USING LOW-LEVEL MOTION TO ESTIMATE GAIT PHASE

Ben Daubney, David Gibson and Neill Campbell
Department of Computer Science, University of Bristol, Bristol, BS8 1UB, UK

Keywords: Dynamic Programming, Gait Phase Estimation, Human Tracking.

Abstract: This paper presents a method that is capable of robustly estimating gait phase of a human walking using the motion of a sparse cloud of feature points extracted using a standard feature tracker. We first learn statistical motion models of the trajectories we would expect to observe for each of the main limbs. By comparing the motion of the tracked features to our models and integrating over all features we create a state probability matrix that represents the likelihood of being at a particular phase as a function of time. By using dynamic programming and allowing only likely phase transitions to occur between consecutive frames, an optimal solution can be found that estimates the gait phase for each frame. This work demonstrates that despite the sparsity and noise contained in the tracking data, the information encapsulated in the motion of these points is sufficient to extract gait phase to a high level of accuracy. Presented results demonstrate our system is robust to changes in height of the walker, gait frequency and individual gait characteristics.

1 INTRODUCTION

There is currently much interest in developing systems that are capable of extracting human body pose from video sequences. Applications for such a system include motion capture, medical analysis and surveillance. The emergence of gait as a biometric has particulary increased interest in being able to recognise and identify gaited motions. The difficulty in this problem is that the human body has many degrees of freedom, each limb alone is capable of a large range of movements. The result of this is a very computationally large search space that is inherently multi modal. Accurately estimating gait phase could be used to constrain this problem given that at each phase certain poses are more likely than others.

Particle filters have been used to successfully search this high dimensional space (Sidenbladh et al., 2000; Deutscher et al., 2000). This approach does not guarantee to find the global minimum and the number of particles needed grows exponentially with the number of dimensions being searched. Attempts to overcome these problems have included the use of annealing (Deutscher et al., 2000) and the use of prior models that can be used to guide particles to explore only likely human poses (Caillette et al., 2005). Techniques such as PCA have also been used to reduce the dimensionality of the problem, thus significantly reducing the size of the search space (Hu and Buxton, 2005; Urtasun et al., 2005; Argawal and Triggs, 2004).

An image sequence contains a temporal dimension that can be exploited with the use of a motion model. Motion models currently used in the literature vary in how much prior knowledge they encapsulate and subsequently how much they constrain the search space. Motion models used to track the upper body include the use of HMMs to ensure smooth trajectory through a feature space (Navaratnam et al., 2005) and simple dynamical models to limit the movement of a part between consecutive frames (Micilotta et al., 2006). These approaches do not constrain the types of motion that can be observed, they are not action specific. Whilst their generality make them attractive they do not offer enough constraint to yield good results whilst tracking the entire human body due to self occlusions and view point ambiguities.

When extracting pose for the entire body it is often necessary to learn a different motion model for each action being performed. Often these motion models can be represented as a curve through a feature space such as those discussed above. Different positions on this motion curve can be seen to represent different phases of a gait cycle and define the expected pose for each phase. Navigation through this feature space has been achieved using Extended Kalman Filters (Hu and Buxton, 2005), Particle Filters (Sidenbladh et al., 2000) and HMMs (Caillette et al., 2005;

Daubney B., Gibson D. and Campbell N. (2008).
USING LOW-LEVEL MOTION TO ESTIMATE GAIT PHASE.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 496-501
DOI: 10.5220/0001073604960501

Lan and Huttenlocher, 2004). These all represent examples of high-level motion models, since the motion represents a change in configuration rather than the movement of individual parts.

The motion of tracked feature points has been used in the analysis and recognition of quadruped gait (Gibson et al., 2003). In this approach tracking errors are overcome by first splitting the foreground object into quadrants and then analysing the average motion of each quadrant. This approach relies on being able to accurately locate the centre of the foreground object making it sensitive to outliers.

Gait has successfully been detected using approaches such as spatio-temporal features (Schuldt et al., 2004), symmetry cues (Havasi et al., 2007) and probabilistic models learnt from sparse motion features (Song et al., 2001). However, these approaches do not estimate the particular phase of gait only that it is present.

