Silhouette Lookup for Automatic Pose Tracking

Nicholas R. Howe
Computer Science
Smith College
Northampton, MA 01063

Abstract

Computers should be able to detect and track the articulated 3-D pose of a human being moving through a video sequence. Current tracking methods often prove slow and unreliable, and many must be initialized by a human operator before they can track a sequence. This paper introduces a simple yet effective algorithm for tracking articulated pose, based upon looking up observed silhouettes in a collection of known poses. The new algorithm runs quickly, can initialize itself without human intervention, and can automatically recover from critical tracking errors made while tracking previous frames in a video sequence.

1. Introduction

Researchers have worked for decades towards the goal of a computer system that can track the articulated pose of a moving human being from monocular video input [15, 7]. Such a system would immediately enable applications in security, ergomonics, human-computer interaction, and many other fields. Yet a recent study concluded that none of the automated tracking methods tested could successfully track a moderately difficult example [6]. Recovery from tracking errors therefore deserves more than the scant research attention it has received [11] to date.

This paper develops an approach to pose tracking based upon silhouette lookup, hereafter referred to as *SiLo tracking*. This approach offers significant advantages over currently popular methods using parameter optimization and particle tracking algorithms. The SiLo tracker described in Section 2 requires no human input for initialization. Even if it makes grave errors during difficult sections of a video, it can automatically recover to track the correct pose on subsequent frames. Furthermore, although the implementation described here is not optimized for speed, it invites significantly faster implementations than approaches based upon optimization and particle tracking.

Several developments contribute to enable these advances. The many-to-one silhouette-to-pose relationship has in the past proved a barrier to the development of silhouette-based trackers. The new technique exploits temporal continuity to choose the best hypothesis among multiple candidate poses at each frame, via a Markov chain formulation. Relieved of the burden of finding the perfect match, simple yet effective metrics make feasible the rapid retrieval of candidate silhouettes. Finally, smoothing and optimization based upon polynomial splines ensure that the tracked output forms a plausible human motion.

The sections that follow describe each of these contributions in more detail. Section 2 describes the SiLo tracking algorithm and places it in the context of previous work. Section 3 describes experimental results using the algorithm. Section 4 concludes with an analysis of the approach's strengths and weaknesses, and a discussion of possible future work.

2. SiLo Tracking

The algorithm described below takes as its input raw video from a fixed viewpoint, assumed for simplicity to contain a single human being entirely within the camera frame and unoccluded by other objects. (Multiple subjects, partial visibility, and camera motions can all be addressed but fall beyond the scope of this paper.) For each frame F_i in the input video, it produces as output a vector of parameters Θ_i, specifying the pose of an articulated model of the human body for that frame. Data from the input video pass through multiple stages during generation of the output pose: background subtraction and silhouette extraction, silhouette lookup, Markov chaining, and smoothing. The sections below describe each of these stages.

2.1. Silhouette Extraction

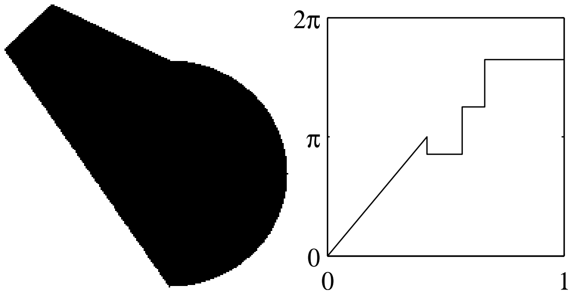
A number of cues distinguish the human being in a video from the background. These may include appearance, motion, and heat emission (if infrared cameras are available [5]). The experiments below use motion segmentation because there exist well-studied techniques that are straightforward to apply under appropriate conditions (i.e., static camera and background). Any of a number of techniques may be used to model the background and perform background subtraction [12, 8], including some that can identify human subjects moving against dynamic backgrounds [25]. The work presented here uses a graph-based static estimation of the background [9], generated by robustly measuring the mean and deviance of each pixel over time while excluding outliers. In applications where temporal batch processing is impractical, one of the dynamically updated background models cited above could be used instead. In either case, comparing the background model with each frame of the video yields a set of pixels that deviate strongly, presumably due to occlusion by the human subject. Simple morphological operations applied to this set of pixels clean up small errors and yield the observed silhouette for that frame. If the set of pixels output is disjoint, then the subsequent processing steps use the largest connected component.

