Finding and Tracking People from the Bottom Up

Deva Ramanan and D. A. Forsyth
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
ramanan@cs.berkeley.edu, daf@cs.berkeley.edu

Abstract

We describe a tracker that can track moving people in long sequences without manual initialization. Moving people are modeled with the assumption that, while configuration can vary quite substantially from frame to frame, appearance does not. This leads to an algorithm that firstly builds a model of the appearance of the body of each individual by clustering candidate body segments, and then uses this model to find all individuals in each frame. Unusually, the tracker does not rely on a model of human dynamics to identify possible instances of people; such models are unreliable, because human motion is fast and large accelerations are common. We show our tracking algorithm can be interpreted as a loopy inference procedure on an underlying Bayes net. Experiments on video of real scenes demonstrate that this tracker can (a) count distinct individuals; (b) identify and track them; (c) recover when it loses track, for example, if individuals are occluded or briefly leave the view; (d) identify the configuration of the body largely correctly; and (e) is not dependent on particular models of human motion.

1. Introduction

A practical person tracker should: track accurately for long sequences; self-start; track independent of activity; be robust to drift; track multiple people; track through brief occlusions; and be computationally efficient. It should also avoid background subtraction; we want to track people who happen to stand still on backgrounds that happen to move.

The literature on human tracking is too large to review in detail. Tracking people is difficult, because people can move very fast. One can use the configuration in the current frame and a dynamic model to predict the next configuration; these predictions can then be refined using image data (see, for example, [9, 13, 3]). Particle filtering uses multiple predictions — obtained by running samples of the prior through a model of the dynamics — which are refined by comparing them with the local image data (the likelihood) (see, for example [14, 3]). The prior is typically quite diffuse (because motion can be fast) but the likelihood function may be very peaky, containing multiple local maxima which are hard to account for in detail. For example, if an arm swings past an "arm-like" pole, the correct local maximum must be found to prevent the track from drifting. Annealing the particle filter is one way to attack this difficulty [6]. An alternative is to apply a strong model of dynamics [14], at the considerable cost of needing to choose the motion model before one can detect or track people. An attractive alternative is to ignore dynamics and find people in each frame independently, using such cues as local motion [15] or appearance [11].

As far as we know, no current person tracker meets all our criteria. This is most likely because of difficulties with *data association* [2]; only a small part of the abundant image data contains any information about the person and one may need to determine which part this is. Particle filters do data association only implicitly which explains their attraction. One may use a variety of templates encoding the appearance of the person being tracked (e.g. [12, 16]). To date, these templates have been learned in advance of tracking, and so cannot use properties like the color of clothing, which change from instance to instance.

People tend not to change "appearance" (color and texture of clothing, etc.) from frame to frame. We describe a people tracker which builds models of the people to be tracked from the video sequence and then tracks them. This has considerable advantages: First, knowing the appearance model of each body part greatly constrains our search and so simplifies data association. Second, we can prevent drift, recover from occlusion relatively easily, and count individuals. We show examples that suggest our tracker meets our criteria.

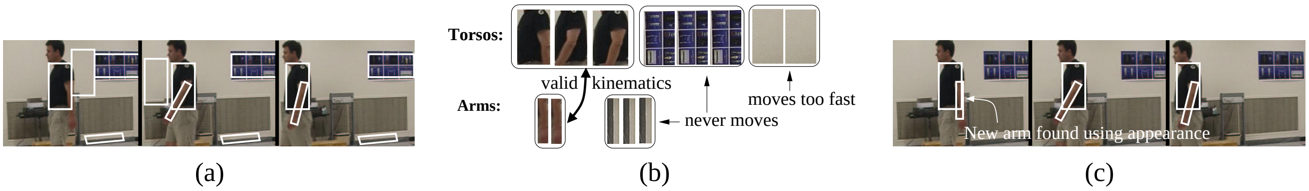


Figure 1. Description of algorithm for a torso-arm person model for 3 frames. We search for candidates in (a), enforce constant appearance by clustering the patches in (b). By connecting good clusters with valid kinematics, we learn the appearance of the torso and arm, which we use to find new candidates in (c).