In this work we present a system that exploits the low-level motion of a sparse set of feature points extracted using the Kanade-Lucas-Tomasi (KLT) feature tracker (Shi and Tomasi, 1994). The feature points track both the foreground and background of the image meaning segmentation must be carried out. The feature points also contain tracking errors that are not gaussian in nature but systematic due to for example edge effects, this is particularly apparent during self occlusion e.g. as one leg passes another.

We initially learn motion models that represents the expected trajectories for each of the main limbs. Given a set of feature points we use our models to simultaneously solve two problems: the first problem is that of labelling the feature points as belonging to the background or foreground, if a feature is classified as a foreground point it is also assigned to the limb that the feature's motion best represents. The second problem is to estimate the phase that the limb must be in to have produced the observed motion.

Once all feature points have been classified we then integrate over all the points and estimate the most likely gait phase for each frame, ensuring that only smooth transitions are allowed between frames. This is achieved without making assumptions about the location of any of the features; each trajectory is classified depending only on its motion, not its position.

2 LEARNING

Our objective is to learn a statistical model for each of the main limbs that represents how we would expect a point located at that limb to move through time. To create a motion model we use a representation similar to (Coughlan et al., 2000) except we make our representation dependant on orientation; we assume that people walk upright. A different motion model is learnt for each limb and is represented as a chain of m vectors, where each vector represents the mean displacement you would expect to observe between frames. Each vector also defines the centre of a Gaussian that represents the variation in motion we expect. This model can be defined by the parameters $\Theta = (R, \Sigma)$, where $R = \{R_1, .., R_m\}$ and $\Sigma = \{\Sigma_1, .., \Sigma_m\}$. R_j is the average motion belonging to the jth point in the chain and Σ_j is the corresponding covariance matrix. This representation is illustrated in Figure 1.

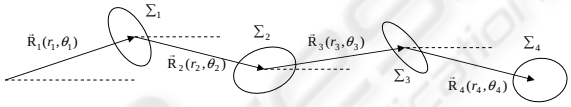

Figure 1: Chain used to represent a motion trajectory. r is the magnitude of the vector; θ is the angle relative to the horizontal; Σ is the covariance matrix.

To learn a model for each limb consider we have a set of example gait cycles $\{g^1, .., g^n\}$ where each gait cycle consists of m temporally ordered vectors $\{v_1, .., v_m\}$. We want to learn a model Θ^{max} that maximises

$$P(g^1, .., g^n|\Theta) = \prod_{i=1}^{n}\prod_{j=1}^{m} p(g_j^i|\Theta_j) \qquad (1)$$

This is a maximisation over all the training examples for every position in the model. We see that equation (1) can be maximised by solving for each Θ_j independently,

$$\Theta_j^{max} = arg\max_{\Theta_j} \prod_{i=1}^{n} p(g_j^i|\Theta_j) \qquad (2)$$

This is the Maximum Likelihood estimate for Θ_j and can be calculated directly from the training examples. Each position j in the model can be seen as representing a different gait phase.

However, our ground truth data consists of coarsely hand labeled x and y positions of the main limbs through the duration of a video clip. To use the method described above we need examples of individual gait cycles and we need all gait cycles to have the same temporal length.

The data can be cut into individual gait cycles by using a reliable heuristic, for example the turning point in the data that corresponds to when the toes are at their maximum height. To make each gait cycle the same temporal length the average length is first calculated. A Cubic spline is then fitted to each individual

gait cycle and each gait cycle is resampled to be the same length as the mean.
These statistical models allow us to calculate a likelihood that an observation was produced by a particular limb in a particular phase. We can then classify the feature point to the limb and phase with the highest likelihood.

