Int J Comput Vis (2008) 80: 337–357
DOI 10.1007/s11263-008-0142-8

Searching for Complex Human Activities with No Visual Examples

Nazlı İkizler · David A. Forsyth

Received: 19 July 2007 / Accepted: 28 April 2008 / Published online: 29 May 2008

Abstract We describe a method of representing human activities that allows a collection of motions to be queried without examples, using a simple and effective query language. Our approach is based on units of activity at segments of the body, that can be composed across space and across the body to produce complex queries. The presence of search units is inferred automatically by tracking the body, lifting the tracks to 3D and comparing to models trained using motion capture data. Our models of short time scale limb behaviour are built using labelled motion capture set. We show results for a large range of queries applied to a collection of complex motion and activity. We compare with discriminative methods applied to tracker data; our method offers significantly improved performance. We show experimental evidence that our method is robust to view direction and is unaffected by some important changes of clothing.

Keywords Human action recognition · Video retrieval · Activity · HMM · Motion capture

1 Introduction

Understanding what people are doing is one of the great unsolved problems of computer vision. A fair solution opens tremendous application possibilities, ranging from medical to security. The major difficulties have been that (a) good kinematic tracking is hard; (b) models typically have too many parameters to be learned directly from data; and (c) for much everyday behaviour, there isn't a taxonomy. Tracking is now a usable, if not perfect technology (Sect. 4). Building extremely complex dynamical models from heterogenous data is now well understood by the speech community, and we borrow some speech tricks to build models from motion capture data (Sect. 3) to minimize parameter estimation. Desirable properties of an activity recognition and retrieval system are:

- it should handle different clothings and varying motion speeds of different actors
- it should accommodate the different timescales over which actions are sustained
- it should allow composition across time and across the body
- there should be a manageable number of parameters to estimate
- it should perform well in presence of limited quantities of training data
- it should be indifferent to viewpoint changes
- it should require no example video segment for querying

Building such a system has many practical applications. For example, if a suspicious behaviour can be encoded in terms of "action word"s—wrt arms and legs separately whenever needed—one can submit a text query and search for that specific behaviour within security video recordings. Similarly, one can encode medically critical behaviours and search for those in surveillance systems.

Understanding activities is a complex issue in many aspects. First of all, there is a shortage of training data, because a wide range of variations of behaviour is possible. A particular nuisance is the tendency of activity to be compositional (below). Discriminative methods on appearance

N. İkizler (✉)
Bilkent University, 06800 Ankara, Turkey
e-mail: inazli@cs.bilkent.edu.tr

D.A. Forsyth
University of Illinois at Urbana-Champaign, 61801 Urbana, IL, USA

may be confounded by intraclass variance. Different subjects may perform the actions with different speeds in various outfits and these nuisance variations make it difficult to work directly with appearance. Training a generative model directly on a derived representation of video is also fraught with difficulty. Either one must use a model with very little expressive power (for example, an HMM with very few hidden states) or one must find an enormous set of training data to estimate dynamical parameters (the number of which typically goes as the square of the number of states). This issue has generated significant interest in variant dynamical models, which we review below.

The second difficulty is the result of the composite nature of activities. Most of the current literature on activity recognition deals with simple actions. However, real life involves more than just simple "walk"s. Many activity labels can meaningfully be composed, both over time—"walk"ing then "run"ing—and over the body—"walk"ing while "wave"ing. The process of composition is not well understood (see the review of animation studies in Forsyth et al. 2006), but is a significant source of complexity in motion. Examples include: "walking while scratching head" or "running while carrying something". Because composition makes so many different actions possible, it is unreasonable to expect to possess an example of each activity. This means we should be able to find activities for which we do not possess examples.

A third issue is that tracker responses are noisy, especially when the background is cluttered. For this reason, discriminative classifiers over tracker responses work poorly. One can boost the performance of discriminative classifiers if they are trained on noise-free environments, like carefully edited motion capture datasets. However, these will lack the element of compositionality.

All these points suggest having a model of activity which consists of *pieces* which are relatively easily learned and are then combined together within a model of *composition*. In this study, we try to achieve this by

- learning local dynamic models for atomic actions distinctly for each body part, over a motion capture dataset
- authoring a compositional model of these atomic actions
- using the emissions of the data with these composite models

To overcome the data shortage problem, we propose to make use of motion capture data. This data does not consist of everyday actions, but rather a limited set of American football movements. There is a form of transfer learning problem here—we want to learn a model in a football domain and apply it to an everyday domain—and we believe that transfer learning is an intrinsic part of activity understanding.

We first author a compositional model for each body part using a motion capture dataset. This authoring is done in a similar fashion to phoneme-word conjunctions in speech recognition: We join atomic action models to have more complex activity models. By doing so, we achieve the minimum of parameter estimation. In addition, composition across time and across body is achieved by building separate activity models for each body part. By providing composition across time and space, we can make use of the available data as much as possible and achieve a broader understanding about what the subject is up to.

After forming the compositional models over 3D data, we track the 2D video with a state-of-the-art full body tracker and lift 2D tracks to 3D, by matching the snippets of frames to motion capture data. We then infer activities with these lifted tracks. By this lifting procedure, we achieve view-invariance, since our body representation is in 3D.

Finally, we write text queries to retrieve videos. In this procedure, we do not require example videos and we can query for activities that have never been seen before. Making use of finite state automata, we employ a simple and effective query language that allows complex queries to be written in order to retrieve the desired set of activity videos. Using separate models for each body part, compositional nature of our system allows us to span a huge query space.

Our particular interest is everyday activity. In this case, a fixed vocabulary either doesn't exist, or isn't appropriate. For example, one often does not know words for behaviours that appear familiar. One way to deal with this is to work with a notation (for example, Laban notation); but such notations typically work in terms that are difficult to map to visual observables (for example, the weight of a motion). We must either develop a vocabulary or develop expressive tools for authoring models. We favour this third approach (Sect. 5).

We compare our method with several controls. Each has a discriminative form, and we justify this choice in Sect. 6.2. First, we built discriminative classifiers over raw 2D tracks. We expect that discriminative methods applied to 2D data perform poorly because intra-class variance overwhelms available training data. In comparison, our method benefits by being able to estimate dynamical models on motion capture dataset. Second, we built classifiers over 3D lifts. Although classifiers applied to 3D data could be view invariant, we expect poor performance because there is not much labelled data and the lifts are noisy. Our third control involves classifiers trained on 3D motion capture data and applied to lifted data. This control also performs poorly, because noise in the lifting process is not well represented by the training data. This also causes problems with the composition. On contrary, our model supports a high level of composition and its generative nature handles different lengths of actions easily. In our experiments section, we evaluate the effect of all these issues and also analyze the view-invariance of our method in greater detail (Sect. 6).

A shorter version of this paper appeared in CVPR 2007 (Ikizler and Forsyth 2007).

2 Related Work

There is a long tradition of research on interpreting activities in the vision community (see, for example, the extensive survey in Hu et al. 2004; Forsyth et al. 2006). There are three major threads. First, one can use motion clusters of the same type and explore the statistics or relative ordering of these clusters. Second, one can use (typically, hidden Markov) models of dynamics or temporal logics to represent the crucial order relations between states that constrain activities. Third, one can use discriminative methods, either with spatio-temporal templates or using 'bag-of-words'.

Timescale A wide range of helpful distinctions is available. Bobick (1997) distinguishes between movements, activity and actions, corresponding to longer timescales and increasing complexity of representation; some variants are described in two useful review papers (Aggarwal and Cai 1999; Gavrila 1999).

2.1 Motion Primitives

A natural method for building models of motion on longer time scales is to identify clusters of motion of the same type and then consider the statistics of how these *motion primitives* are strung together. There are pragmatic advantages to this approach: we may need to estimate fewer parameters and can pool examples to do so; we can model and account for long term temporal structure in motion; and matching may be easier and more accurate. Feng and Perona describe a method that first matches motor primitives at short timescales, then identifies the activity by temporal relations between primitives (Feng and Perona 2002). In animation, the idea dates at least to the work of Rose et al., who describe motion *verbs*—our primitives—and *adverbs*—parameters that can be supplied to choose a particular instance from a scattered data interpolate (Rose et al. 1998). Primitives are sometimes called *movemes*. Matarić et al. represent motor primitives with force fields used to drive controllers for joint torque on a rigid-body model of the upper body (Matarić et al. 1998, 1999). Del Vecchio et al. define primitives by considering all possible motions generated by a parametric family of linear time-invariant systems (Vecchio et al. 2003). Barbič et al. compare three motion segmenters, each using a purely kinematic representation of motion (Barbič et al. 2004). Their method moves along a sequence of frames adding frames to the pool, computing a representation of the pool using the first k principal components, and looking for sharp increases in the residual error of this representation. Fod et al. construct primitives by segmenting motions at points of low total velocity, then subjecting the segments to principal component analysis and clustering (Fod et al. 2002). Jenkins and Mataric segment motions using kinematic considerations, then use a variant of Isomap (detailed in Jenkins and Matarić 2004) that incorporates temporal information by reducing distances between frames that have similar temporal neighbours to obtain an embedding for kinematic variables (Jenkins and Matarić 2003). Li et al. segment and model motion capture data simultaneously using a linear dynamical system model of each separate primitive and a Markov model to string the primitives together by specifying the likelihood of encountering a primitive given the previous primitive (Li et al. 2002).

2.2 Methods with Explicit Dynamical Methods

Hidden Markov Models (HMM's) have been very widely adopted in activity recognition, but the models used have tended to be small (e.g, three and five state models in Brand et al. 1997). Such models have been used to recognize: tennis strokes (Yamato et al. 1992); pushes (Wilson and Bobick 1995); and handwriting gestures (Yang et al. 1997). Feng and Perona (2002) call actions "movelets", and build a vocabulary by vector quantizing a representation of image shape. These codewords are then strung together by an HMM, representing activities; there is one HMM per activity, and discrimination is by maximum likelihood. The method is not view invariant, depending on an image centered representation. There has been a great deal of interest in models obtained by modifying the HMM structure, to improve the expressive power of the model without complicating the processes of learning or inference. Methods include: coupled HMM's (Brand et al. 1997; to classify T'ai Chi moves); layered HMM's (Oliver et al. 2004; to represent office activity); hierarchies (Mori et al. 2004; to recognize everyday gesture); HMM's with a global free parameter (Wilson and Bobick 1999; to model gestures); and entropic HMM's (Brand and Kettnaker 2000; for video puppetry). Building variant HMM's is a way to simplify learning the state transition process from data (if the state space is large, the number of parameters is a problem). But there is an alternative—one could author the state transition process in such a way that it has relatively few free parameters, despite a very large state space, and then learn those parameters; this is the lifeblood of the speech community.

Stochastic grammars have been applied to find hand gestures and location tracks as composites of primitives (Bobick and Ivanov 1998). However, difficulties with tracking mean that there is currently no method that can exploit the potential view-invariance of lifted tracks, or can search for

models of activity that compose across the body and across time.

Finite state methods have been used directly. Hongeng et al. demonstrate recognition of multi-person activities from video of people at coarse scales (few kinematic details are available); activities include conversing and blocking (Hongeng et al. 2004). Zhao and Nevatia use a finite-state model of walking, running and standing, built from motion capture (Zhao and Nevatia 2004). Hong et al. use finite state machines to model gesture (Hong et al. 2000).

2.3 Methods with Partial Dynamical Models

Pinhanez and Bobick (1997, 1998) describe a method for detecting activities using a representation derived from Allen's interval algebra (Allen 1984), a method for representing temporal relations between a set of intervals. One determines whether an event is past, now or future by solving a consistent labelling problem, allowing temporal propagation. There is no dynamical model—sets of intervals produced by processes with quite different dynamics could be a consistent labelling; this can be an advantage at the behaviour level, but probably is a source of difficulties at the action/activity level. Siskind (2003) describes methods to infer activities related to objects—such as throw, pick up, carry, and so on—from an event logic formulated around a set of physical primitives—such as translation, support relations, contact relations, and the like—from a representation of video. A combination of spatial and temporal criteria are required to infer both relations and events, using a form of logical inference. Recently, Ryoo and Aggarwal use context-free grammars to exploit the temporal relationships between atomic actions to define composite activities (Ryoo and Aggarwal 2007).

