Strike a Pose: Tracking People by Finding Stylized Poses

Deva Ramanan[1] and D. A. Forsyth[1] and Andrew Zisserman[2]
[1]University of California, Berkeley – Berkeley, CA 94720
[2]University of Oxford – Oxford, OX1 4AJ, UK
{ramanan,daf}@cs.berkeley.edu, az@robots.ox.ac.uk

Abstract

We develop an algorithm for finding and kinematically tracking multiple people in long sequences. Our basic assumption is that people tend to take on certain canonical poses, even when performing unusual activities like throwing a baseball or figure skating. We build a person detector that quite accurately detects and localizes limbs of people in lateral walking poses. We use the estimated limbs from a detection to build a discriminative appearance model; we assume the features that discriminate a figure in one frame will discriminate the figure in other frames. We then use the models as limb detectors in a pictorial structure framework, detecting figures in unrestricted poses in both previous and successive frames. We have run our tracker on hundreds of thousands of frames, and present and apply a methodology for evaluating tracking on such a large scale. We test our tracker on real sequences including a feature-length film, an hour of footage from a public park, and various sports sequences. We find that we can quite accurately automatically find and track multiple people interacting with each other while performing fast and unusual motions.

1. Introduction

Kinematically tracking people is a task of obvious importance; people are quite concerned about what other people are doing. Large-scale, accurate, and automatic kinematic tracking would allow for data mining of surveillance video, studies of human behavior and bulk motion capture. No current systems are capable of kinematic tracking on a large scale; most demonstrate results on mere hundreds of frames. We develop an algorithm that is accurate and automatic, allowing us to evaluate results on over one hundred thousand frames.

The literature on human tracking is too large to review in detail. Tracking people is difficult, because people can move very fast and configure themselves in many different poses. 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, [2, 7, 17]). 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 [2, 9, 19]). 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 [5] or performing local searches [21] are ways to attack this difficulty. An alternative is to apply a strong model of dynamics [19]; typically one must choose this model *a priori*, but methods for online selection exist [1].

An alternative is to ignore dynamics and find people in each frame independently, using such cues as local motion [22] or appearance [8, 13, 23] or both [25]. This approach is attractive because it self-starts and is robust to drift (since it essentially re-initializes itself at each frame). In general, detecting people is hard because people wear different clothes and can take on many poses; this suggests a bottom-up approach using part detectors [10, 12, 14, 18, 20]. Approaches combining detection and tracking have also proven useful [11, 15].

2. General approach

We follow the approach of [16], which uses the fact that people tend not to change appearance over a track. The authors first (a) cluster candidate limbs detected in a set of frames to learn appearance models for each limb and then (b) track by detecting the appearance models in each frame. The clustering step only works for sequences where limbs are reliably found by low-level detectors and where limbs look different from the background. If the algorithm produces bad clusters, the resulting appearance models will produce poor tracks.

We observe that the initial set of detectors are not trying to detect but rather learn appearance. This is an important

0-7695-2372-2/05/$20.00 (c) 2005 IEEE

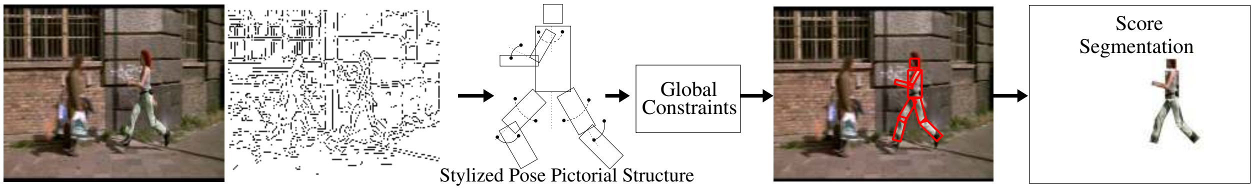

Figure 1. *Our lateral-walking pose finder. Given an edge image on the* **left**, *we search for a tree pictorial structure [6] using rectangle chamfer template costs to construct limb likelihoods. We restrict limbs to be positioned and oriented within bounded intervals consistent with walking left. We set these bounds (designated by the arcs overlaid on the model) by hand. We also search a mirror-flipped version of the image to find people walking right. To enforce global constraints (left and right legs should look similar), we sample from the pictorial structure posterior (using the efficient method of [6]), and re-compute a global score for the sampled configurations. The best configuration is shown on the* **right**. *In general, this procedure also finds walking poses in textured backgrounds; to prune away such false detections, we re-evaluate the score by computing the goodness of a segmentation into person/non-person pixels. We do this by building an appearance model for each limb (as in Fig.2) and then use the model to classify pixels from this image. We define the final cost of a walking-pose detection to be the number of mis-classified pixels.*

distinction because typically one wants detectors with high precision and recall performance. In our case, we want a person detector with rather unique properties: (a) it must accurately localize limbs (since we will use the estimated limbs to build appearance models) and (b) it should have high precision (we want most detections to be of people). Given both, we can tolerate a low recall rate since we can use the learned appearance models to find the figure in those frames where the detector failed.

