Real-Time Pose Estimation of Articulated Objects using Low-Level Motion

Ben Daubney, David Gibson, Neill Campbell
Department of Computer Science
University of Bristol, UK
daubney@cs.bris.ac.uk

Abstract

We present a method that is capable of tracking and estimating pose of articulated objects in real-time. This is achieved by using a bottom-up approach to detect instances of the object in each frame, these detections are then linked together using a high-level a priori motion model. Unlike other approaches that rely on appearance, our method is entirely dependent on motion; initial low-level part detection is based on how a region moves as opposed to its appearance. This work is best described as Pictorial Structures using motion. A sparse cloud of points extracted using a standard feature tracker are used as observational data, this data contains noise that is not Gaussian in nature but systematic due to tracking errors. Using a probabilistic framework we are able to overcome both corrupt and missing data whilst still inferring new poses from a generative model. Our approach requires no manual initialisation and we show results for a number of complex scenes and different classes of articulated object, this demonstrates both the robustness and versatility of the presented technique.

1. Introduction

Psychophysical experiments using the moving light display (MLD) have long demonstrated that motion can be used to extract high level information [11, 3]. These experiments represent each of the main joints as a single point, thus serving to degrade the appearance of a person to an absolute minimum, ensuring that any recognition cannot be based on appearance cues. It is currently unclear whether human perception of the MLD is achieved by considering all the moving lights as one entity (the change in configuration of this entity then represents motion) or whether the trajectories of the points are considered independently to determine whether their motion is gait-like and then the structure of the points is considered. The first of these approaches could be described as top-down and the second bottom-up. In this work we explore the latter; bottom-up estimation of pose using motion.

Bottom-up approaches attempt to break a large problem into smaller sub-problems, in the case of pose estimation this corresponds to first detecting individual parts, then assembling them into the most likely configuration. Current approaches use appearance cues for low-level part detection and then find the most likely configuration using techniques such as Dynamic Programming [8], Loopy Belief Propagation [15] or non-parametric Belief Propagation [19]. Our approach, rather than to use appearance, is to use motion for low-level part detection. We learn motion models that represent how we expect a feature to move if tracking a particular joint. These models are then used as low-level part detectors, to detect candidate limb positions based on the motion of a specific region. We then use Dynamic Programming to find the most likely configuration.

Low-level motion has been exploited in action recognition using techniques such as motion templates [2], spatio-temporal features [5, 17], XYT cubes [16] and probabilistic models learnt from sparse motion features [20]. These approaches use low-level motion to classify different actions using a discriminative model, this allows an action to be recognised but doesn't extract specific information as to how that action was performed. Such a task requires a generative model. The work that comes closest to ours, in their use of motion, is that by Fathi *et al* [6] where motion exemplars are used to match against new image sequences. Once a match is found, the pose from the matched training exemplar can be fitted to the input image using Gibbs sampling. Our approach, rather than using a number of exemplars, is to learn statistical models to represent the expected motion and structure of the articulated object.

Various techniques have been employed to accurately track articulated objects including particle filters [4], optical flow [12] and action specific dynamic models [1, 22]. Whilst these approaches provide a framework to track an object, they require manual initialisation and are prone to accumulate tracking errors. Our method achieves accurate tracking by detecting the most likely instance of the object in each frame independently. This approach, tracking via detection, has been proven to yield encouraging results [13].

978-1-4244-2243-2/08/$25.00

The principal objective of pose estimation is that extracted poses should be unique to the image sequence being observed: this requires inference using a generative model. As observations become increasingly corrupt, reliance on priors increases and this task becomes more difficult. The observational data used in this work consists of a sparse cloud of features extracted from a sequence of images using the Kanade-Lucas-Tomasi (KLT) feature tracker [18]. Difficulties with these features include non-Gaussian noise and tracking failures, where a feature is completely lost. However, the biggest problem is the sparsity of the data, most limbs will not be tracked for a complete gait cycle or even tracked at all meaning approaches such as [21] would fail. Our algorithm will need to overcome missing and corrupt data in every frame. To achieve this we use a three stage approach.

The first stage is to use low-level motion models to calculate the likelihood that an observed feature is tracking a specific limb in a particular phase of gait. Once these likelihoods have been calculated for all points in all frames, a HMM can be used to estimate the most probable gait phase at each frame. By knowing the phase at each frame we can constrain our search space by having phase dependent prior models. The suitability of HMMs for this task have previously been demonstrated [13, 10].