2.4 Methods with Discriminative Methods

Methods Based on Templates The notion that a motion produces a characteristic spatio-temporal pattern dates at least to Polana and Nelson (1993). Spatio-temporal patterns are used to recognize actions in Bobick and Davis (2001). Ben-Arie et al. (2002) recognize actions by first finding and tracking body parts using a form of template matcher and voting on lifted tracks. Bobick and Wilson (1997) use a state-based method that encodes gestures as a string of vector-quantized observation segments; this preserves order, but drops dynamical information. Efros et al. (2003) use a motion descriptor based on optical flow of a spatio-temporal volume, but their evaluation is limited to matching videos only. Blank et al. (2005) define actions as space-time volumes. An important disadvantage of methods that match video templates directly is that one needs to have a template of the desired action to perform a search.

Bag-of-Words Approaches Recently, 'bag-of-words' approaches originated from text retrieval research are being adopted to action recognition. These studies are mostly based on the idea of forming codebooks of 'spatio-temporal' features. Laptev et al. first introduce the notion of 'space-time interest points' (Laptev and Lindeberg 2003) and use SVMs to recognize actions (Schuldt et al. 2004). P. Dollár et al. extract cuboids via separable linear filters and form histograms of these cuboids to perform action recognition (Dollár et al. 2005). Niebles et al. apply a pLSA approach over these patches (i.e. the cuboids extracted with the method of (Dollár et al. 2005)) to perform unsupervised action recognition (Niebles et al. 2006). Recently, Wong et al. propose using pLSA method with and implicit shape model to infer actions from spatio-temporal codebooks (Wong et al. 2007). They also show the superior performance of applying SVMs for action recognition. However, these methods are not viewpoint independent and very likely to suffer from complex background schemes.

Transfer Learning Recently, transfer learning has become a very hot research topic in machine learning community. It is based on transfering the information learned from one domain to the another related domain. In one of the earlier works, Caruana approached this problem by discovering common knowledge shared between tasks via "multi-task learning" (Caruana 1997). Evgeniou and Pontil (2004) utilize SVMs for multi-task learning. Ando and Zhang (2005) generate some artificial auxiliary tasks to use shared prediction structures between similar tasks. A recent application involves transfering American Sign Language (ASL) words learned from a synthetic dictionary to real world data (Farhadi et al. 2007).

3 Representing Acts, Actions and Activities

Timescale In terms of acts and activities, there are many quite different cases. Motions could be sustained (walking, running) or have a localizable character (catch, kick). The information available to represent what a person is doing depends on timescale. We distinguish between short-timescale representations (*acts*), like a forward-step; medium timescale *actions*, like walking, running, jumping, standing, waving, whose temporal extent can be short (but may be long) and are typically composites of multiple acts; and long timescale *activities*, which are complex composites of actions.

Since we want our complex, composite activities to share a vocabulary of base units, we use the kinematic configuration of the body as distinctive feature. We ignore limb velocities and accelerations because actions like reach/wave can be performed at varying speeds. However, one should note that velocity and acceleration is a useful clue when differentiating motion pairs like run and walk.

We want our representation to be as robust as possible to view effects and to details of appearance of the body. Furthermore, we wish to search for activities without possessing an example. All this suggests working with an inferred representation of the body's configuration (rather than, say, image flow templates as in Efros et al. 2003; Blank et al. 2005). An advantage of this approach is that models of activity, etc. can be built using motion capture data, then transferred to use on image observations, and this is what we do.

3.1 Acts in Short Timescales

Individual frames are a poor guide to what the body is up to, not least because transduction is quite noisy and the frame rate is relatively high (15–30 Hz). We expect better behaviour from short runs of frames. At the short timescale, we represent motion with three frame long snippets of the lifted 3D representation. We form one snippet for each leg and one for each arm; we omit the torso, because torso motions appear not to be particularly informative in practice (see Sect. 6). Each limb in each frame is represented with the vector quantized value of the snippet centered on that frame. That is, we apply k-means to the 3D representation of snippets the limbs. We use 40 as the number of clusters in vector quantization, for each limb. One can utilize different levels of quantization, but our experiments show that for this dataset, using 40 for each limb provides good enough generalization.

3.2 Limb Action Models

Using a vague analogy with speech, we wish to build a large dynamical model with the minimum of parameter estimation. In speech studies, in order to recognize words, phoneme models are built and joined together to form word models. By learning phoneme models and joining them together, word models share information within the phoneme framework, and this makes building large vocabularies of word models possible.

By using this analogy, we first build a model of the action of each limb (arms, legs) for a range of actions, using HMM's that emit vector quantized snippets we formed in the previous step. We choose a set of 9 actions by hand, with the intention of modelling our motion capture collection reasonably well; the collection is the research collection of motion capture data released by Electronic Arts in 2002, and consists of assorted football movements. Motion sequences from this collection are sorted into actions using the labelling of Arikan et al. (2003). The original annotation includes 13 action labels; we have excluded actions with the direction information (3 actions named `turn left, turn right, backwards`) and observed that `reach` and `catch` actions do not differ significantly in practice, so we joined the data for these two actions and labelled them as `reach` altogether. Moreover, this labelling is adapted to have separate action marks for each limb. Since actions like `wave` cannot be definable for legs, we only used a subset of 6 actions for labelling legs and 9 for labelling arms.

For each action, we fit to the examples using maximum likelihood, and searching over 3–10 state HMM models. Experimentation with the structures shows that 3-state models represent the data well enough. Thus, we take 3-state HMMs as our smallest unit for action representation. Again, we emphasize that the action dynamics are completely built on 3D motion capture data.

3.3 Limb Activity Models

Having built atomic action models, we now string the limb models into a larger HMM by linking states that have similar emission probabilities. That is, we put a link between states m and n of the different action models A and B if the distance

$$\mathrm{dist}(A_m, B_n) = \sum_{o_m=1}^{N} \sum_{o_n=1}^{N} p(o_m)p(o_n)C(o_m, o_n) \tag{1}$$

is minimal. Here, o_m and o_n are the emissions, $p(o_m)$ and $p(o_n)$ are the emission probabilities of respective action model states A_m and B_n, N is the number of possible emissions and $C(o_m, o_n)$ is the Euclidean distance between the emissions centers, which are the cluster centers of the vector-quantized 3D joint points.

The result of this linkage is a dynamical model for each limb that has a rich variety of states, but is relatively easily learned. States in this model are grouped by limb model, and we call a group of states corresponding to a particular limb model a *limb activity model* (Fig. 1). While linking these states, we assign uniform probability to transition between actions and transition to the same action. That is, the probability of the action staying the same is set equal to the probability of transferring to another action.

4 Transducing the Body

4.1 Tracking

We track motion sequences with the tracker of Ramanan et al. (2005); this tracker obtains an appearance model by detecting a lateral walk pose, then detects instances in each frame using the pictorial structure method of Felzenszwalb and Huttenlocher (2005). The advantage of using this tracker is that it is highly robust to occlusions and complex backgrounds. There is no need for background modelling, and this tracker has been shown to perform well on changing

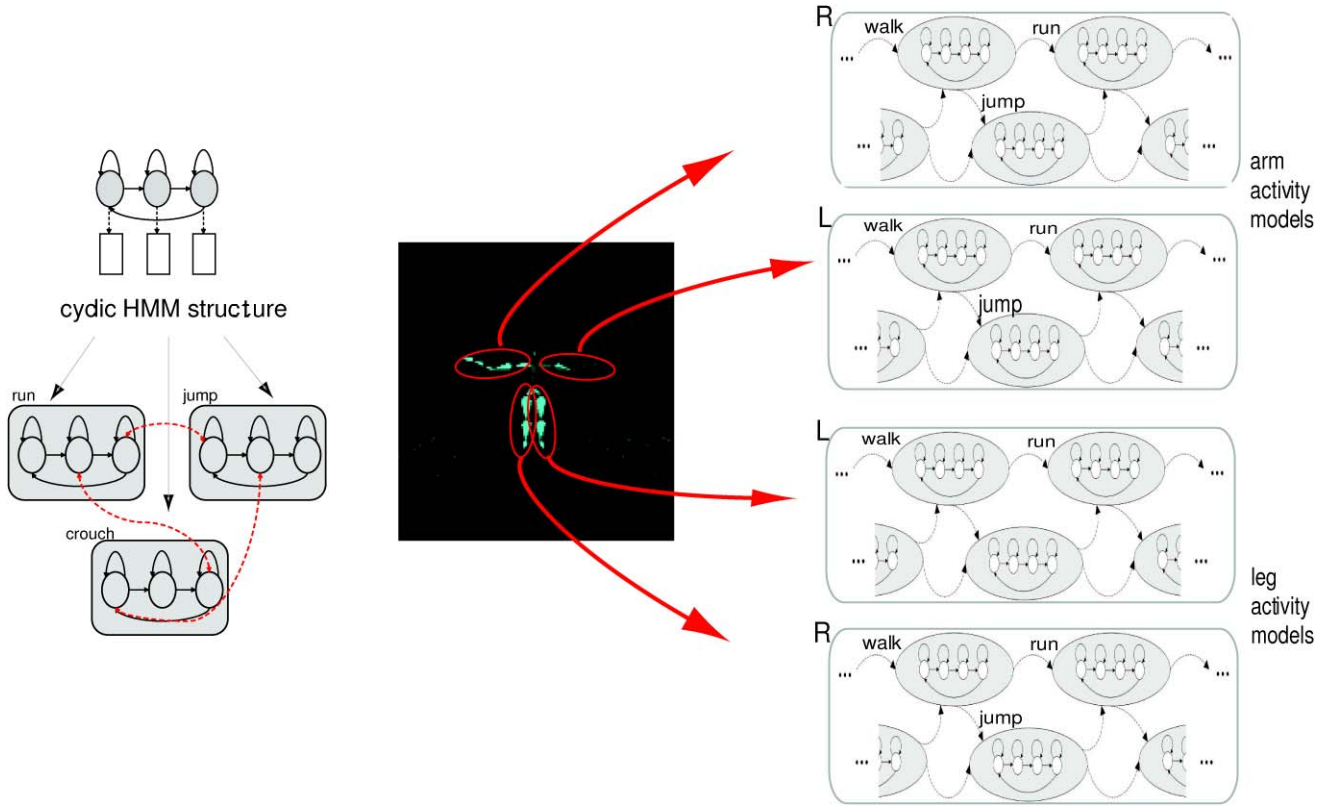


Fig. 1 First, single action HMMs for left leg, right leg, left arm, right arm are formed using motion capture dataset. Actions are chosen by hand to conform with the available actions in this largely synthesized motion capture set (provided by Electronic Arts, consisting of American Football movements). Second, single action HMMs are joint together by linking the states that have similar emission probabilities. This is analogous to joining phoneme models to recognize words in speech recognition. This is loosely a generative model, we compute the probability that each sequence is generated by a certain set of action HMMs

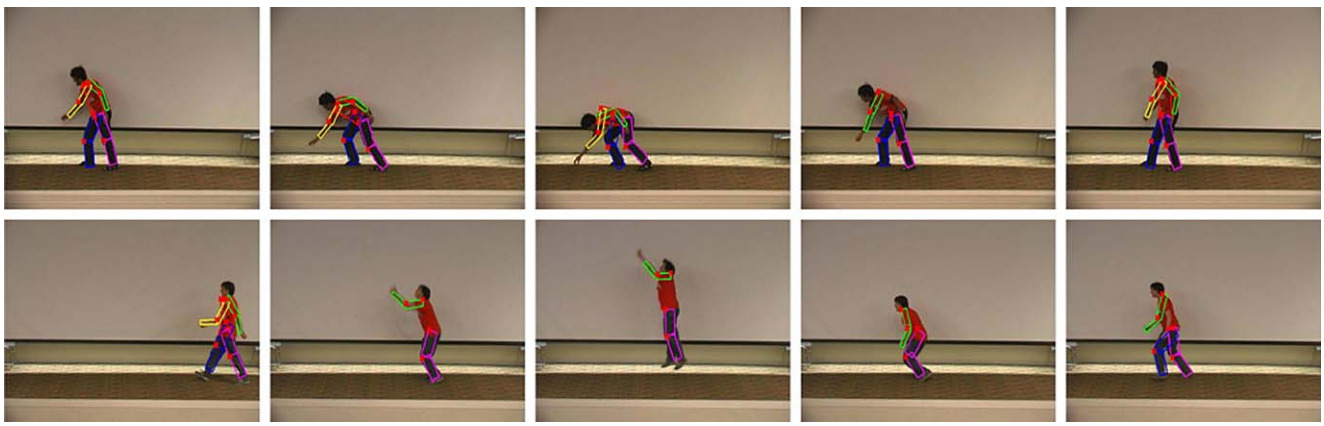


Fig. 2 Here are some example tracks from our video collection. These are two sequences performed by two different actors wearing different outfits. *Top* `stand-pickup` sequence. *Bottom* `walk-jump-reach-walk` sequence. The tracker is able to spot most of the body parts in these sequences. However, in most of the sequences, especially in lateral views, only two out of four limbs are tracked because of the self-occlusions

backgrounds (see also Sect. 6.4). Moreover, it is capable of identifying the distinct limbs, which we need to form our separate limb action models.