2.2. Silhouette Lookup

Successful silhouette lookup requires two ingredients: a knowledge base of silhouettes associated with known poses, and an efficient heuristic for comparing the known silhouettes with those observed in the video input. To populate the knowledge base, this work uses data from the CMU Motion Capture Database artificially rendered from different viewpoints (36 parallel projections taken at 10° intervals around the subject). Although pruning of the stored pose library will probably be necessary in a production system, the current work simply stores all available data. For retrieval of the stored silhouettes, several heuristic similarity measures have been tested, including the turning angle metric and the chamfer distance. Although both work individually, a combination of the two (using summed retrieval status [2]) appears most effective.

The turning angle metric is sensitive to the length and orientation of extended limbs, and has been shown to correlate well with human notions of shape similarity [18]. In brief, the turning angle metric measures the integral of the difference between two normalized functions, where each function is derived from a silhouette by taking the tangent trace made during one complete circuit around the silhouette's border (see Figure 1). The turning angle metric is not rotation invariant; its use here assumes that the vertical axis in physical space coincides with the y axis in the input video. The tangent trace begins at the highest point of the silhouette (typically the head) and proceeds clockwise. The stored silhouettes average around 150 point samples around the silhouette boundary, from which individual comparisons can be computed rapidly.

The chamfer distance compares two sets of pixels (the boundaries of the silhouettes, in this case) by taking the sum of the distances from each pixel in one set to the nearest



(a) (b)

Figure 1: Turning angle representation for a simple shape (a). For this figure, the perimeter trace (b) starts at the bottom of the curved section and proceeds counterclockwise.

pixel in the other set.

$$H(S_1, S_2) = \sum_{p \in S_1} \min_{q \in S_2} d(p, q) \tag{1}$$

Note that this is related to the Hausdorff metric, which takes a maximum rather than a sum. This chamfer distance can also be computed rapidly using a chain-code representation of the boundary.

Using the selected comparison heuristic, each silhouette extracted from the input frames identifies a set of silhouettes in the knowledge base that lie within some threshold of similarity. The poses associated with the selected silhouettes become the candidates in the next processing phase, Markov chaining. Because the quantity and quality of the best matches varies widely at different points in a video clip, it is helpful also to establish minima and maxima on the number of selections k_i, such that $k_{min} < k_i < k_{max}$ at all frames. The experiments described below use between 100 and 500 selections per frame.

2.3. Markov Chaining

Because the mapping of poses to observed silhouettes is many-to-one, and the retrieved poses are only approximate matches to the actual observations, silhouette lookup returns multiple possible poses for a single observed silhouette. Markov chaining exploits the temporal dependency of human motion to weed out unlikely pose sequences, retaining the single chain of poses (one for each frame) that simultaneously maximizes both the per-frame match to the observations and the temporal similarity between successive frames. The problem may be stated in terms of error minimization, with the goal of minimizing the function E stated below.

$$E = \sum_{i=0}^{n} M\left(\Theta_i, S_i\right) + \lambda \sum_{i=1}^{n} \Delta\left(\Theta_i, \Theta_{i-1}\right) \tag{2}$$

Here n represents the number of frames in the video, M represents the matching error between the silhouette corresponding to the pose parameters Θ_i and the observations in a given frame, Δ represents the motion difference between two different sets of pose parameters, and λ serves as a weighting factor.

This stage requires a more sensitive silhouette comparison than the turning angle provides, so M uses a symmetric version of Equation 1 applied over all the pixels in the two silhouettes:

$$\begin{aligned} \text{Let } P_{\Theta_i} &= pixels(render(\Theta_i)) \\ \text{and } P_{S_i} &= pixels(S_i); \\ M(\Theta_i, S_i) &= H(P_{\Theta_i}, P_{S_i}) + H(P_{S_i}, P_{\Theta_i}) \quad (3) \\ & \quad (4) \end{aligned}$$

The choice of motion difference function Δ offers an array of possibilities depending upon the degree of physical realism desired. The simplest functions merely reward solutions that change as little as possible from one frame to the next, perhaps in terms of each joint's angular parameters weighted by the mass and moment of inertia of the affected portions of the body. A more physically realistic criterion would measure the change in linear and angular momentum of body parts in 3-D space, or perhaps the power required to transition between frames. Unfortunately, implementing any criterion based upon change in velocity or momentum requires the use of a stochastic chain with two-state memory in place of a Markov chain. Fortunately, the simpler format yields excellent results, and the extra computation of the more physically plausible models appear unnecessary.