2 Algorithm

2.1 General approach

We model a 2D view of the human body as a puppet of colored, textured rectangles [7], such as those in Fig.7. We use the 9 *segment* tree model of [10], consisting of the torso plus the left/right upper/lower arms/legs (Fig.2-c). We look for candidate body segments in each frame of a sequence and group them to form a sequence of likely puppet configurations.

Our fundamental assumption is that coherence in appearance is a stronger cue to body configuration than dynamics because body segments may move very fast but it takes time to change clothes. This suggests the following 2 part strategy. We first **learn** the appearance of each rectangle in our puppet, and then **find** likely puppet configurations in each frame which link up over time.

We learn an appearance for each body segment in our puppet model by the following:
Detect candidate body segments with detuned detectors.
Cluster the resulting image patches associated with each candidate to identify segments that look similar over time.
Prune clusters whose members have unlikely dynamics.

2.2 Learning appearance

We model segments as cylinders and generate candidates by convolving each frame with a template that responds to parallel lines of contrast (at a variety of orientations and scales), suppressing the non-maximum responses. We expect our local detectors to suffer from false positives and missed detections, such as those shown in Fig.1-(a).

We create a feature vector for each candidate segment capturing its appearance; we use a normalized color histogram (in Lab space) of the associated image patch. We represent the histogram with projections onto the L, a, and b axis, using 10 bins for each projection. We could extend our 30 dimensional feature vector by incorporating further cues such as image texture, but this appears unnecessary for clustering.

We learn appearance by clustering the candidate feature vectors for each body segment, as in Fig.1. Since we do not know the number of people in our sequence *a priori* (or for that matter, the number of segment-like objects in the background), typical parametric methods such as k-means or gaussian mixture models prove difficult.

We opted for a modified mean-shift procedure [4], a non-parametric density estimation technique. We interpret our candidate segments as points lying in a 30 dimensional feature space. We find the mean position of all points within a hypersphere, recenter the hypersphere around the new mean, and repeat until convergence. When encountering multiple points from the same frame, we only use the point closest to the current hypersphere center to calculate the new mean. We initialize this procedure at each original feature point, and denote all points which converge to the same position as belonging to the same cluster. We prune those clusters whose members violate our bounded-velocity motion model or which stay completely still (Fig.1-(b)), arguing that both behaviors are not body segment-like.

The claim that we should only concern ourselves with segments that are coherent over time and that move (a notion we call *foreground enhancement*) is markedly different from traditional background subtraction since it is used to learn appearance and not to find people. Once the appearance is known, we can track people who stand still (so long as they move at some point).

2.3 Finding (multiple) people using appearance

We connect up the remaining clusters which best obey our kinematic constraints to learn the appearance of each body segment. If more than one torso and arm cluster linked up in Fig.1-(b), we could have learned multiple appearance models for different people. Hence tracking multiple people follows naturally from our algorithm.

We can use the learned appearance to build better segment detectors; we now know the arm is a brown rectangle, rather than just two parallel edges. We search for *new* candidates using the medoid image patch of the valid clusters from Fig.1-(b) as a template. We link up those candidates which obey our velocity constraints into the final track in Fig.1-(c). We currently make the simplifying assumption that all people in a sequence have different appearance templates, although instancing a single appearance template multiple times is straightforward.

3 Probabilistic model

We now motivate our algorithm by introducing a probabilistic graphical model. The relevant variables from Fig.2-(a):

C_{seg} – Constant underlying appearance feature vector
P^i_{seg} – Position (and orientation) of segment in frame i
Im^i_{seg} – Collection of observed feature vectors for each image patch in frame i

Im^i_{seg} represents a stack of image patches from each position in an image, indexed by the given image position. One of those patches is the true segment patch, while the others are background (which we omit from Fig.2-(a) for simplicity). P^i_{seg} can be interpreted as a pointer to the correct patch, such that

$$Im^i_{seg}(P^i_{seg})$$

is distributed as

$$\phi_{seg}(C_{seg})$$

In our case, the appearance model $\phi_{seg}(C_{seg})$ is an Epanechnikov (triangle) kernel centered at C_{seg}. Our graphical model explicitly makes obvious the data association issue at hand; since we do not observe P^i_{seg}, we do not know where in the image to look for a segment, and hence must consider all the possible image patches in Im^i_{seg}. In turn, we see that inference is complicated by the fact that P^i_{seg} variables from across frames are dependent on each other; our segments must move with a specific motion model. Since we would like to track people performing a variety of activities, we limit ourselves to a bounded-velocity motion model.