Kinematic tracking is known to be hard (see the review in Forsyth et al. 2006) and, while the tracker is usable, it has some pronounced eccentricities (Fig. 3, Ramanan et al. 2007). Note that the noise introduced by this behaviour is a part of the activity understanding procedure and by lifting 2D tracks to 3D, we want to suppress the effects of such noise as much as possible.

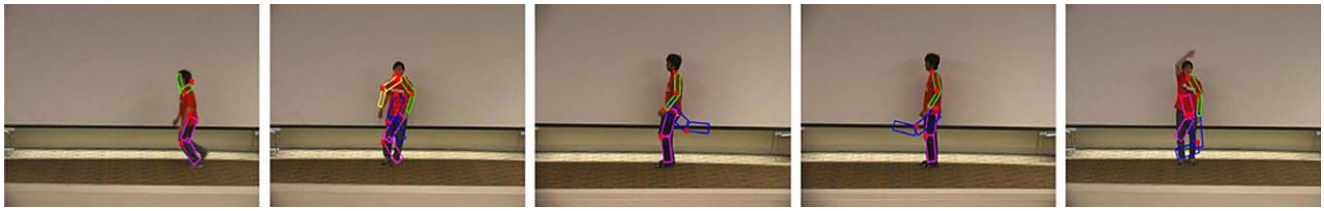


Fig. 3 Due to motion blur and similarities in appearance, some frames are out of track. *First*: appearance and motion blur error. *Second*: legs mixed up because of rectangle search failure on legs. *Third and fourth*: one leg is occluded by the other leg, the tracker tries to find second leg, mistaken by the *solid dark line*. *Fifth*: motion blur causes tracker to miss the waving arm, legs scrambled. Note that all such bad tracks are a part of our test collection and non-perfect tracking introduces considerable amount of noise to our motion understanding procedure

4.2 Lifting 2D Tracks to 3D

The tracker reports a 2D configuration of a puppet figure in the image (Fig. 2), but we require 3D information. Several authors have successfully obtained 3D reconstructions by matching projected motion capture data to image data by matching *snippets* of multiple motion frames (Howe 2004; Howe et al. 2000; Ramanan and Forsyth 2003). A complete sequence incurs a per-frame cost of matching the snippet centered at the frame, and a frame-frame transition cost which reflects (a) the extent of the movement and (b) the extent of camera motion. The best sequence is obtained with dynamic programming. The smoothing effect of matching snippets—rather than frames—appears to significantly reduce reconstruction ambiguity (see also the review in Forsyth et al. 2006).

The disadvantage of the method is that one may not have motion capture that matches the image well, particularly if one has a rich collection of activities to deal with. We use a variant of the method. In particular, we decompose the body into four quarters (two arms, two legs). We then match the legs using the snippet method, but allowing the left and right legs to come from different snippets of motion capture, making a search over 20 camera viewing directions. The per-frame cost must now also reflect the difference in camera position in the root coordinate system of the motion capture; for simplicity, we follow (Ramanan and Forsyth 2003) in assuming an orthographic camera with a vertical image plane. We choose arms in a similar manner conditioned on the choice of legs, requiring the camera to be close to the camera of the legs. In practice, this method is able to obtain lifts to quite rich sequences of motion from a relatively small motion capture collection. Our lifting algorithm is given in Algorithm 1.

Algorithm 1 Lifting 2D Tracks to 3D

for each camera $c \in C$ **do**
  for all pose $p \in mocap$ **do**
    $\sigma_{pc} \leftarrow projection(p, c)$
  end for
  camera_transition_cost $\delta(c_i, c_j) \leftarrow (c_i - c_j) \times \alpha$
end for
for each $l_t \in L$ (leg segments in 2D) **do**
  for all $p \in mocap$ and $c \in C$ **do**
    $\lambda(l_t, \sigma_{pc}) \leftarrow match\_cost(\sigma_{pc}, l_t)$
    $\gamma(l_t, l_{t+w})$
    $\leftarrow transition\_cost(\lambda(l_t, \sigma_{pc}), \lambda(l_{t+w}, \sigma_{pc}))$
  end for
end for
do dynamic programming over δ, λ, γ for L
$c_{legs} \leftarrow$ (minimum cost camera sequence)
for each $a_t \in A$ (arm segments in 2D) **do**
  for $c_\epsilon \leftarrow$ neighborhood ϵ of c_{legs} and pose $p \in mocap$ **do**
    compute $\lambda(a_t, \sigma) \leftarrow match\_cost(\sigma_{pc_\epsilon}, a_t)$
    compute $\gamma(a_t, a_{t+w})$
    $\leftarrow transition\_cost(\lambda(a_t, \sigma_{pc_\epsilon}), \lambda(a_{t+w}, \sigma_{pc_\epsilon}))$
  end for
end for
do dynamic programming over δ, λ, γ for A

4.3 Representing the Body

We can now represent the body's behaviour for any sequence of frames with $P(\text{limb activity model}|\text{frames})$. The model has been built entirely on motion capture data. By computing a forward-algorithm pass of the lifted sequences over the activity models, we get a posterior probability map representation for each video, which indicates the likelihood of each snippet to be in a particular state of the activity HMMs over the temporal domain.

The posterior probability of a set of action states $\lambda = (s_1, \ldots, s_t)$ given a sequence of observations $\sigma_k = o_1, o_2, \ldots, o_t$ and model parameters θ can be computed from the joint. In particular, note

$$P(\lambda|\sigma_k, \theta) \propto P(\lambda, \sigma_k|\theta) = P(s_1)\left(\prod_{j=1}^{t-1} P(o_j|s_j)P(s_{j+1}|s_j)\right)P(o_t|s_t) \quad (2)$$

where the constant of proportionality $P(\sigma_k)$ can be computed easily with the forward-backward algorithm (for more details, see, for example Rabiner and Juang 1993). We follow convention and define the forward variable $\alpha_t(i) = P(q_t = i, o_1, o_2, \ldots, o_T|\theta)$, where q_t is the state of the HMM at time t and T is the total number of observations. Similarly, the backward variable $\beta_t(i) = P(o_{t+1}, \ldots, o_T|q_t = i, \theta)$. We write $b_j(o_t) = P(o_t|q_t = j)$, $a_{ij} = P(q_t = j|q_{t-1} = i)$ and so have the recursions

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] \tag{3}$$

$$\beta_t(j) = \sum_{i=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \tag{4}$$

where a_{ij} is the transition probability from state i to j, π_i and b_is are the initial state and observation probabilities, N is the number of states of the HMM and

$$\alpha_1(i) = \pi_i b_i(o_1) \tag{5}$$

$$\beta_T(i) = 1 \tag{6}$$

This gives

$$P(\sigma_k) = \sum_{i=1}^{N} \alpha_T(i) \tag{7}$$

Our activity model groups states with an equivalence relation. For example, several different particular configurations of the leg might correspond to walking. We can compute a posterior over these groups of states in a straightforward way. We assume we have a total of $M \leq N$ groups of states. Now assume we wish to evaluate the posterior probability of a sequence of state groups $\lambda_g = (g_1, \ldots, g_t)$ conditioned on a sequence of observations $\sigma_k = (o_1, \ldots, o_t)$. We can regard a sequence of state groups as a set of strings Λ_g, where a string $\lambda \in \Lambda_g$ if and only if $s_1 \in g_1, s_2 \in g_2, \ldots, s_t \in g_t$. Then we have

$$P(\lambda_g, \sigma_k) = \sum_{\lambda \in \Lambda_g} P(\lambda, \sigma_k) \tag{8}$$

This allows us to evaluate the posterior on activity models (see, for example, Fig. 4).

As example sequences in Figs. 5 and 6 indicate, this representation is quite competent at discriminating between different labellings for motion capture data. In addition, we achieve automatic segmentation of activities using this representation. There is no need for explicit motion segmentation, since transitions between action HMM models simply provide this information.

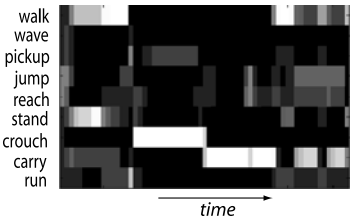


Fig. 4 Posterior probability map of a `walk-pickup-carry` video of an arm. This probability map corresponds to a run of forward algorithm through the activity HMM for this particular video. The action models are quite discriminative, therefore we can expect a good search for a composition. Moreover, the action models give a good segmentation in and of themselves. Despite some noise, we can clearly observe transitions between different actions within the video

5 Querying for Activities

We can compute a representation of what the body is doing from a sequence of video. By using this representation, we would like to be able to build complex queries of composite activities, such as carrying while standing, or waving while running. We can address composition across the body because we can represent different limbs doing different things; and composition in time is straightforward with our representation.

This suggests thinking of querying as looking for strings, where the alphabet is a product of possible activities at limbs and locations in the string represent locations in time. Generally, we do not wish to be precise about the temporal location of particular activities, but would rather find sequences where there is strong evidence for one activity, followed by strong evidence for another, and with a little noise scattered about. In turn, it is natural to start by using regular expressions for motion queries (we see no need for a more expressive string model at this point).

An advantage of using regular expressions is that it is relatively straightforward to compute

$$\sum_{\text{strings matching RE}} P(\text{string}|\text{frames}) \tag{9}$$

which we do by reducing the regular expression to a finite state automaton and then computing the probability this automaton reaches its final state using a straightforward sum-product algorithm.

This query language is very simple: Suppose we want to find videos where the subject is walking and waving his arms at the same time. For legs, we form a walk automaton. For arms, we form a wave automaton. We simultaneously query both limbs with these automata. Figures 8 and 9 show the corresponding automata for example queries.

Finite State Representation for Activity Queries A finite state automaton (FSA) is defined with the quintuple $(Q, \Sigma, \delta, s_0, F)$, where Q is the finite non-empty set of states of the FSA, Σ is the input alphabet, δ is the state