As noted above, using a simplified Δ makes the sequence of frame poses into a Markov chain, where the likelihood of a particular pose in frame i depends only upon the pose assigned for frame $i-1$, and not on the pose in any preceding frames. Efficient dynamic-programming algorithms exist for finding the minimum-energy solution of Equation 2, given the finite set of k_i possible solutions at each frame generated during silhouette lookup. This minimum-energy solution serves as the basis for further smoothing and optimization.

2.4. Smoothing and Optimization

Markov chain minimization produces a solution that is consistent both from frame to frame and with observations made at each frame. However, it is still made up of poses retrieved from the knowledge base, which typically cannot express the true solution exactly. A rendering of the proposed solution may appear jerky and occasionally inconsistent with the input video where no pose in the knowledge base exactly matches the true pose. Two final processing steps address these concerns.

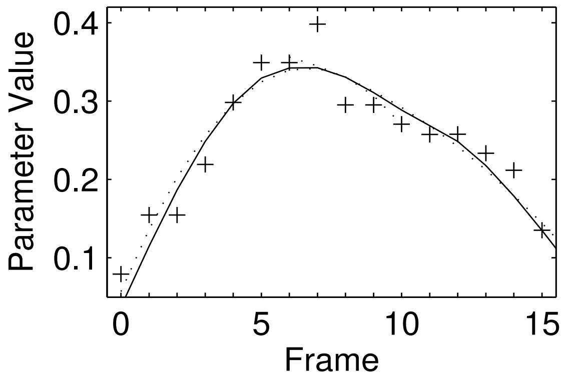


Figure 2: Sample smoothing of single parameter dimension $\theta_j(i)$. The final curve (solid line) smooths out the individual Markov chain points (crosses). Two of the overlapping spline curves are shown (dotted lines).

The first step eliminates jerkiness through a temporal smoothing of the Markov chain solution. The vector Θ_i of pose parameters at frame i can be decomposed into its individual components, each viewed as deriving from a one-dimensional function of the frame number $\theta_j(i)$ plus some error $\epsilon_j(i)$. Assuming that the underlying component functions θ_j should be smooth, $\theta_j(i)$ can be modeled as a series of overlapping polynomial splines (see Figure 2). Taking $\Theta'_i = (\theta_1(i), \theta_2(i), ..., \theta_m(i))$ yields a smoothed solution. The experiments described in this paper build the $\theta_j(i)$ using quadratic splines of eleven frames in length and smoothly overlapped by five frames. Given a frame rate of 30 Hz, this enforces smoothness over a timescale of about one-third of a second. (Note that the use of splines in this capacity depends upon a careful choice of the joint angle representation, to ensure that the range of motion for each individual joint does not include any singularities.)

The result of the process described above may still not exactly match the observed silhouettes in all places, depending upon the density with which poses in the knowledge base cover the range of poses observed. The faithfulness between the observations and the proposed solution may be increased through parametric optimization (currently implemented via Matlab's *fminsearch* function). This final step takes much longer than the preceding ones, so applications (such as activity/gait recognition) not needing extreme precision may choose to forego it. To maintain the smoothness of the solution, the optimization proceeds on the parameters of the m polynomial splines (created during the smoothing process) that generate a smoothed block of 11 frames at once. Equation 2 gives the energy criterion.

2.5. Related Work

A large body of work on pose tracking precedes this paper, dating back to the early 1980's [15, 7]; a 2001 survey lists many recent contributions [13]. Only the most relevant works can be cited here due to space limitations. In particular, this section will focus on other research into full *3-D articulated pose reconstructions from monocular video input*. Recent efforts in this area have used models of probable poses and motions and sophisticated optimization routines together with particle-based tracking algorithms and motion models [10, 20, 22]. As mentioned in the introduction, these present difficulties with initialization and error recovery, and can be slow to operate. There has been some prior interest in using silhouettes for pose recognition [4, 17, 21], but the reported results do not present completed 3-d reconstructions of video clips. One exception does include results for a single very short (19-frame) sequence [14]. The latter work is similar in spirit to that described in this paper, using edge images instead of silhouettes to retrieve poses from a library. It applies a completely different retrieval metric (shape context [3]) and does not address the frame-to-frame issues considered herein. Without further examples of its performance, it is difficult to compare with the current work.

