Silhouette Lookup for Monocular 3D Pose Tracking

Nicholas R. Howe[*]

Smith College, Northampton, MA 01063, USA

Abstract

Computers should be able to detect and track the articulated 3-D pose of a human being moving through a video sequence. Incremental tracking methods often prove slow and unreliable, and many must be initialized by a human operator before they can track a sequence. This paper describes a simple yet effective algorithm for tracking articulated pose, based upon looking up observations (such as body silhouettes) within 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.

Key words: monocular tracking, articulated tracking, pose tracking, silhouette lookup, failure recovery

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 (1; 2; 3). An effective pose tracking 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 (4). Recovery from tracking errors therefore deserves more than the scant research attention it has received (5) to date. Furthermore, currently popular

[*] Based on "Silhouette Lookup for Automatic Pose Tracking", by Nicholas R. Howe which appeared in *IEEE Workshop on Articulated and Nonrigid Motion.* ©2004 IEEE.

Email address: nhowe@cs.smith.edu (Nicholas R. Howe).
URL: http://www.cs.smith.edu/ nhowe (Nicholas R. Howe).

Preprint submitted to Elsevier Science *2 February 2006*

approaches based upon particle tracking are slowed by the need to propagate multiple samples at each frame. Research into non-incremental, recoverable tracking mechanisms therefore fills a pressing need.

This paper develops a lookup-based approach to pose tracking, focusing in particular on silhouette lookup. This approach, hereafter referred to as *SiLo tracking*, 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, the approach invites significantly faster implementations than those 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 that matches the observations.

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 future work.

2 SiLo Tracking

The algorithm described below takes as its input raw video from a single 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 make the already challenging problem more difficult. Although this paper will at times indicate how such additional complications might be addressed, they fall beyond its focus, and the experiments will all use input that conforms to the assumptions listed above.

For each frame F_i in the input video, the algorithm produces as output a vector Θ_i, specifying the pose of a parameterized articulated model of the human body. (The model includes head, neck, torso, upper and lower arms and legs, hands and feet. It uses 35 parameters to specify the angular orientations of the fifteen rigid body parts, plus two for translation in the image plane and one for scaling.) 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 (6)). 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 (7; 8), including some that can identify human subjects moving against dynamic backgrounds (9). The experiments presented below use a static estimation of the background, 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 can be used instead without other significant changes to the algorithm. Regardless of the background model chosen, comparing that model with each frame of the video yields a set of pixels that deviate strongly; these are labeled as foreground and the remaining pixels as background. Further operations on the pixel labels mitigate small errors and yield the observed silhouette for that frame. Simple morphological operations have commonly been used for such cleanup, but this work instead uses a graph-based method that yields slightly cleaner silhouette boundaries (10). Under the assumptions stated above, foreground pixels should correspond to the human subject in the frame. However, there are occasionally small errors due to poor contrast, reflection, shadowing, and other effects. If the set of foreground pixels is disjoint, then the subsequent processing steps work with the largest connected foreground component. Figure 1 shows some sample silhouettes, including some examples of the (rare) failures.

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. This work uses

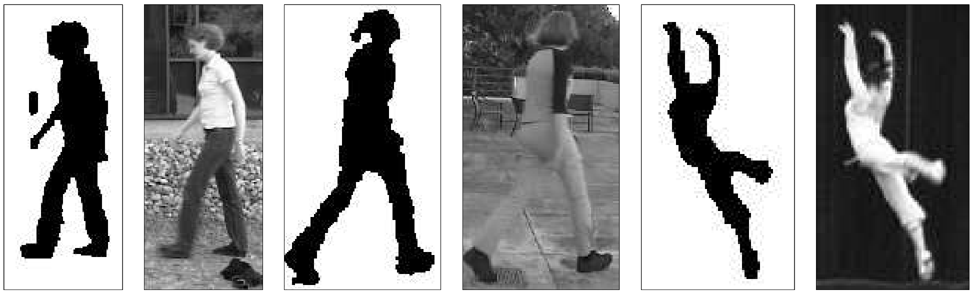


Fig. 1. Sample silhouette extractions, showing some of the failure modes. At left, reflection causes an extra spot (removed in postprocessing). At center and right, hair is labeled as background.

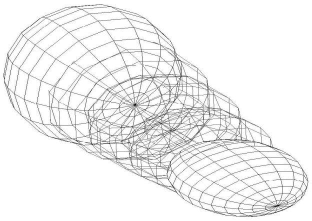
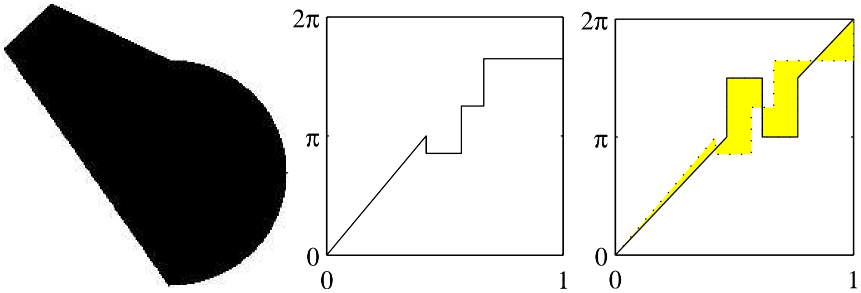


Fig. 2. Body model for artificial renderings (shown: foot). Parts are rendered as smoothly-varying extrusions between ellipsoidal endpoints with aligned axes, generating realistic body shapes from simple pose parameters.

data from the CMU Motion Capture Database to populate the knowledge base, or *silhouette library*. Motion-captured poses are transformed to a standard scale and orientation, then artificially rendered from different viewpoints using a generic body model. (See Figure 2.) For each pose, the library stores silhouettes computed from multiple views. The experiments described herein use 36 parallel projections at 10° intervals of azimuth around the subject and 0° of elevation, but other views can easily be included depending upon the anticipated camera viewpoint.