3 ESTIMATING GAIT PHASE

Given that we have learnt a motion model for each of the main limbs $\{\Theta^1,..,\Theta^l\}$ we now want to compare the models against observed trajectories to estimate the gait phase. We do this in two steps: first we compare each motion trajectory to all of the models and classify each feature point as being associated with a particular limb and gait phase. We then integrate over all the feature points at every frame enforcing smooth phase transitions to find the actual gait phase.
The probability of an observed vector v being a member of a particular limb k and gait phase j is given by

$$p(v|\Theta_j^k) \propto \frac{1}{|\Sigma_j^k|} e^{-\left(\frac{1}{2}(v-R_j^k)^T(\Sigma_j^k)^{-1}(v-R_j^k)\right)} \quad (3)$$

Where equation (3) is a Gaussian with mean R_j^k and covariance Σ_j^k. Taking the natural logarithm of this we get the log-likelihood

$$l(v|\Theta_j^k) = -\log|\Sigma_j^k| - \frac{1}{2}(v-R_j^k)^T(\Sigma_j^k)^{-1}(v-R_j^k) \quad (4)$$

However, we want to compare trajectories that are more than just one vector in length, given that we observe the ith vector of a trajectory we classify the trajectory by solving

$$L(v_i|\Theta_j^k) = \max_{k,j} \frac{1}{\lambda}\left(l(v_i|\Theta_j^k) + L(v_{i-1}|\Theta_{j-1}^k)(\lambda-1)\right) \quad (5)$$

The term $L(v_{i-1}|\Theta_{j-1}^k)$ is the likelihood of being in the previous phase in the previous frame and acts as a prior for the current frame. The constant λ acts as a decay constant, this means that $L(v_i|\Theta_j^k)$ is calculated as a weighted mean, where recent observations are given a higher weight than old observations. The value of λ effectively determines how large a temporal window to integrate over.
We assume in equation (5) that the next consecutive phase of the model must be moved into at each new frame. If the frequency of the model and the observed gait are the same this is valid. However, if they are not this assumption is invalid the two will eventually become out of phase.
Our approach then is to find a value of λ such that we are integrating over a small enough temporal window that errors introduced due to frequency differences are small; yet large enough that we allow enough past observations to be used for accurate classifications.
Each feature point has now been assigned to it's most likely state (k, j), where k is the limb and j is the gait phase. To calculate the global gait phase, each feature votes for the phase it has been classified as. Since opposing limbs will be half a cycle out of phase the resultant votes will have a bimodal distribution. To overcome this we reduce the number of phases by a half, shifting any votes for phases that have been eliminated by half a gait cycle. This is normalised for each frame and can now be seen as representing a state probability matrix containing the likelihood of being in a specific phase at a given frame.
To enforce smooth transitions between states we learn state transition probabilities from the ground truth data. Rather than making these dependent on the specific state of the system we make them more general, we learn the probabilities of remaining in the same state, moving to the next state and skipping a state.
One of the difficulties with using gait phase is that it's not possible to hand label ground truth data explicitly for each frame, this is as we have no method to accurately recognise one gait phase over another. This makes it difficult to calculate state transitional probabilities. However, provided we have well defined start and end points of a gait cycle these can be hand labeled and then used as ground truth data. From this we can calculate the number of frames needed to complete a gait cycle, by comparing this to the number of states in our model we can estimate the transitions that were necessary to be able to traverse all the states in a given number of frames. This doesn't permit us to learn a full set of state transitional probabilities but it does allow us to create an approximation. This approximation is based on the assumption that the transitional probabilities are not dependent on the particular state, but are dependent on the relative state transition. We learn the probability of remaining in the same state, moving to the next consecutive state and skipping a state. We assume all other transitions have a zero probability.
The Likelihood of being in state S_m in the current frame given that you were previously in the state S_n in the previous frame is defined as

$$P(S_m) = p(S_m)p(S_m|S_n) + P(S_n) \quad (6)$$

Where $p(S_m)$ is the probability of currently being in the mth state, this is contained in the state probability matrix. $p(S_m|S_n)$ is the probability of being in state m given that you were previously in state n. $P(S_n)$ is

the likelihood of being in the nth state in the previous frame. The optimal route is found by maximising equation (6) over n for each m in every frame, this problem can be efficiently solved using dynamic programming.

