Chapter 5
Language-Motivated Approaches to Action Recognition

Manavender R. Malgireddy, I. Nwogu and V. Govindaraju

Abstract We present language-motivated approaches to detecting, localizing and classifying activities and gestures in videos. In order to obtain statistical insight into the underlying patterns of motions in activities, we develop a dynamic, hierarchical Bayesian model which connects low-level visual features in videos with poses, motion patterns and classes of activities. This process is somewhat analogous to the method of detecting topics or categories from documents based on the word content of the documents, except that our documents are dynamic. The proposed generative model harnesses both the temporal ordering power of dynamic Bayesian networks such as hidden Markov models (HMMs) and the automatic clustering power of hierarchical Bayesian models such as the latent Dirichlet allocation (LDA) model. We also introduce a probabilistic framework for detecting and localizing pre-specified activities (or gestures) in a video sequence, analogous to the use of filler models for keyword detection in speech processing. We demonstrate the robustness of our classification model and our spotting framework by recognizing activities in unconstrained real-life video sequences and by spotting gestures via a one-shot-learning approach.

Keywords Dynamic hierarchical Bayesian networks · Topic models · Activity recognition · Gesture spotting · Generative models

Editors: Isabelle Guyon and Vassilis Athitsos.

M.R. Malgireddy (✉) · I. Nwogu · V. Govindaraju
Department of Computer Science and Engineering, University at Buffalo, SUNY,
Buffalo, NY 14260, USA
e-mail: mrm42@buffalo.edu

I. Nwogu
e-mail: inwogu@buffalo.edu

V. Govindaraju
e-mail: govind@buffalo.edu

S. Escalera et al. (eds.), *Gesture Recognition*, The Springer Series
on Challenges in Machine Learning, DOI 10.1007/978-3-319-57021-1_5

5.1 Introduction

Vision-based activity recognition is currently a very active area of computer vision research, where the goal is to automatically recognize different activities from a video. In a simple case where a video contains only one activity, the goal is to classify that activity, whereas, in a more general case, the objective is to detect the start and end locations of different specific activities occurring in a video. The former, simpler case is known as *activity classification* and latter as *activity spotting*. The ability to recognize activities in videos, can be helpful in several applications, such as monitoring elderly persons; surveillance systems in airports and other important public areas to detect abnormal and suspicious activities; and content based video retrieval, amongst other uses.

There are several challenges in recognizing human activities from videos and these include videos taken with moving background such as trees and other objects; different lighting conditions (day time, indoor, outdoor, night time); different view points; occlusions; variations within each activity (different persons will have their own style of performing an activity); large number of activities; and limited quantities of labeled data amongst others.

Recent advances in applied machine learning, especially in natural language and text processing, have led to a new modeling paradigm where high-level problems can be modeled using combinations of lower-level segmental units. Such units can be learned from large data sets and represent the universal set of alphabets to fully describe a vocabulary. For example, in a high-level problem such as speech recognition, a phoneme is defined as the smallest segmental unit employed to form an utterance (speech vector). Similarly, in language based documents processing, words in the document often represent the smallest segmental unit while in image-based object identification, the bag-of-words (or bag-of-features) technique learns the set of small units required to segment and label the object parts in the image. These features can then be input to generative models based on hierarchical clustering paradigms, such as topic modeling methods, to represent different levels of abstractions.

Motivated by the successes of this modeling technique in solving general high-level problems, we define an activity as a sequence of contiguous sub-actions, where the sub-action is a discrete unit that can be identified in a action stream. For example, in a natural setting, when a person waves goodbye, the sub-actions involved could be (i) raising a hand from rest position to a vertical upright position; (ii) moving the arm from right to left; and (iii) moving the arm from left to right. The entire activity or gesture[1] therefore consists of the first sub-action occurring once and the second and third sub-actions occurring multiple times. Extracting the complete vocabulary of sub-actions in activities is a challenging problem since the exhaustive list of sub-actions involved in a set of given activities is not necessarily known beforehand. We therefore propose machine learning models and algorithms to (i) compose a compact, near-complete vocabulary of sub-actions in a given set of activities; (ii) recognize the

[1] When referring to activity spotting purposes, we use the term gestures instead of activities, only to be consistent with the terminology of the *ChaLearn Gesture Challenge*.

specific actions given a set of known activities; and (iii) efficiently learn a generative model to be used in recognizing or spotting a pre-specified action, given a set of activities.

We therefore hypothesize that the use of sub-actions in combination with the use of a generative model for representing activities will improve recognition accuracy and can also aid in activity spotting. We will perform experiments using various available publicly available benchmark data sets to evaluate our hypothesis.

5.2 Background and Related Work

Although extensive research has gone into the study of the classification of human activities in video, fewer attempts have been made to spot actions from an activity stream. A recent, more complete survey on activity recognition research is presented by Aggarwal and Ryoo (2011). We divide the related work in activity recognition into two main categories: activity classification and activity spotting.

5.2.1 Activity Classification

Approaches for activity classification can be grouped into three categories: (i) space-time approaches: a video is represented as a collection of space-time feature points and algorithms are designed to learn a model for each activity using these features; (ii) sequential approaches: features are extracted from video frames sequentially and a state-space model such as a hidden Markov model (HMM) is learned over the features; (iii) hierarchical approaches: an activity is modeled hierarchically, as combination of simpler low level activities. We will briefly describe each of these approaches along with the relevant literature, in sections below.

5.2.1.1 Space-Time Approaches

Space-time approaches represent a video as a collection of feature points and use these points for classification. A typical space-time approach for activity recognition involves the detection of interest points and the computation of various descriptors for each interest point. The collection of these descriptors (bag-of-words) is therefore the representation of a video. The descriptors of labeled training data are presented to a classifier during training. Hence, when an unlabeled, unseen video is presented, similar descriptors are extracted as mentioned above and presented to a classifier for labeling. Commonly used classifiers in the space-time approach to activity classification include support vector machines (SVM), K-nearest neighbor (KNN), etc.

Spatio-temporal interest points were initially introduced by Laptev and Lindeberg (2003) and since then, other interest-point-based detectors such as those based on spatio-temporal Hessian matrix (Willems et al. 2008) and Gabor filters (Bregonzio et al. 2009; Dollár et al. 2005) have been proposed. Various other descriptors such as those based on histogram-of-gradients (HoG) (Dalal and Triggs 2005) or histogram-of-flow (HoF) (Laptev et al. 2008), three-dimensional histogram-of-gradients (HoG3D) (Kläser et al. 2008), three-dimensional scale-invariant feature transform (3D-SIFT) (Scovanner et al. 2007) and local trinary patterns (Yeffet and Wolf 2009), have also been proposed to describe interest points. More recently, descriptors based on tracking interest points have been explored (Messing et al. 2009; Matikainen et al. 2009). These use standard Kanade-Lucas-Tomasi (KLT) feature trackers to track interest points over time.

In a recent paper by Wang et al. (2009), the authors performed an evaluation of local spatio-temporal features for action recognition and showed that dense sampling of feature points significantly improved classification results when compared to sparse interest points. Similar results were also shown for image classification (Nowak et al. 2006).

5.2.1.2 Sequential Approaches

Sequential approaches represent an activity as an ordered sequence of features, here the goal is to learn the order of specific activity using state-space models. HMMs and other dynamic Bayesian networks (DBNs) are popular state-space models used in activity recognition. If an activity is represented as a set of hidden states, each hidden state can produce a feature at each time frame, known as the observation. HMMs were first applied to activity recognition in 1992 by Yamato et al. (1992). They extracted features at each frame of a video by first binarizing the frame and dividing it into $(M \times N)$ meshes. The feature for each mesh was defined as the ratio of black pixels to the total number of pixels in the mesh and all the mesh features were concatenated to form a feature vector for the frame. An HMM was then learned for each activity using the standard Expectation-Maximization (EM) algorithm. The system was able to detect various tennis strokes such as forehand stroke, smash, and serve from one camera viewpoint. The major drawback of the conventional HMM was its inability to handle activities with multiple persons. A variant of HMM called coupled HMM (CHMM) was introduced by Oliver et al. (2000), which overcame this drawback by coupling HMMs, where each HMM in the CHMM modeled one person's activity. In their experiments they coupled two HMMs to model human-human interactions, but again this was somewhat limited in its applications. An approach to extend both HMM and CHMMs by explicitly modeling the duration of an activity using states was also proposed by Natarajan and Nevatia (2007). Each state in a coupled hidden semi-Markov model (CHSMMs) had its own duration and the sequence of these states defined the activity. Their experiments showed that CHSMM modeled an activity better than the CHMM.