Early versions of the system simply stored all views for every frame of motion-captured data, leading to redundancy in the silhouette library. Storing the silhouettes of multiple nearly-identical poses degrades performance in several ways. Clearly, it increases search time. More subtly, it can decrease the independence of the top hits retrieved from the library for a particular query, so that the effective number of candidate poses retrieved is lower than the number requested. This means that the correct pose is less likely to be among the k top hits retrieved, for any fixed k. For this reason, during library construction each new pose must be compared against all those currently stored, and discarded if it fails to differ significantly from some pose already in the library. Since the motion of a single body part can change the silhouette, a pose is considered significantly different if either endpoint of any body part differs by more than a chosen threshold. (The amount of frame-to-frame change observed in typical motion capture data motivates the choice of both this threshold and



(a) (b)

Fig. 3. 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. The turning angle metric measures the area between two such traces, (c).

the number of different views generated for each pose.)

Once stored in the library, silhouettes must be retrieved using one or more shape retrieval heuristics. Many common heuristics used for general shape retrieval prove insufficiently sensitive to the precise orientation of arms and legs, which are crucial in this application. Furthermore, while many shape retrieval heuristics strive for rotation invariance, gravity ensures that rotation is highly significant for human poses. This work incorporates two heuristic similarity measures: the turning angle metric and the chamfer distance. Although both work individually, a combination of the two (using summed retrieval ranks (11)) 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 (12). 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 3). A single trace records the tangent angle at equally-spaced points around the silhouette, beginning at the highest point (usually the top of the head) and proceeding clockwise. Typically around 150 points are sampled, depending on the compactness of the shape. Because the turning angle metric is not rotation invariant, its use here assumes that a common rotational reference point exists; standard filming convention justifies making this assumption. (The vertical axis in physical space corresponds with the y axis in most videos.) The expedient of using the topmost point to start implies a minor instability if two separate body parts are close to topmost, but this difficulty may be addressed by ensuring that the silhouette library includes separate examples of similar poses where each distinct part gets to be topmost. Using this approach, no unusual problems appear in the experiments for sequences where the arms rise above the head.

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 pixel in the other set.

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

Note that this is related to the Hausdorff metric, which takes a maximum rather than a sum. It is asymmetric, i.e., $\xi(S_1, S_2) \neq \xi(S_2, S_1)$. For silhouette retrieval, the experiments below use the chamfer distance between the boundary points of the observed silhouette and the boundary of the library silhouettes. This one-way chamfer distance can be found rapidly by precomputing the distance transform of the observed silhouette boundary and using a chain-code representation of each library silhouette boundary to sample from it.

Using the selected comparison heuristic or combination of heuristics, each silhouette extracted from the input frames identifies a set of silhouettes in the knowledge base that appear closest to the observed silhouette: the library "hits". The poses associated with these hits become the candidates in the next processing phase (Markov chaining). Depending on the coverage density in the silhouette library, the number of hits within a fixed similarity threshold varies widely at different points in a video clip. As a supplement or alternative to using a fixed threshold, the number of hits can be k_i confined to lie within bounds k_{min} and k_{max}, such that $k_{min} < k_i < k_{max}$ at all frames. In practice, setting $k_{min} = k_{max} \approx 50$ often works satisfactorily, but this depends on the density of coverage in the library.

If the library population is sparse (due to a scarcity of relevant motion capture data), it can be augmented via the notion of *kinetic jumps* (14). Simply put, a family of related poses can generate identical silhouettes under orthograhic projection, and thus a number of candidate poses can be generated from a single library pose. The simplest example involves a simultaneous left-right inversion of the pose and reflection of the line-of-sight axis (see Figure 4). Thus a set of library retrievals may be augmented by computing the family of kinetic jumps, then pruning physically impossible and duplicate poses. In practice, this can be useful if the library coverage is sparse, but produces little or no benefit when the poses of interest are densely represented in the library.

The discussion above has assumed that simple linear search will suffice for retrieval of the candidate silhouettes. Indeed, the metrics used can be computed rapidly enough that linear search is used for all of the experiments presented below. However, retrieval speed becomes significant if operation near real-time speeds is desired, or if the pose library becomes sufficiently large. Including many different types of motion in the pose library will swell its size, as will

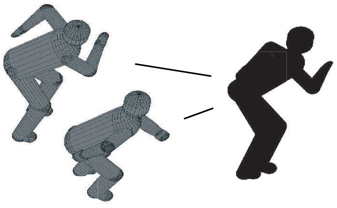


Fig. 4. Right-left ambiguity for silhouettes. The two poses on the left produce exactly the same silhouette when viewed from the side under orthographic projection. The transformation from one pose to the other is called a *kinetic jump*.

including sihouettes seen from nonzero camera elevations.

Some research has looked at sub-linear retrieval using parameter-sensitive hashing (15) or the triangle inequality (16), resulting in techniques which can be applied here. For the tracking application one can also take advantage of the temporal connections between frames to restrict the number of poses that must be examined. Poses very dissimilar to the candidates for the previous frame will be rejected in the chaining phase (as described below), and therefore need not be considered for retrieval. If the pose library is augmented during creation so that each entry contains a table of pointers to all other poses within a chosen similarity threshold, then these pointer tables may be used to quickly search the subset of poses that are most similar to the candidates from the previous frame. Consideration of the maximum plausible human motion in a single frame suggests a suitable threshold choice. Judicious use of these strategies should allow the implementation of much larger pose libraries in practical applications.

