Video-Based Tracking of 3D Human Motion Using Multiple Cameras

by
Ben Sigelman

A Thesis submitted in partial fulfillment of the requirements for Honors
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2003

This thesis by Ben Sigelman is accepted in its present form by
the Department of Computer Science as satisfying the research requirement
for the awardment of Honors.

Date _____ _____

Michael Black, Reader
Thesis Advisor

Date _____ _____

John F. Hughes, Reader

Acknowledgements

I am deeply indebted to Professor Michael Black for his guidance during the past year. He has taught me a great deal about computer vision in little time, and has also served as an important mentor figure for me. Whenever I entered his office confused, I would exit enlightened, inspired, and enthusiastic; what more could you ask from an advisor? I would also like to thank John Hughes for his invaluable assistance with various and sundry loose ends in my implementation.

I would like to thank my parents for deciding to have a second child, and also for choosing not to throw me to the wolves. My sister — despite dropping me on my head when I was two — has been supportive throughout, and this document would not exist but for the generous cornucopia of thesis-writing snack food she gave to me.

Tommy Jarrell, Kyle Creed, John Salyer, Clyde Davenport, Ed Haley, Gid Tanner, and the Skillet Lickers kept me entertained throughout, and their music served as the soundtrack to most of my efforts in computer vision — so perhaps they deserve some recognition. (Regardless of that condition, I cannot resist the temptation to formally mention my old-time musical idols)

Of course I would like to thank my wonderful friends, who have, at every step of the way, reminded me that my work will never be more than the second-most-important thing.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Framework

The goal of human motion tracking, in as few words as possible, is to know where someone is. Specifically, we want to know where someone is in three dimensions given a series of video sequences from multiple cameras. Commercial motion tracking systems usually employ various optical or magnetic markers, and thus are solving a less demanding subproblem: instead of tracking human beings, they're tracking bright points and inferring the human being. We are attempting to track human motion with minimal assumptions about clothing or image backgrounds.

1.1 The Problem of Human Motion Estimation

Tracking with minimal assumptions is a demanding task. In Figure 1.1, we see an illustration of many prominent difficulties.

- **Poor image quality:** Grainy images result in noisy measurements, and motion blur obscures limb edges.
- **Self-Occlusion:** Even when a subject is in plain view, limbs are often obscured by other parts of the body. By consequence, any effective tracking system must be robust to the momentary disappearance and reappearance of limbs.
- **Inaccurate body model:** At a certain level of detail, any model of the human body will be inaccurate. People come in varying proportions, and a good model must be robust to wide variation in human appearance.

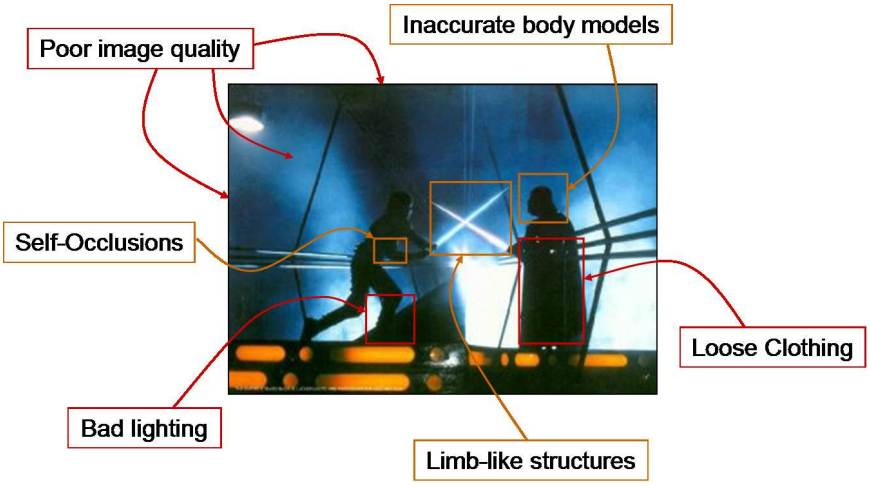


Figure 1.1: **Human motion tracking is a demanding task.**

- **Loose clothing:** Even with an accurate body model, loose clothing ambiguates limb location and muddles appearance.
- **Limb-like structures:** Without constraints on scene background characteristics for a capture sequence, it is easy to misidentify miscellaneous scene elements as subject substructure.
- **Bad lighting:** Excessively dim or excessively bright lighting conditions make feature detection more challenging.