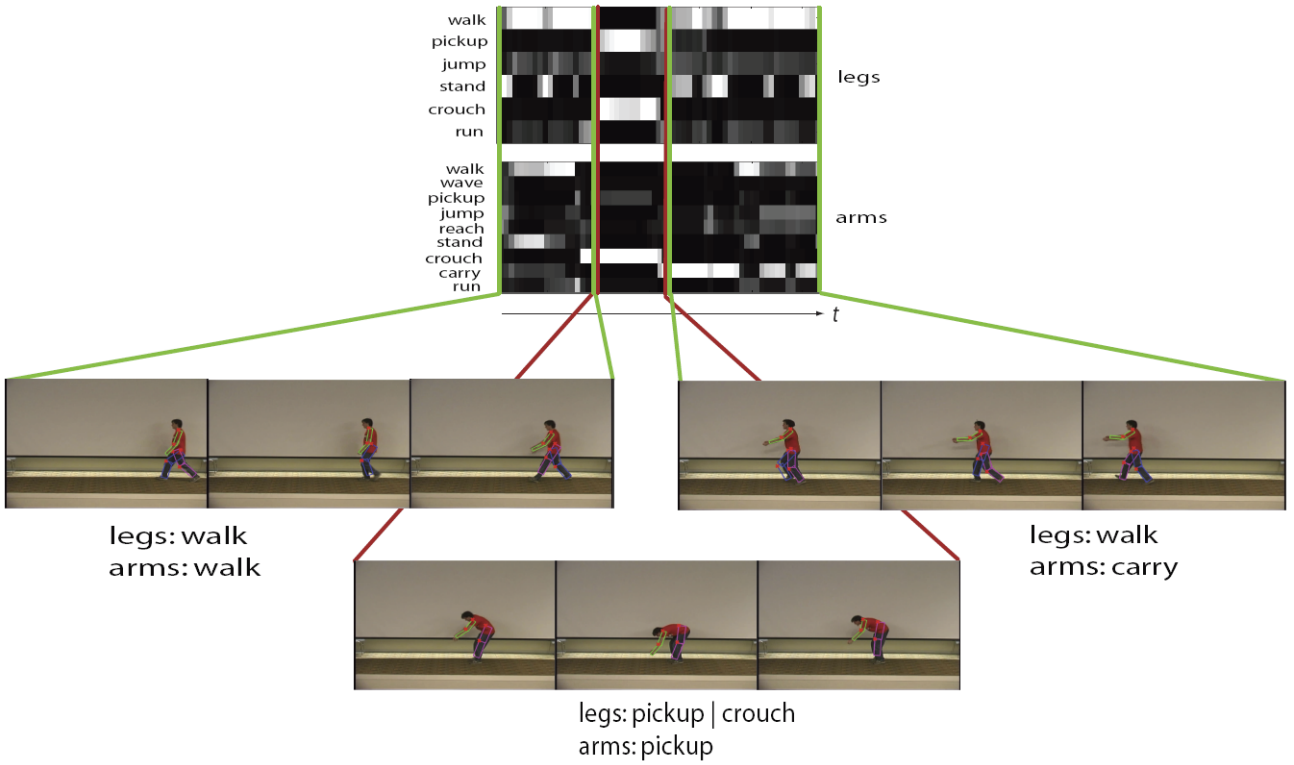


Fig. 5 Using activity models for each body part, we compute posteriors of sequences. After that, HMM posteriors for right and left parts of the body are queried together using finite state automata of the query string. *Top*: Average HMM posteriors for the legs and arms of sequence `walk-pickup-carry` (performed by a male subject) are shown. As it can be seen, maximum likelihood goes from one action HMM to the other within the activity HMM as the action in the video changes. This way, we achieve automatic segmentation of activities and there is no need to use other motion segmentation procedures. *Bottom*: Corresponding frames from the subsequences are shown. This sequence is correctly labeled and segmented as `walk-pickup-carry` as the corresponding query is evaluated

transition function where $\delta : Q \times \Sigma \to Q$, s_0 is the (set) of initial states, and F is the set of final states. In our representation, each state $q_i \in Q$ corresponds to the case where the subject is inside a particular action. Transitions between states (δ) represent the actions taking place. Transitions of the form x^{u_x} means action x sustained for u_x length, which means that actions shorter than their specified unit length do not cause the FSA to change its state. More specifically, each x^{u_x} (shown over the transition arrows) represents a smaller FSA on its own, as shown in Fig. 7. This small FSA reaches in its end state when the action is sustained for u_x number of frames. This regulation is needed in order to eliminate the effect of short-term noise.

While forming the finite state automata, as in Fig. 8, each action is considered to have a unit length u_x. A query string is converted to a regular expression, and then to an FSA based on these unit lengths of actions. Unit action length is based on two factors: first is the fps of the video, second is the action's level of sustainability. Actions like walking and running are sustainable; thus their unit length is chosen to be longer than those of localizable actions, like jump and reach.

We have an FSA F, and wish to compute the posterior probability of any string accepted by this FSA, conditioned on the observations. We write Σ_F for the strings accepted by the FSA. We identify the alphabet of the FSA with states—or groups of states—of our model, and get

$$P(F|o_1, \ldots, o_T, \theta) \propto \sum_{\sigma \in \Sigma_F} P(\sigma, o_1, \ldots, o_T|\theta) \tag{10}$$

where the constant of proportionality can be obtained from the forward-backward algorithm, as in Sect. 4.3. The term $\sum_{\sigma \in \Sigma_F} P(\sigma, o_1, \ldots, o_T|\theta)$ requires some care. We label the states in the FSA with indices $1, \ldots, Q$. We can compute this sum with a recursion in a straightforward way: Write

$$\begin{aligned} Q_{ijs} &= P\{\text{a string of length } i \text{ that takes } F \text{ to state } j \text{ and} \\ &\qquad \text{has last element } s, \text{ joint with } o_1, \ldots, o_i\} \\ &= \sum_{\substack{\sigma \in \text{strings of length } i \text{ with} \\ \text{last character } s \text{ that take } F \text{ to } j}} P(\sigma, o_1, \ldots, o_i|\theta) \end{aligned} \tag{11}$$

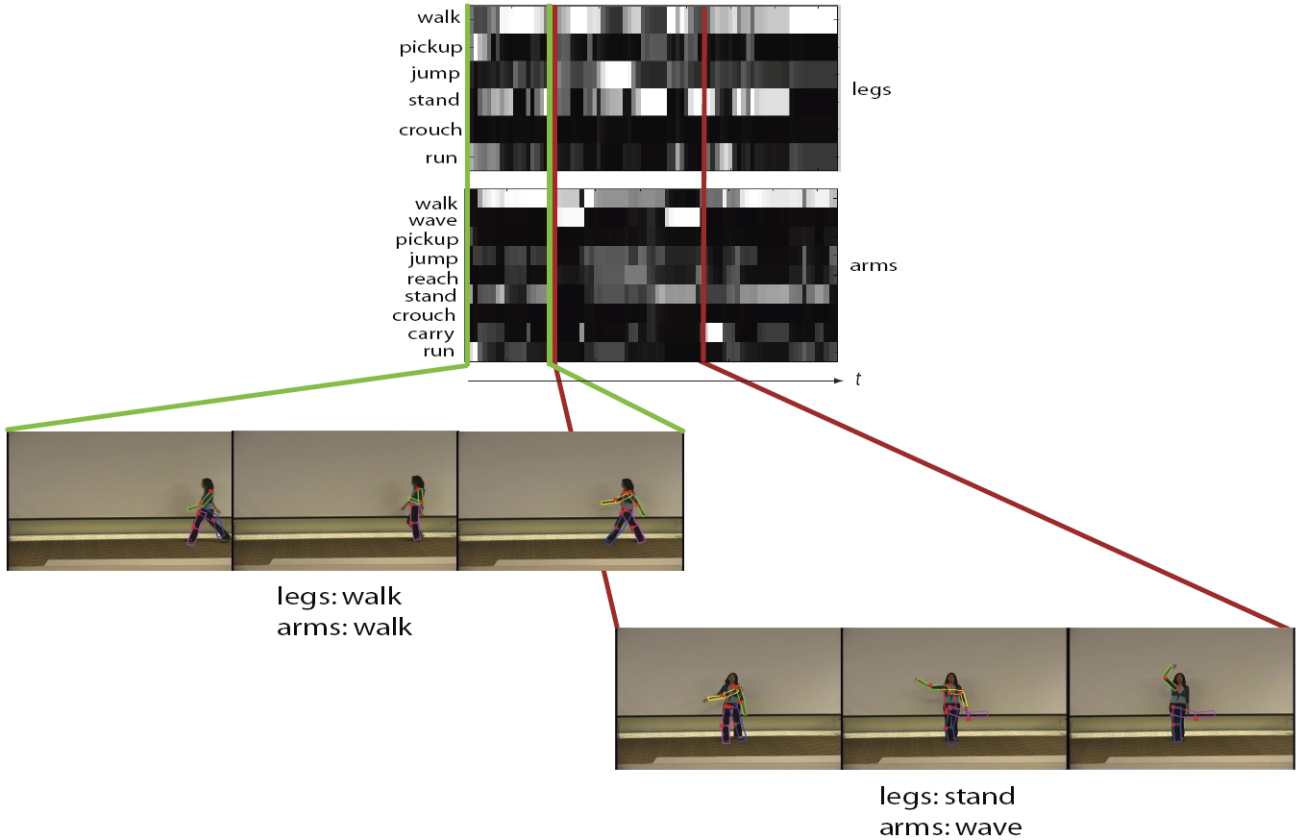


Fig. 6 Another example sequence from our system, performed by a female subject. In this sequence, the subject first walks into the scene, stops and waves for some time, and then walks out of the sequence. A query for `walk-wave-walk` for arms and `walk-stand-walk` for legs returned this sequence as top one, despite the noise in tracking. Again, by examining the posterior maps for each limb, we can identify the transitions between actions. *Top*: Posterior probability maps for legs and arms. *Bottom*: Corresponding frames from the correctly identified subsequences are shown

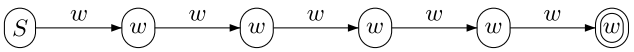


Fig. 7 The FSA for a single action is constructed based on its unit length. Here the expansion of the `walk` FSA is shown (w represents walk). As an example, unit length of `walk` is set to 5 frames ($u_w = 5$). So the corresponding FSA consist of five states and the probability of it reaching its final state requires that we observe five consecutive frames of `walk`.

Write $\mathrm{Pa}(j)$ for the parents of state j in the FSA (that is, the set of all states such that a single transition can take the FSA to state j). Write $\delta_{i,s}(j) = 1$ if F will transition from state i to state j on receiving s and zero otherwise; then we have

$$Q_{1js} = \sum_{u \in s_0} P(s, o_1|\theta)\delta_{u,s}(j) \tag{12}$$

and

$$Q_{ijs} = \sum_{k \in \mathrm{Pa}(j)} \delta_{k,s}(j) P(o_i|s,\theta) \left[\sum_{u \in \Sigma} P(s|u,\theta) Q_{(i-1)ku} \right] \tag{13}$$

Then

$$\sum_{\sigma \in \Sigma_F} P(\sigma, o_1, \ldots, o_T|\theta) = \sum_{u \in \Sigma, v \in s_e} Q_{Tvu} \tag{14}$$

and we can evaluate Q using the recursion. Notice that nothing major changes if each item u of the FSA's alphabet represents a set of HMM states (as long as the sets form an equivalence relation). We must now modify each expression to sum states over the relevant group. So, for example, if we write s_u for the *set* of states represented by the alphabet term u, we have

$$Q_{1ju} = \sum_{u \in s_0} \sum_{v \in s_u} P(v, o_1|\theta)\delta_{u,s}(j) \tag{15}$$

and

$$Q_{iju} = \sum_{k \in Pa(j), v \in s_u} \delta_{k,v}(j) P(o_i|v,\theta) \times \left[\sum_{u \in \Sigma, w \in s_u} P(v|w,\theta) Q_{(i-1)ku} \right] \tag{16}$$

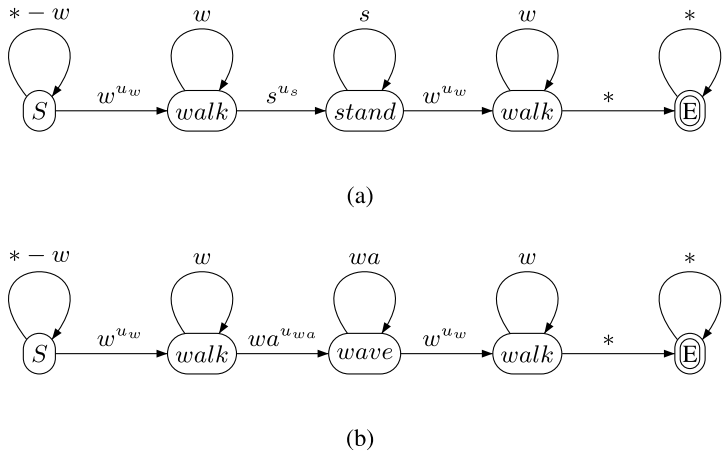


Fig. 8 To retrieve complex composite activities, we write separate queries for each of the body parts. Here, example query FSAs for a sequence where the subject walks into the view, stops and waves and then walks out of the view are shown. *Top*: FSA formed for the legs `walk-stand-walk`. *Bottom*: The corresponding query FSA for the arms with the string `walk-wave-walk`. Here, w is for walk, s for stand, wa for wave and u_x's are the corresponding unit lengths for each action x

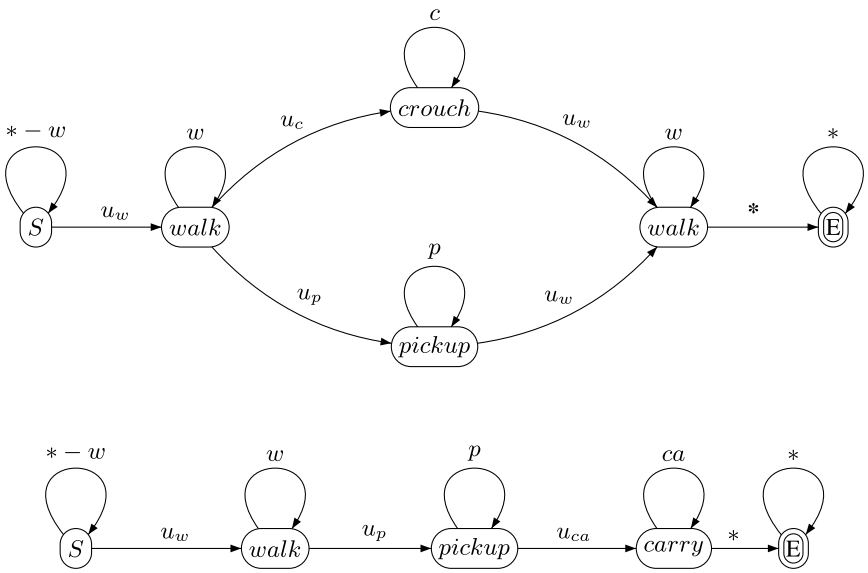


Fig. 9 Query for a video where the person walks, pickups something and carries it. Here, w is for walk, c for crouch, p for pickup and ca is for carry actions. Notice the different and complex representation achievable by writing queries in this form. Arms and legs are queried separately, composited across time and body. Also note that, since `pickup` and `crouch` actions are very similar in dynamics for the legs, we can form an `OR` query and do more wide-scale searches

A tremendous attraction of this approach is that no visual example of a motion is required to query; once one has grasped the semantics of the query language, it is easy to write very complex queries which are relatively successful.