We build a person detector that only detects people in typical poses. Even though the detector will not fire on atypical poses, we can use the appearance learned from the standard poses to track in those atypical frames. This notion of *opportunistic detection* states that we can choose those poses we want to detect. This way we concentrate our efforts on easy poses rather than expending considerable effort on difficult ones. Convenient poses are ones that are (a) easy to detect and (b) easy to learn appearance from. For example, consider a person walking in a lateral direction; their legs form a distinctive scissor pattern that one tends not to find in backgrounds. The same pose is also fairly easy to learn appearance from since there is little self-occlusion; both the legs and arms are swinging away from the body. Following our observations, we build a single-frame people detector that finds people in the mid-stance of a lateral-walk.

Once we have detected a lateral-walking pose with our detector (Sec.3), we build a discriminative model of appearance of each limb (Sec.4). We assume the features that help discriminate the figure in one frame will help detect the figure in other frames. We finally track by detecting the appearance model in other frames where the figure can be in any pose (Sec.5). We develop and apply a methodology for evaluating data on a large scale in Sec.6.

3 Detecting Lateral Walking Poses

An overview of our approach to people detection is found in Fig.1. We will use a sequence from the film 'Run Lola Run' as our running example (pun intended). Our basic representation is a tree pictorial structure that decomposes a person model into a shape model and appearance model [6, 8, 16]. If we write the configuration of a limb as $P_i = [x, y, \theta]$, we can write the posterior configuration for a person given an image as

$$\Pr(P_1 \ldots P_n|Im) \propto \prod_{(i,j)\in E} \Pr(P_i|P_j) \prod_{i=1}^{n} \Pr(Im(P_i)) \tag{1}$$

where i ranges over set of limbs (head, torso, upper/lower arm, and left/right upper/lower legs) and E is the set of edges that defines the tree structure. $\Pr(P_i|P_j)$ is the shape model, and $\Pr(Im(P_i))$ is the local image likelihood given the limb appearance model. We search for only one arm since we assume the other will be occluded in our lateral walking pose. When convenient we refer to the terms above as costs rather than probabilities (implying we are in negative log space).

$\mathbf{Pr(\boldsymbol{P_i}|\boldsymbol{P_j})}$: We manually set our kinematic shape potentials to be uniform within a bounded range consistent with walking laterally (Fig.1). For example, we force θ for our upper legs to be between 45 and 15 degrees with respect to the torso axis. We do not allow them to be 0 degrees because we want to detect people in a distinctive scissor-leg pattern. Learning these potentials automatically from data is interesting future work.

$\mathbf{Pr(\boldsymbol{Im}(\boldsymbol{P_i}))}$: We evaluate the local image likelihood with a chamfer template edge mask [24]. We use rectangles as our edge templates (Fig.1). Given an image, the chamfer cost of a edge template is the average distance between each edge in the template and the closest edge in the image. We compute this efficiently by convolving the distance-transformed edge image with the edge template. To exploit edge orientation cues, we quantize edge pixels into one of 12 orientations, and compute the chamfer cost separately for each orientation (and add the costs together). To capture the deformations from Fig.1, we convolve using rotated versions of our templates. We assume the figure is at a fixed scale, and as such do not search over scale.