5.2.1.3 Hierarchical Approaches

The main idea of hierarchical approaches is to perform recognition of higher-level activities by modeling them as a combination of other simpler activities. The major advantage of these approaches over sequential approaches is their ability to recognize activities with complex structures. In hierarchical approaches, multiple layers of state-based models such as HMMs and other DBNs are used to recognize higher level activities. In most cases, there are usually two layers. The bottom layer takes features as inputs and learns atomic actions called *sub-actions*. The results from this layer are fed into the second layer and used for the actual activity recognition. A layered hidden Markov model (LHMM) (Oliver et al. 2002) was used in an application for office awareness. The lower layer HMMs classified the video and audio data with a time granularity of less than 1 s while the higher layer learned typical office activities such as phone conversation, face-to-face conversation, presentation, etc. Each layer of the HMM was designed and trained separately with fully labeled data. Hierarchical HMMs (Nguyen et al. 2005) were used to recognize human activities such as person having “short-meal”, “snacks” and “normal meal”. They also used a 2-layer architecture where lower layer HMM modeled simpler behaviors such as moving from one location in a room to another and the higher layer HMM used the information from layer one as its features. The higher layer was then used to recognize activities. A method based on modeling temporal relationships among a set of different temporal events (Gong and Xiang 2003) was developed and used for a scene-level interpretation to recognize cargo loading and unloading events.

The main difference between the above mentioned methods and our proposed method, is that these approaches assume that the higher-level activities and atomic activities (sub-actions) are known a priori, hence, the parameters of the model can be learned directly based on this notion. While this approach might be suitable for a small number of activities, it does not hold true for real-word scenarios where there is often a large number of sub-actions along with many activities (such as is found in the HMDB data set which is described in more detail in Sect. 5.6.2). *For activity classification, we propose to first compute sub-actions by clustering dynamic features obtained from videos, and then learn a hierarchical generative model over these features, thus probabilistically learning the relations between sub-actions, that are necessary to recognize different activities including those in real-world scenarios.*

5.2.2 Activity Spotting

Only a few methods have been proposed for activity spotting. Among them is the work of Yuan et al. (2009), which represented a video as a 3D volume and activities-of-interest as sub-volumes. The task of activity spotting was therefore reduced to one of performing an optimal search for activities in the video. Another work in spotting by Derpanis et al. (2010) introduced a local descriptor of video dynamics based on visual spacetime oriented energy measures. Similar to the previous work, their input was also a video which was searched for a specific action. The limitation of these techniques is their inability to adapt to changes in view points, scale, appearance etc. Rather than being defined on the motion patterns involved in an activity, these methods performed template matching type techniques, which do not readily generalize to new environments exhibiting a known activity. Both methods reported their results on the KTH and CMU data sets (described in more detail in Sect. 5.6), where the environment in which the activities were being performed did not readily change.

5.3 A Language-Motivated Hierarchical Model for Classification

Our proposed language-motivated hierarchical approach aims to perform recognition of higher-level activities by modeling them as a combination of other simpler activities. The major advantage of this approach over the typical sequential approaches and other hierarchical approaches is its ability to recognize activities with complex structures. By employing a hierarchical approach, multiple layers of state-based dynamic models can be used to recognize higher level activities. The bottom layers take observed features as inputs in order to recognize atomic actions (sub-actions). The results from these lower layers are then fed to the upper layers and used to recognize the modular activity.

5.3.1 Hierarchical Activity Modeling Using Multi-class Markov Chain Latent Dirichlet Allocation (MCMCLDA)

We propose a supervised dynamic, hierarchical Bayesian model, the multi-class Markov chain latent Dirichlet allocation (MCMCLDA), which captures the temporal information of an activity by modeling it as sequence of motion patterns, based on the Markov assumption. We develop this generative learning framework in order to obtain statistical insight into the underlying motions patterns (sub-actions) involved in an activity. An important aspect of this model is that motion patterns are shared across activities. So although the model is generative in structure, it can act discriminatively as it specifically learns which motion patterns are present in each activity. The fact that motion patterns are shared across activities was validated empirically (Messing et al. 2009) on the University of Rochester activities data set. Our proposed generative model harnesses both the temporal ordering power of DBNs and the automatic clustering power of hierarchical Bayesian models. The model correlates these motion patterns over time in order to define the signatures for classes of activities. Figure 5.1 shows an overview of the implementation network although we do not display *poses*, since they have no direct meaningful physical manifestations.

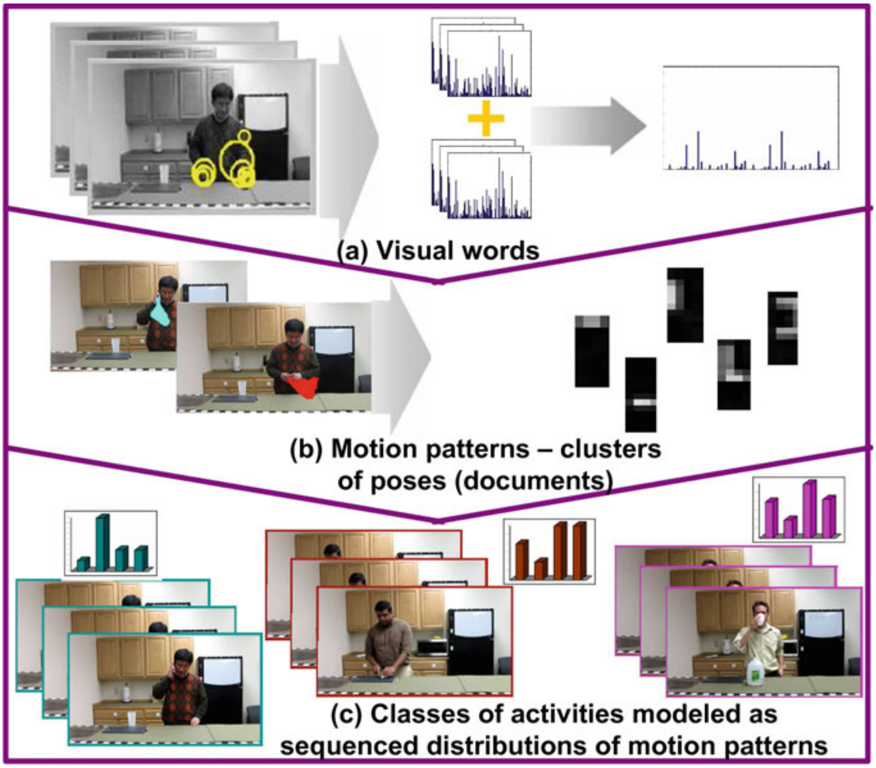

(a) Visual words
(b) Motion patterns – clusters of poses (documents)
(c) Classes of activities modeled as sequenced distributions of motion patterns

Fig. 5.1 Our general framework abstracts low-level visual features from videos and connects them to poses, motion patterns and classes of activity. **a** A video sequence is divided into short segments with a few frames only. In each segment, the space time interest points are computed. At the interest points, HoG and HoF are computed, concatenated and quantized to represent our low-level *visual words*. We discover and model a distribution over visual words which we refer to as *poses* (not shown in image). **b** Atomic motions are discovered and modeled as distributions over poses. We refer to these atomic motions as *motion patterns* or *sub-actions*. **c** Each video segment is modeled as a distribution over motion patterns. The time component is incorporated by modeling the transitions between the video segments, so that a complete video is modeled as a dynamic network of motion patterns. The distributions and transitions of underlying motion patterns in a video determine the final activity label assigned to that video

A given video is broken into motion segments comprising of either a combination of a fixed number of frames, or at the finest level, a single frame. Each motion segment can be represented as bag of vectorized descriptors (visual words) so that the input to the model (at time t) is the bag of visual words for motion segment t. Our model is similar in sprit to Hospedales et al. (2009), where the authors mine behaviors in video data from public scenes using an unsupervised framework. A major difference is that our MCMCLDA is a supervised version of their model in which motion-patterns/behaviors are shared across different classes, which makes it possible to handle a large number of different classes. If we assume that there exists only one class, then the motion-patterns are no longer shared, our model also becomes unsupervised and will thus be reduced to that of Hospedales et al. (2009).

We view MCMCLDA as a generative process and include a notation section before delving into the details of the LDA-type model:

$m =$ any single video in the corpus,
$z_t =$ motion pattern at time t (a video is assumed to be made up of motion patterns),
$y_{t,i} =$ the hidden variable representing a pose at motion pattern i, in time t (motion patterns are assumed to be made up of poses),
$x_{t,i} =$ the slices of the input video which we refer to as visual words and are the only observable variables,
$\phi_y =$ the visual word distribution for pose y,
$\theta_z =$ motion pattern specific pose distribution,
c_m is the class label for the video m (for one-shot learning, one activity is represented by one video ($N_m = 1$)),
$\psi_j = j$th class-specific transition matrix for the transition from one motion pattern to the next,
$\gamma_c =$ the transition matrix distribution for a video,
α,$\beta =$ the hyperparameters of the priors.