We can simplify our model by turning to the undirected case in Fig.2-b. Note that since we observe Im^i_{seg}, we only use a 2-dimensional slice of the 3-dimensional "table" $\Pr(Im^i_{seg}|P^i_{seg}, C^i_{seg})$. Hence the image observations specify a potential between P^i_{seg} and C_{seg} (this is the standard moralization that results from conversion of a directed graphical model to an undirected one). Note that both our appearance model $\phi_{seg}(C_{seg})$ and our image observations Im^i_{seg} are now implicitly represented in the potentials $\psi_i(C_{seg}, P^i_{seg})$, while our motion model lives in the potential $\psi_{link}(P^i_{seg}, P^{i-1}_{seg})$.

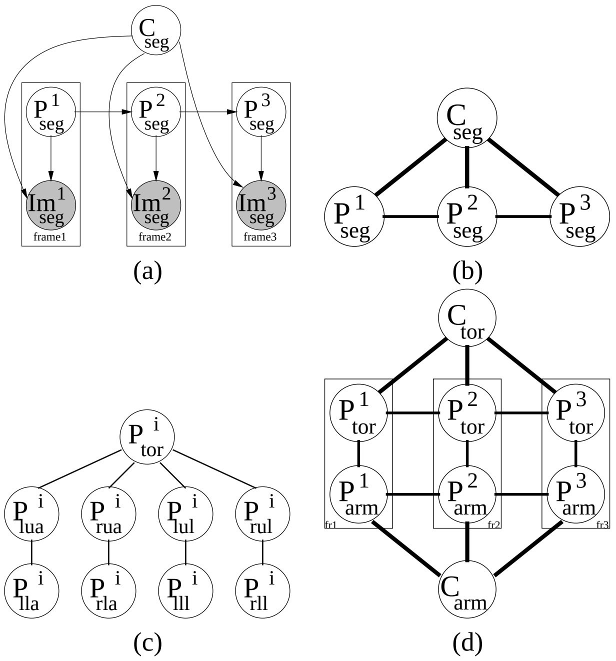


Figure 2. Graphical model. The model in (a) encodes the fact that each image instance of a segment has the same appearance (encoded in C_{seg}), but appears in different places. In (b), the undirected form of the model. A complete person model (c) contains additional kinematic dependencies. We show the full imaging model for a limited torso and arm subset in (d).

Our model becomes more complicated for the full human body, consisting of 9 segments with additional *kinematic* constraints between them (Fig.2-c). Although [10] learns these potentials from training data, we construct them by hand assuming a fully deformable connected kinematic model. We show the full model for a two-segment (torso and arm) subset in Fig.2-d.

Local segment detectors fit into our model as an additional potential $\psi_{seg}(C_{seg})$; we favor rectangles in our puppet with a specific edge profile (we might also favor rectangles with skin-colored pixels). Hence our initial segment detection step prunes away those image patches with a low ψ_{seg} potential.

Note that most people trackers model the appearance as changing over time by replacing the constant C_{seg} with a temporally varying copy, C^i_{seg}, in each frame plate. Although this alternative model is fully Markovian and might appear to make inference simpler, the constant appearance assumption proves key for data association. We quickly prune away those parts of the image which do not look like our underlying appearance model.

3.1 Tracking as inference

Finding an optimal track given a video sequence now corresponds to finding the maximum *a posteriori* (MAP) estimate of both the C_{seg} and P^i_{seg} nodes from Fig.2-d. Exact inference is difficult for two reasons; one, the graph contains large induced cliques and two, the domains of the nodes are typically large (they are discretized versions of underlying continuous quantities).

If our model was a tree, we could find the MAP estimate by dynamic programming, also known as max product belief propagation (BP). This is in fact true for any model lacking undirected cycles (which we will also refer to as trees, with a slight abuse of terminology). However, we can still apply the updates to nodes in a cyclic graph, pretending that the local neighborhood of each node is a tree. Many researchers have found loopy max product BP and its more common sum product variant to be empirically successful [5]. In particular, fixed points are guaranteed to yield solutions which are optimal within a large neighborhood of alternative assignments [8].