2.3 Markov Chaining

Because the relationship of possible 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. There exist techniques that implement one-to-one mappings of silhouettes to poses (17), but these will have difficulty when they encounter a pose that differs from the one embodied by their mapping, as must happen sooner or later. Nevertheless, any tracker must ultimately weed through the profusion of possible poses to settle on a single most likely pose in each frame, and Markov chaining provides the appropriate mechanism.

Markov chaining exploits the temporal dependency of human motion to eliminate 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.

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

Here n represents the number of frames in the video, ζ represents the matching error between the silhouette corresponding to the pose parameters Θ_i and the observations in a given frame S_i, Δ represents the motion difference between two different sets of pose parameters, and λ serves as a weighting factor. The remainder of this section discusses the choice of functions for ζ and Δ.

One potential choice for ζ is the energy function used to rank silhouettes for retrieval. Although an asymmetric chamfer distance was used for rapid retrieval, at this stage the number of poses to be considered is small enough to allow the use of the symmetric function, and this provides greater sensitivity to the precise silhouette observations. Thus the chaining stage uses a ζ that applies Equation 1 symmetrically over the border pixels of the two silhouettes:

$$\begin{aligned} &\text{Let } P_{\Theta_i} = BorderPoints(Render(\Theta_i)) \\ &\text{and } P_{S_i} = BorderPoints(S_i); \\ \zeta\left(\Theta_i, S_i\right) &= \xi\left(P_{\Theta_i}, P_{S_i}\right) + \xi\left(P_{S_i}, P_{\Theta_i}\right) \end{aligned} \tag{3}$$

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. However, properly 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, which increases the complexity of the computation. In most cases, the simpler format yields excellent results, and the extra computation of the more physically plausible models appear unnecessary. Except for the synthetic-data experiment, all the results presented below use the simpler difference function.

$$\Delta^{(1)} = \sum_{j \in Parts} M_j \left|\partial x_j(\Theta_i, \Theta_{i-1})\right| + I_j \left|\partial \varphi_j(\Theta_i, \Theta_{i-1})\right| \tag{4}$$

Here M_j and I_j are the mass and moment of inertia, respectively, of the jth body part, while ∂x_j and $\partial \varphi_j$ are the translation and rotation of the part between the poses specified by parameters Θ_{i-1} and Θ_i. By contrast, the two-frame function looks like the following.

$$\Delta^{(2)} = \sum_{j \in Parts} M_j \left| \partial x_j(\Theta_i, \Theta_{i-1}) - \partial x_j(\Theta_{i-1}, \Theta_{i-2}) \right| \\ + I_j \left| \partial \varphi_j(\Theta_i, \Theta_{i-1}) - \partial \varphi_j(\Theta_{i-1}, \Theta_{i-2}) \right| \quad (5)$$

As noted above, using the simplified $\Delta^{(1)}$ 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. The minimum of Equation 2 can be found efficiently using a forward-backward (Viterbi) dynamic programming algorithm, 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.

As a final note, although the Markov-chain approach works for most normal cases, it can run into trouble if poor quality video input leads to intermittent failures of the foreground segmentation. A single frame with a gross error in the observed silhouette can cause all the poses retrieved for that frame to be far from any of the retrieved poses in the surrounding frames. This situation will be readily apparent as a spike in the Markov chain energy at that frame. If detected (perhaps by examining the per-frame transition energy $\Delta(\Theta_i, \Theta_{i-1})$), the situation can be dealt with in one of two ways. One possibility is to allow "skipped" frames in the chaining process, computing the energy for the skipped frames by interpolation. (This can be viewed as augmenting the set of retrieved poses for a particular frame with additional candidates generated via linear combinations of two retrieved poses, one from a preceding frame and one from a succeeding frame.) Allowing skipped frames increases the complexity of the chaining process, but has been implemented and run successfully.

Another approach is simply to require some or all of the candidate poses to lie near the poses generated for the previous frame, by choosing the best matches from a restricted set of silhouettes selected for their proximity. Including a near-neighbor for each pose from the preceding frame will ensure continuity, but may waste resources on candidate solutions with low probability, while leaving insufficient resources for the more likely solutions. This is analogous to the problem faced by particle-tracking methods (18), which solve it by weighting the chance of perpetuating any given trajectory according to how well it fits the data available. For the experiments herein, a pool of candidates is chosen from the subset of all poses lying near some candidate from the preceding frame, without requiring a nearby candidate for every preceding pose. This allows for continuity of the best trajectories while allowing the more unlikely trajectories to die out. Nevertheless, if all the new candidates

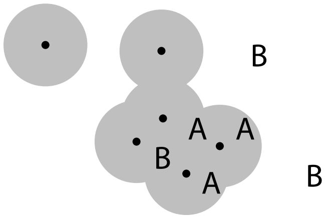


Fig. 5. Schematic illustration of candidate pose selection for a new frame. One pool of candidates (A) is chosen from those lying near the candidates for the preceding frame (dots; gray area shows eligible neighborhood). A second pool is chosen without restriction (B), and may be any distance from the preceding frame's candidates.

are selected in this manner, then the ability to recover from tracking errors may suffer. (The solution may get trapped in an unlikely area of parameter space far from the true solution.) Thus it also makes sense to include in the pool of candidates some poses chosen without restriction, to allow recovery from gross errors. Figure 5 illustrates these considerations schematically. (The reader may recall that Section 2.2 also proposed restricting the candidate pool to poses near the previous frame's candidtates, for faster library searches. If speed is essential, then the inclusion of unrestricted candidate poses for error recovery may be performed only periodically, rather than at every frame.)