Recent research has also looked at the use of silhouettes for tracking hand pose [24, 1]. The hand-tracking work differs from the results presented herein by making the assumption that only a small number of key poses (e.g., sign-language symbols) need be precisely identified, with intervening frames filled in via interpolation. By contrast, this work uses a knowledge base with broad coverage to retrieve the best matches for *every* frame, allowing the motion to develop arbitrarily without having to pass through key poses. The large number of degrees of freedom in the human body inhibits identification of key poses. However, key poses have been applied to full-body pose estimation in certain limited domains such as the analysis of tennis serves [23].

Others have looked at alternate approaches to the problem of automatic initialization. Ramanan and Forsyth first identify clusters of candidate features that might indicate the presence of a person, and then track those features through the video [16]. This avoids the use of background subtraction, but introduces other assumptions about the appearance of the tracking subjects (e.g., body parts have coherent appearance). Their work also differs in producing only two-dimensional information on body part location, while the use of silhouette lookup to make three-dimensional inferences lies at the heart of this paper's contribution.

The use of silhouette lookup here shares some ideas in common with recent work by Shakhnarovich et. al. on lookup-based approaches to pose estimation [19]. Their work uses edge features rather than silhouettes, applied to the rapid estimation of upper-body pose from single images rather than videos. They use parameter-sensitive hashing to achieve sub-linear retrieval speeds, and increase the precision of the retrieval prediction, by interpolating between the top retrieval results. Both of these ideas should prove useful with silhouette lookup, although the Markov chaining and smoothing steps achieve results similar to those of the interpolation process.

3. Experimental Results

Quantitative evaluation of 3-d pose reconstruction is notoriously difficult. This section shows the results of experiments with the methods described in Section 2 on multiple sample video clips of varying degrees of difficulty. *Walk* shows the subject walking from right to left, while *Circle* shows the same subject walking in a circle. Both clips were generated and used to test other tracking algorithms [20], although lack of a ground truth prevents any quantitative comparisons. A third clip, *Dancer*, shows a ballet dancer performing a short routine. The turning of the dancer's body in this clips makes it difficult for many tracking algorithms to follow.

Figures 3-5 summarize the tracking results for the trial clips. The system tracks *Walk* well, making no significant errors. On the other two clips the system tracks the bulk of the sequence with high fidelity, but tracking failures appear at several points. Analysis of the failures reveals two distinct modes: ambiguity problems (where the silhouette cannot distinguish between two or more plausible solutions) and retrieval problems (where lookup in the knowledge base returns no poses matching the actual motion). The discussion below examines each in turn.

3.1 Error Analysis

Ambiguity problems appear in the latter third of *Circle*: the tracked motion and the true motion suffer from a right-left reversal. This cannot be avoided in any system based solely upon silhouette measurements; mathematically, a simultaneous left-right inversion of the pose and reflection about the line-of-sight axis produces an identical silhouette, as illustrated in Figure 6. Similar ambiguities cause problems in the *Dancer* clip when the dancer's body turns. The tracked silhouette matches the observations, but close inspection shows that in about half the cases the tracked direction of rotation does not match reality. Ultimately the use of additional cues beyond silhouette matching (such as optical flow) should control this source of error.

Retrieval failure appears in the *Circle* clip around frame 30, as the subject turns away from the camera. Close investigation of the frames immediately following the point of error indicates that none of the poses returned during the

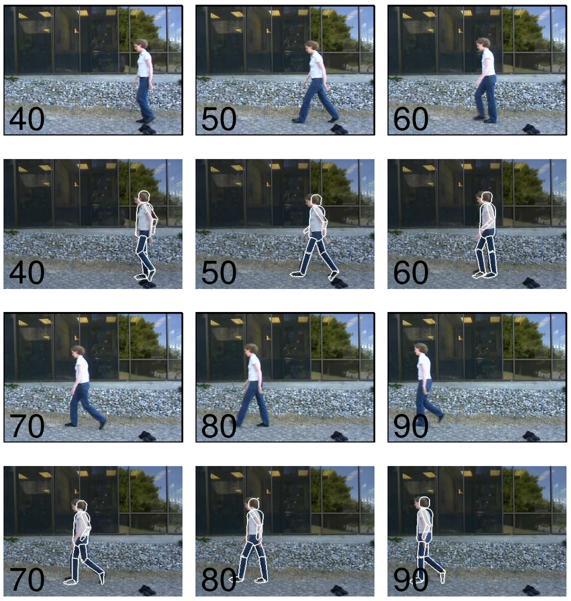


Figure 3: *Walk* clip and its reconstructed pose. The corresponding videos may be viewed in the supplementary materials.