Applying these updates asynchronously, we can interpret this procedure as performing inference on a set of embedded trees within the model (similar to [17]). The algorithm described in section 2 performs approximate inference on Fig.2-(d) using the trees in Fig.3. We cluster to learn a torso appearance in Fig.3-(a), learn an arm appearance in (b), connect up the kinematically valid clusters in (c), and enforce our velocity bounds in (d) to create our approximate MAP estimate.

3.2 Clustering as loopy inference

Although clustering may not seem like inference on trees (a) and (b) in Fig.3, it is an approximate procedure to obtain likely values of C_{seg}. Assume messages are initialized to '1'. Passing max product messages on trees (a) and (b) is equivalent to finding values of C_{seg} and P^i_{seg} that maximize $\psi_1(C_{seg}, P^1_{seg})\psi_2(C_{seg}, P^2_{seg}) \ldots \psi_k(C_{seg}, P^k_{seg})$, where the image information is implicit in the ψ_i's (whence the subscript).

Now this corresponds to choosing a C_{seg} and a set of P^i_{seg} such that all the image segments identified by P^i_{seg} look like C_{seg}. If both variables were defined over small, discrete domains, all we'd be doing is dynamic programming; for each value of C_{seg}, we'd choose the best P^i_{seg} for each i, form the product, and then choose the C_{seg} with the best product. This search is not easy for continuous or large domains (e.g. [1]).

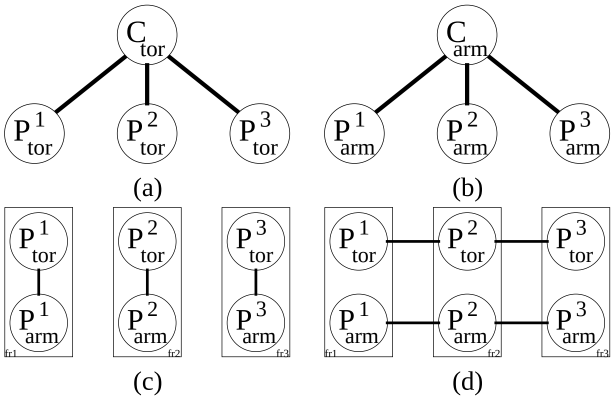


Figure 3. A set of trees for loopy inference on Fig.2-(d). Parts (a) and (b) correspond to the learning and application of torso and arm appearance templates. Part (c) corresponds to selecting an arm and torso cluster which obey our kinematic constraints. Part (d) corresponds to the application of the motion model.

However, we know that we are looking for, in essence, a point in the domain of C_{seg} such that there are many image segments that look like that point. Now assume we have a detuned but usable segment detector. It will detect many, but not all, instances of the relevant segment and some background, too. The instances of the relevant segment will look like one another. This means that, by clustering the representations of the segment appearances, we are obtaining a reasonable approximation to the true max marginal of C_{seg}. In particular, finding local modes of C_{seg} using a Parzen's window estimate with a kernel ϕ_{seg} is equivalent to the mean-shift algorithm. In fact, using a more sophisticated appearance kernel ϕ_{seg} reduces to using a weighted mean in each iteration of the mean-shift procedure. We possess no formal information on the quality of the approximation.

Likewise, inferring the max marginal of P^i_{seg} from trees (a) and (b) in Fig.3 can be approximated by querying the i^{th} image using the inferred appearance C_{seg}. Hence the second phase of our algorithm where we use the inferred appearance to build new segment detectors also falls from this loopy framework.

3.3 Algorithm details

Our algorithm infers with a single pass through each link in our model, rather than repeatedly passing messages until convergence. We experimented with various orderings of the trees in Fig.3. For example, we could apply tree (d) immediately after inferring on trees (a) and (b). This suggests a procedure more sophisticated that our initial one; rather than naively pruning a cluster if it contains invalid dynamics, we can extract a sequence of dynamically valid segments from it (and then prune it if the sequence is too small). This is in practice what we do.

Alternatively, inferring on tree (c) before tree (b) equates to only searching for candidate arms near the clustered torsos (i.e., we can use a truncated version of our kinematic potential $\psi_{kin}(P^i_{tor}, P^i_{arm})$ to limit our search space for arms given the found torsos). We use the following ordering; first, we learn the torso appearance template by clustering candidate torsos, and then use it to find a dynamically valid torso track. Next we do the same for lower limbs only searching for candidates near the found torsos. Finally, we repeat the procedure for upper limbs. We found this ordering sensible since it reflected the quality of our initial segment detectors (upper limbs are hard to detect, so we constrain their position by first detecting torsos and lower limbs).