The second stage is to use *a priori* spatial models to search for the most likely pose in each frame. This is performed using the gait phase estimate and likelihoods calculated in the previous stage. Our spatial model is based on the pictorial structures representation [9] and efficient searches can be performed over these models using Dynamic Programming [8].

Given the previous pose estimates for each frame the third stage is to perform a temporal search over all frames. This is achieved using high-level motion models that describe how the articulated object should deform over time. This search is also performed via Dynamic Programming. Methods such as Loopy Belief Propagation (LBP) present a framework such that the second and third stage could be integrated as has been previously demonstrated [14]. We opt not to do this since we find that the temporal search can be performed in a smaller search space than the spatial search. This makes two separate searches more computationally efficient than one search using LBP.

In summary, the first stage detects parts and provides an estimation of gait phase. The second stage uses this information to estimate the most likely pose for each frame. The third stage links together individual detections so that they are temporally coherent. Each layer of the algorithm operates on the results of the previous layer.

To the best of our knowledge, this work presents the first method that estimates pose of articulated objects using only motion cues in a bottom-up manner, this work is biologically inspired by the psychophysical experiments referenced above. However, the interest in using motion for pose estimation is not purely academic. Our approach has a significant advantage over others as no training is ever performed on descriptors derived from an image or sequence of images. The result of this is that models could be learnt from MoCap data and applied directly to an image sequence without using any additional training data. No current approach can achieve this and this highlights the real potential of our approach and why using only motion deserves further exploration in the computer vision community.

2. Part Detection and Gait Phase Estimation

Our observational data used for part detection consists of tracked features extracted from a sequence of images using the KLT feature tracker.

Given the motion of a feature, we wish to calculate the likelihood that it is tracking a specific joint in a particular phase. As we also know the positions of the features we can use this information to build sparse probability maps, this will give us spatial information about where a particular joint may be located in the image. In addition to this, we can use the likelihoods to estimate the gait phase for each frame, this information can then be used to guide our spatial and temporal search as these models are both designed to be dependent on phase.

We represent low-level motion as the motion a feature would make if it was tracking a particular joint. This is defined as a vector measuring the relative change in position over consecutive frames.

Consider we have a different motion model for each joint $\{\Theta^1, .., \Theta^n\}$, where n is the number of joints. Each model is defined by $\Theta = \{r, \Sigma\}$ where $r = \{r_1, .., r_m\}$ are vectors that represent the expected motion between consecutive frames, $\Sigma = \{\Sigma_1, .., \Sigma_m\}$ are the corresponding covariance matrices that model the variance in that motion and m is the number of phases. The model is then represented by a chain of vectors as shown in Figure 1.

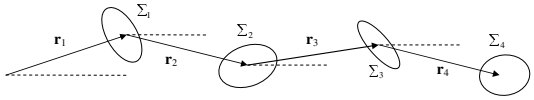

Figure 1. Chain used to represent a motion trajectory. r is the average vector; Σ is the covariance matrix.

Given that we now observe the motion of a feature, v_t, we calculate the probability of that feature tracking the ith joint in the jth phase as $p(v_t|\Theta_j^i) = \mathcal{N}(v_t, r_j^i, \Sigma_j^i)$. Given that we have tracked a feature over a number of frames the negative log likelihood is calculated as

$$L(v_t|\Theta_j^i) = \frac{1}{\lambda}\left(l(v_t|\Theta_j^i) + L(v_{t-1}|\Theta_{j-1}^i)(\lambda - 1)\right) \quad (1)$$

where $l(v_t|\Theta^i_j) = -\log\left(p(v_t|\Theta^i_j)\right)$ and $L(v_{t-1}|\Theta^i_{j-1})$ is the likelihood of being in the previous phase in the previous frame. λ is a constant that effectively determines how large a temporal window to integrate over.

We calculate this likelihood for each feature at every joint and every phase in the model. A background model is learnt by applying RANSAC to the motion of the features. Each feature is then compared to this model, if a feature is classed as being part of the background it is eliminated from further use.

To estimate gait phase over the entire sequence of images, we classify a feature to a specific limb and phase by minimising eq. (1) over i and j and allow each feature to vote for the phase it has been classed as. This vote space is representative of the states of a HMM where each phase represents a different state.