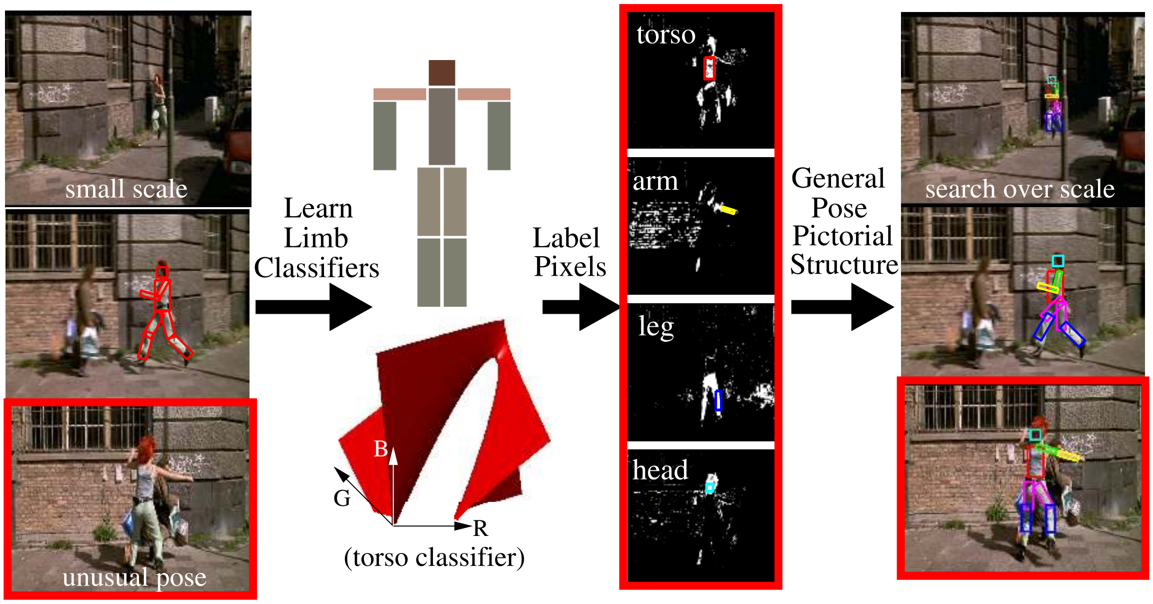

Figure 2. *An overview of our approach; given a video sequence, we run a single-scale walking pose detector on each frame. Our detector fails on the small scale figure and the on a-typical pose, but correctly detects the walking pose (***left***). Given the estimated limb positions from that detection, we learn a quadratic logistic regression classifier for each limb in RGB space, using the masked limb pixels as positives and all non-person pixels as negatives. In the* **middle left**, *we show the learned decision boundary for the torso and crudely visualize the remaining limb classifiers with a gaussian fit to the positive pixels. Note the visual models appear to be poor; many models look like the background because some of the limb pixels happen to be in shadow. The classifiers are successful precisely because they learn to ignore these pixels (since they do not help discriminate between positive and negative examples). We then run the classifiers on* all *frames from a sequence to obtain limb masks on the* **middle right** *(we show pixels from the third frame classified as torso, lower arm, lower leg, and head). We then search these masks for candidate limbs arranged in a pictorial structure (searching over general pose deformations at multiple scales) [6]. This yields the recovered configurations on the* **right**. *We show additional frames in Fig.4*

Since we use our lateral-walking detector in a high-precision/low-recall regime, we need to only look at those configurations where all the limbs have high likelihoods. Before evaluating the kinematic potentials, we perform non-maximum suppression on the chamfer likelihood response functions (and only keep candidate limbs above a likelihood threshold). We also throw away arm candidates that are vertical or horizontal (since there tends to be many vertical and horizontal rectangles in images of man-made structures). This is again justified for high-precision/low-recall detection; even though the arm of a person may in fact be horizontal or vertical, we *choose* not to learn their appearance in this pose, since we would encounter many false positive detections (and build incorrect appearance models).

Global constraints: We found it useful to enforce global constraints in our person model. For example, left and right legs tend to be similar in appearance. Also, our kinematic leg potentials still allow for overlap if the left leg happens to be translated over onto the right leg. Instead of finding the MAP estimate of Eq.1, we generate samples from the posterior (using the efficient method of [6]), and throw away those samples that violate our global constraints. We generate 2000 samples per image. To find configurations where the left and right legs look similar, we add the disparity in leg appearance (as measure by the L_2 distance between color histograms) to the negative log probability of the sampled configuration. To force left and right legs to be far apart, we discard samples where leg endpoints are within a distance d of each other, where d is the width of the torso. We finally keep the sample with the lowest cost.

Segmentation score: Given an image with a laterally walking person, the procedure above tends to correctly localize the limbs of the figure. But it does not perform well as a people detector; it fires happily on textured regions. We add a region-based cue to the detection score. We can interpret the recovered figure as a proposed segmentation of the image (into person/non-person pixels), and directly evaluate the segmentation [14] as the final detection cost. Rather than use a standard segmentation measure, we adopt a simpler approach.

We build classifiers (in RGB space) for each limb, as described in Sec.4. For each limb classifier, we create a test pool of limb pixels (from inside the corresponding limb mask) and background pixels (from identically-sized rectangles flanking both sides of the true limb). We then classify the all test pixels, and define the cost of the segmentation to be the total number of misclassified pixels. Note this strategy would not work if we used classifiers with high VC dimension (a nearest neighbor classifier always returns 0 errors when training and testing on the same data). Restricting

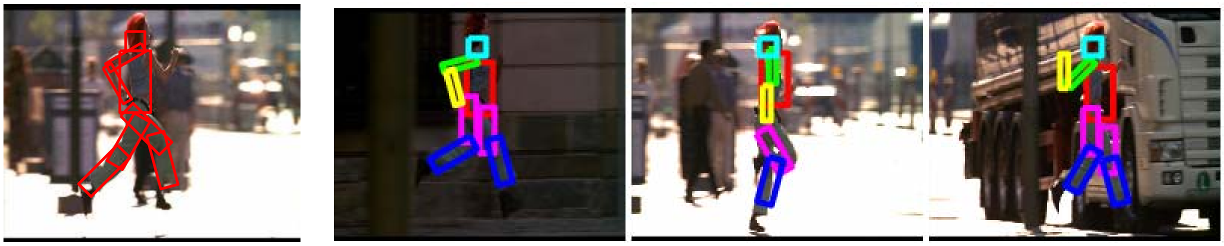

Figure 3. *We show tracking results for a sequence with large changes in the background. On the* **left***, we show the frame on which our walking pose detector fired. By learning discriminative limb appearance models from that* single *frame, we are still able to track the figure when the background changes (***right***). This suggests that our logistic regression model is quite generalizable.*

ourselves to a near-linear classifier (such as quadratic logistic regression) seems to address this issue. We threshold this final segmentation score to obtain good stylized-pose detections.