We update our appearance kernel ϕ_{seg} to reflect the clustered segments; after selecting the correct segment clusters from Fig.1-(b), we set ϕ_{seg} to be a gaussian with the sample mean and variance of the corresponding cluster. We assume the left and right limbs have the same appearance, and as such only learn a single appearance for both.

For long sequences, we only cluster the first K out of the total N frames.

4 Experimental results

Typically, a primary criterion for evaluating the performance of a tracker is the number of successive frames a person can be tracked. However, this is a poor measure because people may become occluded or leave the view. A better measure is whether a tracker finds a person given one is present and does not find a person given one is absent [15]. We extend this analysis to the detection of individual segments.

In Table 1, we show such results for four different sequences. "Jumping Jacks" and "Walk" were both taken indoors while the subjects were simultaneously motion captured. Ground truth for those sequences was obtained by registering the motion capture data with the video. The sequence lengths are N = 100 and 288, respectively, captured at 15 frames per second. The appearance templates were learned using the first K = 75 and 150 frames. "Street Pass" and "Weave Run" were both taken outdoors, and ground truth was hand-labeled for a random subset of frames. For these sequences, we clustered the first K = 200 and 150 out of the total of N = 380 and 300 frames, captured at 30 frames per second. We define a "detection" to occur when the estimated and ground truth segment overlap. Our torso detection rates in Table 1 are quite good, similar to those in [15]. However, one key difference is that we do not need activity specific motion models to obtain them. Detecting the limbs proves to be more difficult, especially in the more challenging outdoor sequences. We show example frames from each sequence and discuss additional results below.

Sequence	Torso		Arms		Legs	
	D	FA	D	FA	D	FA
J. Jacks	94.6	0.00	87.4	0.56	91.8	0.19
Walking	99.5	0.41	84.3	0.41	68.2	1.01
Street Pass	100	0.00	66.7	0.93	42.4	3.02
Weave Run	92.9	0.00	23.3	2.89	63.0	1.92

Table 1. Tracker performance on various sequences. Our tracker will work on long sequences if our learned appearance models are sufficiently general. As such, we measure percentage detections (D) and false alarms (FA) of segments. Our torso rates are quite good, while rates for limbs suffer in the challenging outdoor sequences. In general, the appearance models learned from the data are tight, allowing for very little false positives.

Self-starting: None of these tracks were hand initialized. However, we do optimize thresholds for the segment detectors and the bandwidth for the mean-shift procedure for the sequences shown. More sophisticated segment detection and clustering algorithms may eliminate the need for tweaking.

Multiple activities: In Fig.4, we see frames from the two indoor sequences; "Jumping Jacks" and "Walk". In both top rows, we show the original edge-based candidates which cluster together. That is, we have pruned away those candidates not in the optimal cluster, but we have yet to use the learned appearance template to search the image again. Note when limbs are close to the body or surrounded by a weak-contrast background, few candidates were found due to weak edges. In the bottom rows, we search for limbs using an appearance template (learned using surrounding frames when limbs were clearly defined) and now track over traditionally difficult kinematic configurations and background conditions. Note that the activities in both these sequences are quite different; trying to track a walking person with a jumping jack motion model (and vice-versa) may prove very difficult. However, our weak bounded-velocity motion model proved broad enough to track both.

Lack of background subtraction: In Fig.5, we examine our foreground enhancement claim on the 'Street Pass"