As a note, the experimental results presented in Section 3.4 suggest that the set of retrieved candidate solutions tends to converge to the most likely possibilities rather than diverge. Indeed, there is usually substantial overlap between the candidates chosen with and without restriction. If a low-quality foreground segmentation is ambiguous enough to permit many different candidate poses, then tracking may not be feasible in any case.

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 do not exactly match the poses in the true solution. Usually, a rendering of the proposed solution appears 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.

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

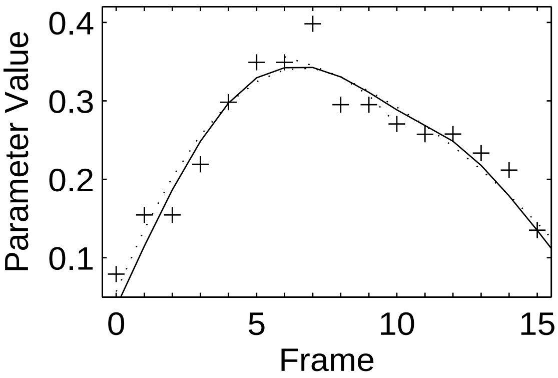


Fig. 6. Sample smoothing of single parameter dimension $\theta_j(i)$. The final curve (solid line) smooths out the individual Markov chain points (crosses). Two of the overlapping spline curves are shown (dotted lines).

underlying component functions θ_j should be smooth, $\theta_j(i)$ can be modeled as a series of smoothly interpolated overlapping polynomial splines (see Figure 6). Taking $\Theta'_i = (\theta_1(i), \theta_2(i), ..., \theta_m(i))$ eliminates the error term and 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. (Examination of real motion capture data indicates that splines of this sort can readily model real human motion with negligible error.) Given a frame rate of 30 Hz, eleven-frame splines enforce smoothness over a timescale of about one-third of a second.

It is worth noting that the use of splines in this capacity causes trouble if the mapping from parameters Θ_i to joint positions contains singularities. In such cases, pose represented by the smoothed spline curve may differ greatly from the unsmoothed pose despite the small distance in parameter space. This work uses an Euler-like representation of the joint angles, and encountered such problems in its early stages. The joint angle repesentation employed herein has since been modified to ensure that the singularities occur outside the range of motion for each individual joint. (As an alternative, one might use quaternions or another singularity-free joint angle representation, but this would increase the number of parameters needed and thus significantly slow the optimization step described below.)

The result of the smoothing process may still not exactly match the observed silhouettes in all places, depending upon how closely the poses in the knowledge base can match the actual poses observed. Parametric optimization can increase the match between the observations and the proposed solution. This is implemented as a simplex search via Matlab's *fminsearch* function, since the discreteness of the silhouette observations makes gradient-based search unreliable. To maintain the smoothness of the solution, the optimization proceeds on the parameters of the m polynomial splines (created during the smoothing process described above) that generate a smoothed block of eleven frames at once. The energy criterion is simply Equation 3 summed over all 11 frames:

$$E = \sum_{i=i_0}^{i_0+10} \zeta\left(\Theta_i, S_i\right) \tag{6}$$

Optimization typically takes much longer than the preceding steps for a relatively small improvement in the accuracy of the results, so applications (such as activity and gait recognition) not needing extreme precision may choose to forego it. Furthermore, if allowed to proceed to a conclusion, optimization may in certain cases discover physically unlikely poses that nevertheless exhibit slightly lower energy than much more physically probable ones. Incorporating a term in the energy equation to evaluate *a priori* pose likelihood would discourage such behavior. Unfortunately, good probabilistic models of human pose are still a subject of research. A more expedient way to avoid the problem is to cut off optimization before it has the chance to explore poses too far from the initial solution. In this manner, optimization corrects gross mismatches between the proposed solution and the observations without straying too far from known poses.

2.5 Related Work

A large body of work on pose tracking precedes this paper, dating back to the early 1980's (1; 2; 19; 20; 21; 22; 23). A 2001 survey lists many contributions (3), and divides the work into categories according to the problem addressed and approach taken. This section will focus on other research into full *3-D articulated pose reconstructions from monocular video input*, since that is the target of the current work.

Recent efforts have used models of probable poses and motions, appearance models for body parts, and sophisticated optimization routines together with particle-based tracking algorithms (21; 22; 14). As mentioned previously, approaches of this sort encounter difficulties with initialization and error recovery, and can be slow to operate due to the number of samples that need to be maintained. More recent work uses bottom-up detection of body parts based upon appearance priors to locate and track subjects (24; 25). This can allow for automatic initialization without the use of background subtraction, but usually introduces other assumptions about the appearance of the tracking subjects (e.g., body parts have coherent appearance). Assembling people from their parts is difficult, and while the initial results appear promising, more research is necessary to define the full capabilities of the approach.

There has been some prior interest in using silhouettes for pose recognition (26; 27; 28), 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 (23). The latter work is similar in spirit to that described here, using edge images instead of silhouettes to retrieve a single pose per frame. It applies a completely different retrieval metric (shape context (29)) and does not address the issues of frame-to-frame chaining considered herein. A more complete comparison of the two methods would make an interesting subject for future research. Another recent paper develops an elegant method for regression from silhouettes to poses (17), but by implicitly defining the silhouette-to-pose relationship as one-to-one, it limits the variety of human poses that it can handle.

Recent research has also looked at the use of silhouettes for tracking hand pose (30; 31). The hand-tracking work makes the one-to-one assumption, and further differs from the results presented herein by presuming that only a small number of key poses (e.g., sign-language symbols) need be precisely identified, with intervening frames filled in via interpolation. In a similar vein, repetitive full-body motions such as walking have been reduced to a small number of key poses, with the problem further constrained by a learned model of the transition probabilities (32). 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 would seem to inhibit the identification of key poses in free-form motion such as dance. (On the other hand, key poses have also been applied for full-body estimation in certain limited domains, for example in the analysis of tennis serves (33).)