The alphabet from which queries are formed consists in principle of $6^2 \times 9^2$ terms (one has one choice each for each leg and each arm). We have found that the tracker is not sufficiently reliable to give sensible representations of both legs (resp. arms). It is often the case that one leg is tracked well and the other poorly, mainly because of the occlusions. We therefore do not attempt to distinguish between legs (resp. arms), and reduce the alphabet to terms where either leg (resp. either arm) is performing an action; this gives an alphabet of 6×9 terms (one choice at the leg and one at the arm). This is like a noisy OR operation over the signals coming from top and bottom parts of the body. When any of the signals are present we take the union of them to represent the body pose.

Using this alphabet, we can write complex composite queries, for example, searching for strings that have several (`l-walk; a-walk`)'s followed by several `(l-stand; a-wave)` followed by several `(l-walk; a-walk)` yields sequences where a person walks into view, stands and waves, then walks out of view (see Fig. 8 for corresponding FSAs).

6 Experimental Results

Using limb activity models, we can do complex activity search with fair accuracy.

Clothing presents a variety of problems. We know of no methods that behave well in the presence of long coats, puffy jackets or of skirts. Our subjects wear a standard uniform of shirt and trousers. However, as Fig. 12 shows, the colour, arm-length and looseness of the shirts varies, as does the cut of the trousers and the presence of accessories (a jersey).

These variations are a fairly rich subset of those that preserve the silhouette. Our method is robust to these variations, and we expect it to be robust to any silhouette preserving change of clothing.

Datasets We collected our own set of motions, involving three subjects wearing a total of five different outfits in a total of 73 movies (15 Hz). Each video shows a subject instructed to produce a complex activity. The sequences differ in length. The complete list of activities collected is given in Table 1.

For viewpoint evaluation, we collected videos of 5 actions: jog, jump, jumpjack, wave and reach. Each action is performed in 8 different directions to the camera, making a total dataset of 40 videos (30 Hz). Fig. 14 shows example frames of this dataset.

For evaluating our system on complex backgrounds and also on football movements, we used video shootage from the TV series Friends. We have extracted 19 sequences of varying activities from the episode in which the characters play football in the park. The result is an extremely challenging dataset; the characters change orientation frequently, the camera moves, there are zoom-in and zoom-out effects and a complex and changing background. Different scales and occlusions make tracking even harder. In Fig. 17, we show example frames from this dataset with superimposed tracks.

Performance over a set of queries is evaluated using mean average precision (MAP) of the queries. Average precision of a query is defined as the area under the precision-recall curve for that query and a higher average precision value means that more relevant items are returned earlier.

More formally, average precision *AveP* over a set S is defined as

$$AveP = \frac{\sum_{r=1}^{N}(P(r) \times rel(r))}{\text{number of relevant documents in S}}$$

Here, r is the rank of the item, N is the number of retrieved items and $rel(r)$ is the binary relevance vector for each item in S and $P(r)$ precision at a given rank.

Limb activity models were fit using a collection of 10938 frames of motion capture data released by Electronic Arts in 2002, consisting of assorted football movements. To model our motion capture collection reasonably well, we choose a set of 9 actions. While these actions are abstract building blocks, the leg models correspond reasonably well to: run, walk, stand, crouch, jump, pickup (total of 6 actions). Similarly, the arm models correspond reasonably well to: run, walk, stand, reach, crouch, carry, wave, pickup, jump motions (total of 9 actions). Figure 10 shows the posterior for each model applied to labelled motion capture data; this can be interpreted as a class confusion matrix within the motion capture dataset itself. Limb activity models require that 3D coordinates of limbs be vector quantized. The choice of procedure has some effect on the outcome (details in Sect. 6.5).

Table 1 Our collection of video sequences, named by the instructions given to actors

Context	# videos	Context	# videos
crouch-run	2	run-backwards-wave	2
jump-jack	2	run-jump-reach	5
run-carry	2	run-pickup-run	5
run-jump	2	walk-jump-carry	2
run-wave	2	walk-jump-walk	2
stand-pickup	5	walk-pickup-walk	2
stand-reach	5	walk-stand-wave-walk	5
stand-wave	2	crouch-jump-run	3
walk-carry	2	walk-crouch-walk	3
walk-run	3	walk-pickup-carry	3
run-stand-run	3	walk-jump-reach-walk	3
run-backwards	2	walk-stand-run	3
walk-stand-walk	3		

Controls In order to analyse the performance of our approach, we have implemented three controls. Control 1 is single action SVM classifiers over raw 2D tracks (details in Sect. 6.2.1). We expect that discriminative methods applied to 2D data perform poorly because intra-class variance overwhelms available training data. In comparison, our method benefits by being able to estimate dynamical models on motion capture dataset. Control 2 is action SVMs built on 3D lifts of the 2D tracks (for details see Sect. 6.2.2). Although they have view-invariance aspect, we also expect them performing poorly, because they suffer from data shortage and noise in lifts. And finally, Control 3 is the SVM classifiers over 3D motion capture dataset (details in Sect. 6.2.3). They are also insufficient in tolerating the different levels of sustainability and different speeds of activities. This also causes problems with the composition. On contrary, our model supports high level of composition and its generative nature handles different lengths of activities easily.

6.1 Searching

We evaluate our system by first identifying an activity to search for, then marking relevant videos, then writing a regular expression, and finally determining the recall and precision of the results ranked by P(FSA in end state|sequence). On the traditional simple queries (`walk, run, stand`), MAP value is 0.9365; only a short sequence of `run` action is confused with `walk` action. Figures 12 and 13 show search results for more complex queries. Our method is able to respond to complex queries quite effectively. The biggest difficulty we faced was to find an accurate track for each limb

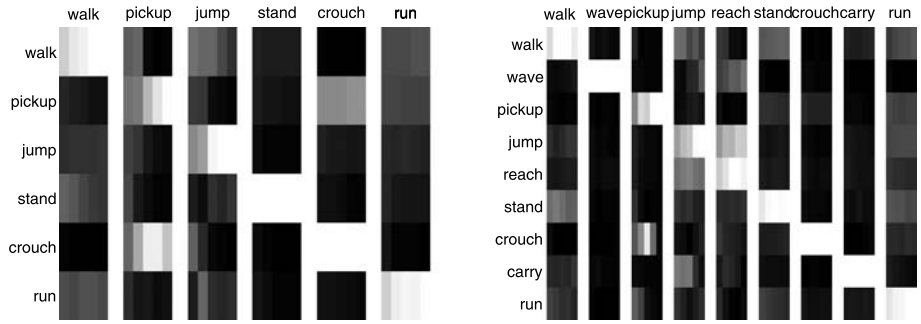


Fig. 10 Local dynamics is quite a good guide to a motion in the motion capture data set. Here we show HMM interpretation of these dynamics. Each column represents 5 frame average HMM posteriors for the motion capture sequences (*left*: legs, *right*: arms). These images represent the expressive and generative power of each action HMM. For example, `pickup` HMM for the legs gives high likelihood for `pickup` and `crouch` action, whereas `crouch` HMM for the legs is more certain when it observes a `crouch` action, therefore it produces a higher posterior as opposed to `pickup`. The asymmetry present in this figure is due to the varying number of training examples available in motion capture dataset for each action. The higher the number of examples for an action, the better HMMs are fit. This image can also be interpreted as a confusion matrix between actions. Most of the confusion occurs between dynamically similar actions. For example, for `pickup` motion, the leg HMMs may fire `pickup` or `crouch` motions. These two actions are in fact very similar in dynamics. Likewise, for `reach` motion, arm HMMs show higher posteriors for `reach`, `wave` or `jump` motions

due to the discontinuity in track paths and left/right ambiguity of the limbs. That's why some sequences are identified poorly.

We have evaluated several different types of search. In Type I queries, we encoded activities where legs and arms are doing different actions simultaneously, for instance "walking while carrying". In Type II queries, we evaluated the cases where there are two consecutive actions, same for legs and arms (like a `crouch` followed by a `run`). Type III queries search for activities that are more complex; these are activities involving three consecutive actions where different limbs may be doing different things (ex: `walk-stand-walk` for legs; `walk-wave-walk` for arms). MAP value for these sets of complex queries is 0.5636 with our method.

The performance over individual type of activities is presented in Table 2. Based on this evaluation, we can say that our system is more successful in retrieving complex activities as in Type III queries. That's mostly because complex activities occur within longer sequences which are less affected by the short-term noise of tracking and lifting.

Torso Exclusion In our method, we omit the torso information and query over the limbs only. This is because we found that torso information is not particularly useful. The results demonstrating this case is given in Fig. 11. When we query using the whole body, including torso, we get an Mean Average Precision of 0.501, whereas if we query using limbs only, we get a MAP of 0.5636. We conclude that using torso is not particularly informative. This is mostly because in our set of actions, the torso HMMs fire high posteriors for more than one action, and therefore, they don't help much in discriminating between actions.

Table 2 The Mean Average Precision values for different types of queries. We have three types of query here. Type I: single activities where there is a different action for legs and arms (ex: walk-carry). Type II: two consecutive actions like `crouch` followed by a `run`. Type III: activities that are more complex, consisting of three consecutive actions where different body parts may be doing different things (ex: `walk-stand-walk` for legs; `walk-wave-walk` for arms)

Query type	MAP
Type I	0.5562
Type II	0.5377
Type III	0.5902

6.2 Controls

We cannot fairly compare to HMM models because complex activities require large numbers of states (which cannot be learned directly from data) to obtain a reasonable search vocabulary. However, discriminative methods are rather good at classifying activities without explicit dynamical models, and it is by no means certain that dynamical models are necessary (see Sect. 2.4 in the discussion of related work). Discriminative models regard changes in the temporal structure of an action as likely to be small, and so well covered by training data. For this reason, we choose to compare with discriminative methods. There are three possible strategies, and we compare to each. First, one could simply identify activities from image-time features (like, for example, the work of Blank et al. 2005; Efros et al. 2003; Schuldt et al. 2004). Second, one could try to identify activities from lifted data, using lifted data to train models. Finally, one could try to identify activities from lifted data, but perform training

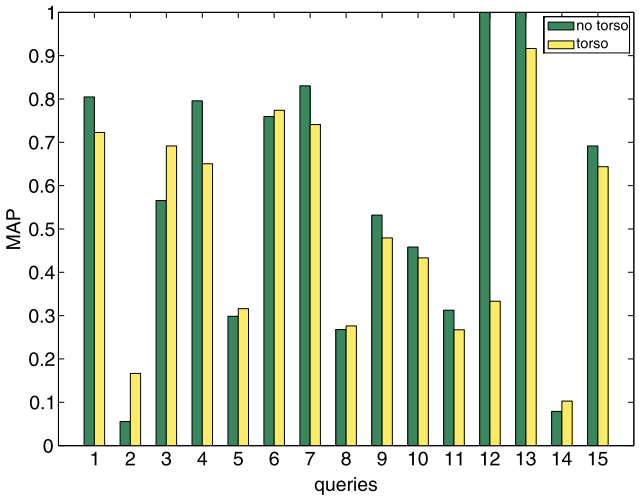


Fig. 11 Mean Average Precision values of our method with respect to torso inclusion. The MAP of our method over the whole body is 0.501 when we query with the torso, whereas it is 0.5636 when we query over the limbs only. For some queries, including torso information increases performance slightly, however, on the overall, we see that using torso information is not very informative

using motion capture data.