retrieval step are close matches for the actual pose. Indeed, the next 40 frames or so consist of poses for which the retrieval metric used cannot adequately distinguish the correct pose among a multitude of incorrect poses with similar silhouettes. There are simply too many competing candidates. The recovered pose track for this period is correspondingly confused. However, around frame 80 the tracker miraculously recovers: a sequence of frames provide good matches, and the tracked motion closely resembles the actual motion once more. The spontaneous recovery shows that the system can indeed regain the correct track even after essentially losing it completely.

3.2 Behavior Analysis

For any algorithm that processes frames in batch, one may reasonably ask how much the processing mode influences the solution. In particular, how far down the Markov chain does a choice made at one frame show any effect in practice? The experiments in this section investigate this question empirically for the *Walk* clip, and find that the answer in most cases is fewer than ten frames.

Figure 7 shows the results of an experiment designed to test how quickly the Markov chain solution converges from an erroneous initial starting point, chosen at random from the pose library and repeated over 1000 trials. The plot shows that after only ten frames, all starting points converge on two fairly stable solutions, and after 45 frames all reach

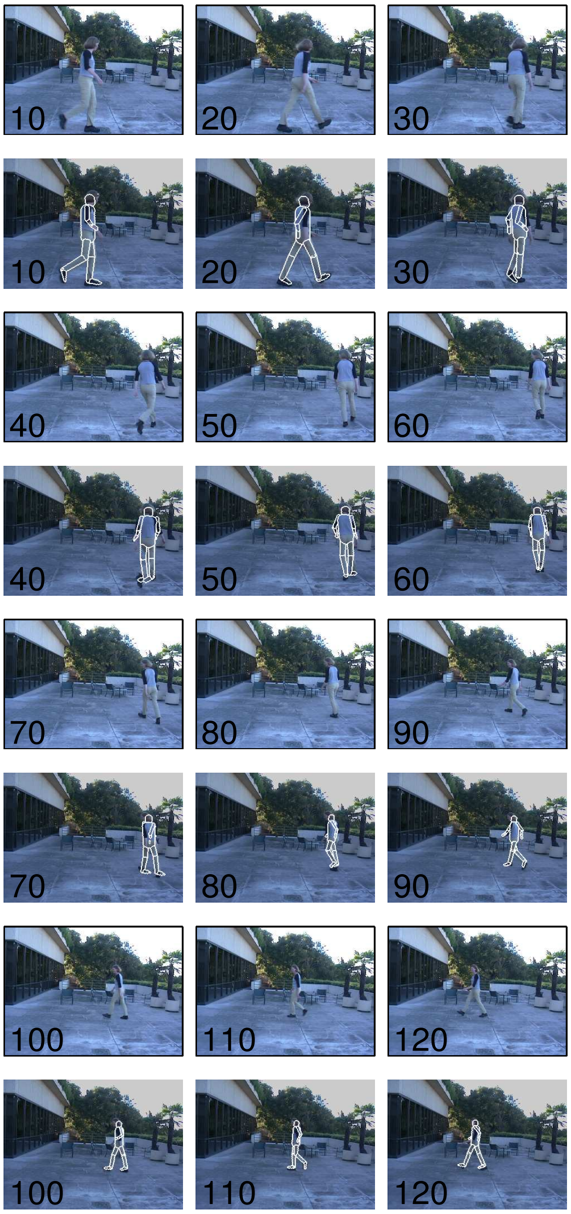


Figure 4: *Circle* clip and its reconstructed pose. The corresponding videos may be viewed in the supplementary materials.