4. Discriminative Appearance Models

Since our person detector localizes a complete person in a single frame, we know both the person pixels *and* the non-person pixels. This suggests we can build a discriminative model of appearance. We assume each limb is (more or less) constant colored, and train a quadratic logistic regression classifier. We use all pixels inside the estimated limb rectangle as positives, and use all non-person pixels (not inside any limb mask) as negatives. Our appearance model for each limb is a quadratic surface that splits RGB space into limb/non-limb pixels (Fig.2). Recall our set of limbs are the head, torso, upper/lower arm, and left/right upper/lower leg. We fit one model for the upper leg using examples from both left and right limbs (and similarly for the lower leg). We find our appearance models to be quite generalizable (Fig.3).

5. Tracking as Model Detection

We track by detecting the limb appearance models (built from Sec.4) in other frames (both prior to and after the walking pose detection). We use the same pictorial structure framework as Sec.3, but use the *appearance model to compute the image likelihood* $\Pr(Im(P_i))$ (as opposed to a chamfer edge template). We score the likelihood that a limb is at given configuration by counting the number of misclassified pixels for that configuration. For example, given a torso rectangle at a certain position and orientation, we count the number of non-torso pixels inside that rectangle and the number of torso pixels inside rectangles flanking either side. To compute this score, we first classify each pixel to obtain a limb mask (Fig.2). We then perform two convolutions with rectangular masks; one to compute the number of limb pixels misclassified as background and another to compute the number of background pixels misclassified as limb (and add the 2 together appropriately).

We find the MAP estimate of Eq.1 by dynamic programming (working in log space for convenience). With our improved image likelihoods, we no longer need to restrict our shape potentials to lateral walking poses; we enlarge the interval bounds for our shape model to respect reasonable joint limits. However, this introduces a difficulty; the estimated left and right legs tend to overlap, since they are both attracted to regions with high likelihood. A related problem is that in some poses arms and legs are occluded.

Occlusion: We must take care to prevent the estimated configuration to be drawn toward an awkward/incorrect pose just to minimize the image likelihood of an occluded limb. Rather than building an explicit model of occlusion, we found the following simple strategy to be effective. We observe that in almost all poses, the head, torso, and one upper/lower leg is visible. We create a pictorial structure model just with these limbs, and directly find the MAP estimate. This tends to result in good localizations (even when an arm occludes much of the torso) because of the quality of our limb masks and likelihoods. We search for the remaining limbs (again using the pictorial structure framework) holding the torso fixed at the estimated location. When searching for a new leg, we mask out the already-estimated leg from the new leg mask; this prevents the left and right leg from lying on the same image region. The same approach can be used for estimating left/right arms. We finally disregard those limbs that fall below a detection threshold. We show results for our running example in Fig.4.

Multiple People: In general, we must account for multiple people in a video. Given a set of walking-pose detections from across the video, we need to automatically establish the number of different people that are actually present. For each detection, we learn a *generative* appearance model (by fitting a gaussian in RGB space for each limb). This returns a vector of RGB values. We cluster these vectors to obtain sets of people models with similar appearance. We use the mean-shift clustering procedure since we do not know the number of people in a video *a priori* [4]. After obtaining clusters of similar looking people, we use positive and negative examples from across the cluster when training the logistic regression for each limb appearance. We then use these people models as described in the next two paragraphs.

Multiple instances: If a video has multiple people that look similar, walking-pose detections for different people