A HMM is defined by three probability measures, A, B and π, where A represents the state transitional probability matrix, B is the observational probability distribution, which is dependent on the number of votes a particular state received, and π is the initial state probability distribution. We allow only three types of state transition, to remain in the current state, move to the next consecutive state, or skip a state. These probabilities are set as $\{0.1, 0.8, 0.1\}$ respectively. We use a flat prior π since all phases are equally likely. The optimal sequence of phases can then be calculated using the Viterbi algorithm.

3. Estimating Pose

Our objective in this section is to estimate pose in each frame independently given the likelihoods and optimal sequence of gait phases calculated in Section 2. As we know the position of each tracked feature we know the likelihood of a specific joint being at that position in the image, this allows us to construct a probability map for each joint. However, the tracked features are very sparse meaning that the likelihoods at most of the pixels in the image are missing. To avoid limiting our search over image locations where features are present we need to infer likelihoods between features to construct a dense probability map.

To achieve this consider a set of locations on a grid $l \in \mathcal{G}$, where the grid represents the image pixels. The observed features lie on a subset of the grid $\mathcal{B} \subset \mathcal{G}$, at these locations is the calculated likelihood of that feature tracking the joint in which we are interested $m(l)$. As you move to a location (l_i) further away from an observed feature ($l_j \in \mathcal{B}$) the probability should decrease to reflect the increased uncertainty in that observation at your current position. This is represented by a zero mean gaussian $p(l_i, l_j) = \mathcal{N}(l_i - l_j, 0, \sigma)$. The inferred likelihood at each location l_i of the grid is calculated as

$$\mathcal{P}(l_i) = \min_{l_j \in \mathcal{B}} \left(m(l_j) - \log(p(l_i, l_j))\right) \tag{2}$$

Since $p(l_i, l_j)$ is defined as a zero mean Gaussian, eq. (2) can be efficiently calculated as a distance transform using the techniques described in [7]. An example of a calculated probability map with the features overlaid is shown in Figure 2. Regions with a higher likelihood are represented by darker colours, this is as the probability maps actually represent the negative log of the probability. There is a local minima around every feature point, this makes it preferable for a limb to be located in the neighbourhood of a pixel containing a feature.

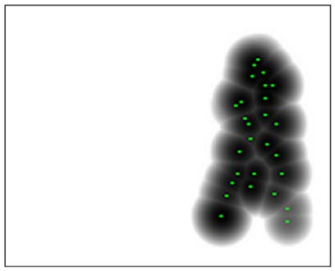

Figure 2. Example of a likelihood map for the hip location with KLT features overlaid. Darker regions represent areas with a higher likelihood.

We construct a probability map for each limb $\mathcal{P} = \{\mathcal{P}_1, .., \mathcal{P}_n\}$ over which we can perform our spatial search. To define our articulated object we follow the notation from [8] and we refer the reader to this work for a full description. Consider the graph $G = (V, E)$, where $V = \{v_1, .., v_n\}$ defines the set of n vertices (or nodes), and $(v_i, v_j) \in E$ define the set of edges connecting the vertices. The vertices represent the joints of the articulated object and the edges represent the dependence between connected joints. A particular configuration of this graph can be described by $L = \{l_1, .., l_n\}$, where l_i specifies the image location of v_i. There is an associated cost of placing v_i at l_i which is defined as $m_i(l_i)$. The set of edges represent the dependence between connected vertices, where $d_{ij}(l_i, l_j)$ is the deformation cost of placing v_i at l_i and v_j at l_j. The quality of a specific configuration is calculated as

$$Q = \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \tag{3}$$

Finding the best configuration L^* is found by minimising eq. (3). Provided that the graph $G = (V, E)$ has no loops this can be solved using Dynamic Programming (DP). This is achieved by starting at the leaf nodes of the graph and iteratively working towards the root, calculating at each node

$$B_j(l_i) = \min_{l_j}(m_j(l_j) + d_{ij}(l_i, l_j) + \sum_{v_c \in C_j} B_c(l_j)) \quad (4)$$

where $v_c \in C_j$ are the children of v_j. The cost function for the root node is then calculated as:

$$B_r(l_r) = m_r(l_r) + \sum_{v_c \in C_r} B_c(l_r) \quad (5)$$

Finding the best root position l_r^* is found by minimising eq. (5) over l_r. Given a set of observations $\mathcal{O} = \{\mathcal{O}_1, .., \mathcal{O}_n\}$ it can be shown that minimising eq. (3) is equivalent to maximising the posterior distribution $p(L|\mathcal{O})$ given by Bayes rule

$$p(L|\mathcal{O}) \propto p(\mathcal{O}|L)p(L) \quad (6)$$

where

$$m_j(l_j) = -\log\left(p(\mathcal{O}_j|l_j)\right) \quad (7)$$

$$d_{ij}(l_i, l_j) = -\log\left(p(l_i, l_j|c)\right) \quad (8)$$

and c represents a connection parameter between l_i and l_j.