6.2.1 Control 1: SVM Classifier over 2D Tracks

To evaluate the effectiveness of our approach, we implemented an SVM-based action classifier over the raw 2D tracks. Using the tracker outputs for 17 videos as training set (chosen such that 2 different video sequences are available for each action), we built action SVMs for each limb separately. We used RBF kernel and 7 frame snippets of tracks to build the classifiers for this setting has given the best results for this control. A grid search over parameter space of the SVM is done and best classifiers are selected using 10-fold cross-validation. The performance of these SVMs are then evaluated over the remaining 56 videos. Figures 12 and 13 show the results. MAP value over the sets of queries is 0.3970 with Control 1. Note that for some queries, SVMs are quite successful in marking relevant documents. However, on the overall, SVMs are penalized by the noise and variance in dynamics of the activities. Our HMM limb activity models, on the other hand, deal with this issue by the help of the dynamics introduced by synthesized motion capture data. SVMs would need a great deal of training data to discover such dynamics.

6.2.2 Control 2: SVM Classifier Over 3D Lifts

We have also trained SVM classifiers over 3D lifted track points. Mean average precision of the whole query set in this case is 0.3963. This is not surprising, since there is some noise introduced by lifting 2D tracks, causing the performance of the classifier to be low. In addition, HMM method still has the advantage of using the dynamics introduced by motion capture dataset. The corresponding results are presented in Figs. 12 and 13. These results support the fact that motion capture dataset dynamics is a good clue for human action detection in our case.

6.2.3 Control 3: SVM Classifier Over 3D Motion Capture Set

Our third control is based on SVM classifiers built over 3D motion capture data set. We used the same vector-quantization as in building our HMM models, for generalization purposes. Mean average precision of the query set here is 0.3538. Although they rely on extra information added with the presence of motion capture data set, we observed that, these SVMs are also insufficient in tolerating the different levels of sustainability and different speeds of activities. This also causes problems with the composition. Generative nature of HMMs eliminates such difficulties and handles with varying length actions/activities easily.

6.3 Viewpoint Evaluation

To evaluate our method's invariance to viewpoint, we queried 5 single activities (`jog, jump, jumpjack, reach, wave`) over the data set that has 8 different view directions of the subjects (Fig. 14). We assume that if these simple sequences produce reliable results, the complex sequences will be accurate as well. Results of this evaluation are shown in Figs. 15 and 16.

As Fig. 15 shows, the performance is not significantly affected by the change in viewpoint, however there is slight lost of precision in some angles due to tracking and lifting difficulties in those view directions. Examples of non-reliable tracks are also shown in Fig. 15. Due to occlusions and motion blur, the tracker tends to miss the moving arms quite often, making it hard to discriminate between actions.

Figure 16 shows the overall precisions averaged w.r.t. angles for each action. Not surprisingly, most confusion occurs between `reach` and `wave` actions. If the tracker misses the arm during these actions, it is highly likely that the dynamics of these actions will not be recovered and those two actions will resemble each another. On the other hand, `jumpjack` action is a combination of wave and jump actions, which is also subject to high confusion.

6.4 Activity Retrieval Over Football Sequences with Complex Backgrounds

In order to see how well our algorithm will behave in football sequences with complicated settings, we tested our

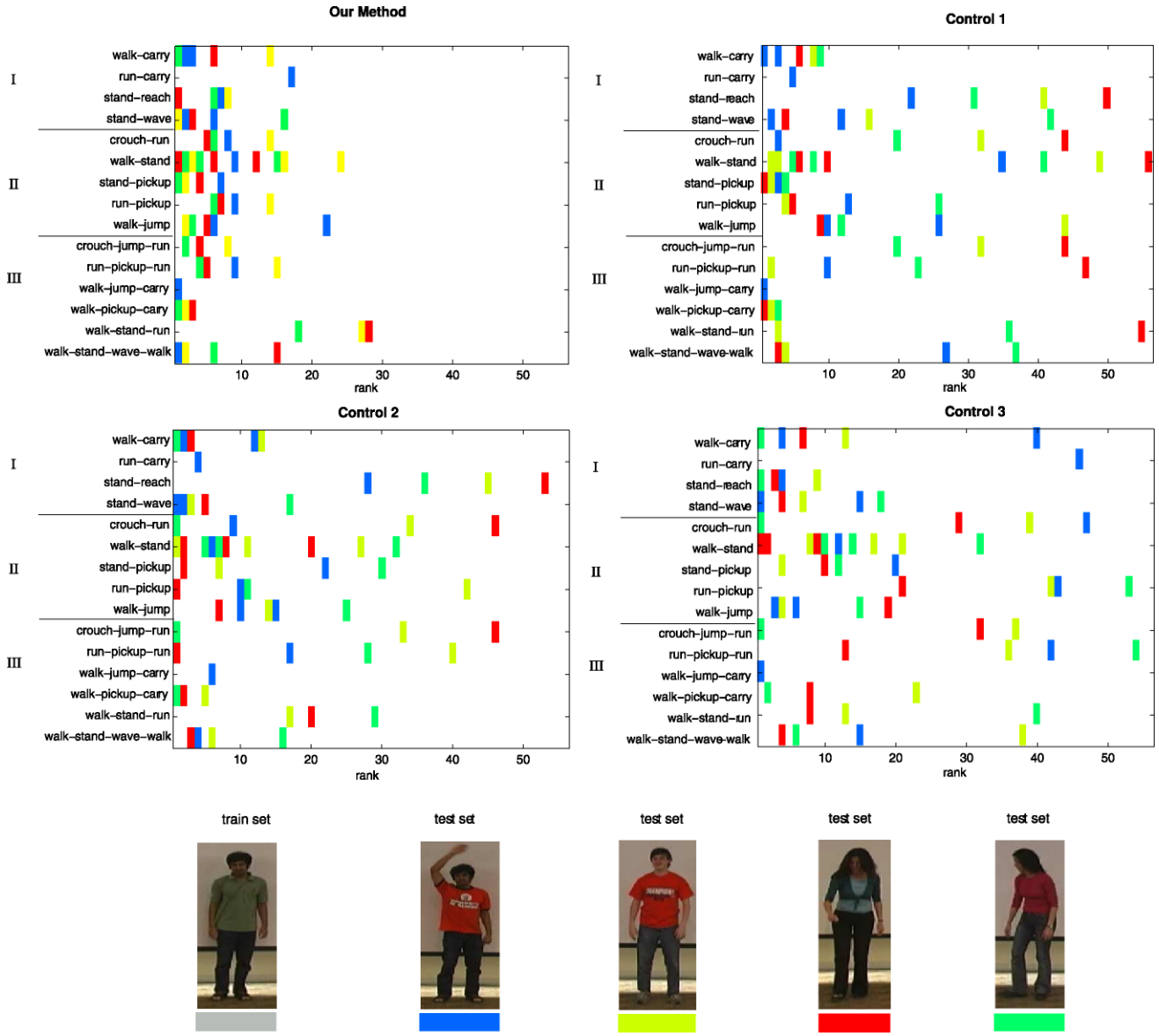


Fig. 12 Our representation can give quite accurate results for complex activity queries, regardless of the clothing worn by the subject. The results of ranking for 15 queries over our video collection. In these images, a colored pixel indicates a relevant video. An ideal search would result in an image where all the colored pixels are on the left of the image. Each color represents a different outfit. We have three types of query here (see text for details). *Top left*: The ranking results of our activity modeling based on joint HMMs and motion capture data. We have used $k = 40$ in vector quantization. Note that the videos retrieved in top columns are more likely to be relevant and the retrieval results are more condensed to the left. Note that the choice of the outfit doesn't affect the performance. *Top right*: Control 1: Separate SVM classifiers for each action over the 2D tracks of the videos. Composite queries built on top of a discriminative (SVM) based representation are not as successful as querying with our representation. Again, clothing does not affect the result. *Bottom left*: Control 2: SVM classifiers over 3D lifted tracks. *Bottom right*: Control 3: SVM classifiers over 3D motion capture data. While these classifiers benefit from dynamics of mocap data, they suffer due to lack of composition. For some queries, SVM performances are good, however, on the overall, precision and recall rate is low. Also, note that the relevant videos are all scattered through the retrieval list

approach over football sequences taken from Friends TV Show. We have constructed a dataset, consisting of 19 short clips, in which characters play football in park (from Episode 9 of Season 3). We then annotated the actions of a single person in these clips by our available set of actions. This dataset is extremely challenging; the characters change orientation frequently, the camera moves, there are zoom-in and zoom-out effects and a complex and changing background. Examples frames from these sequences are shown in Fig. 17.

Since we built our activity models using a dataset of motion captured American football movements, we expect to

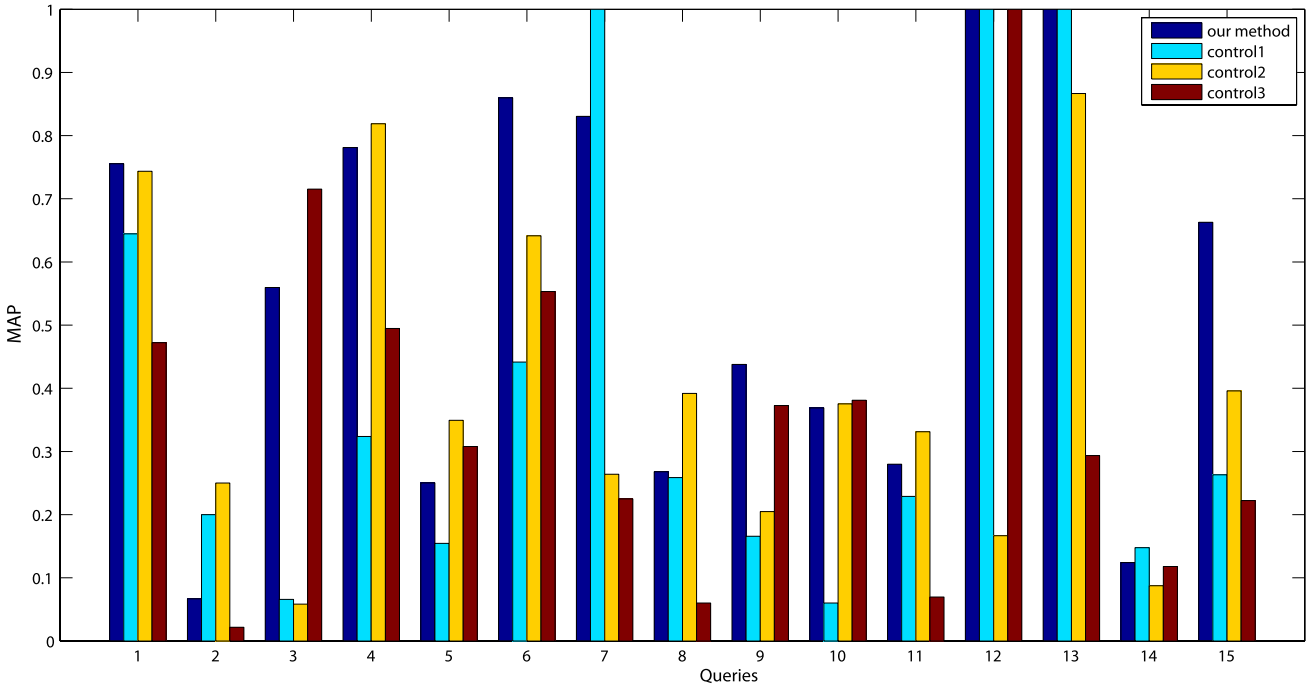


Fig. 13 Average precision values for each query. Our method gives a mean average precision (MAP) of 0.5636 over the whole query set. Control 1's MAP value is 0.3970. Control 2 acquires a MAP of 0.3963, while it is 0.3538 for Control 3

Fig. 14 Example frames from our dataset of single activities with different views. *Top row*: Jogging 0 degrees, Jump 45 degrees, jumpjack 90 degrees, reach 135 degrees. *Bottom row*: wave 180 degrees, jog 225 degrees, jump 270 degrees, jumpjack 315 degrees

have a higher accuracy in domains with similar actions. We test our system using 10 queries, ranging from simple to complex, and results are given in Fig. 18. For 9 out of 10 queries, the top retrieved video is a relevant video which includes the queried activity. Our MAP of 0.8172 over this dataset shows that our system is quite good in retrieving football movements, even in complicated settings.