The complete generative model is given by:

$$\begin{aligned}
\boldsymbol{\psi}_j^z &\sim Dir(\psi_j^z|\gamma_j),\\
\boldsymbol{\theta}_z &\sim Dir(\theta_z|\alpha),\\
\boldsymbol{\phi}_y &\sim Dir(\phi_y|\beta),\\
z_t &\sim Mult(z_t|\boldsymbol{\psi}_j^{z_{t-1}}),\\
y_{t,i} &\sim Mult(y_{t,i}|\boldsymbol{\theta}_{z_t}),\\
x_{t,i} &\sim Mult(x_{t,i}|\boldsymbol{\phi}_{y_{t,i}}),
\end{aligned}$$

where $Mult(\cdot)$ refers to a multinomial distribution.

Now, consider the Bayesian network of MCMCLDA shown in Fig. 5.2. This can be interpreted as follows: For each video m in the corpus, a motion pattern indicator z_t is drawn from $p(z_t|z_{t-1}, \boldsymbol{\psi}_{c_m})$, denoted by Mult($\boldsymbol{\psi}_{c_m}^{z_{t-1}}$), where c_m is the class label for the video m. Then the corresponding pattern specific pose distribution $\boldsymbol{\theta}_{z_t}$ is used to draw visual words for that segment. That is, for each visual word, a pose indicator $y_{t,i}$ is sampled according to pattern specific pose distribution $\boldsymbol{\theta}_{z_t}$, and then the corresponding pose-specific word distribution $\boldsymbol{\phi}_{y_{t,i}}$ is used to draw a visual word. The poses $\boldsymbol{\phi}_y$, motion patterns $\boldsymbol{\theta}_z$ and transition matrices $\boldsymbol{\psi}_j$ are sampled once for the entire corpus.

The joint distribution of all known and hidden variables given the hyperparameters for a video is:

$$p\left(\{x_t, y_t, z_t\}_1^T, \phi, \psi_j, \theta|\alpha, \beta, \gamma_j\right) = p(\phi|\beta)p(\theta|\alpha)p(\psi|\gamma_j)\prod_t\prod_i p(x_{t,i}|y_{t,i})p(y_{t,i}|z_t)p(z_t|z_{t-1}).$$

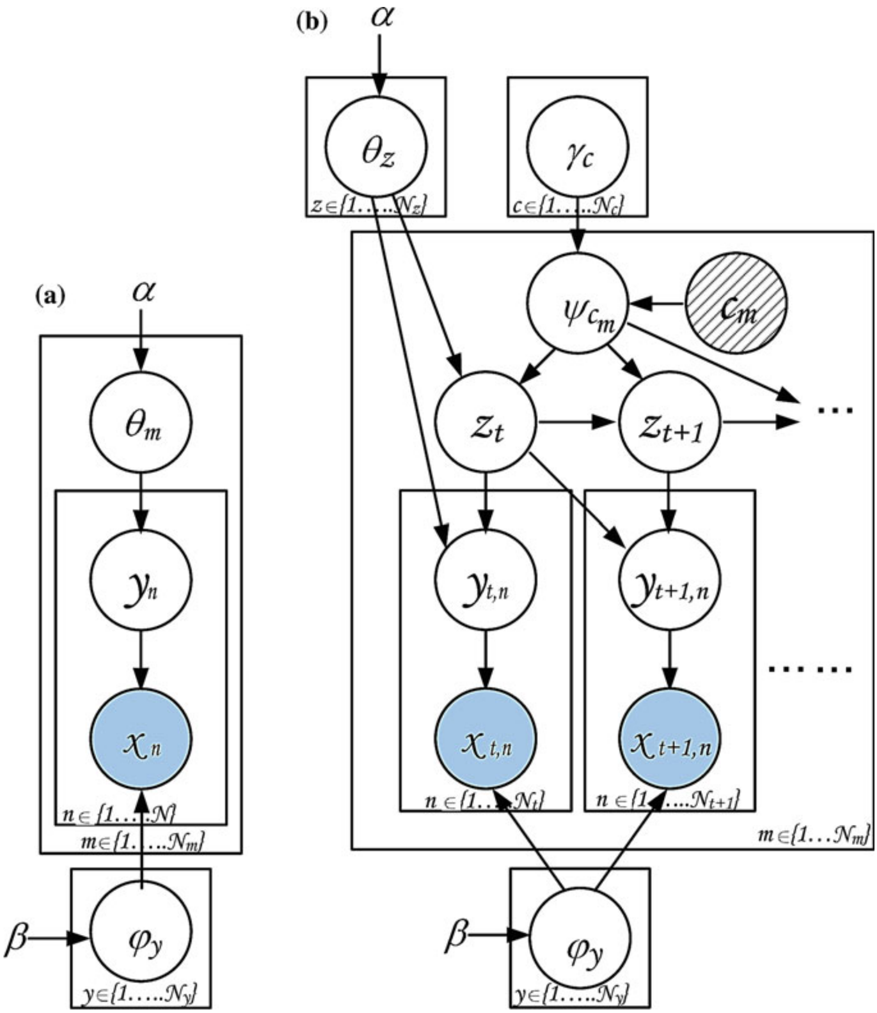


Fig. 5.2 *Left* plates diagram for standard LDA; *right* plates diagram for a dynamic model which extends LDA to learning the states from sequence data

5.3.2 Parameter Estimation and Inference of the MCMCLDA Model

As in the case with LDA, exact inference is intractable. We therefore use collapsed Gibbs sampler for approximate inference and learning. The update equation for pose from which the Gibbs sampler draws the hidden pose $y_{t,i}$ is obtained by integrating out the parameters θ, ϕ and noting that $x_{t,i} = x$ and $z_t = z$:

$$p\left(y_{t,i}=y|\mathbf{y}_{\neg(\mathbf{t},\mathbf{i})},\mathbf{z},\mathbf{x}\right)\propto\frac{n_{x,y}^{\neg(t,i)}+\beta}{\sum_{x=1}^{N_x}n_{x,y}^{\neg(t,i)}+N_x\beta}\left(n_{y,z}^{\neg(t,i)}+\alpha\right),\qquad(5.1)$$

where $n_{x,y}^{\neg(t,i)}$ denote the number of times that visual word x is observed with pose y excluding the token at (t,i) and $n_{y,z}^{\neg(t,i)}$ refers to the number of times that pose y is associated with motion pattern z excluding the token at (t,i). N_x is size of codebook and N_y is the number of poses.

The Gibbs sampler update for motion-pattern at time t is derived by taking into account that at time t, there can be many different poses associated to a single motion-pattern z_t and also the possible transition from z_{t-1} to z_{t+1}. The update equation for z_t can be expressed as:

$$p\left(z_t=z|\mathbf{y},\mathbf{z}_{\neg\mathbf{t}}\right)\propto p\left(y_t|z_t=z,z_{\neg t},y_{\neg t}\right)p\left(z_t=z|z_{\neg t}^{c_m},c_m\right).\qquad(5.2)$$

The likelihood term $p(y_t|z_t=z,z_{\neg t},y_{\neg t})$ cannot be reduced to the simplified form as in LDA as the difference between $n_{y,z}^{\neg t}$ and $n_{y,z}$ is not one, since there will be multiple poses associated to the motion-pattern z_t. $n_{y,z}$ denotes the number of times pose y is associated with motion-pattern z and $n_{y,z}^{\neg t}$ refers to the number of times pose y is observed with motion-pattern z excluding the poses (multiple) at time t. Taking the above condition into account, the likelihood term can be obtained as below:

$$p\left(y_t|z_t=z,z_{\neg t},y_{\neg t}\right)=\frac{\prod_y\Gamma(n_{y,z}+\alpha)}{\prod_y\Gamma(n_{y,z}^{\neg t}+\alpha)}\frac{\Gamma\left(\sum_y n_{y,z}^{\neg t}+N_y\alpha\right)}{\Gamma\left(\sum_y n_{y,z}+N_y\alpha\right)}.$$

Prior term $p(z_t=z|z_{\neg t}^{c_m},c_m)$ is calculated as below depending on the values of z_{t-1}, z_t and z_{t+1}.

$$\begin{aligned}
&if\quad z_{t-1}\neq z:\\
&=\frac{n_{z_{t-1},z,\neg t}^{(c_m)}+\gamma_{c_m}}{\sum_z n_{z_{t-1},z,\neg t}^{(c_m)}+N_z\gamma_{c_m}}\frac{n_{z,z_{t+1},\neg t}^{(c_m)}+\gamma_{c_m}}{\sum_{z_{t+1}}n_{z,z_{t+1},\neg t}^{(c_m)}+N_z\gamma_{c_m}},\\
&if\quad z_{t-1}=z=z_{t+1}:\\
&=\frac{n_{z_{t-1},z,\neg t}^{(c_m)}+1+\gamma_{c_m}}{\sum_z n_{z_{t-1},z,\neg t}^{(c_m)}+1+N_z\gamma_{c_m}}\frac{n_{z,z_{t+1},\neg t}^{(c_m)}+\gamma_{c_m}}{\sum_{z_{t+1}}n_{z,z_{t+1},\neg t}^{(c_m)}+N_z\gamma_{c_m}},\\
&if\quad z_{t-1}=z\neq z_{t+1}:\\
&=\frac{n_{z_{t-1},z,\neg t}^{(c_m)}+\gamma_{c_m}}{\sum_z n_{z_{t-1},z,\neg t}^{(c_m)}+N_z\gamma_{c_m}}\frac{n_{z,z_{t+1},\neg t}^{(c_m)}+\gamma_{c_m}}{\sum_{z_{t+1}}n_{z,z_{t+1},\neg t}^{(c_m)}+1+N_z\gamma_{c_m}}.
\end{aligned}$$

