without normalization. The last row shows results for the combination of the reduced MAP adaptation and the fast estimation of optimal parameters covering the proposed Fast-SHMM (FSHMM) are shown.

All the experiments follow the strict-one-shot-learning paradigm and, except in Weizmann dataset where differences are not meaningful, the $\overline{K}$ Reduction method improves the original accuracy and Fast-SHMM increases this improvement. Thanks to these experiments we can conclude that the proposed acceleration of Fast-SHMM not only reduces the computational cost but also improves the accuracy.

Finally, using the validated Fast-SHMM, final batches of ChaLearn gesture dataset are tested. Table 4 shows the Levenshtein distance, an error measure used in the ChaLearn challenges [6], for the 20 batches provided for validation (*valid*) and the two challenges of the ChaLearn gesture dataset (*final1* and *final2*) providing the average of the results as final score. Fast-SHMM is compared with the 15 best groups presented in the challenges and the baseline methods provided by the organizers. The rows are ordered from the lower Levenshtein distance (better) to the higher (worse) in the average column.

Table 4. Levenshtein distances in ChaLearn gesture dataset for the 15 best groups of the challenges [6], the baseline methods provided by the organization and our proposal.

	valid	final1	final2	Average
Alfnie2	9,95	7,34	7,10	8,13
Alfnie1	14,26	9,96	9,15	11,12
Pennect	17,97	16,52	12,31	15,60
TurtleTamers	20,84	17,02	10,98	16,28
Joewan	18,24	16,80	14,48	16,51
Immortals	24,88	18,47	18,53	20,63
OneMillionMonkeys	28,75	16,85	18,19	21,26
Manavender	25,59	21,64	19,25	22,16
WayneZhang	28,19	23,03	16,08	22,43
FSHMM-T	**26,37**	**21,08**	**20,69**	**22,71**
SkyNet	28,25	23,30	18,41	23,32
Zonga	27,14	23,03	21,91	24,03
BalazsGodeny	27,14	23,14	26,79	25,69
FSHMM-M	**28,08**	**25,09**	**24,68**	**25,95**
HITCS	32,45	28,25	20,08	26,93
XiaoZhuWudi	29,30	25,64	26,07	27,00
FSHMM-K	**29,30**	**26,37**	**25,46**	**27,04**
Vigilant	30,90	28,09	22,35	27,11
Baseline method 2	38,14	29,97	31,72	33,28
Baseline method 1	59,76	62,51	56,46	59,58

Results for our proposal are divided into three rows depending on the data used for the model. (K) uses only depth data, (M) uses only RBG data and (T) combines both by adding the computed log-likelihood of their respective Simplex-HMM. It is important to note that our experiments do not use the sequences in the development batches of the ChaLearn dataset. The UBM trained with human activities, previously explained, does not contain samples of gestures. Moreover, our proposal is strict-one-shot-learning, which is more restrictive than the relaxed one-shot-learning approach used by other methods. Finally, FSHMM approaches use a basic segmentation where all gestures has the same length, as the baseline methods do. Nevertheless, we can observe how results from FSHMM are among these 15 teams out of 85 and overcome both baseline methods, being FSHMM-T exactly the 10th best, despite of the above experimental limitations.

5. Conclusions

Results have shown how the proposed algorithm modifications have reduced significantly the computational cost. Computational times can be further improved, if more efficient languages as C or C++ are used for the implementation, instead of Matlab.

Accuracy improvement obtained in Fast-SHMM can be explained as follows: On the one hand, most of the UBM Gaussians only introduce noise in the encoding as a reduced number of them provide better results in general. On the other hand, optimizing the Simplex-HMM for one class using only one training example might over-fit the model as using an approximation gives better results.

While this paper focuses on one-shot learning, the Fast-SHMM could also be used if several training sequences are available in the target domain. However, in this case, future work should resolve the issue that the computational cost would increase linearly with the increase of training sequences in the target scenario. Another challenge is to make the algorithm adapt to new sequences, while they are being recorded.

Finally, it is worth noting how the Simplex-HMM with an MAP adaptation can be used in several contexts as the case of gestures, even if the features space has been trained in a different domain (namely human activities). Better results should be obtained in the ChaLearn datasets if the development batches are used to create the UBM.

6. Acknowledgments

This work was partially supported by Spanish Grant TIN2013- 45312-R (MINECO), Gobierno de Aragon and the European Social Found.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] A. Bobick and J. Davis. Real-time recognition of activity using temporal templates. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 39–42, Dec 1996.

[3] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, Mar. 2001.

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *International Conference on Computer Communications and Networks (ICCCN)*, pages 65–72, 2005.

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[6] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante. The ChaLearn gesture dataset (cgd 2011). *Machine Vision and Applications*, 25(8):1929–1951, 2014.

[7] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *European Conference on Computer Vision (ECCV)*, 2012.

[8] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[9] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64:107–123, 2005.

[10] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2008.

[11] T. P. Minka. Estimating a Dirichlet distribution. Technical report, 2009.

[12] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision (ECCV)*, pages 392–405, 2010.

[13] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3153–3160, 2011.

[14] C. Orrite, M. Rodriguez, and M. Montañes. One-sequence learning of human actions. In *Human Behavior Unterstanding*, pages 40–51, 2011.

[15] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, feb 1989.

[16] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[17] M. Rodriguez, C. Medrano, E. Herrero, and C. Orrite. Transfer learning of human poses for action recognition. In *Human Behavior Unterstanding*, 2013.

[18] M. Rodriguez, C. Orrite, C. Medrano, and D. Makris. One-shot learning of human activity with an map adapted gmm and simplex-hmm. *IEEE Transactions on Cybernetics*, PP(99):1–12, 2016.

[19] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *International Conference on Pattern Recognition (ICPR)*, 2004.

[20] H. J. Seo and P. Milanfar. Action recognition from one example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):867–882, 2011.

[21] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[22] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103:60–79, 2013.

[23] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[24] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.

[25] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 461–470, New York, NY, USA, 2015. ACM.

[26] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1635–1648, July 2013.

[27] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 984–989, June 2005.