The spatial model is represented as a set of joints, where the position of a joint with respect to it's parent is defined by an angle measured relative to the horizontal $\phi(l_i, l_j)$ and a fixed distance L_{ij}. In current approaches the relative angle between two limbs (three joints) is typically used, this allows the conditional dependance between them to be modeled and stops unlikely poses being inferred. However, we learn a different spatial model for each gait phase, this means our model is well enough constrained so that unlikely poses do not occur. This assumption allows us to reduce our search space as we are not concerned with the orientation of a parent joint, only it's position.

The prior for each joint's configuration $p(l_i, l_j|c)$ is defined by a Von-Mises distribution:

$$\mathcal{M}(\phi(l_i, l_j), \mu, \kappa) \propto e^{\kappa \cos(\phi(l_i, l_j) - \mu)} \quad (9)$$

where μ represents the mean angle of the distribution and κ defines how constrained the joint is. Learning a different prior for each phase consists of estimating different values for the parameters μ and κ for each limb.

Given the set of probability maps $\mathcal{P} = \{\mathcal{P}_1, .., \mathcal{P}_n\}$ the best configuration L^* for the model is calculated through eq. (4) where $m_j(l_j) = \mathcal{P}_j(l_j)$ and

$$d_{ij}(l_i, l_j) = \begin{cases} -\log p(l_i, l_j|c) & \text{if } \; l_j \in l_i + T_{ij}(\theta) \\ \infty & \text{otherwise} \end{cases} \quad (10)$$

$\theta = \{\theta_1, .., \theta_k\}$ represents a set of k angles and $T_{ij}(\theta)$ is a function that calculates possible positions of l_j given the angle set θ. This function is defined since limb lengths are fixed in our model. Given a location l_i there are only a small number of possible locations for l_j. In practice we minimise eq. (4) over l_j through the parameter θ. In general k is much smaller than the number of locations in the image, the result is that the complexity of finding the minimum of eq. (3) increases linearly with the number of grid locations.

Calculating L^* corresponds to calculating the Maximum A Posterior (MAP) estimate of the probability distribution $p(L|\mathcal{O})$. The MAP estimate is just one example of how to evaluate a posterior distribution, a more robust measure is the expectation value $\langle L \rangle$. To calculate the expectation value we needed to have calculated the full posterior distribution $p(L|\mathcal{O})$. Through DP we have maximised the posterior but this was achieved without having to actually calculate it. However, we can use the approximation that $B_r(l_r) \approx -\log(p(l_r|\mathcal{O}))$ to calculate the expectation value for the root location $\langle l_r \rangle$. The location of the other nodes in the graph can then be extracted using the MAP estimate conditioned on the root position $\langle l_r \rangle$.

One of the difficulties with calculating an expectation value is that if the posterior distribution is sharply peaked then $\langle l_r \rangle = l_r^*$. A solution to this is to smooth the posterior using techniques from simulated annealing [4] where

$$p'(l_r|\mathcal{O}) = p(l_r|\mathcal{O})^{\frac{1}{\gamma}} \quad (11)$$

The value of γ affects how smooth the resultant posterior will be. Whilst γ would normally be set to a constant we define it by

$$\gamma = \frac{\log(p(l_r|\mathcal{O})_{max}) - \log(p(l_r|\mathcal{O})_{min})}{\rho} \quad (12)$$

where ρ is a constant that specifies the order of magnitude between the lowest probability and the highest. This makes our approach more robust since the degree of smoothing γ is calculated for each frame depending on the quality of the current observational data. The resultant probability distribution for the hip location is shown in Figure 3 (a). The distribution is very broad, this is expected as we have a large uncertainty in the exact position of the root node because our observational data was very sparse. The two horizontal lines in Figure 3 (a) are because if the root node was located on either of these lines the outermost joints of the object could not be placed in the image, this has a zero probability.

Using the techniques described, pose is estimated for each frame in the sequence independently. We learn a different *a priori* model for each phase of the gait cycle so that the search is performed using the prior model for the current gait phase estimated in Section 2.