Here $n_{z_{t-1},z,\neg t}^{(c_m)}$ denotes the count from all the videos with the label c_m where motion-pattern z is followed by motion-pattern z_{t-1} excluding the token at t. $n_{z,z_{t+1},\neg t}^{(c_m)}$ denotes the count from all the videos with label c_m where motion-pattern z_{t+1} is followed by motion-pattern z_t excluding the token at t. N_z is the number of motion-patterns. The Gibbs sampling algorithm iterates between Eqs. 5.1 and 5.2 and finds the approximate posterior distribution. To obtain the resulting model parameters $\{\phi, \theta, \phi\}$ from the Gibbs sampler, we use the expectation of their distribution (Heinrich 2008), and collect N_s such samples of the model parameters.

For inference, we need to find the best motion-pattern sequence for a new video. The Gibbs sampler draws N_s samples of parameters during the learning phase. We assume that these are sufficient statistics for the model and that no further adaptation of parameters is necessary. We then adopt the Viterbi decoding algorithm to find the best motion-pattern sequence. We approximate the integral over ϕ, θ, ψ using the point estimates obtained during learning. To formulate the recursive equation for the Viterbi algorithm, we can define the quantity

$$\begin{aligned}\delta_t(i) &= \max_{z_1,\dots,z_{t-1}} \int_{\phi,\theta,\psi_{cm}} p\left(z_{1:(t-1)}, z_t = i, x_{1:t} | \phi, \theta, \psi_{c_m}\right),\\ &\approx \max_{z_1,\dots,z_{t-1}} \left(\frac{1}{N_s}\sum_s p\Big(z_{1:(t-1)}, z_t = i, x_{1:t} | \phi^s, \theta^s, \psi^s_{c_m}\Big)\right),\end{aligned}$$

that is $\delta_t(i)$ is the best score at time t, which accounts for first t motion-segments and ends in motion-pattern i. By induction we have

$$\delta_{t+1}(j) \approx \max_i \delta_t(i) \frac{1}{N_s} \sum_s p\left(z_{t+1} = j | z_t = i, \psi^s_{c_m}\right) p\left(x_{t+1} | z_{t+1} = j, \theta^s, \phi^s\right). \tag{5.3}$$

To find the best motion-pattern, we need to keep track of the arguments that maximized Eq. 5.3. For the classification task we calculate the likelihood $p^\star$ for each class and assign the label which has maximum value in:

$$p^\star = \max_{1 \le j \le N_z} \delta_T(j).$$

5.4 Experiments and Results Using MCMCLDA

In this section, we present our observations as well as the results of applying our proposed language-motivated hierarchical model to sub-action analysis as well as to activity classification, using both simulated data as well as a publicly available benchmark data set.

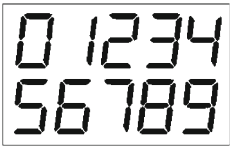


Fig. 5.3 Digital digits for simulations

5.4.1 Study Performed on Simulated Digit Data

To flesh out the details of our proposed hierarchical classification model, we present a study performed on simulated data. The ten simulated dynamic activity classes were the writing of the ten digital digits, 0–9 as shown in Fig. 5.3. The word vocabulary was made up of all the pixels in a 13 × 5 grid and the topics or poses represented the distribution over the words. An activity class therefore consisted of the steps needed to simulate the writing of each digit and the purpose of the simulation was to visually observe the clusters of motion patterns involved in the activities.

5.4.1.1 Analysis of Results

A total of seven clusters were discovered and modeled, as shown in Fig. 5.4. These represent the simulated strokes (or topics) involved in writing each digit. There were fourteen motion patterns discovered, as shown in the two bottom rows of Fig. 5.4. These are the probabilistic clusters of the stroke motions. An activity or digit written was therefore classified based on the sequences of distributions of these motion patterns over time.

5.4.2 Study Performed on the Daily Activities Data Set

The *Daily Activities data set* contains high resolution (1280 × 760 at 30 fps) videos, with 10 different complex daily life activities such as *eating banana, answering phone, drinking water, etc.*. Each activity was performed by five subjects three times, yielding a total of 150 videos. The duration of each video varied between 10 and 60 s.

We generated visual words for the MCMCLDA model in a manner similar to Laptev (2005), where the Harris3D detector (Laptev and Lindeberg 2003) was used to extract space-time interest points at multiple scales. Each interest point was described by the concatenation of HoF and HoG (Laptev 2005) descriptors. After the extraction of these descriptors for all the training videos, we used the k-means clustering algorithm to form a codebook of descriptors (or visual words (VW)). Furthermore, we vector-quantized each descriptor by calculating its membership with respect to the

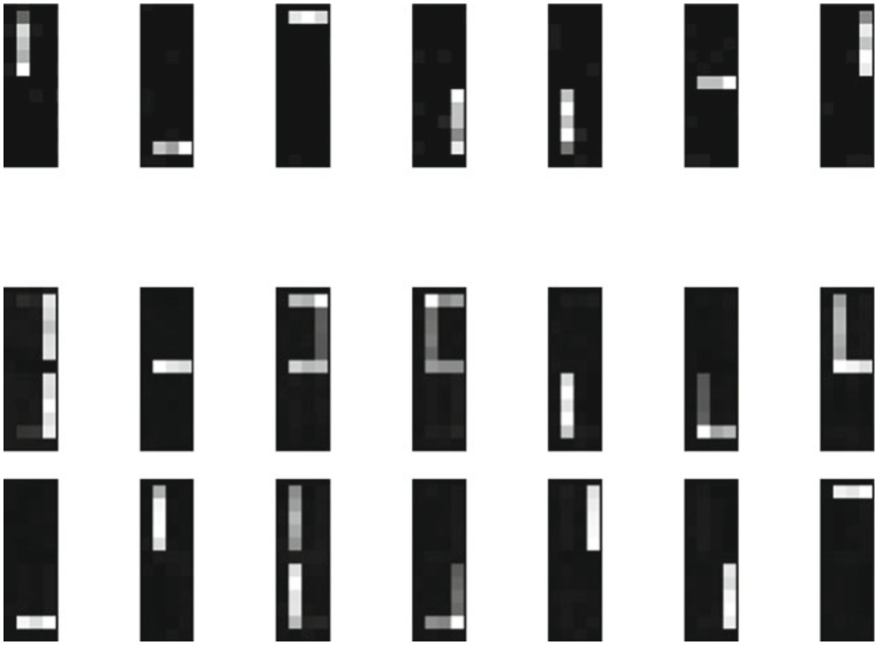
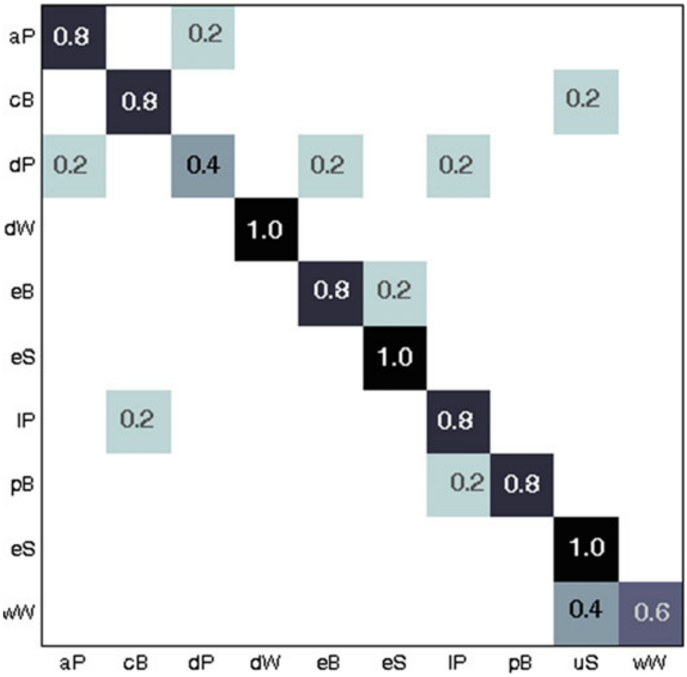


Fig. 5.4 The *top row* shows seven poses discovered from clustering words. The *middle* and *bottom rows* show the fourteen motion patterns discovered and modeled from the poses. A motion pattern captures one or more strokes in the order they are written

codebook. We used the original implementation available online[2] with the standard parameter settings to extract interest points and descriptors.