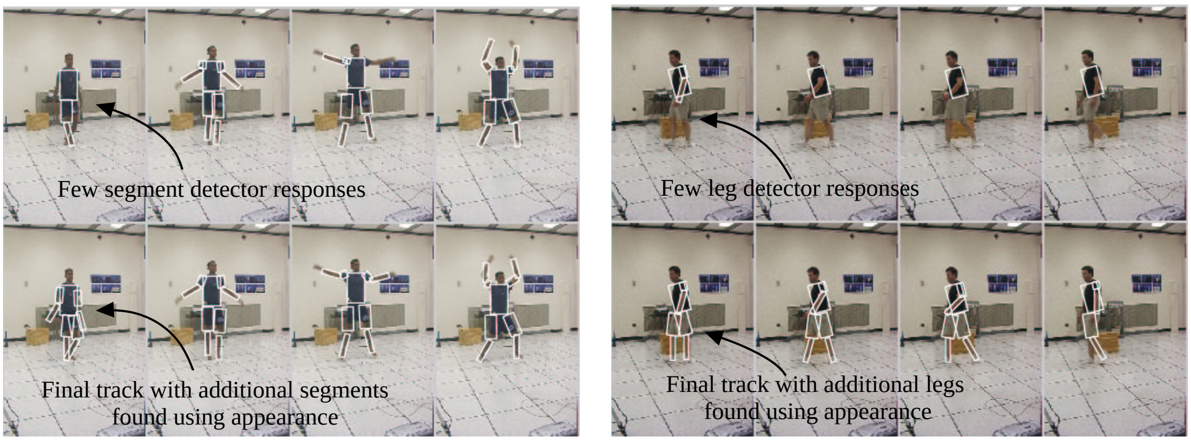

Few segment detector responses
Final track with additional segments found using appearance
Few leg detector responses
Final track with additional legs found using appearance

Figure 4. Self-starting tracker on "Jumping Jacks" (*left*) and "Walk" (*right*) sequences. Recall our algorithm has two basic phases; we learn appearance by clustering candidates, and then find people using appearance. Both *top* rows show the original edge candidates which clustered together; both *bottom* rows show the final tracks. In frames where the limbs are close to the body or surrounded by a weak-contrast background, we find additional segments using the learned appearance templates.

sequence, which contains multiple moving objects. We still learn fairly accurate appearance models, although varying lighting conditions often result in poor matches (as in the high missed detection rates for arms and legs in Table 1). By using a metric more robust to lighting changes, the appearance models learned in the clustering and the segments found may be more accurate. Alternatively, one might add explicit illumination variables to C_{seg} to deal with temporal changes in brightness.

Multiple people, recovery from occlusion and error In Fig.7, we see frames from the "Weave Run" sequence, in which three figures are running in a weave fashion. In the top row, we see two tracks crossing. When the two figures lie on top of each other, we correctly disambiguate who is in front, and furthermore, recover the interrupted track of the occluded figure. In the bottom row, a track finds a false arm in the background but later recovers. We also see a new track being born, increasing the count of tracked people to three.

4.1 Discussion

Our algorithm (1) learns segment appearances by clustering, (2) finds assemblies in each frame using the appearance and finally (3) links up the temporally consistent assemblies into a track.

More generally, this approach reduces tracking to the problem of inference on a dynamic Bayes net. Other approximate inference algorithms may yet prove more successful. However, our approximations exploit the close link between tracking and learning appearance. Note we are able to learn appearance due to the underlying constancy of segment image patches; other features commonly used for tracking such as edges [11] or motion vectors [15] do not exhibit this phenomenon. Similarly, we can extend our framework to other features that might be constant over time, such as texture.

How many frames does one need to cluster to obtain a working appearance model? Certainly with more frames we would expect better *localization* of segments (by having a lower variance estimate of appearance), but the question of the number of frames needed to *detect* the correct segments is rather interesting. We evaluate appearance model performance versus the number of K frames used for learning in Fig.6 for the "Walk" sequence. We also show results for models learned with local detectors augmented with a skin classifier.

Since we are using the first K frames to build a model which we hope will generalize for the entire N frame sequence, we also use Fig.6 to examine the generalizability of our model as a function of K. We plot performance of the learned appearance templates on the the first K training frames in (a) and (b), while we look at the performance on the entire N frame sequence in (c) and (d). Note that the performance is virtually identical for either measure. This suggests that our model of stationary appearance is indeed appropriate; after the first 60 frames, we pretty much know the segment appearances in the next 222. Because we have learned accurate appearance models, we should be able to track for arbitrarily long.