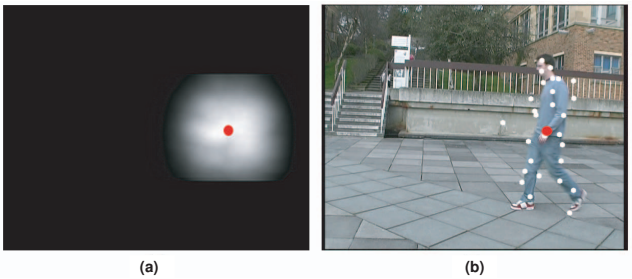

Figure 3. The expectation value is shown as a red dot. (a) Resultant probability distribution for the root node (hip), lighter regions have a higher probability. (b) Corresponding image with KLT features overlaid.

4. Temporal Search

The high-level motion model is represented by the change in angle between adjacent joints' position. As this is measured relative to just the parent node's position we can perform temporal searches separately for each joint. The purpose of the temporal search is to refine pose estimates from the previous section by making limb movements temporally coherent over the sequence of frames. We are not interested in the position of the root node as this was robustly estimated in the previous section. A simple low-pass filter is adequate to make the motion of the root node temporally coherent.

Since the root node's position is already predetermined the temporal search is carried out over the possible angles a joint may be relative to it's parent's position. This search space is much smaller than that used in Section 3 and can be performed more efficiently in comparison.

There is a high-level motion model for each joint except the root joint. This describes how a joint will move relative to it's parent as a function of phase, each model is defined by a set of angles that represent the expected motion between frames $\phi = (\phi_1, .., \phi_m)$, where m is the number of phases in the model.

The temporal search is also performed via Dynamic Programming and as such can be defined using the notation introduced in Section 3. The graph used for the temporal search consists of n vertices, where each vertex represents a frame of the sequence. The possible locations for a vertex now correspond to different angles. The observational data used for a joint $\mathcal{O} = \{\mathcal{O}_1, .., \mathcal{O}_n\}$ is the angle of that joint estimated in the previous section for each frame. We also make use of the estimated gait phase from Section 2 defined as $S = \{S_1, ..S_n\}$. The temporal search is performed over the entire sequence using eq. (4), once we have defined

$$p(l_i, l_j|c) = \mathcal{M}(l_i, l_j + \phi_{s_j}, \kappa_{s_j}) \tag{13}$$

and

$$p(\mathcal{O}_j|l_j) = \mathcal{M}(l_j, \mathcal{O}_j, \alpha\kappa_{s_j}) \tag{14}$$

The deformation term makes it most probable to move through the angle ϕ_{s_j} across consecutive frames. The observational likelihood is defined so that the probability of a particular location l_j is lower the further it is away from the observed angle $\mathcal{O}_j$. α is a constant that defines the weighting between observations and model. For a low value of α the model will dominate and a high value the observations will dominate. In our experiments we set $\alpha = 0.5$. The results using different values of α are shown in Figure 4. When $\alpha = 0.5$ the motion model acts as a template which is deformed to fit the observations, notice in particular that the amplitude of the observed gait is maintained.

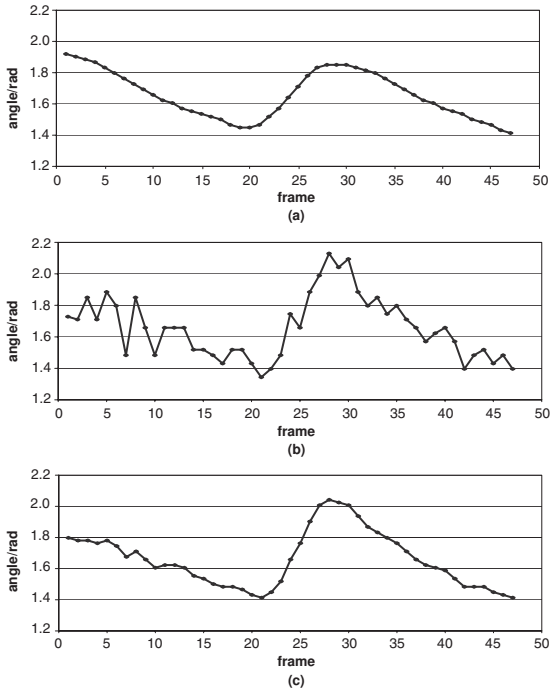

Figure 4. Results of temporal search for knee joint using different values of α. (a) $\alpha = 0.001$, motion model dominates. (b) $\alpha = 1000$, observations dominate. (c) $\alpha = 0.5$, the model is deformed to the observations.