4 EXPERIMENTS

We initially tested our algorithm on footage of people on a treadmill that was recorded using a standard camcorder at PAL 25fps. Models were learnt using about 450 frames of hand labeled data of a person walking on a treadmill, this equates to about 14 complete gait cycles. The limbs that were labeled were the head, shoulder, elbow, wrist, hip, knee, ankle and toes. The model consisted of 32 phases. The state transitional probabilities were calculated from 12 clips of different people walking on a treadmill, the start and end points of each complete gait cycle were manually labeled. The transitional probabilities were calculated as:

$$p(S_m|S_m) = 0.01$$
$$p(S_m|S_{m-1}) = 0.91$$
$$p(S_m|S_{m-2}) = 0.08$$

We see that the probability of moving into the next state is most probable and the probability of remaining in the same state is the most unlikely.
The KLT feature tracker was used to extract 150 feature points per frame. A background model was learnt by performing RANSAC on the motion of the feature points from the first 10 frames of data, the background was assumed to be the dominant model. The average velocity and covariance matrix of the background points were then calculated. Trajectories were compared to the models of the limbs and the background model, if a point was classed as being in the background it was discarded.

The algorithm was tested on 12 video clips of different people walking on a treadmill, an example of the state probability matrix and calculated optimal route is shown in Figure 2. There are only 16 states as the number of phases was reduced by a half as discussed in Section 3. The lighter a box is the higher the probability of that state. Notice there is very close agreement between the extracted data and the ground truth. The graph appears as a sawtooth due to the cyclic nature of gait, once the last state is reached it will then return back to the first state.
The average error in estimating gait phase as a function of λ is shown in Figure 3. All errors that are used to quantify the accuracy of our method are calculated as the average difference between the hand

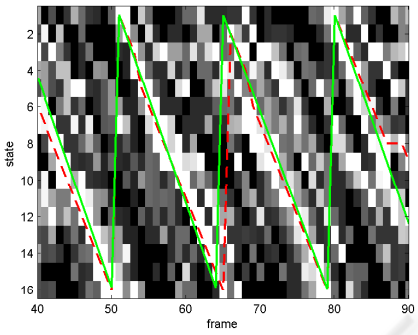

Figure 2: Example of a state probability matrix, green solid line shows ground truth, red dashed lines shows extracted optimal path. The lighter a box the probable that state is.

labeled ground truth and the estimated gait phase (the average difference between the red and green line in Figure 2). Error bars show the calculated standard deviation of the errors for all the clips the algorithm was tested on.

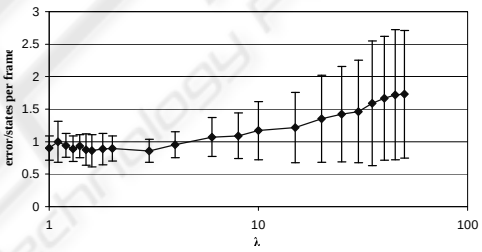

Figure 3: Average error of gait phase estimation as a function of λ.

The lowest average error is when $\lambda = 3.0$, however there is not much difference between this and when $\lambda = 1.0$ implying the trajectories for each frame could be classified independently to observations made in previous frames. Figure 3 shows that the standard deviation of the results gets larger as λ increases, this is because for a larger value of λ we assume a feature can only move into the next consecutive phase for longer, meaning that the phase of the person being observed and that estimated by our model will become offset. Consider the results shown in Figure 4, the errors when $\lambda = 2$ appear independent of gait cycle length, however when $\lambda = 50$ the error is generally greater the bigger the difference between the length of gait cycle of the walker and the model.
To visually demonstrate the accuracy of our method a simple stick man model was created by learning the average pose of each phase from the ground truth data. Some sample frames are shown in Figure 6.
Our results show that when $\lambda = 3.0$ we are able to achieve an average error in estimating the correct state

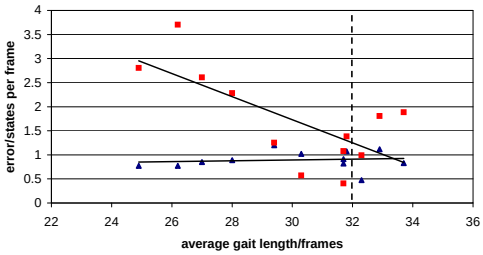

Figure 4: Error of gait phase estimation compared to gait length; blue diamonds $\lambda = 2.0$, red squares $\lambda = 50.0$, the dashed vertical line shows the number of phases in the model.

of 0.9 ± 0.3 states per frame, this corresponds to a temporal error of about 0.05 seconds.