Due to the limitations of the distributed implementation of space-time interest points (Laptev et al. 2008), we reduced the video resolution to 320 × 180. In our experimental setup, we used 100 videos for training and 50 videos for testing exactly as pre-specified by the original publishers of this data set (Messing et al. 2009). Both the training and testing sets had a uniform distribution of samples for each activity. We learned our MCMCLDA model on the training videos, with a motion segment size of 15 frames. We ran a Gibbs sampler for a total of 6000 iterations, ignoring the first 5000 sweeps as burn-in, then took 10 samples at a lag of 100 sweeps. The hyperparameters were fixed initially with values ($\alpha = 5$, $\beta = 0.01$, $\gamma = 1$) and after burn-in, these values were empirically estimated using maximum-likelihood estimation (Heinrich 2008) as ($\alpha = 0.34$, $\beta = 0.001$ and $\gamma = \{0.04, 0.05, 0.16, 0.22, 0.006, 0.04, 0.13, 0.05, 0.14, 0.45\}$). We set the number of motion-patterns, poses and codebook size experimentally as $N_z = 100$, $N_y = 100$ and $N_x = 1000$.

[2] Implementation can be found at http://www.irisa.fr/vista/Equipe/People/Laptev/download.html#stip.



	aP	cB	dP	dW	eB	eS	lP	pB	uS	wW
aP	0.8		0.2							
cB		0.8							0.2	
dP	0.2		0.4		0.2		0.2			
dW				1.0						
eB					0.8	0.2				
eS						1.0				
lP		0.2					0.8			
pB							0.2	0.8		
eS									1.0	
wW									0.4	0.6

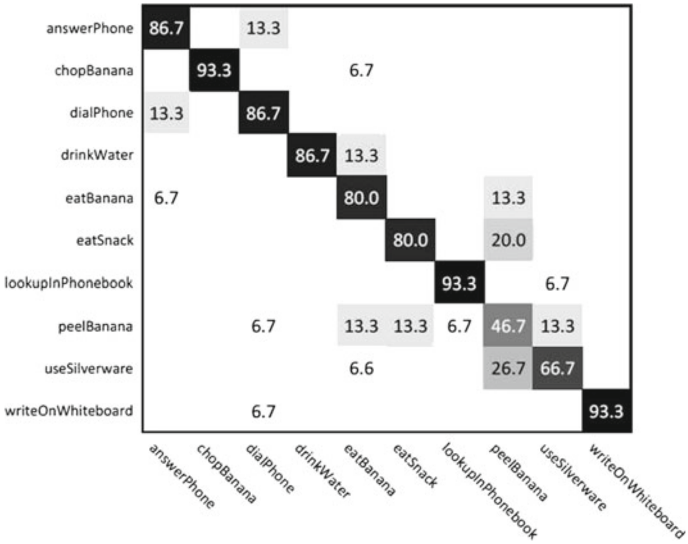
Fig. 5.5 Confusion matrix for analyzing the University of Rochester daily activities data set. Overall accuracy is 80.0%. Zeros are omitted for clarity. The labels and their corresponding meaning are: ap-answer phone; cB-chop banana; dP-dial phone; dW-drink water; eB-eat banana; eS-eat snack; lP-lookup in phonebook; pB-peel banana; uS-use silverware; wW-write on whiteboard

The confusion matrix computed from this experiment is given in Fig. 5.5 and a comparison with other activity recognition methods on the Daily Activities data set is given in Table 5.1. Because the data set was already pre-divided, the other recognition methods reported in Table 5.1 were trained and tested on the same sets of training and testing videos.

Qualitatively, Fig. 5.7 pictorially illustrates some examples of different activities having the same underlying shared motion patterns.

5.4.2.1 Analysis of Results

We present comparative results with other systems in Table 5.1. The results show that the approach based on computing a distribution mixture over motion orientations at each spatial location of the video sequence (Benabbas et al. 2010), slightly outperformed our hierarchical model. Interestingly, in our test, one activity, the *write on whiteboard (wW)* activity is quite confused with *use silverware (uS)* activity, significantly bringing down the overall accuracy. The confusion matrix for Benabbas et al.



	answerPhone	chopBanana	dialPhone	drinkWater	eatBanana	eatSnack	lookupInPhonebook	peelBanana	useSilverware	writeOnWhiteboard
answerPhone	86.7		13.3							
chopBanana		93.3			6.7					
dialPhone	13.3		86.7							
drinkWater				86.7	13.3					
eatBanana	6.7				80.0			13.3		
eatSnack						80.0		20.0		
lookupInPhonebook							93.3		6.7	
peelBanana			6.7		13.3	13.3	6.7	46.7	13.3	
useSilverware					6.6			26.7	66.7	
writeOnWhiteboard			6.7							93.3

Fig. 5.6 Confusion matrix results from Benabbas et al. (2010) on the University of Rochester daily activities data set

(2010) is presented in Fig. 5.6 and it shows several of the classes being confused, no perfect recognition scores and also one of the class recognition rates being below

Table 5.1 The accuracy numbers reported in literature from applying different activity recognition techniques on the daily activities data set

Technique	Focus	Accuracy (%)
Latent velocity trajectory features (Messing et al. 2009)[a]	Motion feature enhancement	67
Naive-Bayes pairwise trajectory features (Matikainen et al. 2010)	Motion feature enhancement	70
Salient region tracking features (Bilen et al. 2011)	Motion feature enhancement	74
Video temporal cropping technique	Motion feature enhancement	80
Our supervised dynamic hierarchical model	Dynamic hierarchical modeling	80
Direction of motion features (Benabbas et al. 2010)	Motion feature enhancement	81

[a] The authors also reported velocity trajectory feature augmented with prior spatial layout information, resulting in an accuracy of 89%

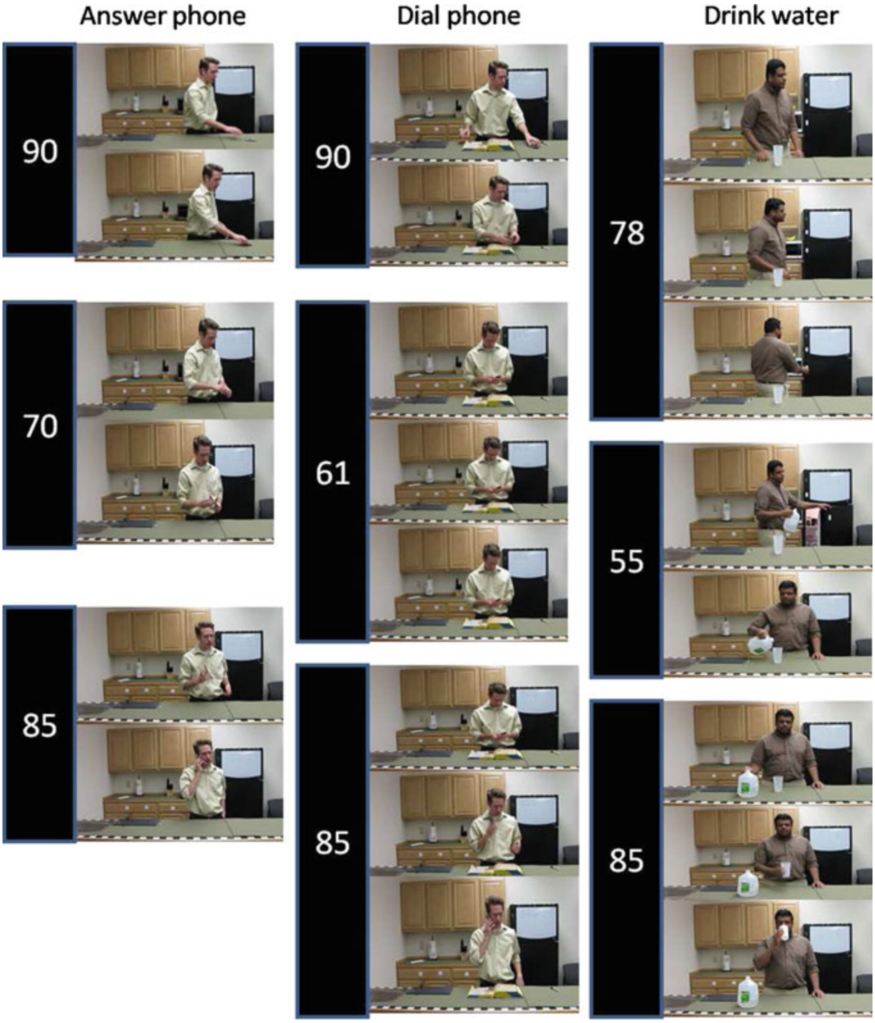


Fig. 5.7 Different activities showing shared underlying motion patterns. The shared motion patterns are 85 and 90, amidst other underlying motion patterns shown

50%. Being a generative model, the MCMCLDA model performs comparably to other discriminative models in a class labeling task.