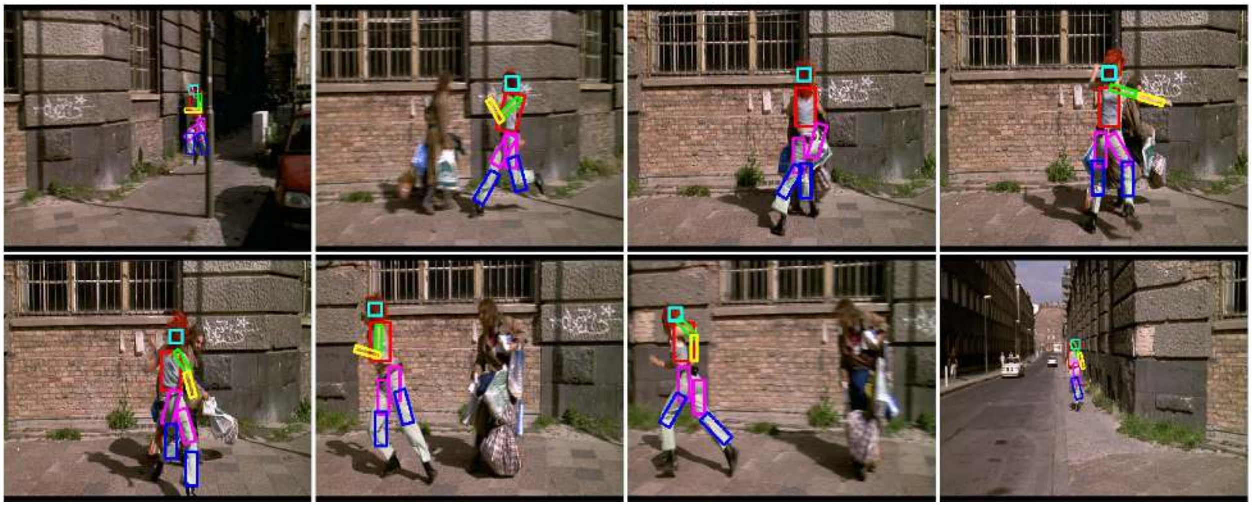

Figure 4. *A sequence from 'Lola' of Lola running around corner and bumping into a character while undergoing extreme scale changes. Note the character is wearing bulky clothing and so our person detector has no hope of finding him. Our initial walking detector is run at a single scale; once we learn an appearance model (as shown in Fig.2), we track over multiple scales by searching an image pyramid at each frame.*

might cluster together (consider a video of a soccer team). In this case, when searching for people using an appearance model, we might need to instance that model multiple times in a single frame. We do this by first finding the best matching pictorial structure, as described previously. We then mask away those pixels covered by all the estimated limbs, and find the best match in the remaining pixels, and repeat again. We repeat until the posterior falls below a threshold.

In general, we will have multiple appearance models, each possibly instanced multiple times. For each model, we independently find all instances of it in a frame. Many models will compete to explain the same or overlapping image regions. We use a simple greedy assignment; we first assign the best-scoring instance to the image pixels it covers. For all the remaining instances that do not overlap, we find the best-scoring one, assign it, and repeat.

6. Results

We have run our tracker on hundreds of thousands of frames. Our dataset includes the feature length film 'Run Lola Run', an hour of footage of a local park, and long sequences of sports footage. This presents an interesting challenge in evaluating our system; we cannot mark up every frame. Our system consists of 2 main components; (a) we run a stylized pose detector in each frame to find people and then (b) use the learned appearance from the detection to find people in other frames. We evaluate each component separately.

Lateral-Walking Pose Detection: Evaluating our walking pose detector is a difficult task by itself. Labeling false positives is straightforward; if the detector finds a person in the background, that is incorrect. But labeling missed detections is difficult because our detector is not trying to detect all people; only people in certain configurations.

In the case of 'Lola', we can exploit *shots* (sequences where the camera is filming continuously) in our evaluation. We label each shot (as determined by a histogram-based shot detector) as containing a full-body figure or not. We define the score of a shot to be the best score from our walking detector on its set of frames; the implicit assumption is that if a shot contains a person, our walking pose detector will fire at some point. In Fig.5, we show precision-recall curves for the task of detecting shots with full-body figures using our walking detector. We do quite reasonably at high-precision/low-recall regions of the graph, and significantly better than chance. We show detections returned by our detector from the 'Lola' frames in Fig. 9.

For the park sequence, there are no natural shots, making recall measurements awkward to define. Since this video contains multiple people (many of which look similar), we cluster the appearance models to obtain a set of different-looking people models, and then use them to search the entire video. In this case, we would like to select a detector threshold for our walking detector where most of the accepted detections are correct and we still accept enough different looking models to capture most people in the video. As such, we plot precision versus number of appearance model clusters spanned by the accepted detections in Fig.6. We do quite reasonably; we can find most of the different looking people in the video while still maintaining about 50% precision. In other words, if our appearance model detection was perfect and we were willing to deal with 50% of the tracks being junk, we could track all the people in the video. We show the top 5 detections returned by our walking detector in Fig.10.

We look at the ability of our detector to find peo-

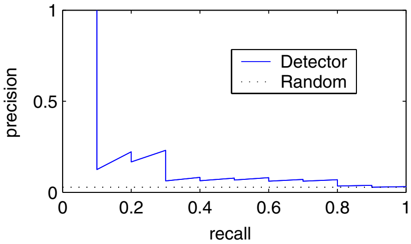

Figure 5. *Precision recall curves for our walking pose detector on shots from 'Lola'. At low-recall, high-precision regions, our detector performs quite well. Many running shots of 'Lola' show her running toward or away from the camera, for which our detector does not fire. If we use the model learned from one shot across the whole video, we would expect to do significantly better (since the central figure never changes clothes!)*