To improve the generality of our algorithm we want it to be able to work on people walking in real world situations, these scenes will typically have more background clutter and as a result the tracking will contain more errors. Unlike when people are walking on a treadmill and have no net motion, people walking in the real world have a translational motion in the direction they are walking. Our approach is to compensate for this translation so that the walker is in a globally stationary frame of reference.

We use a particle filter to propagate multiple hypothesis of the position of the bounding box. We design a likelihood function that will favour positions that contain larger numbers of foreground points over positions that have fewer. After each frame we resample our distribution of particles so that those with low likelihoods will typically be replaced by those with higher likelihoods, we then propagate the particles by allowing them to perform a random walk. Eight frames are allowed to let the particle filter initialise before phase estimation commences. We also use a low pass filter to smooth the motion of the bounding box.

To minimise the effect of foreground outliers we only consider feature points that are located inside the bounding box.

We tested this method on 6 clips of different people walking in an outdoors scene. In three clips the camera remained stationary and in the remaining clips the camera panned to follow the person as they walked. The error as a function of λ is shown in Figure 5. In contrast to Figure 3 the error now initially decreases as λ gets larger, this is as there are now additional errors introduced through the estimation of the foreground object's motion, a larger λ is needed so that a larger temporal window is used to integrate out errors. When $\lambda = 20$ we obtain an average error of 0.9 ± 0.2 states per frame.

By learning a different pose for each gait phase we

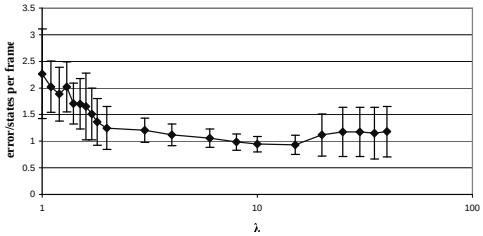

Figure 5: Average error of gait phase estimation as a function of λ.

have been able to crudely estimate pose. However the poses that our system is capable of extracting is limited to the number of phases in our model. These estimates of pose could be used to initialise a further search using methods such as deterministic optimisation (Urtasun et al., 2005) or dynamic programming (Lan and Huttenlocher, 2004).

The height of the person used as ground truth was about 380 pixels, the largest walker in our data set was 420 pixels in height and the smallest was 310 pixels in height. This demonstrates that our approach can cope with large changes in scale without having to adjust the original models to compensate.

5 CONCLUSIONS

We have presented a system that is capable of robustly and accurately estimating gait phase using the motion of a sparse set of feature points. This has been achieved by building low-level motion models from ground truth data obtained from a single person walking for 14 complete gait cycles. Despite this small amount of training data our system has been shown to be robust to changes in frequency, walker height and gait characteristics. This has been achieved despite a large amount of noise and uncertainty in the observed tracking data. This work has demonstrated the large amount of extractable information present in low-level motion that is currently not being exploited. The next step in the work is to integrate spatial information to the models, so as well as being concerned with how points move we consider where they move relative to one another.

REFERENCES

Argawal, A. and Triggs, B. (2004). Tracking articulated motion with piecewise learned dynamical models. In *ECCV*, pages 54–65.

Caillette, F., Galata, A., and Howard, T. (2005). Real-Time 3-D Human Body Tracking using Variable Length Markov Models. In *BMVC*, pages 469–478.