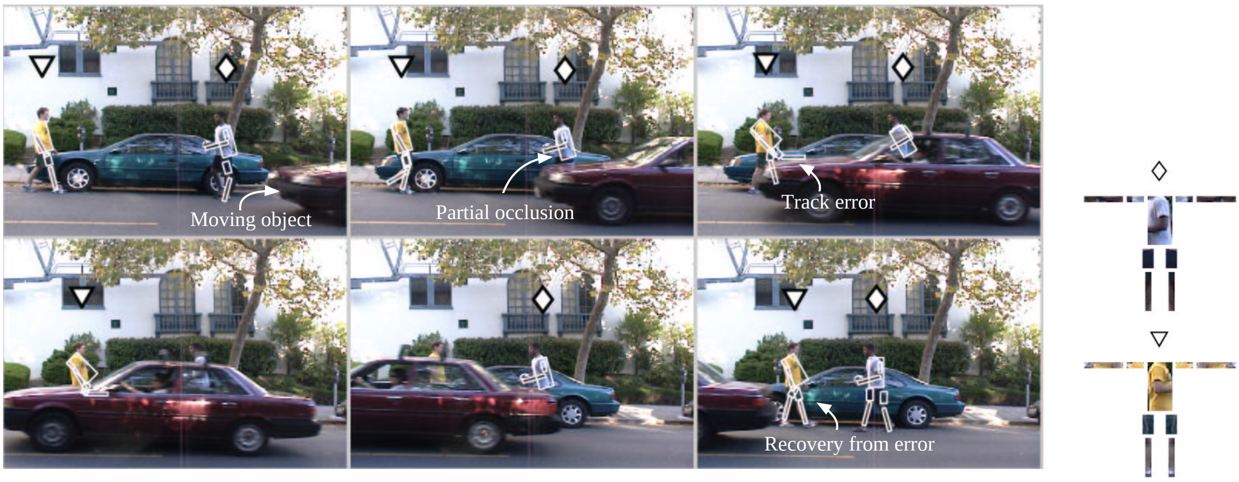


Figure 5. Self-starting tracker on “Street Pass” sequence containing multiple moving objects. Both rows are select frames from the final track. The learned appearance templates are also shown on the *right*. We denote individual tracks by a token displayed above the figure. The tracker successfully learns the correct number of appearance models, and does not mistake the moving car for a new person. We are also able to recover from partial and complete occlusion, as well as from errors in configuration (which Markovian state models would typically fail on).

Let us now consider how the quality of the low-level segment detectors affects the learned appearance templates. Both detectors in Fig.6 happen to perform well for small K since our subject is initially against a uncluttered background. As we increase K and our subject walks into the room, our edge detectors pick up extraneous background candidates which cluster into poor appearance models. However, both perform well as K is increased sufficiently. This result suggests that high-level clustering can compensate for poor low-level detection, given we cluster over enough frames.

References

[1] D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[2] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[3] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE T. Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[5] J. Coughlan and S. J. Ferreira. Finding deformable shapes using loopy belief propogation. In *European Conference on Computer Vision*, 2002.

[6] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.

[7] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.

[8] W. T. Freeman and Y. Weiss. On the fixed points of the max-product algorithm. *IEEE T. Information Threory*, 2000.

[9] D. Hogg. Model based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[10] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *Int. Conf. on Computer Vision*, 2001.

[11] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision*, 2002.

[12] C. Pal, B. Frey, and N. Jojic. Learning montages of transformed latent images as representations of objects that change in appearance. In *European Conference on Computer Vision*, 2002.

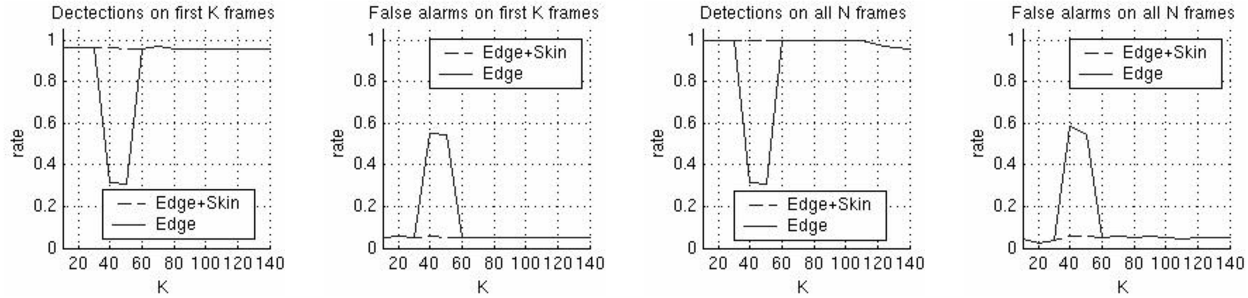
[13] K. Rohr. Incremental recognition of pedestrians from image sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 9–13, 1993.