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 (15). 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 equally useful with silhouette lookup, although the Markov chaining and smoothing steps already achieve results similar to those of the interpolation process.

Finally, some previous work has looked at the use of temporal Markov chains for simpler problems. Kwatra et. al. use temporal chains to label body parts protruding from the edges of human silhouettes, and for generating simple pose descriptors such as standing, sitting, bowing, etc. (34). This work uses similar ideas, but takes them much further.

3 Experimental Results

Quantitative evaluation of 3-d pose reconstruction is notoriously difficult, and standard test sets have yet to emerge. It is difficult to obtain ground truth calibrated with real video. This section therefore begins with quantitative results for synthetic input for which ground truth is known. Further experiments apply the methods described above to real video clips without ground truth, but representing a wider range in difficulty.

3.1 Synthetic Data

Synthetic input can be easily generated from motion capture data, using the same renderer that produces library silhouettes. This experiment uses a fresh set of motion-capture data, not made available to the system during library generation. The renderer generates a sequence of foreground silhouettes, one for each frame, and these then serve as the queries for the lookup stage. Because the test sequence was not seen by the system during library creation, the library should not contain any exact matches to the test silhouettes. The chosen sequence shows one hundred frames of a person walking in side view (much like the *Walk-Straight* clip described later).

Figure 7 summarizes a comparison of the tracking results with ground truth. Twenty points of interest on the body form the basis of the comparison. The figure displays the root-mean-square deviation of the tracked solution from the ground truth in the camera plane and along the line of sight. Units are pixels (by comparison, the human figure is around 150 pixels tall). Because the reconstruction of coordinates along the line of sight can only be determined only up to a constant term, the reconstuction is normalized along this axis so that the mean position over the entire sequence equals that of the ground truth. The errors displayed are therefore residuals reflecting differences in limb position and orientation between the reconstructed pose and the ground truth.

The figure reveals several interesting patterns. While the line-of-sight error is the greatest, it is within a factor of two of the visible dimensions. It rises at the extremities of the body, due to accumulation of errors at previous joints in the kinematic chain. The two image-plane dimensions exhibit the highest error in the arms, which are frequently occluded by the body and therefore more difficult to reconstruct accurately.

Synthetic data can also answer questions about the importance of the body model used for tracking. Figure 8 shows three variant inputs created from the ground truth sequence, simulating both under- and overweight subjects, and increased noise in the background subtraction. Below each input appears

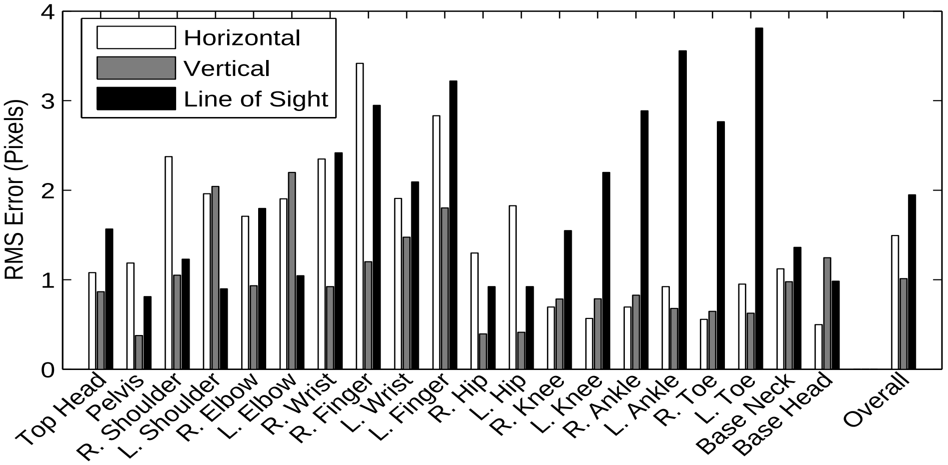
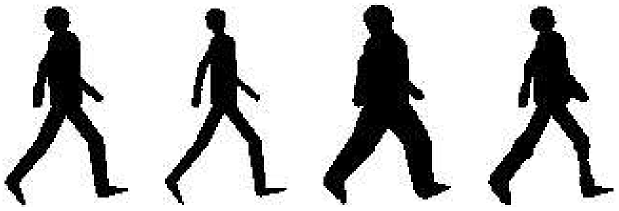


Fig. 7. RMS error in the reconstruction of selected body points for a 100-frame sequence of synthetic data, as compared to ground truth. By comparison, the human figure used for this experiment is approximately 150 pixels tall.



	Normal	Thin	Heavy	Noisy
Horizontal	1.5	1.2	3.4	1.1
Vertical	1.0	1.4	1.2	0.9
Line-of-sight	1.9	1.9	3.4	1.7

Fig. 8. Synthetic data with varying body models: Sample silhouettes (top), with RMS error averaged over all body points (bottom).

the mean error observed in a reconstruction using the standard body model. All the experiments yield qualitatively correct results, clearly reproducing the walking behavior. The quantitative results are also mostly similar, with minor variation for three of the four inputs. Only the overweight walker gave the system noticeable difficulty, evident in a greater variation from the ground truth.