6.5 Vector Quantization for Action Dynamics

We vector quantize 3D coordinates of the limbs when forming the action models. This quantization step is useful to have a more general representation of the domain. We use k-means as our quantization method. Since k-means is very dependent on the initial cluster centers, we run each clustering 10 times and choose the best clusters such that the inter-cluster distance is maximized and intra-cluster distance is minimized. Our experiments show that when we choose number of clusters k in k-means as low as 10, the retrieval process suffers from information loss due to excessive generalization. Using $k = 40$ gives the best results over this dataset. Note that, one can try different levels of quantization for different limbs, however, our empirical evaluation shows that doing so does not provide a significant performance improvement.

7 Discussions and Conclusion

There is little evidence that a fixed taxonomy for human motion is available. However, research to date has focused on multi-class discrimination of simple actions. Everyday activities are more complex in nature. People tend to perform

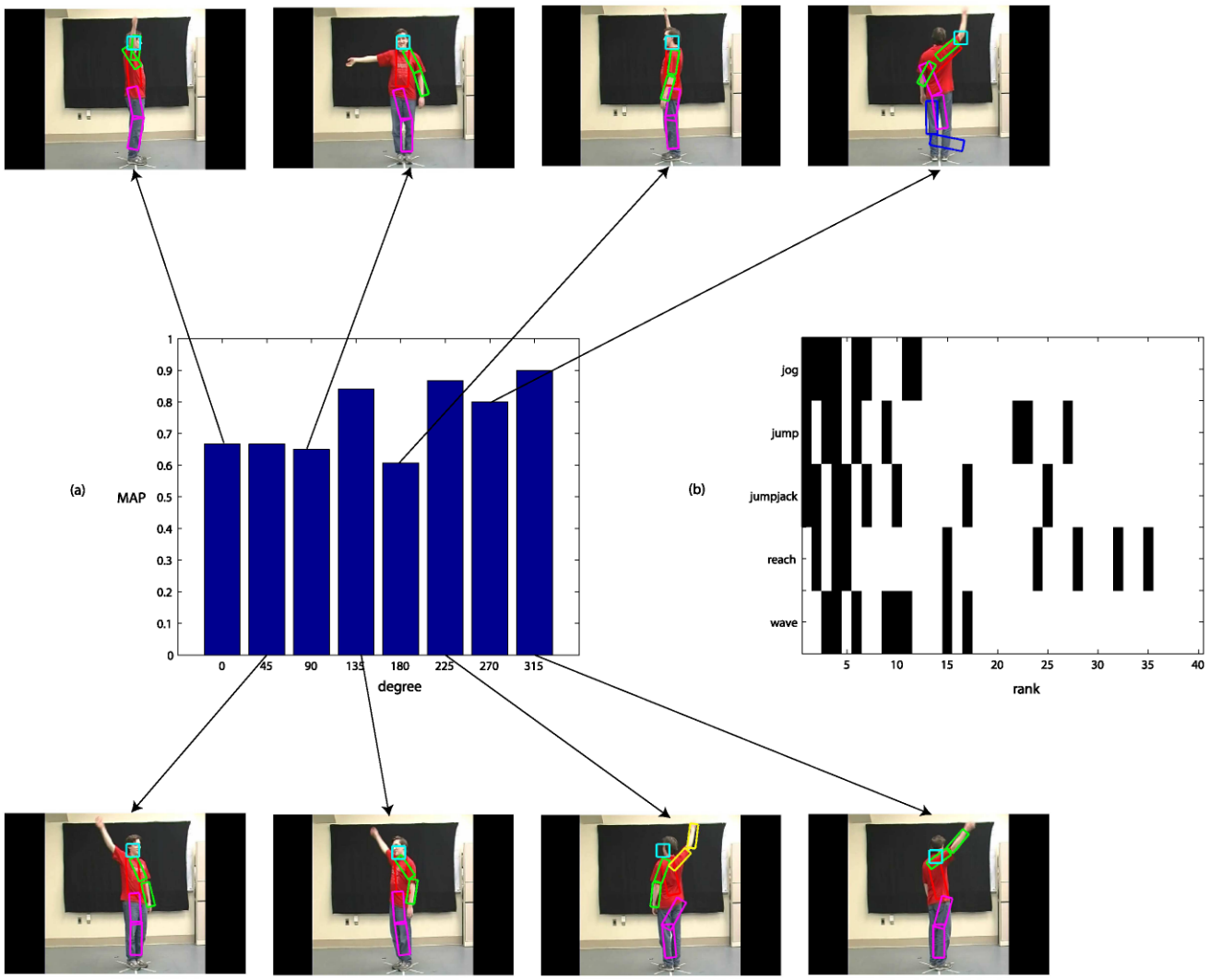


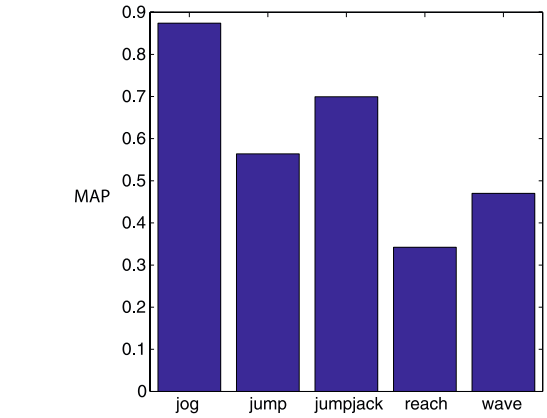
Fig. 15 For evaluating our methods invariance to view direction change, we have a separate dataset of single activities `1-jog 2-jump 3-jumpjack 4-reach 5-wave`. **(a)** Average precision values for each viewing direction. Some viewing directions has slightly worse performance due to the occlusion of the limbs and poor tracking response to bendings of the limbs in some view directions. Here, we show some representative frames with tracks for the `wave` action. As it can be seen, tracker sometimes misses the moving arm, causing the performance of the system to degrade. However, we can say that on the overall, performance is not significantly affected with the change in viewpoint. **(b)** The ranking of the five queries of single actions separately. The poorest response comes from `reach` action, which inevitably confuses with `wave`, especially when the arms are out of track in the middle of the action. Here, note that SVMs would need to be retrained for each viewing direction, while our method does not

composite activities both on the spatio and temporal dimensions.

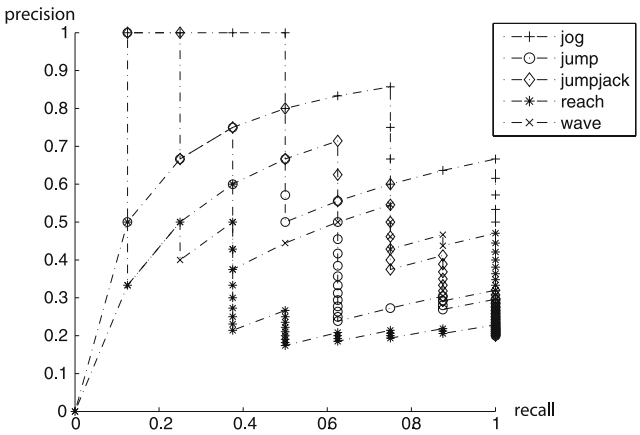
We have demonstrated a representation of human motion that can be used to query for complex activities in a large collection of video. We build our queries using finite state automata and for each limb, we write separate queries. We are aware of no other method that can give comparable responses for such queries.

Our representation uses a generative model, built using motion capture and applying it over video data. By joining models of atomic actions to form activity models, we perform minimum parameter estimation. This can also be thought as an instance of transfer learning; we transfer the knowledge we gain from 3D motion capture data, to 2D everyday activity data.

We expect these HMM's to simulate rendered activity extremely poorly, as they are not constructed to produce good transitions between frames. We are not claiming that the generative model concentrates probabilities only on correct human actions, and we don't believe that any other work in activity makes this claim; the standards of performance required to do credible human animation are now extremely high (e.g. Kovar et al. 2002; Lee et al. 2002; Arikan and Forsyth 2002; review in Forsyth et al. 2006), and it is



(a)



(b)

Fig. 16 **(a)** The mean precisions of each action averaged over the viewpoint change. The most confusion occurs between `reach` and `wave` actions. **(b)** Respective precision-recall curves for each action averaged over the angles. SVMs would need to be retrained for each viewing direction, while our method does not

known to be very difficult to distinguish automatically between good and bad animations of humans (Ren et al. 2005; Ikemoto et al. 2007; Forsyth et al. 2006). Instead, we believe that the probability that appears on actions that are not natural, does not present difficulties as long as the models are used for inference, and our experimental evidence bears this out. Crucially, when one infers activity labels from video, one can avoid dealing with sequences that do not contain natural human motion.

One of the strengths of our method is that, when searching for a particular activity, no example activity is required to formulate a query. We use a simple and effective query language; we simply search for activities by formulating sentences like "Find action X followed by action Y" or "Find videos where legs doing action X and arms doing action Y" via finite state automata. Matches to the query are evaluated and ranked by the posterior probability of a state representation summed over strings matching the query. Using a strategy like ours, one can search for activities that have never been seen before.

As our results show, query responses are unaffected by clothing, and our representation is robust to aspect. Our representation significantly outperforms discriminative representations built using image data alone. It also outperforms models built on 3D lifted responses, meaning that the dynamics transferred from motion capture domain to real world domain helps in retrieval of complex activities. In addition, the generative nature of HMM models helps to compensate the different levels of sustainability of the actions and makes composition across time easier.

Moreover, since our representation is in 3D, we don't need to retrain our models separately for each viewing direction. We show that our representation is mostly invariant to change in viewing direction.

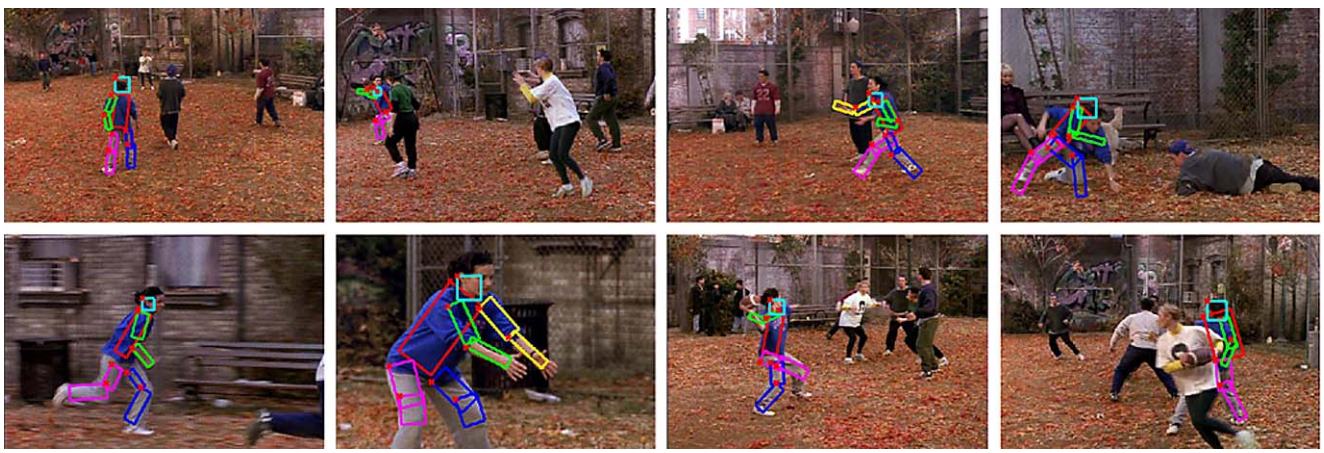


Fig. 17 Example frames from the Friends dataset. This dataset consists of 19 short clips compiled from the Friends TV show (from Episode 9 of Season 3). This is a challenging dataset, in which there are lots of camera movement, scale and orientation changes, zoom-in and out effects. In addition, occlusions make the tracking harder in this dataset. In this figure, frames with relatively good tracks (which are superimposed) are shown