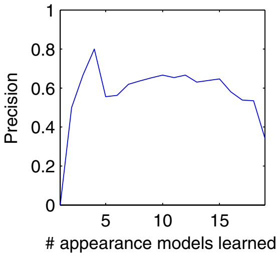

Figure 6. *Precision curves for our walking pose detector on 30000 frames from the park sequence. We define a correct detection to occur when* all *the limbs are correctly localized. Recall is awkward to define since we are not trying to detect people in every frame; rather we want to fire at least once on different looking people in the sequence. We obtain a set of models of different looking people by clustering the correct detections of our walking detector (which we validate by hand). For a given detector threshold, we can explicitly calculate precision (how many of the reported detections are correct) and the number of different models spanned by correct detections. As we lower the threshold, we span more models.*

ple performing unusual activities in Fig.13. Perhaps surprisingly, we are able to find frames where our detector fires. This means our algorithm is capable of tracking long and challenging sports footage, where people are moving fast and taking on extreme poses. We show results on a baseball pitch from the 2002 World Series and Michelle Kwan's medal winning performance from the 1998 Winter Olympics.

Appearance model detection: In the second phase of our algorithm, we track by detecting the learned appearance models in each frame. Since we are implementing our tracker as a detector, we evaluate the final track by looking at precision and recall rates. For 'Lola', we evaluate performance on two shots (shown in Fig.3 and Fig.4) where the walking detector correctly fired. Our algorithm learns models from the single detected frame, and uses them to detect configurations in other frames of the shot. For a random set of 100 frames from each shot, we manually mark correct detections of the torso, arm and leg. We define a correct torso detection to occur when the majority of pixels covered by the estimated torso can be labeled as a torso.

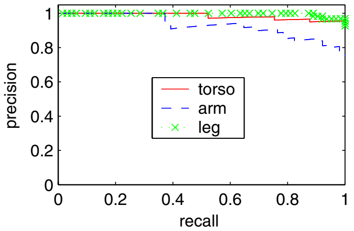

Figure 7. *'Lola' precision/recall curves for limb detection in the appearance model-based person detector. We manually mark correct torsos, arms, and legs, for shots where our walking detector correctly fired. Each limb has its own segmentation score, obtained from its corresponding logistic regression limb mask. We threshold the score to obtain the above curves. We do almost perfect torso and leg detection, and near perfect arm detection. This is because the central character is visually distinctive, and that the discriminative appearance model captures this fact.*

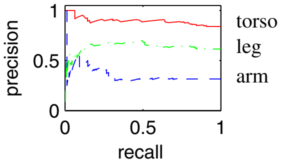

Figure 8. *Precision/recall curves for limb detection for the park sequence, using the same procedure as Fig.7. Our torso detection is quite good, but our arm detection is poor because arms are small and fast, making them hard to detect. Our performance is good considering the difficulty of this dataset; the video has weak color quality, there are strong shadow effects, and many small figures are interacting, performing quick motions.*

We define a correct arm detection to occur when most of the pixels masked by the estimated lower arm can be labeled as an upper or lower arm (and likewise for leg detections). For the arm and leg scoring, we only look at the first arm and leg found (from the two-step MAP estimation procedure described in Sec.5). We generate the final precision/recall curves by thresholding the segmentation score for the torso, lower arm, and lower leg limbs. Looking at Fig.7, we do extremely well; torsos and legs are found almost perfectly, with quite good performance on the arms as well. These results imply that if the walking pose detector performs ideally, than we can *track with near perfect accuracy*. These (startlingly) good results are really an artifact of the way films are photographed; characters are often intentionally dressed to be visually distinctive. A wonderful example is Lola's hair (Fig.3); once our head model learns to look for something red, it is essentially impossible for it to loose track (since there is nothing red in the background).

We use a similar criteria for 200 random frames from the park sequence in Fig.8. Here, we consider a detection to be correct if it fires on any person in the image; we do not look for consistency of detections from one frame to the next. Since many people in the sequence look like each other, we need additional constraints to pull out individual tracks (such as motion). Our results are also good, though not near the performance we achieve on 'Lola'. We do quite well

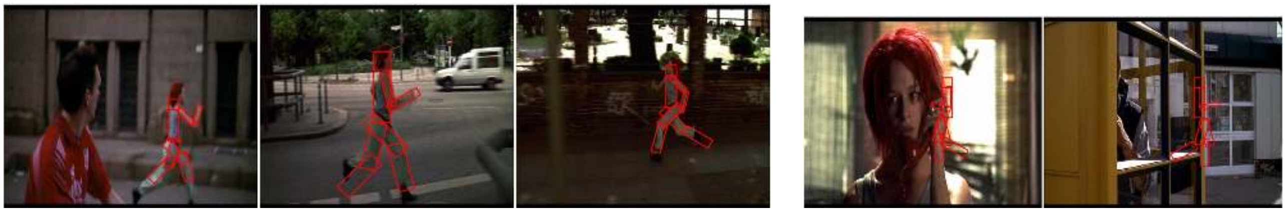

Figure 9. *The top detections for our walking pose detector on 30000 frames from 'Lola'. On the* **left**, *we show the top 3 correct detections (at positions 1, 9, and 13 in a ranked list). On the* **right**, *we show the top 2 false positives (at positions 2 and 3). Even though in the second correct detection, one leg is incorrectly localized, we still learn the correct appearance model since we use both leg masks to train the logistic regression classifier.*

at detecting torsos, with about 90% accuracy, while arms are still difficult because they are small and move fast. This data is hard for many reasons; the video is washed out, there are significant shadow effects, there are many small people interacting with each other (Fig.12).