3.2 Real Video Data

Four video clips range from easy to more difficult to track. *Walk-Straight* shows a subject walking from right to left, while *Walk-Circle* shows the same subject walking in a circle. Both clips were originally generated and used to test other

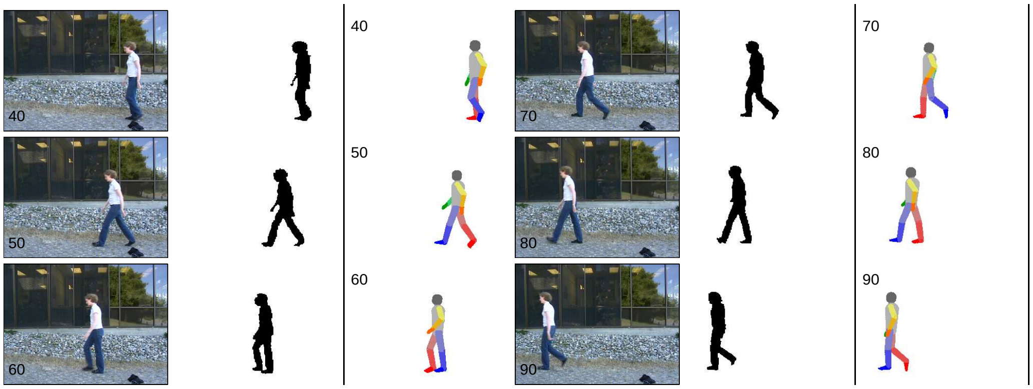


Fig. 9. *Walk-Straight* clip and its reconstructed pose in selected frames.

tracking algorithms (22), although lack of a ground truth precludes quantitative comparisons. Two other clips, *Dancer-Leap and Dancer-Arabesque*, show a ballet dancer performing short routines. The turning of the dancer's body in these clips makes them difficult for many tracking algorithms to follow.

Figures 9-12 summarize the tracking results for the trial clips. The system tracks *Walk-Straight* 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 a multitude of 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.

3.3 Error Analysis

Ambiguity problems appear in the latter third of *Walk-Circle*: the tracked motion and the true motion suffer from a right-left reversal. This cannot be completely 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 4. Similar ambiguities cause problems in the *Dancer-Leap* clip when the dancer's body turns. The tracked silhouette matches the observations, but close inspection shows that the tracked direction of rotation does not match reality. It is possible that the use of additional cues beyond silhouette matching (such as optical flow) could control this source of error.

Retrieval failure appears in the *Walk-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 retrieval step are close matches for the actual pose. Indeed, the next 40 frames or so consist of poses for which the retrieval metric does not adequately dis-

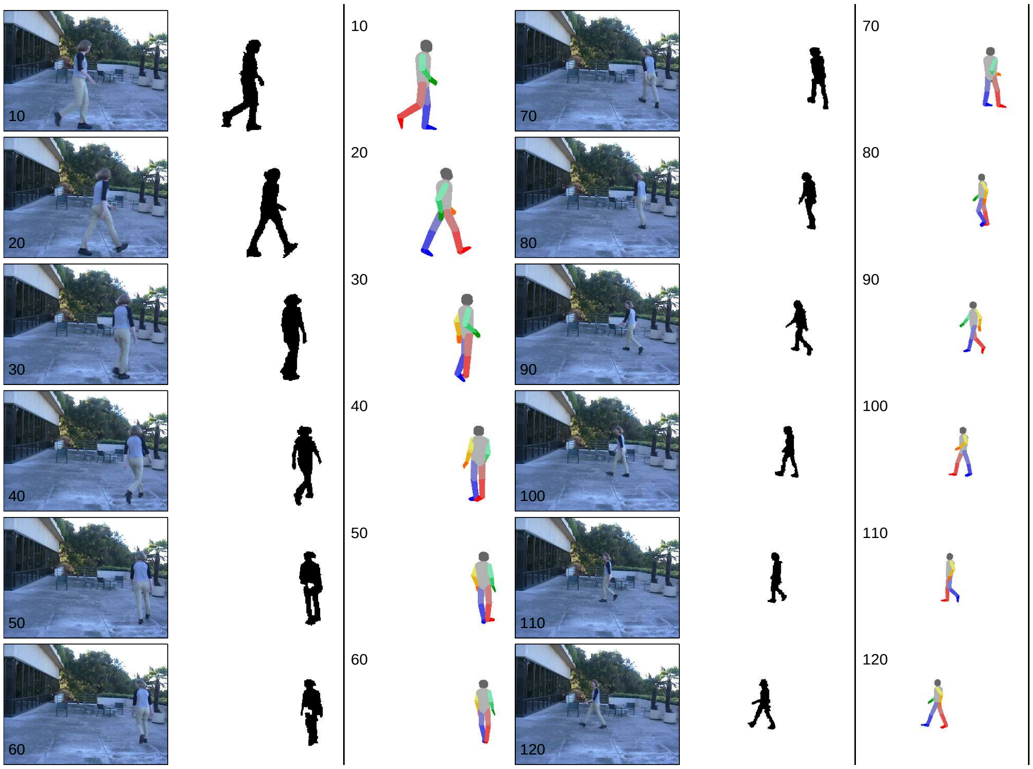


Fig. 10. *Walk-Circle* clip and its reconstructed pose in selected frames.