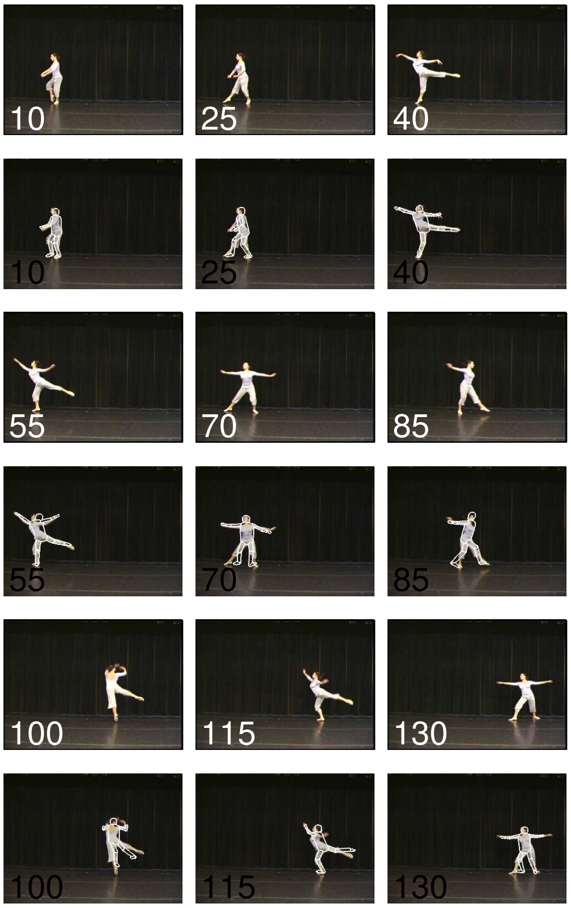


Figure 5: *Dancer* clip and its reconstructed pose. The corresponding videos may be viewed in the supplementary materials.

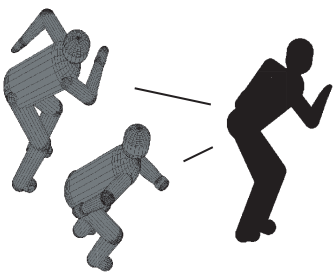


Figure 6: Right-left ambiguity for silhouettes. The two poses on the left produce exactly the same silhouette when viewed from the side under orthographic projection.

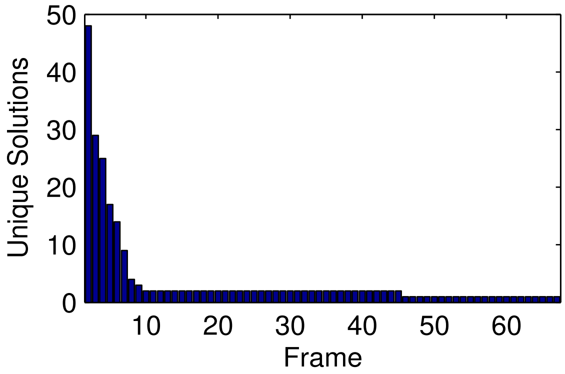


Figure 7: Convergence of the SiLo tracker under divergent starting conditions. The bars show the number of unique solutions on the *Walk* clip decreasing rapidly over time, despite the initial frame's constraint to a randomly chosen pose on each of 1000 trials.

the same solution, regardless of initial conditions.

Given that the Markov chain solution converges quickly regardless of the starting point, one might also ask how the endpoint of the chain can affect the final result. Although most of the experiments described in this paper evaluate the frames in a single batch, some applications require incremental processing. Theoretically, the addition or deletion of a few frames at the end of a clip could change the entire Markov chain solution back to the initial frame, making incremental processing risky. Fortunately, Figure 8 shows empirically that choosing a different endpoint affects at most the last ten frames or so. This opens the door to incremental processing with a roughly half-second delay, as the solution for the final frames awaits the arrival of additional data before commitment.

4. Conclusion

The SiLo tracker demonstrates successful self-initialization and error-recovery for three-dimensional pose tracking. It infers realistic-looking depth information missing from the two-dimensional video input. Like other current algorithms for monocular 3-D pose tracking, it makes some errors, but unlike many techniques it can recover automatically and regain the correct track on subsequent frames relatively quickly and without human intervention.

Despite the positive results presented in this paper, silhouette lookup remains an essentially simple approach to a difficult problem. The tracker described in the preceding sections uses no models of motion or body appearance (other than those implicit in the knowledge base). Any method based upon silhouettes alone lacks the ability to explicitly track body parts with no edges incident on the

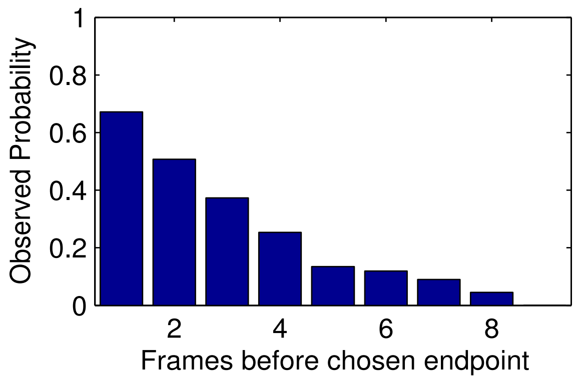


Figure 8: End effects for the SiLo tracker. Running the Markov chain reconstruction on prefix subclips of *Walk* yields a solution that may be compared to that for the full clip. No solutions differed by more than eight final frames.