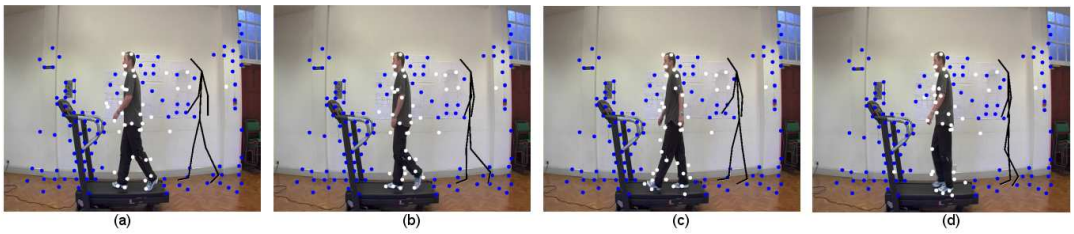

Figure 6: Sample frames with extracted phase illustrated by a stick man; white points represent KLT features classified as foreground points, blue points represent KLT features classified as background points.

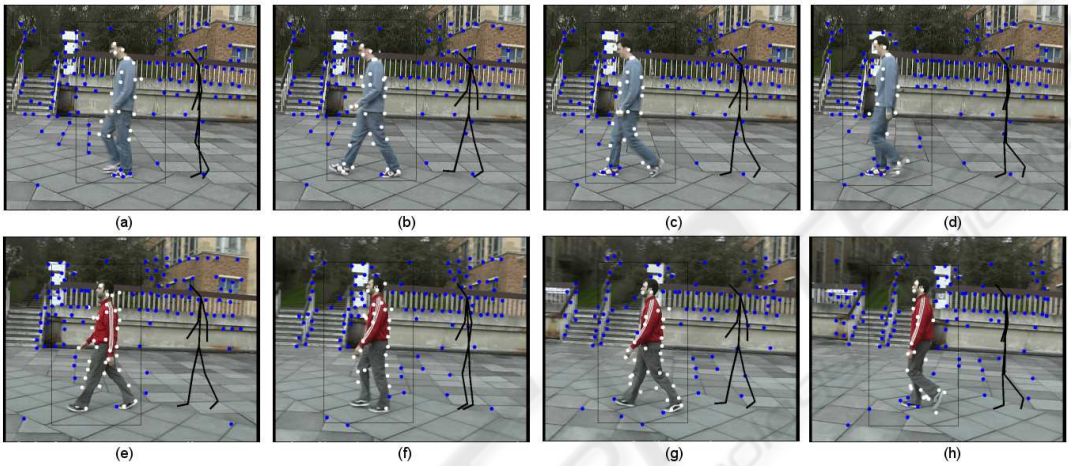

Figure 7: Figures (a) to (d) are frames taken from a sequence where the camera was stationary, Figures (e) to (g) are frames taken from a sequence where the camera panned to follow the walker; white points represent KLT features classified as foreground points, blue points represent KLT features classified as background points.

Coughlan, J., Yuille, A., English, C., and Snow, D. (2000). Efficient deformable template detection and localization without user initialization. *CVIU*, 78(3):303–319.

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

Gibson, D., Campbell, N., and Thomas, B. (2003). Quadruped gait analysis using sparse motion information. In *ICIP*, pages 333–336.

Havasi, L., Szlavik, Z., and Sziranyi, T. (2007). Detection of gait characteristics for scene registration in video surveillance system. *IEEE Trasactions on Image Processing*, 16(2):503–510.

Hu, S. and Buxton, B. F. (2005). Using temporal coherence for gait pose estimation from a monocular camera view. In *BMVC*, pages 449–558.

Lan, X. and Huttenlocher, D. P. (2004). A unified spatio-temporal articulated model for tracking. In *CVPR*, pages 722–729.

Micilotta, A. S., Ong, E. J., and Bowden, R. (2006). Real-Time Upper Body Detection and 3D Pose Estimation in Monoscopic Images. In *ECCV*, pages 139–150.

Navaratnam, R., Thayananthan, A., Torr, P., and Cipolla, R. (2005). Hierarchical part-based human body pose estimation. In *BMVC*, pages 479–488.

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

Shi, J. and Tomasi, C. (1994). Good features to track. In *CVPR*, pages 593 – 600.

Sidenbladh, H., Black, M. J., and Fleet, D. J. (2000). Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In *ECCV*, pages 702–718.

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

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