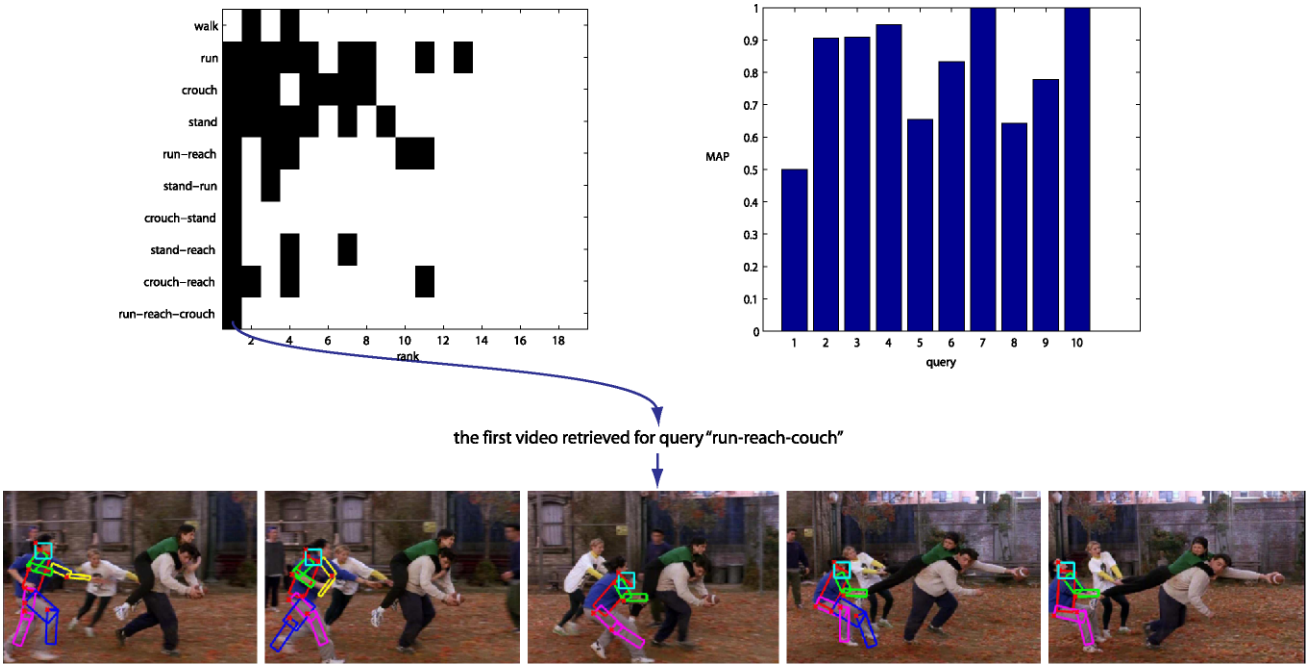


Fig. 18 Results of our retrieval system over the Friends dataset. Our system is quite successful over this dataset. Since our activity models are formed using motion capture dataset which consists of American football movements, this dataset is a natural application domain for our system. In 9 out of 10 queries, our system returns a relevant video as the top result and we achieve a MAP of 0.8172 over this dataset

The biggest difficulty we faced was to properly track the fast moving limbs and then lifting to 3D in the presence of such tracking errors and ambiguities. That's why we can say that there is much room for improvement; a better tracker would give better results immediately. Further improvements would involve a richer vocabulary of actions, or some theory about how a canonical action vocabulary could be built; a front-end of discriminative features (after Sminchisescu et al. 2005a, 2005b); improved lifting to 3D; and, perhaps, a richer query interface.

References

Aggarwal, J., & Cai, Q. (1999). Human motion analysis: A review. *Computer Vision and Image Understanding*, *73*(3), 428–440.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence*, *23*(2), 123–154.

Ando, R. K., & Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, *6*, 1817–1853.

Arikan, O., Forsyth, D., & O'Brien, J. (2003). Motion synthesis from annotations, In *Proc. of SIGGRAPH*, 2003.

Arikan, O., & Forsyth, D. A. (2002). Interactive motion generation from examples. In *Proceedings of the 29th annual conference on computer graphics and interactive techniques* (pp. 483–490). New York: ACM.

Barbič, J., Safonova, A., Pan, J.-Y., Faloutsos, C., Hodgins, J. K., & Pollard, N. S. (2004). Segmenting motion capture data into distinct behaviors. In *GI '04: Proceedings of the 2004 conference on graphics interface* (pp. 185–194), School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

Ben-Arie, J., Wang, Z., Pandit, P., & Rajaram, S. (2002). Human activity recognition using multidimensional indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(8), 1091–1104.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., & Basri, R. (2005). Actions as space-time shapes. In *Int. conf. on computer vision* (pp. 1395–1402).

Bobick, A. (1997). Movement, activity, and action: The role of knowledge in the perception of motion. *Proceedings of the Royal Society B*, *352*, 1257–1265.

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

Bobick, A. F., & Ivanov, Y. A. (1998). Action recognition using probabilistic parsing. In *CVPR* (p. 196).

Bobick, A., & Wilson, A. (1997). A state based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(12), 1325–1337.

Brand, M., & Kettnaker, V. (2000). Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 844–851.

Brand, M., Oliver, N., & Pentland, A. (1997). Coupled hidden Markov models for complex action recognition. In *IEEE conf. on computer vision and pattern recognition* (pp. 994–999).

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*(1), 41–75.

Dollár, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.

Efros, A. A., Berg, A. C., Mori, G., & Malik, J. (2003). Recognizing action at a distance. In *ICCV'03* (pp. 726–733).

Evgeniou, T., & Pontil, M. (2004). Regularized multi-task learning. In *17th SIGKDD conf. on knowledge discovery and data mining*, 2004.

Farhadi, A., Forsyth, D., & White, R. (2007). Transfer learning in sign language. In *IEEE conf. on computer vision and pattern recognition*, June 2007.

Felzenszwalb, P., & Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1), 55–79.

Feng, X., & Perona, P. (2002). Human action recognition by sequence of movelet codewords. In *3D data processing visualization and transmission* (pp. 717–721).

Fod, A., Matarić, M. J., & Jenkins, O. C. (2002). Automated derivation of primitives for movement classification. *Autonomous Robots*, *12*(1), 39–54.

Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., & Ramanan, D. (2006). Computational studies of human motion I: Tracking and animation. *Foundations and Trends in Computer Graphics and Vision*, *1*(2/3), 1–255.

Gavrila, D. M. (1999). The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, *73*(1), 82–98.

Hong, P., Turk, M., & Huang, T. (2000). Gesture modeling and recognition using finite state machines. In *Int. conf. automatic face and gesture recognition* (pp. 410–415).

Hongeng, S., Nevatia, R., & Bremond, F. (2004). Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, *96*(2), 129–162.

Howe, N. (2004). Silhouette lookup for automatic pose tracking. In *IEEE workshop on articulated and non-rigid motion* (p. 15).

Howe, N. R., Leventon, M. E., & Freeman, W. T. (2000). Bayesian reconstruction of 3d human motion from single-camera video. In *Proc. neural information processing systems* (pp. 820–826).

Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, *34*(3), 334–352.

Ikemoto, L., Arikan, O., & Forsyth, D. (2007). Quick transitions with cached multi-way blends. In *ACM symposium on interactive 3D graphics and games (I3D)*, 2007.

Ikizler, N., & Forsyth, D. (2007). Searching video for complex activities with finite state models. In *IEEE conf. on computer vision and pattern recognition*, June 2007.

Jenkins, O. C., & Matarić, M. J. (2003). Automated derivation of behavior vocabularies for autonomous humanoid motion. In *AAMAS '03: proceedings of the second international joint conference on autonomous agents and multiagent systems* (pp. 225–232). New York: ACM.

Jenkins, O. C., & Matarić, M. J. (2004). A spatio-temporal extension to isomap nonlinear dimension reduction. In *ICML'04: proceedings of the twenty-first international conference on machine learning* (p. 56). New York: ACM.

Kovar, L., Gleicher, M., & Pighin, F. (2002). Motion graphs. In *Proceedings of the 29th annual conference on computer graphics and interactive techniques* (pp. 473–482). New York: ACM.

Laptev, I., & Lindeberg, T. (2003). Space-time interest points. In *ICCV* (p. 432). Washington D.C., USA, 2003. Los Alamitos: IEEE Comput. Soc.

Lee, J., Chai, J., Reitsma, P., Hodgins, J., & Pollard, N. (2002). Interactive control of avatars animated with human motion data. In *Proc. of SIGGRAPH*, 2002.

Li, Y., Wang, T., & Shum, H.-Y. (2002). Motion texture: a two-level statistical model for character motion synthesis. In *Proceedings of the 29th annual conference on computer graphics and interactive techniques* (pp. 465–472). New York: ACM.

Matarić, M. J., Zordan, V. B., & Mason, Z. (1998). Movement control methods for complex, dynamically simulated agents: Adonis dances the macarena. In *AGENTS '98: proceedings of the second international conference on autonomous agents* (pp. 317–324). New York: ACM.

Matarić, M. J., Zordan, V. B., & Williamson, M. M. (1999). Making complex articulated agents dance. *Autonomous Agents and Multi-Agent Systems*, *2*(1), 23–43.

Mori, T., Segawa, Y., Shimosaka, M., & Sato, T. (2004). Hierarchical recognition of daily human actions based on continuous hidden Markov models. In *Int. conf. automatic face and gesture recognition* (pp. 779–784).

Niebles, J. C., Wang, H., & Fei-Fei, L. (2006). Unsupervised learning of human action categories using spatial-temporal words. In *British machine vision conference*, 2006.

Oliver, N., Garg, A., & Horvitz, E. (2004). Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding*, *96*(2), 163–180.

Pinhanez, C., & Bobick, A. (1997). Pnf propagation and the detection of actions described by temporal intervals. In *DARPA IU workshop* (pp. 227–234).

Pinhanez, C., & Bobick, A. (1998). Human action detection using pnf propagation of temporal constraints. In *IEEE conf. on computer vision and pattern recognition* (pp. 898–904).

Polana, R., & Nelson, R. (1993). Detecting activities. In *IEEE conf. on computer vision and pattern recognition* (pp. 2–7).

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. New York: Prentice Hall.

Ramanan, D., & Forsyth, D. (2003). Automatic annotation of everyday movements. In *Proc. neural information processing systems*, 2003.

Ramanan, D., Forsyth, D., & Zisserman, A. (2005). Strike a pose: Tracking people by finding stylized poses. In *IEEE conf. on computer vision and pattern recognition* (pp. 271–278).

Ramanan, D., Forsyth, D., & Zisserman, A. (2007). Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(1), 65–81.

Ren, L., Patrick, A., Efros, A. A., Hodgins, J. K., & Rehg, J. M. (2005). A data-driven approach to quantifying natural human motion. *ACM Transactions on Graphics*, *24*(3), 1090–1097.

Rose, C., Cohen, M. F., & Bodenheimer, B. (1998). Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, *18*(5), 32–40.

Ryoo, M. S., & Aggarwal, J. K. (2007). Recognition of composite human activities through context-free grammar based representation. In *IEEE conf. on computer vision and pattern recognition*, June 2007.

Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local svm approach. In *ICPR* (pp. 32–36). Washington D.C., USA, 2004. Los Alamitos: IEEE Comput. Soc.

Siskind, J. M. (2003). Reconstructing force-dynamic models from video sequences. *Artificial Intelligence*, *151*, 91–154.

Sminchisescu, C., Kanaujia, A., Li, Z., & Metaxas, D. (2005a). Conditional random fields for contextual human motion recognition. In *ICCV* (pp. 1808–1815).

Sminchisescu, C., Kanaujia, A., Li, Z., & Metaxas, D. (2005b). Discriminative density propagation for 3d human motion estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, *1*, 390–397.

Vecchio, D. D., Murray, R., & Perona, P. (2003). Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automatica*, *39*(12), 2085–2098.

Wilson, A., & Bobick, A. (1995). Learning visual behavior for gesture analysis. In *IEEE symposium on computer vision* (pp. 229–234).

Wilson, A., & Bobick, A. (1999). Parametric hidden Markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(9), 884–900.

Wong, S.-F., Kim, T.-K., & Cipolla, R. (2007). Learning motion categories using both semantic and structural information. In *IEEE conf. on computer vision and pattern recognition*, June 2007.

Yamato, J., Ohya, J., & Ishii, K. (1992). Recognising human action in time sequential images using hidden Markov model. In *IEEE conf. on computer vision and pattern recognition* (pp. 379–385).

Yang, J., Xu, Y., & Chen, C. S. (1997). Human action learning via hidden Markov model. *IEEE Transactions on Systems Man and Cybernetics*, *27*, 34–44.

Zhao, T., & Nevatia, R. (2004). Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(9), 1208–1221.