Figure 5.7 pictorially illustrates some examples of different activities having the same underlying shared motion patterns. For example, the activity of answering the phone shares a common motion pattern (#85) with the activities of dialing the phone and drinking water. Semantically, we observe that this shared motion is related to the *lifting* sub-action.

5.5 A Language-Motivated Model for Gesture Recognition and Spotting

Few methods have been proposed for gesture spotting and among them include the work of Yuan et al. (2009), who represented a video as a 3D volume and activities-of-interest as sub-volumes. The task of gesture spotting was therefore reduced to performing an optimal search for gestures in the video. Another work in spotting was presented by Derpanis et al. (2010) who introduced a local descriptor of video dynamics based on visual space-time oriented energy measures. Similar to the previous work, their input was also a video in which a specific action was searched for. The limitation in these techniques is their inability to adapt to changes in view points, scale, appearance, etc. Rather than being defined on the motion patterns involved in an activity, these methods performed a type of 3D template matching on sequential data; such methods do not readily generalize to new environments exhibiting the known activity. *We therefore propose to develop a probabilistic framework for gesture spotting that can be learned with very little training data and can readily generalize to different environments*.

Justification: Although the proposed framework is a generative probabilistic model, it performs comparably to the state-of-the-art activity techniques which are typically discriminative in nature, as demonstrated in Tables 5.2 and 5.3. An additional benefit of the framework is its usefulness for gesture spotting based on learning from only one, or few training examples.

Background: In speech recognition, unconstrained keyword spotting refers to the identification of specific words uttered, when those words are not clearly separated from other words, and no grammar is enforced on the sentence containing them. Our proposed spotting framework uses the Viterbi decoding algorithm and is motivated by the *keyword-filler HMM for spotting keywords in continuous speech*. The current state of the art keyword filler HMM dates back to the seminal papers of Rohlicek et al. (1989) as well as Rose and Paul (1990), where the basic idea is to create one HMM of the keyword and a separate HMM of the filler or non keyword regions. These two models are then combined to form a composite filler HMM that is used to annotate speech parts using the Viterbi decoding scheme. Putative decisions arise when the Viterbi path crosses the keyword portion of the model. The ratio between the likelihood of the Viterbi path that passes through the keyword model and the

Table 5.2 Comparison of our proposed model and features for KTH data set

Method	Accuracy (%)
(Laptev et al., 2008)	91.8
(Yuan et al., 2009)	93.3
(Wang et al., 2011)	94.2
(Gilbert et al., 2011)	94.5
(Kovashka and Grauman, 2010)	94.53
Proposed mcHMM	94.67

Table 5.3 Comparison of our proposed model and features for the HMDB data set

Method	Accuracy (%)
Best results on 51 activities (original)	
(Kuehne et al. 2011)	23.18
Proposed mcHMM on 51 activities (original)	25.64
Best results on 10 activities (original)	
(Kuehne et al. 2011)	54.3
Proposed mcHMM on 10 activities (original)	57.67
Proposed mcHMM on 10 activities (stabilized)	66.67

likelihood of an alternate path that passes solely through the filler portion can be used to score the occurrence of keywords. In a similar manner, we compute the probabilistic signature for a gesture class, and using the filler model structure, we test for the presence of that gesture within a given video. For one-shot learning, the parameters of the single training video are considered to be sufficiently representative of the class.

5.5.1 Gesture Recognition Using a Multichannel Dynamic Bayesian Network

In a general sense, the spotting model can be interpreted as an HMM (whose random variables involve hidden states and observed input nodes) but unlike the classic HMM, this model has multiple input channels, where each channel is represented as a distribution over the visual words corresponding to that channel. In contrast to the classic HMM, our model can have multiple observations per state and channel, and we refer to this as the multiple channel HMM (mcHMM). Figure 5.8 shows a graphical representation of the mcHMM.

5.5.2 Parameter Estimation for the Gesture Recognition Model

To determine the probabilistic signature of an activity class, one mcHMM is trained for each activity. The generative process for mcHMM involves first sampling a state from an activity, based on the transition matrix for that activity; then a frame-feature comprising of the distribution of visual words is sampled according to a multinomial distribution for that state[3] and this is repeated for each frame. Similar to a classic HMM, the parameters for the mcHMM are therefore:

[3] States are modeled as multinomials since our input observables are discrete values.

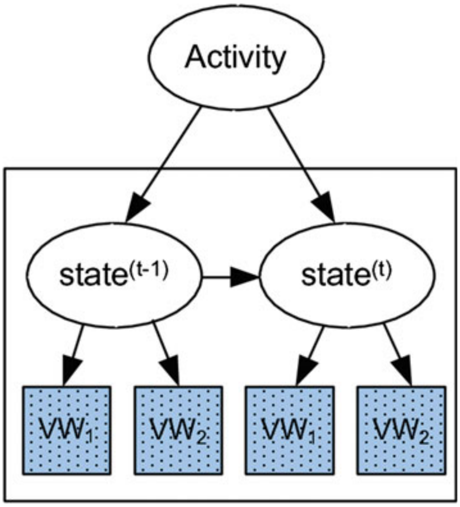


Fig. 5.8 Plates model for mcHMM showing the relationship between activities or gestures, states and the two channels of observed visual words (VW)

1. Initial state distribution $\pi = \{\pi_i\}$,
2. State transition probability distribution $A = \{a_{ij}\}$,
3. Observation densities for each state and descriptor $B = \{b_i^d\}$.

The joint probability distribution of observations (O) and hidden state sequence (Q) given the parameters of the multinomial representing a hidden state (λ) can be expressed as:

$$P(O, Q|\lambda) = \pi_{q_1} b_{q_1}(O_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} \cdot b_{q_t}(O_t),$$

where $b_{q_t}(O_t)$ is modeled as follows:

$$\begin{aligned} b_{q_t}(O_t) = &\quad \textstyle\prod_{d=1}^{D} b_q^d(O_t^d), \\ &= \textstyle\prod_{d=1}^{D} Mult(O_t^d|b_q^d), \end{aligned}$$

and D is the number of descriptors.

EM is implemented to find the maximum likelihood estimates. The update equations for the model parameters are:

$$\hat{\pi} = \sum_{r=1}^{R} \gamma_1^r(i),$$

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T} \eta_t^r(i, j)}{\sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_t^r(i)},$$

$$\hat{b}_j^d(k) = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_t^r(j) \cdot \frac{n_t^{d,k}}{n_t^{d,\cdot}}}{\sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_t^r(j)},$$

where R is number of videos and $\gamma_1(i)$ is the expected number of times the activity being modeled started with state i;

$\eta_t^r(i, j)$ is the expected number of transitions from state i to state j and $\gamma_t^r(i)$ is the expected number of transitions from state i;

$n_t^{d,k}$ is the number of times that visual word k occurred in descriptor d at time t and $n_t^{d,\cdot}$ is the total number of visual words that occurred in descriptor d at time t.

5.5.3 Gesture Spotting via Inference on the Model

The gesture spotting problem is thus reduced to an inference problem where, given a new not-previously-seen test video, and the model parameters or probabilistic signatures of known activity classes, the goal is to establish which activity class distributions most likely generated the test video. This type of inference can be achieved using the Viterbi algorithm.

We constructed our spotting network such that there could be a maximum of five gestures in a video. This design choice was driven by our participation in the Chalearn competition where there was a maximum of five gestures in every test video. Each of these gesture classes was seen during training, hence, there were no random gestures inserted into the test video. This relaxed our network, compared to the original filler model in speech analysis, where there can exist classes that have not been previously seen. Figure 5.9 shows an example of the stacked mcHMMs involved the gesture spotting task. This toy example shown in the figure can spot

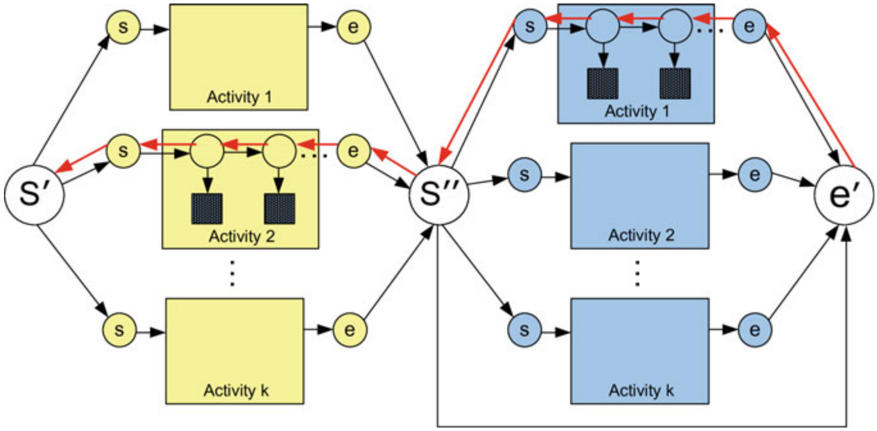


Fig. 5.9 Activity spotting by computing likelihoods via Viterbi decoding. The toy example shown assumes there are at most two activities in any test video, where the first activity is from the set of activities that start from s' and end at s'', followed by one from the set that start from s'' and end at e'. The image also shows an example of a putative decision path from e' to s', after the decoding is completed

gestures in a test video comprised of at most two gestures. This network has a non-emitting start state S'. This state does not have any observation density associated with it. From this state, we can enter any of K gestures, which is shown by having edges from S' to K mcHMMs. All the gestures are then connected to non-emitting state S'' which represents the end of first gesture. Similarly we can enter the second gesture from S'' and end at e' or directly go from S'' to e' which handles the case for a video having only one gesture. This can be easily extended to the case where there are at most five gestures.