We obtain quite good results (similar to Lola) for the sport sequences from Fig.13, though we omit quantitative evaluation for lack of space. The observation that movie characters are visually distinctive applies to athletes as well; teams tend to wear uniforms that are not the same color as an athletic field. This suggests that our algorithm can be used to automatically track interesting and historic sports footage.

7. Discussion

We present a simple and effective algorithm for tracking on extremely large datasets. We also present and apply a methodology for evaluation on a large scale. Our algorithm is based on two observations. First, detecting and tracking objects in video can be done opportunistically; we choose to find people in certain stylized poses that are easy to detect and build appearance from. One could also look for stylized motions over short frames; such a detector might perform better since it pools information from across frames. Secondly, discriminative appearance models learned from a few frames can discriminate the object in other frames. Discriminative features for tracking are not new [3], but by learning them from select frames where we trust our detections, we make them quite powerful.

Acknowledgments

This work was supported by NSF award no. 0098682 and by Office of Naval Research grant no. N00014-00-1-0890, as part of the MURI program. D. R. is supported by a NSF fellowship.

References

[1] A. Agarwal and B. Triggs. Tracking articulated motion with piecewise learned dynamic models. In *ECCV*, 2004.

[2] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR*, pages 8–15, 1998.

[3] R. Collins and Y. Liu. On-line selection of discriminitive tracking features. In *ICCV*, 2003.

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

[5] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, 2000.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, 2000.

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

[8] S. Ioffe and D. A. Forsyth. Human tracking with mixtures of trees. In *ICCV*, 2001.

[9] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *ICCV*, pages 34–41, 2001.

[10] K. Mikolajcyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, 2004.

[11] K. Mikolajczyk, R. Choudhury, and C. Schmid. Face detection in a video sequence - a temporal approach. In *CVPR*, 2001.

[12] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. In *PAMI*, 2001.

[13] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *ECCV*, 2002.

[14] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004.

[15] K. Okumna, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

[16] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. In *CVPR*, 2003.

[17] K. Rohr. Incremental recognition of pedestrians from image sequences. In *CVPR*, pages 9–13, 1993.

[18] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse picture of people. In *ECCV*, 2002.

[19] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, 2000.

[20] L. Sigal, M. Isard, B. Sigelman, and M. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *NIPS*, 2003.

[21] C. Sminchisescu and B. Triggs. Building roadmaps of local minima of visual models. In *ECCV*, 2002.

[22] Y. Song, X. Feng, and P. Perona. Towards detection of human motion. In *CVPR*, pages 810–17, 2000.

[23] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *ECCV*, 2002.

[24] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, 2003.

[25] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, 2003.

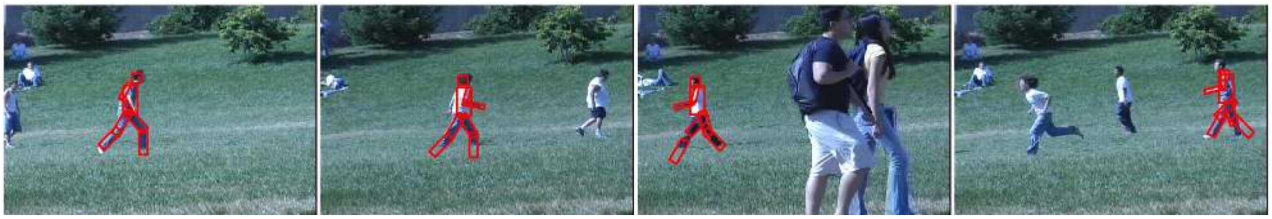

Figure 10. *The top 5 people detections for our walking pose detector on 30000 frames from an unscripted park video. Even though multiple people are frequently interacting (see Fig.12), our walking pose detector tends to fire on frames where the figures are well separated, since they have a better detection score. Even though most detections are not perfect, we find that the logistic model learned can compensate for small errors. Note we do not use any form of background subtraction.*

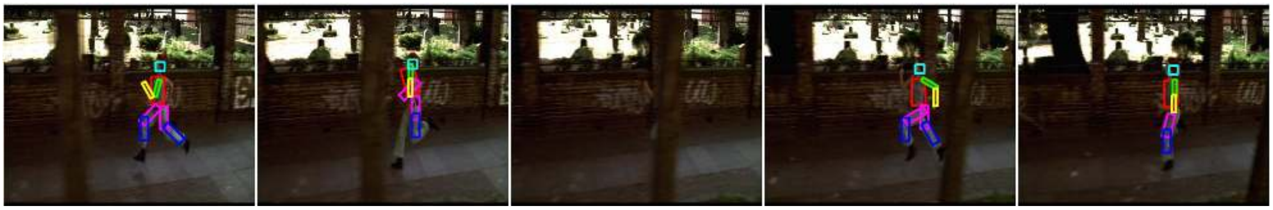

Figure 11. *Automatic tracking of one of the shots from the 30000 frame 'Lola' sequence. Even though the figure is performing a fast running motion, the arms and legs are quite well localized. The track recovers from the full occlusion of the telephone pole and handles self-occlusion as well.*