silhouette's outline, and cannot distinguish between some classes of solution (such as those in Figure 6). For this reason, research is needed on hybrid approaches that augment silhouette lookup with motion models and incremental, texture-based tracking of individual parts. The two approaches have complementary strengths, and each may support the other where it is weak.

The experiments in this paper use activity-specific knowledge bases tailored towards walking and dancing. Even so, the gaps in the knowledge base sometimes impact negatively on the final tracked pose. For the future, generating a general-purpose library of poses that achieves even coverage of the parameter space without redundancy will prove a significant research challenge. Another related challenge will be to control the time required for silhouette lookup by investigating and incorporating algorithms that offer sublinear retrieval speeds [19].

The key contribution of this work lies in the message it carries about approaches to pose tracking: nice results can be achieved by comparatively simple methods based upon retrieval rather than prediction. Instead of generating results by incremental frame-after-frame processing, the SiLo tracker combines simultaneous recognition/retrieval at every frame with subsequent Markov-based temporal reconciliation. This allows the stronger portions of the input to dominate the result, rather than the weakest. The SiLo tracker demonstrates impressive reliability in tracking difficult motions of a single subject in monocular video. With further research, this may prove only the beginning of what lookup-based trackers can achieve.

5 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0328741. The training data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0186217.

References

[1] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[2] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Information Processing and Management*, 31(3):431–448, 1995.

[3] M. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, April 2002.

[4] M. Brand. Shadow puppetry. In *International Conference on Computer Vision*, pages 1237–1244, September 1999.

[5] J. W. Davis and A. F. Bobick. A robust human-silhouette extraction technique for interactive virtual environments. In N. Magnenat-Thalmann and D. Thalmann, editors, *International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, pages 12–25. Springer, 1998.

[6] D. DiFranco, T.-J. Cham, and J. M. Rehg. Reconstruction of 3-d figure motion from 2-d correspondences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 307–314, 2001.

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

[8] T. Horprasert, D. Harwood, and L.S. Davis. A robust background subtraction and shadow detection. In *Proceedings of the Asian Conference on Computer Vision*, 2000.

[9] N. Howe and A. Deschamps. Better foreground segmentation through graph cuts. Technical report, Smith College, 2004. `http://arxiv.org/abs/cs.CV/0401017`.

[10] N. R. Howe, M. E. Leventon, and W. T. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 820–826, Cambridge, MA, 2000. MIT Press.

[11] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *International Conference on Computer Vision*, pages 690–695, 2001.

[12] K.-P. Karmann and A. von Brandt. Moving object recognition using an adaptive background memory. In *Time-Varying Image Processing and Moving Object Recognition*, Amsterdam, 1990. Elsevier.

[13] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, March 2001.

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

[15] J. O'Rourke and N. I. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):522–536, November 1980.

[16] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 467–474, 2003.

[17] R. Rosales, M. Siddiqui, J. Alon, and S. Sclaroff. Estimating 3d body pose using uncalibrated cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[18] B. Scassellati, S. Alexopoulos, and M. Flickner. Retrieving images by 2d shape: A comparison of computation methods with human perceptual judgments. In *Storage and Retrieval for Image and Video Databases*, pages 2–14, 1994.

[19] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *International Conference on Computer Vision*, pages 750–757, 2003.

[20] H. Sidenbladh. *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences*. PhD thesis, Royal Institute of Technology, Stockholm, 2001.

[21] C. Sminchisescu and A. Telea. Human pose estimation from silhouettes. a consistent approach using distance level sets. In *WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, 2002.

[22] C. Sminchisescu and B. Triggs. Kinetic jump processes for monocular 3d human tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 69–76, 2003.

[23] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *European Conference on Computer Vision*, 2002.

[24] C. Tomasi, S. Petrov, and A. Sastry. 3d tracking = classification + interpolation. In *International Conference on Computer Vision*, pages 1441–1448, 2003.

[25] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic, textured background via a robust kalman filter. In *International Conference on Computer Vision*, pages 44–50, 2003.