The temporal search is performed for each limb over every frame. Whilst assuming an independent search can be performed for each limb may seem a somewhat crude approximation, our model is well enough constrained such that unlikely poses will not occur.

5. Experimental Results

We tested our method on two classes of articulated object; people and quadrupeds. Ground truth data for both classes consisted of the hand labeled positions of the main

joints over a sequence of images, we also label the start and end frame of each gait cycle. The structure of how the joints are assembled was manually defined.

The ground truth was preprocessed in the following way before models could be learnt. Firstly the sequence of ground truth data was cut into individual gait cycles, then a cubic spline was fitted to each gait cycle and they were re-sampled so that all individual examples had the same temporal length (in frames). Once the ground truth was in this form all model parameters can be learnt directly from this data.

The grid that we performed the spatial search over was reduced to 180 by 144 locations. The angular range searched over for each joint was set as $\pi/6$ radians. This search was represented as 10 discrete angles centered on the average angle for that joint in the given phase.

For the temporal search we used a space spanning 2π radians represented as 180 discrete values. The image sequences were reduced to a size of 360 by 288 pixels before the KLT feature tracker was applied. For all experiments λ was set as 3.0 and ρ was set to 40, both these values were determined empirically to yield satisfactory results.

The model representing a human was learnt from a person walking on a treadmill side on to the camera for about 13 complete gait cycles. The hip was defined as the root node. The learnt model consisted of 32 phases, a different spatial prior was learnt for each phase.

As the motion models were learnt from a person walking on a treadmill, when applying the models to people in real world scenes we have to compensate for their translational motion. We achieved this by using a bounding box propagated using a particle filter to track the dominant object, represented by the largest cluster of foreground features. From this we could calculate the foreground objects motion over consecutive frames.

We tested our method on a number of different scenes. The rough height of the person being tracked was manually set before pose estimation commenced. Some sample frames of a scene and the calculated poses that we tested our approach on are shown in Figure 5 (a) - (d). The calculated poses are very similar to that of the person being tracked.

In Figure 5 (e) - (h) the features used are shown with the corresponding estimated poses. The features are very sparse making estimating pose from just these data points alone impossible unless strong priors are used.

Another problem with learning gait from a person walking on a treadmill is that people walk differently on a treadmill to how they would normally. Particularly noticeable is their stride length, people tend to have a much smaller stride when walking on a treadmill. The effect of this can be seen in Figure 5 (d) where the observed person is at full stride, in the absence of well tracked features the spatial prior is relied on which does not reflect the actual pose of the person being observed. In Figure 5 (a) the observed person is again at full stride, here as a feature is accurately tracking the foot the model is allowed to deform to fit this observation.

We also show results for another sequence, obtained from Michael Black, in Figure 6. Again the results show that pose is estimated accurately apart from in Figure 6 (b) where both legs are trying to deform to the same feature. Whilst the prior tries to prevent this the resultant pose is not very probable.

In both Figure 5 and Figure 6 the arm pose is poorly estimated. This is largely because the person used as ground truth did not move their arms whilst walking. The consequences of this is that there will be a high deformation cost associated with extending the arm beyond the body and also that the motion model for the hand would expect small motions, if a large motion is being made this motion will be deemed unlikely to be the motion of a hand.

To quantify the accuracy of the presented method we tested our approach on 8 sequences of a person walking on a treadmill. For each sequence ground truth was hand labeled and our approach was tested over 50 frames. Table 1 presents the average error and operating speed for two different grid sizes. Notice that the operating speed is linearly dependent on the number of grid locations. These results also demonstrate the trade-off between speed and accuracy, using a smaller grid results in larger errors due to the reduction in resolution.

In Table 1 we also compare our results to those using motion exemplars from [6]. Whilst direct comparison is difficult as the two sets of results were obtained using different data sets, both consisted of a treadmill viewed from the side-on and the walkers are a similar height in pixels. The gait length of the walkers from our data set ranged between 29 and 34 frames and the average height was 400 pixels. The results show that our technique presents an overall improvement, particulary when using a high resolution grid. A further advantage of our approach is that no learning was ever performed using descriptors derived from image sequences, meaning that a model can be applied to an object with a different shape and appearance, provided the structure and motion of the object is similar.

To further demonstrate this we learnt a model of a quadruped from 6 complete gait cycles of a cheetah walking side on. The model was then applied to a lion, without any further learning or tuning of parameters, as shown in Figure 7. The lion's appearance and shape is significantly different to that of a cheetah. This sequence is also challenging since there is also a lot of clutter present, such as moving grass, and the colour of the lion is also similar to that of the background. However, our method is able to overcome all of these problems.