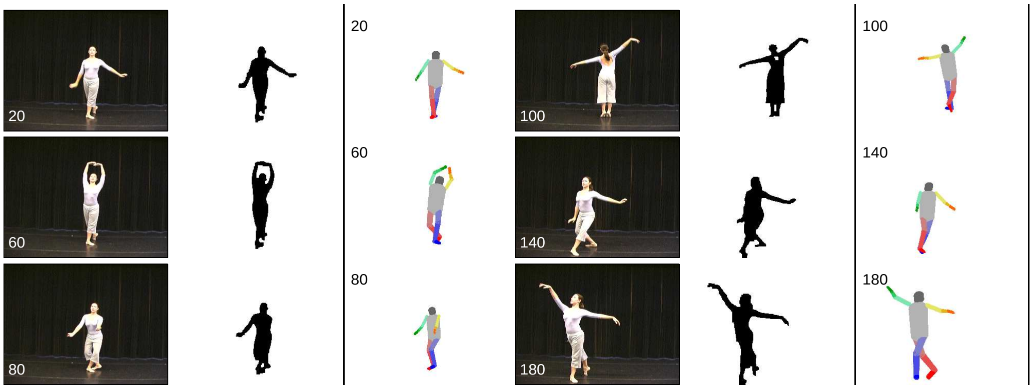


Fig. 11. *Dancer-Arabesque* clip and its reconstructed pose in selected frames.

tinguish the correct pose amongst a multitude of incorrect poses with similar silhouettes. The continuity term of Equation 2 dominates, and the recovered pose track is correspondingly confused. Research into different silhouette retrieval mechanisms might address this problem, but there also appears to be a particular problem with figures moving toward or away from the camera: the limb movements in such cases often do not produce silhouettes that are very distinct from one another. Nevertheless, around frame 80 the tracker 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 regain the correct track even after essentially losing it completely. The *Dancer-Arabesque* clip also shows intermittent tracking failures

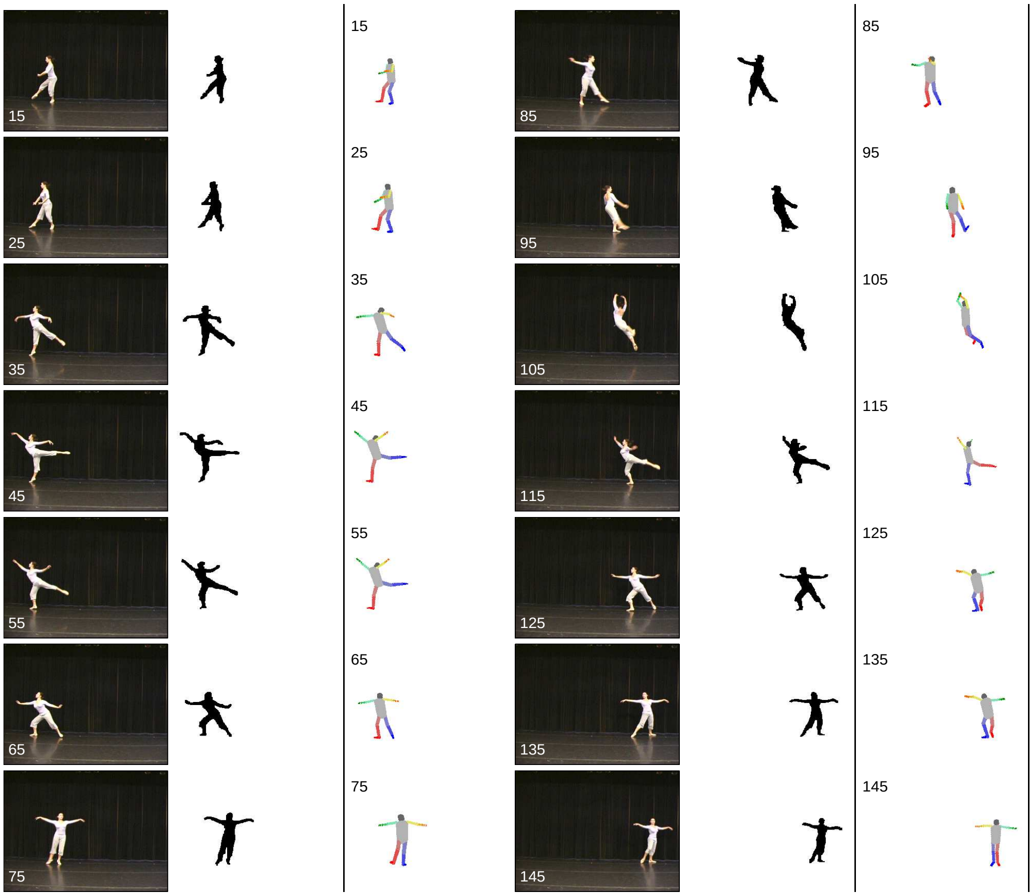


Fig. 12. *Dancer-Leap* clip and its reconstructed pose in selected frames.

and recoveries. However, in this case it is caused not by a failure to retrieve the best existing poses from the library, but simply because some of the poses in this sequence have no close correspondances in the pose library. Expanding the amount of motion capture footage available at library creation would presumably address this problem.

3.4 Incremental Behavior Analysis

The algorithm used for these experiments processes video clips in batch mode, rather than incrementally frame by frame. Although batch analysis offers computational advantages (4), the algorithm can also be modified to allow incremental processing. One may reasonably ask whether such a change will influence 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-Straight* clip, and find that the answer in most cases is fewer than ten frames.

Figure 13 shows the results of an experiment designed to test how quickly

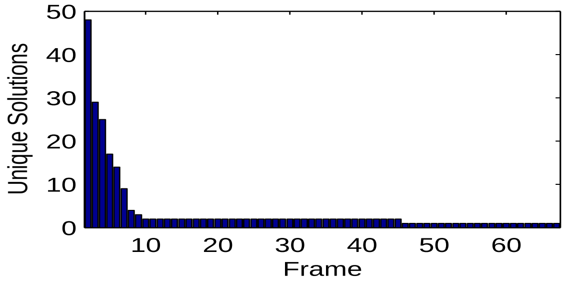


Fig. 13. Convergence of the SiLo tracker under divergent starting conditions. The bars show the number of unique solutions on the *Walk-Straight* clip decreasing rapidly over time, despite the initial frame's constraint to a randomly chosen pose on each of 1000 trials.