The Viterbi decoding algorithm was implemented to traverse the stacked network and putative decisions arose when the Viterbi path crosses the keyword portion of the model. The ratio between the likelihood of the Viterbi path that passed through the keyword model and the likelihood of an alternate path that passes through the non-keyword portion was then used to score the occurrence of a keyword, where a keyword here referred to a gesture class. An empirically chosen threshold value was thus used to select the occurrence of a keyword in the video being decoded.

5.6 Experiments and Results Using mcHMM

In this section, we present our approach on generating visual words and our observations as well as the results of applying proposed mcHMM model to activity classification and gesture spotting, using publicly available benchmark data sets.

5.6.1 Generating Visual Words

An important step in generating visual words is the the need to extract interest points from frames sampled from the videos at 30 fps. Interest points were obtained from the KTH and HMDB data set by sampling dense points in every frame in the video and then tracking these points for the next L frames. These are known as dense trajectories. For each of these tracks, motion boundary histogram descriptors based on HoG and HoF descriptors were extracted. These features are similar to the ones used in dense trajectories (Wang et al. 2011), although rather than sampling interest points at every L frames or when the current point is lost before being tracked for L frames, we sampled at every frame. By so doing, we obtained a better representation for each frame, whereas the original work used the features to represent the whole video and was not frame-dependent.

Because the HMDB data set is comprised of real-life scenes which contain people and activities occurring at multiple scales, the frame-size in the video was reduced by a factor of two repeatedly, and motion boundary descriptors were extracted at multiple scales. In the Chalearn data set, since the videos were comprised of RGB-depth frames, we extracted interest points by (i) taking the difference between two consecutive depth frames and/or (ii) calculating the centroid of the depth foreground in

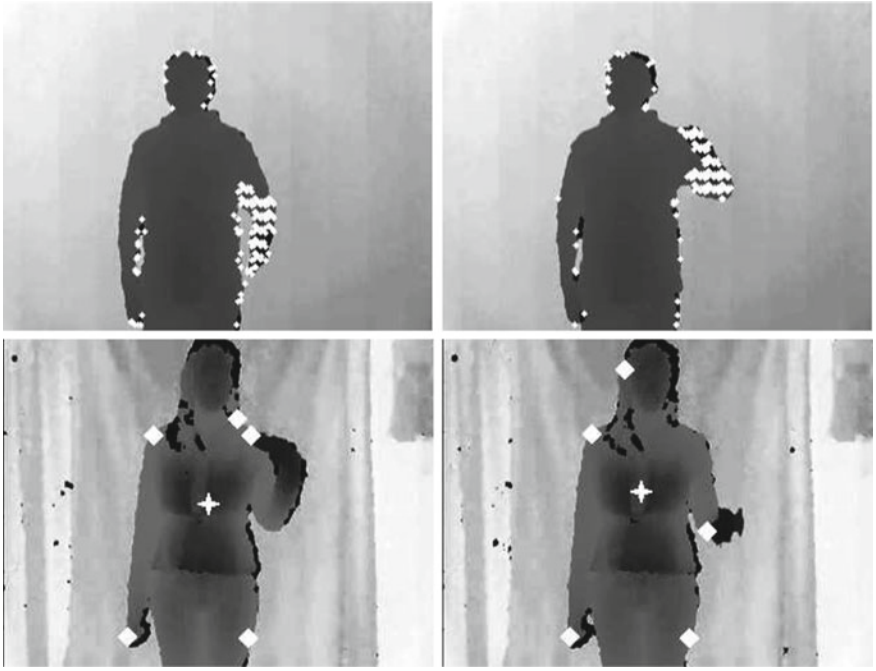


Fig. 5.10 Interest points for 2 consecutive video frames. *Top* Depth-subtraction interest points; *bottom* extrema interest points (with centroid)

every frame and computing the extrema points (from that centroid) in the depth foreground. The second process ensured that extrema points such as the hands, elbows, top-of-the-head, etc., were always included in the superset of interest points. The top and bottom image pairs in Fig. 5.10 show examples of consecutive depth frames from the Chalearn data set, with the interest points obtained via the two different methods, superimposed. Again, HoG and HoF descriptors were extracted at each interest point so that similar descriptors could be obtained in all the cases. We used a patch size of 32 × 32 and a bin size of 8 for HoG and 9 for HoF implementation.

The feature descriptors were then clustered to obtain visual words. In general, from the literature (Wang et al. 2011; Laptev et al. 2008), in order to limit complexity, researchers randomly select a finite number of samples (roughly in the order of 100,000) and cluster these to form visual words. This could prove reasonable when the number of samples is a few orders of magnitude greater than 100,000. But in dealing with densely sampled interest points at every frame, the amount of descriptors generated especially at multiple scales become significantly large. We therefore divided the construction of visual words for HMDB data set into a two step process where visual words were first constructed for each activity class separately, and then the visual words obtained for each class were used as the input samples to cluster the final visual words. For the smaller data sets such as KTH and Chalearn Gesture Data Set, we randomly sampled 100,000 points and clustered them to form the visual words.

5.6.2 Study Performed on the HMDB and KTH Data Sets

In order to compare our framework to the other current state-of-the-art methods, we performed activity classification on video sequences created from the KTH database (Schüldt et al. 2004); KTH is a relatively simplistic data set comprised of 2391 video clips used to train/test six human actions. Each action is performed several times by 25 subjects in various outdoor and indoor scenarios. We split the data into training set of 16 subjects and test set of 9 subjects, which is exactly the same setup used by the authors of the initial paper (Schüldt et al. 2004). Table 5.2 shows the comparison of accuracies obtained.

Similarly, we performed activity classification tests on Human Motion Database (HMDB) (Kuehne et al. 2011). HMDB is currently the most realistic database for human activity recognition comprising of 6766 video clips and 51 activities extracted from a wide range of sources like YouTube, Google videos, digitized movies and other videos available on the Internet. We follow the original experimental setup using three train-test splits (Kuehne et al. 2011). Each split has 70 video for training and 30 videos for testing for each class. All the videos in the data set are stabilized to remove the camera motion and the authors of the initial paper (Kuehne et al. 2011) report results on both original and stabilized videos for 51 activities. The authors also selected 10 common actions from HMDB data set that were similar to action categories in the UCF50 data set (University of Central Florida 2010) and compared the recognition performance. Table 5.3 summarizes the performance of proposed mcHMM method on 51 activities as well as 10 activities for both original and stabilized videos.

5.6.2.1 Analysis of Results

For both the case of simple actions as found in the KTH data set and the case of significantly more complex actions as found in the HMDB data set, the mcHMM model performs comparably with other methods, outperforming them in the activity recognition task. Our evaluation against state-of-the-art data sets suggest that performance is not significantly affected over a range of factors such as camera position and motion as well as occlusions. This suggests that the overall framework (combination of dense descriptors and a state-based probabilistic model) is fairly robust with respect to these low-level video degradations. At the time of this submission, although we outperformed the only currently reported accuracy results on the HMDB data set, as shown by the accuracy scores reported, the framework is still limited in its representative power to capture the complexity of human actions.

5.6.3 Study Performed on the ChaLearn Gesture Data Set

Lastly, we present our results of gesture spotting from the ChaLearn gesture data set (ChaLearn 2011). The ChaLearn data set consisted of video frames with RGB-Depth information. Since the task-at-hand was gesture spotting via one-shot learning, only one video per class was provided to train an activity (or gesture). The data set was divided into three parts: development, validation and final. In the first phase of the competition, participants initially developed their techniques against the development data set. Ground truth was not provided during the development phase. Once the participants had a working model, they then ran their techniques against the validation data set and uploaded their predicted results to the competition website, where they could receive feedback (scores based on edit distances) on the correctness of the technique. In the last phase of the competition, the final data set was released so that participants could test against it and upload their predicted results. Similarly, edit scores were used to measure the correctness of the results and the final rankings were published on the competition website.