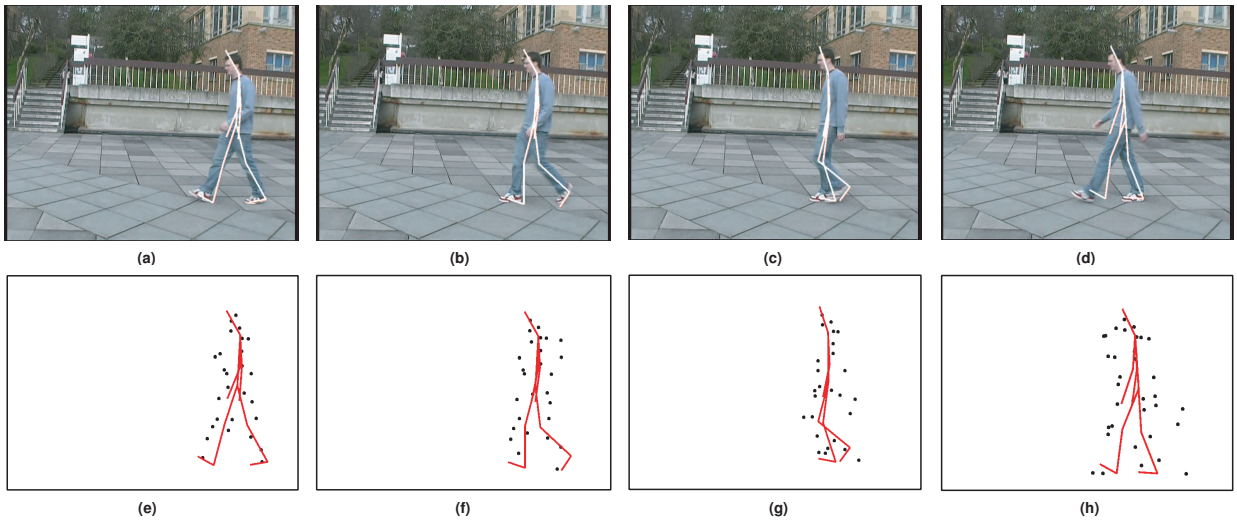

Figure 5. Resultant estimated pose. (a) - (d) sample frames from sequence with pose plotted. (e) - (h) shows the corresponding observational data from which pose was estimated.

Figure 6. Sample frames from sequence with estimated pose shown. The features used are also plotted. Sequence courtesy of Michael Black.

6. Conclusions

We have presented a method that uses only motion to estimate pose using a bottom-up approach. We have demonstrated our approach on different classes of articulated object in challenging scenes. In the presence of sparse and noisy data we have still been able to extract new poses using our generative model learnt using a small amount of ground truth data. The presented method currently operates at real-time, this has been achieved as part detection is computationally cheap and we have assumed there is no probabilistic dependency between the orientation of adjacent joints. Our model is currently constrained to the viewpoint from which models were learnt, future work will focus on integrating models learnt from different viewpoints and developing more general representations of motion. This work has demonstrated the potential of using only motion for pose estimation and the end goal is an approach where 3D models can be learnt from MoCap data and applied directly to image sequences with no additional training data.

Acknowledgements

This work was supported under EPSRC grant EP/D506549/1.

References

[1] A. Agarwal and B. Triggs. Tracking articualted motion with piecewise learned dynamical models. In *ECCV*, pages 54–65, 2004. 1

[2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001. 1

[3] J. Cutting and L. Kozlowski. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin Psychonometric Society*, 9:353–356, 1977. 1

[4] J. Deutscher, A. Blake, and I. D. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, pages 126–133, 2000. 1, 4

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE*

Grid Size	Operating Speed (FPS)	Head	Shoulder	Elbow	Hand	Hip	Knee	Ankle	Toe	Mean
180×144	5	10.1	12.9	14.7	27.9	11.9	11.2	15.3	16.2	15.0
90×72	20	12.9	13.5	16.9	31.5	14.6	15.3	20.2	17.9	17.8
N/A	N/A	N/A	17.5	23.1	30.0	15.1	13.2	15.0	N/A	19.0

Table 1. The error for each joint measured as the average difference between the ground truth and extracted joint position (measured in pixels). Results for the presented method tested on 8 sequences of different people walking on a treadmill using different grid sizes (top two rows). Results using motion exemplars [6] tested on the 'Fast walk side-view' sequences from the Mobo database (bottom row). The operating speed was obtained by running the proposed method on a 2.6 GHz processor and excludes KLT feature extraction.