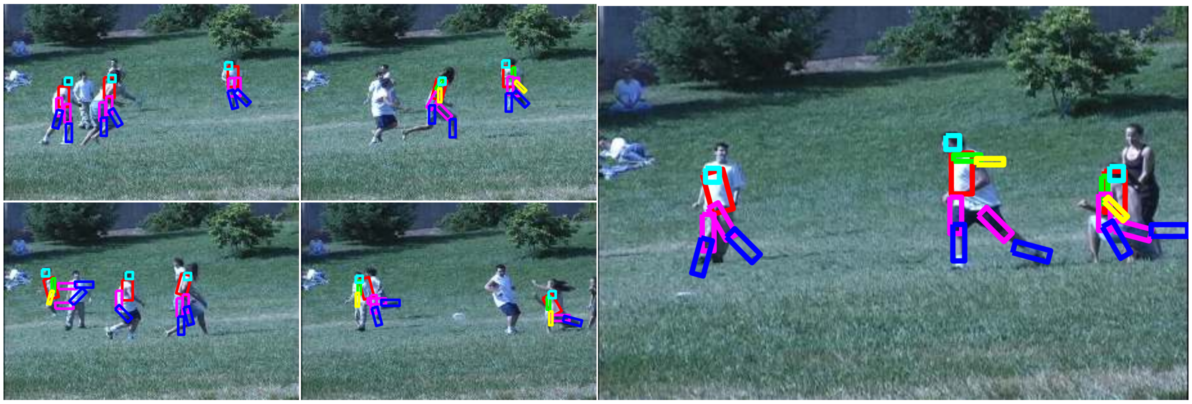

Figure 12. *Automatic tracking of a sequence from the 30000 frame park sequence. This sequence is harder than our 'Lola' sequence; we have poor color resolution, many small figures are in shadow and many are occluding each other while performing fast motions. We still obtain good detections and reasonable localization of arms and legs. Since many of the learned models look similar, we do not try to disambiguate instances from frame to frame. One might do that using motion constraints.*

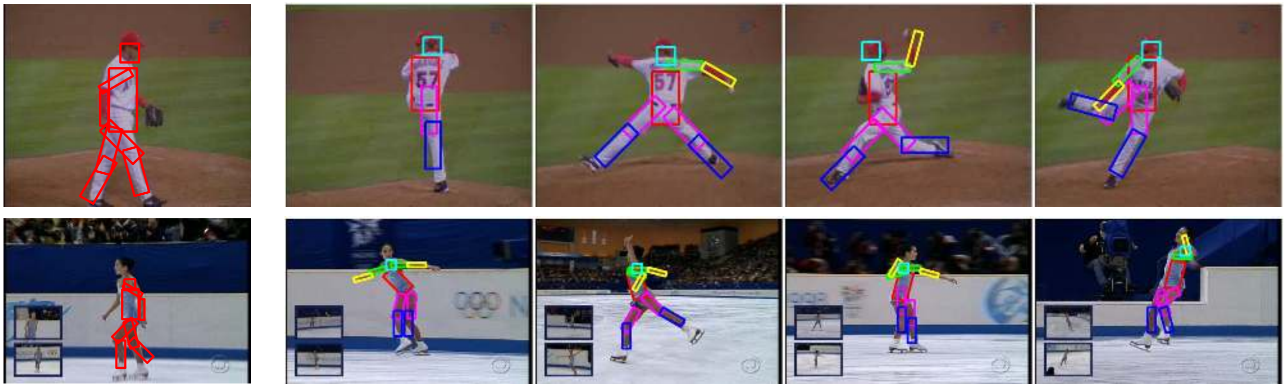

Figure 13. *Our automatic tracker on legacy sports footage with fast and extreme motions. We show the walking detection on the* **left** *and example frames from the final track (obtained using the learned appearance) on the* **right**. *On the top, we use a 300 frame sequence of a baseball pitch from the 2002 World Series. On the* **bottom**, *we run our algorithm on the* complete *(7600 frame) medal-winning performance of Michelle Kwan from the 1998 Winter Olympics. For each sequence, we run our walking pose finder and use the single frame with the best score (shown on the* **left***) to train the logistic models. In the skating sequence, the walking detection does not have a correctly localized head. The tracker learns an appearance model for the wrong image region, and this same mistake is repeated in the tracked frames. We still however obtain reasonable kinematic estimates.*