We reported results using two methods (i) mcHMM (ii) mcHMM with LDA (Blei et al. 2003). For mcHMM method, we constructed visual words as described in Sect. 5.6.1 and represented each frame as two histograms of visual words. This representation was input to the model to learn parameters of the mcHMM model. In the augmented framework, mcHMM + LDA, the process of applying LDA to the input data can be viewed as a type of dimensionality reduction step since the number of topics are usually significantly smaller than the number of unique words. In our work, a frame is analogous to a document and visual words are analogous to words in a text document. Hence, in the combined method, we performed the additional step of using LDA to represent each frame as a histogram of topics. These reduced-dimension features were input to the mcHMM model. Gesture spotting was then performed by creating a spotting network made up of connected mcHMM models, one for each gesture learned, as explained in Sect. 5.5.3.

For the mcHMM model, we experimentally fixed the number of states to 10. The number of visual words was computed as the number of classes multiplied by a factor of 10, for example if the number of classes is 12, then number of visual words generated will be 120. The dimensionality of the input features to the mcHMM model was the number of visual words representing one training sample. For the augmented model the dimension of the features was reduced by a factor of 1.25, that is in the previous example, the length of feature vector would be reduced from 120 to 96. All the above parameters were experientially found using the development set. The same values were then used for the validation and final sets.

5.6.3.1 Analysis of Results

Table 5.4 shows the results of one-shot-learning on the ChaLearn data at the three different stages of the competition. We present results based on the two variants of our framework—the mcHMM model framework and the augmented mcHMM + LDA framework. Our results indicate that the framework augmented with LDA outperforms the unaugmented one, two out of three times. During implementation, the computational performance for the augmented framework was also significantly better than the unaugmented model due the reduced number of features needed for

Table 5.4 Results for ChaLearn gesture data set

Method	Data set	Edit distance
Proposed mcHMM	Development	0.26336
Proposed mcHMM + LDA	Development	0.2409
Baseline	Validation	0.59978
Proposed mcHMM	Validation	0.26036
Proposed mcHMM + LDA	Validation	0.23328
Top ranking participant	Validation	0.20287
Top ranking participant	Final	0.09956
Proposed mcHMM + LDA	Final	0.18465

training and for inference. It is also interesting to observe how the edit distances reduced from the development phase through the final phase, dropping by up to six percentage points, due to parameter tuning.

5.7 Conclusion and Future Work

In the course of this paper, we have investigated the use of motion patterns (representing sub-actions) exhibited during different complex human activities. Using a language-motivated approach we developed a dynamic Bayesian model which combined the temporal ordering power of dynamic Bayesian networks with the automatic clustering power of hierarchical Bayesian models such as the LDA word-topic clustering model. We also showed how to use the Gibbs samples for rapid Bayesian inference of video segment clip category. Being a generative model, we can detect abnormal activities based on low likelihood measures. This framework was validated by its comparable performance on tests performed on the daily activities data set, a naturalistic data set involving everyday activities in the home.

We also investigated the use of a multichannel HMM as a generative probabilistic model for single activities and it performed comparably to the state-of-the-art activity classification techniques which are typically discriminative in nature, on two extreme data sets—the simplistic KTH, and the very complex and realistic HMDB data sets. An additional benefit of this framework was its usefulness for gesture spotting based on learning from only one, or few training examples. We showed how the use of the generative dynamic Bayesian model naturally lent itself to the spotting task, during inference. The efficacy of this model was shown by the results obtained from participating in ChaLearn Gesture Challenge where an implementation of the model finished top-5 in the competition.

In the future, we will consider using the visual words learned from a set of training videos to automatically segment a test video. The use of auto-detected video segments could prove useful both in activity classification and gesture spotting. It will also be interesting to explore the use of different descriptors available in the literature, in order to find those best-suited for representing naturalistic videos.

Acknowledgements The authors wish to thank the associate editors and anonymous referees for all their advice about the structure, references, experimental illustration and interpretation of this manuscript. The work benefited significantly from our participation in the ChaLearn challenge as well as the accompanying workshops.

References

J.K. Aggarwal, M.S. Ryoo, Human activity analysis: a review. ACM Comput. Surv. **43**, 1–16 (2011)

Y. Benabbas, A. Lablack, N. Ihaddadene, C. Djeraba, Action recognition using direction models of motion, in *Proceedings of the 2010 International Conference on Pattern Recognition*, 2010, pp. 4295–4298

H. Bilen, V.P. Namboodiri, L. Van Gool, Action recognition: a region based approach, in *Proceedings of the 2011 IEEE Workshop on the Applications of Computer Vision*, 2011, pp. 294–300

David M. Blei, Andrew Y. Ng, Michael I. Jordan, Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)

M. Bregonzio, S. Gong, T. Xiang, Recognising action as clouds of space-time interest points, in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1948–1955

ChaLearn. ChaLearn Gesture Dataset (CGD2011), ChaLearn, California, 2011. http://gesture.chalearn.org/2011-one-shot-learning

N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893

K.G. Derpanis, M. Sizintsev, K. Cannons, R.P. Wildes, Efficient action spotting based on a spacetime oriented structure representation, in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1990–1997

P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in *Proceedings of the 2005 IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65–72

A. Gilbert, J. Illingworth, R. Bowden, Action recognition using mined hierarchical compound features. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 883–897 (2011)

S. Gong, and T. Xiang, Recognition of group activities using dynamic probabilistic networks, in *Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 742–749

G. Heinrich, Parameter estimation for text analysis. Technical report, University of Leipzig, 2008

T. Hospedales, S.G. Gong, T. Xiang, A Markov clustering topic model for mining behaviour in video, in *Proceedings of the 2009 International Conference on Computer Vision*, 2009, pp. 1165–1172

A. Kläser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in *Proceedings of the 2008 British Machine Vision Conference* (2008)

A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2046–2053

H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in *Proceedings of the 2011 International Conference on Computer Vision* 2011

I. Laptev, On space-time interest points. Int. J. Comput. Vis. **64**, 107–123 (2005)

I. Laptev, T. Lindeberg, Space-time interest points, in *Proceedings of the 2003 International Conference on Computer Vision*, 2003, pp. 432–439

I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8

P. Matikainen, M. Hebert, R. Sukthankar, Trajectons: Action recognition through the motion analysis of tracked features, in *Proceedings of the 2009 IEEE Workshop on Video-Oriented Object and Event Classification* (2009)

P. Matikainen, M. Hebert, R. Sukthankar, Representing pairwise spatial and temporal relations for action recognition, in *Proceedings of the 2010 European Conference on Computer Vision* 2010

R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in *Proceedings of the 2009 International Conference on Computer Vision* 2009

P. Natarajan, R. Nevatia, Coupled hidden semi Markov models for activity recognition, in *Proceedings of the IEEE Workshop on Motion and Video Computing* 2007

N.T. Nguyen, D.Q. Phung, S. Venkatesh, Learning and detecting activities from movement trajectories using the hierarchical hidden Markov models, in *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 955–960

E. Nowak, F. Jurie, B. Triggs, Sampling strategies for bag-of-features image classification, in *Proceedings of the 2006 European Conference on Computer Vision*, 2006, pp. 490–503

N. Oliver, E. Horvitz, A. Garg, Layered representations for human activity recognition, in *Proceedings of the 2002 IEEE International Conference on Multimodal Interfaces*, 2002, pp. 3–8

Nuria M. Oliver, Barbara Rosario, Alex P. Pentland, A Bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 831–843 (2000)

J.R. Rohlicek, W. Russell, S. Roukos, H. Gish, Continuous hidden Markov modeling for speaker-independent word spotting, in *Proceedings of the 1989 International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 627–630

R. Rose, D. Paul, A hidden Markov model based keyword recognition system, in *Proceedings of the 1990 International Conference on Acoustics, Speech, and Signal Processing* 1990

C. Schüldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in *Proceedings of the 2004 International Conference on Pattern Recognition*, 2004, pp. 32–36

P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in *Procedings of the ACM International Conference on Multimedia*, 2007, pp. 57–360

University of Central Florida. *University of Central Florida, Computer Vision Lab*, 2010. URL http://server.cs.ucf.edu/~vision/data/UCF50.rar

H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid,Evaluation of local spatio-temporal features for action recognition, in *Proceedings of the 2009 British Machine Vision Conference* 2009

H. Wang, A. Kläser, C. Schmid, L. Cheng-Lin, Action recognition by dense trajectories. in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3169–3176

G. Willems, T. Tuytelaars, L. Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in *Proceedings of the 2008 European Conference on Computer Vision*, 2008, pp. 650–663

J. Yamato, J. Ohya, K. Ishii, Recognizing human action in time-sequential images using hidden Markov model, in *Proceedings of the 1992 IEEE Conference on Computer Vision and Pattern Recognition*, 1992, pp. 379–385

L. Yeffet, L. Wolf, Local trinary patterns for human action recognition, in *Proceedings of the 2009 International Conference on Computer Vision* 2009

J. Yuan, Z. Liu, Y. Wu, Discriminative subvolume search for efficient action detection, in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition* 2009