[14] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, 2000.

[15] Y. Song, X. Feng, and P. Perona. Towards detection of human motion. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 810–17, 2000.

[16] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. J. Computer Vision*, 48(1):9–19, 2002.

[17] M. Wainwright, T. Jaakola, and A. Willsky. Tree-based reparameterization for approximate inference on loopy graphs. In *NIPS*, volume 14, 2001.



(a) (b) (c) (d)

Figure 6. Appearance model performance for "Walk" sequence. We plot performance for models constructed with edge and edge+skin based detectors for varying K frames. In (a) and (b) we consider the performance of the models on the K training frames. In (c) and (d) we look at performance on the entire sequence. Our models perform similarly on both the training and test frames, so they seem to generalize well. For small K, both detectors fortuitously learn a good model due to a lack of background clutter. As K increases, background clutter leads our edge detectors to construct poor appearance models. For large K, clustering yields working models irrespective of the detectors.

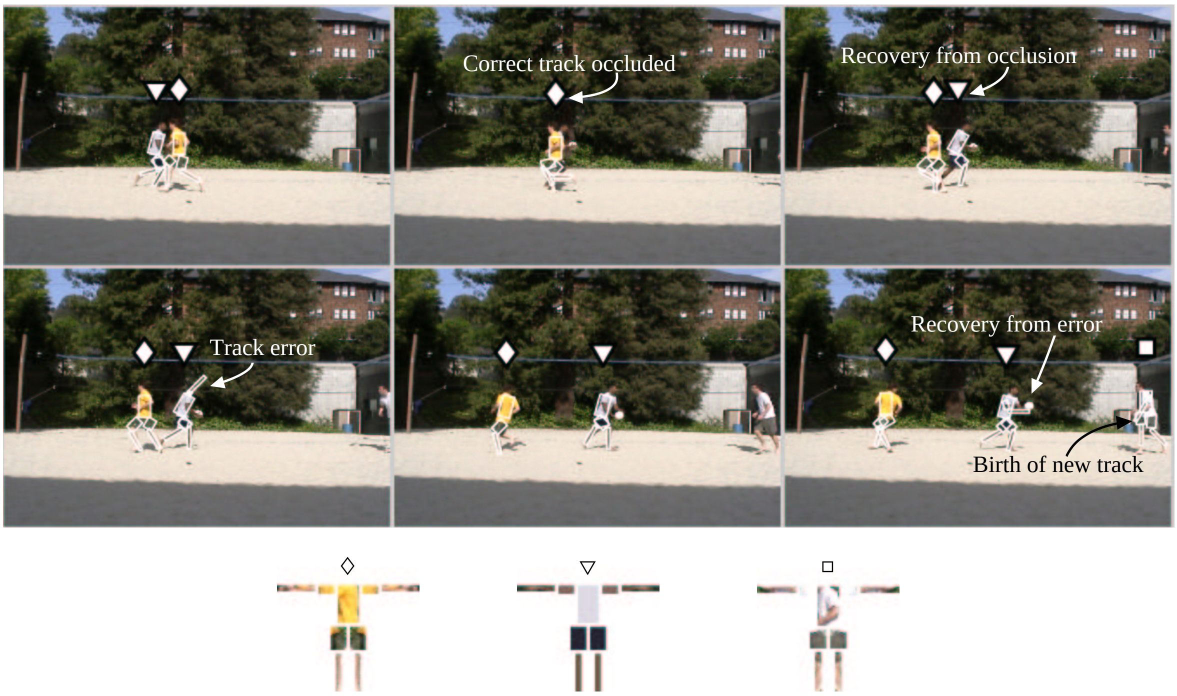


Figure 7. Self-starting tracker on "Weave Run". We show a subset of frames illustrating one figure passing another, with the learned appearance templates below. Note the correct figure is occluded, and furthermore the track is recovered once it reappears. An earlier incorrect arm estimate is also fixed (this would prove difficult assuming a drifting appearance model). Finally, we show a new track being born, increasing the count of found people to three.