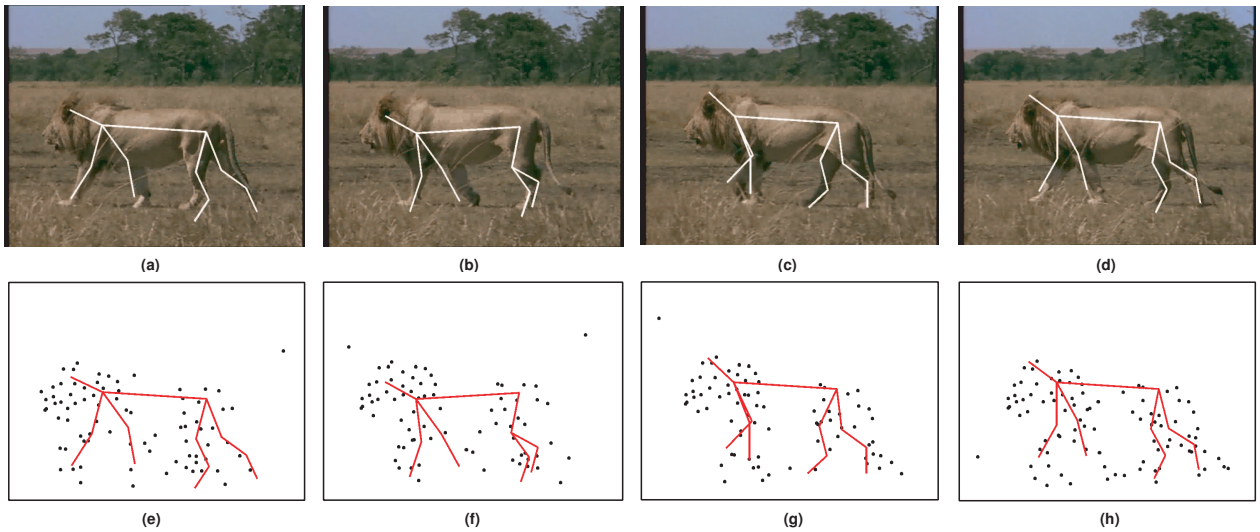

Figure 7. Sample frames with estimated pose plotted. (a) - (d) show the original images. (e) - (h) shows the set of sparse features used.

international workshop on: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pages 65–72, 2005. 1

[6] A. Fathi and G. Mori. Human pose estimation using motion exemplars. In *ICCV*, pages 1–8, 2007. 1, 6, 8

[7] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. In *Cornell Computing and Information Science Technical Report TR2004-1963*, 2004. 3

[8] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, pages 55–79, 2005. 1, 2, 3

[9] M. Fischler and R. Elsclager. The representation and matching of pictorial structures. In *IEEE Transactions on Computer, 22(1)*, pages 67–92, 1973. 2

[10] A. Fossati, M. Dimitrijevic, V. Lepetit, and P. Fua. Bridging the gap between detection and tracking for 3d monocular video-based motion capture. In *CVPR*, pages 1–8, 2007. 2

[11] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973. 1

[12] S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996. 1

[13] X. Lan and D. P. Huttenlocher. A unified spatio-temporal articulated model for tracking. In *CVPR*, pages 722–729, 2004. 1, 2

[14] M. W. Lee and R. Nevatia. Human pose tracking using multi-level structured models. In *ECCV*, pages 368–381, 2006. 2

[15] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *CVPR*, pages 467–474, 2003. 1

[16] J. Rittscher, A. Blake, A. Hoogs, and G. Stein. Mathematical modelling of animate and intentional motion. In *Philos Trans R Soc Lond B Biol Sci. 2003 March 29; 358(1431): 475490.* 1

[17] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, pages 32–36, 2004. 1

[18] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593–600, 1994. 2

[19] L. Sigal, B. Sidharth, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. In *CVPR*, pages 421–428, 2004. 1

[20] Y. Song, L. Goncalves, and P. Perona. Learning probabilistic structure for human motion detection. In *CVPR*, pages 771–777, 2001. 1

[21] L. Torresani and C. Bregler. Space-time tracking. In *ECCV*, pages 801–812, 2002. 2

[22] R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *ICCV*, pages 403–410, 2005. 1