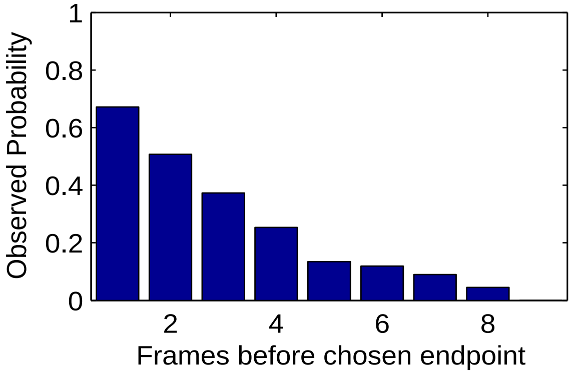


Fig. 14. End effects for the SiLo tracker. Running the Markov chain reconstruction on prefix subclips of *Walk-Straight* yields a solution that may be compared to that for the full clip. No solutions differed by more than eight final frames.

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 (and high-quality) solutions, and after 45 frames all reach 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. This is particularly important for incremental processing, since 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. Fortunately, Figure 14 shows empirically that choosing a different endpoint affects at most the last ten frames or so. This suggests that incremental processing is feasible, with the proviso that results less than one-half second old should be considered temporary. Applications requiring stable results can implement a half-second delay, as the solution for the most recent 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 from monocular video. It infers realistic depth information missing from the two-dimensional input. Like many other current algorithms for monocular 3-D pose tracking, it makes some errors, but unlike most techniques it can recover automatically and regain the correct track on subsequent frames 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 silhouette's outline, and cannot distinguish between some classes of solution (such as those in Figure 4). For this reason, future work should examine 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 reduce the time required for silhouette lookup by investigating and incorporating algorithms that offer sublinear retrieval speeds (15).

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. That database was created with funding from NSF EIA-0186217.

References

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

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

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

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

[5] S. Ioffe, D. Forsyth, Human tracking with mixtures of trees, in: International Conference on Computer Vision, 2001, pp. 690–695.

[6] J. W. Davis, A. F. Bobick, A robust human-silhouette extraction technique for interactive virtual environments, in: N. Magnenat-Thalmann, D. Thalmann (Eds.), International Workshop on Modelling and Motion Capture Techniques for Virtual Environments, Springer, 1998, pp. 12–25.

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

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

[9] J. Zhong, S. Sclaroff, Segmenting foreground objects from a dynamic, textured background via a robust Kalman filter, in: International Conference on Computer Vision, 2003, pp. 44–50.

[10] N. Howe, A. Deschamps, Better foreground segmentation through graph cuts, Tech. rep., Smith College, `http://arxiv.org/abs/cs.CV/0401017` (2004).

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

[12] B. Scassellati, S. Alexopoulos, 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, 1994, pp. 2–14.

[13] N. R. Howe, Silhouette lookup for automatic pose tracking, in: IEEE Workshop on Articulated and Nonrigid Motion, IEEE Computer Society, 2004.

[14] C. Sminchisescu, B. Triggs, Kinetic jump processes for monocular 3d human tracking, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, pp. 69–76.

[15] G. Shakhnarovich, P. Viola, T. Darrell, Fast pose estimation with parameter-sensitive hashing, in: International Conference on Computer Vision, 2003, pp. 750–757.

[16] A. P. Berman, L. G. Shapiro, A flexible image database system for content-based retrieval, Computer Vision and Image Understanding 75 (1-2) (1999) 175–195.

[17] A. Agarwal, B. Triggs, 3d human pose from silhouettes by relevance vector regression, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. II, 2004, pp. 882–888.

[18] M. Isard, A. Blake, Condensation – conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.

[19] C. Bregler, J. Malik, Tracking people with twists and exponential maps, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santa Barbera, 1998.

[20] S. Wachter, H.-H. Nagel, Tracking of persons in monocular image sequences, in: Nonrigid and Articulated Motion Workshop, 1997.

[21] N. R. Howe, M. E. Leventon, W. T. Freeman, Bayesian reconstruction of 3d human motion from single-camera video, in: S. Solla, T. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems 12, MIT Press, Cambridge, MA, 2000, pp. 820–826.

[22] H. Sidenbladh, Probabilistic tracking and reconstruction of 3d human motion in monocular video sequences, Ph.D. thesis, Royal Institute of Technology, Stockholm (2001).

[23] G. Mori, J. Malik, Estimating human body configurations using shape context matching, in: European Conference on Computer Vision, 2002.

[24] D. Ramanan, D. Forsyth, Finding and tracking people from the bottom up, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, pp. 467–474.

[25] L. Sigal, S. Bhatia, S. Roth, M. Black, M. Isard, Tracking loose-limbed people, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. I, 2004, pp. 421–428.

[26] M. Brand, Shadow puppetry, in: International Conference on Computer Vision, 1999, pp. 1237–1244.

[27] R. Rosales, M. Siddiqui, J. Alon, S. Sclaroff, Estimating 3d body pose using uncalibrated cameras, in: IEEE Computer Society Conference on

Computer Vision and Pattern Recognition, 2001.

[28] C. Sminchisescu, 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.

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

[30] C. Tomasi, S. Petrov, A. Sastry, 3d tracking = classification + interpolation, in: International Conference on Computer Vision, 2003, pp. 1441–1448.

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

[32] X. Lan, D. Huttenlocher, A unified spatio-temporal articulated model for tracking, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. I, 2004, pp. 722–729.

[33] J. Sullivan, S. Carlsson, Recognizing and tracking human action, in: European Conference on Computer Vision, 2002.

[34] V. Kwatra, A. Bobick, A. Johnson, Temporal integration of multiple silhouette-based body-part hypotheses, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. II, 2001, pp. 758–764.