An effective tracking system must model a great deal of uncertainty. In this work we present a system designed to track humans using video sequences from multiple cameras. We frame the problem as a Bayesian inference task (Section 1.4) and use a particle set (Section 1.5) to model the rather ornery distribution of potential body configurations given our input video sequences. First, however, we will discuss the body and camera models at the heart of our tracking system.

1.2 The Body Model

Limbs in the body are represented as tapered cylinders, scaled independently along each basis vector. The hands and feet are not included in the body model

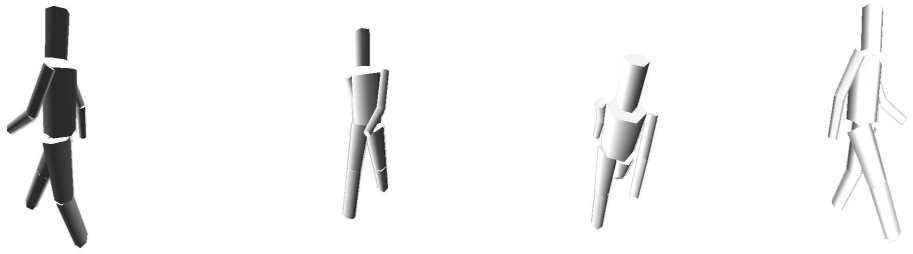


Figure 1.2: **The body model:** The body model as seen from four viewing angles in the visualizer.

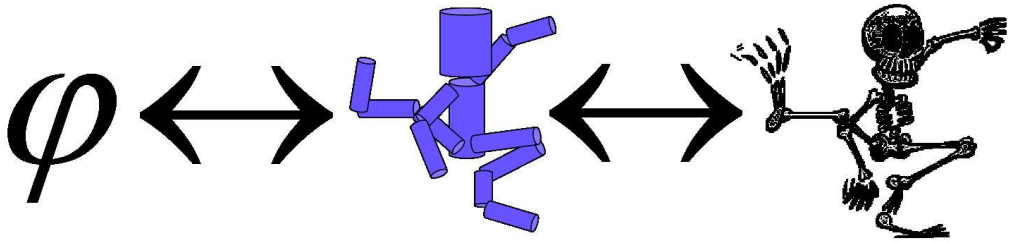


Figure 1.3: **Parameterizing the body:** ϕ describes the rotational and translational quantities used to describe a configuration of tapered cylinders which, in turn, serves to model the body of the subject.

used throughout this work. The skeleton is arranged hierarchically, and each joint is parameterized by a number of Euler angle rotations. (The specifics of the parameterization can be seen in Table 1.1) This kinematic model is prone to infamous singularities,[11] and tracking results would benefit from a switch to a well-behaved parameterization. Quaternions, which effectively eliminate the kinematic singularities, are one possibility; the twists and exponential maps proposed by Bregler and Malik[1] are another.

In addition to the dynamic rotational parameters, each limb may be translated some static distance from its parent joint. This translational offset is useful when modeling limbs that are connected schematically but separated from one another in space (such as an arm and the torso, or the head and the torso).

Joint	# prms	Joint	# prms	Joint	# prms
World Position	3	R. Shoulder	1	L. Shoulder	1
World Rotation	3	R. Elbow	1	L. Elbow	1
R. Hip	1	L. Hip	1	R. Knee	1
L. Knee	1	Neck	2		

Table 1.1: **Specific body model parameterization:** The 16-dimensional state space is broken into 6 global translation and rotation parameters and 10 intrinsic model parameters. Because the subject is walking in the tracking sequence, we need not model the full range of motion at the hip or shoulder.

1.3 The Camera Model

In the field of synthetic computer graphics, a linear camera model is typically sufficient for generated imagery. Specifically, a line in space will typically project to a line in the film plane. However, even special-purpose lenses designed for use in motion capture applications have significant defects, especially in the corners, and the linear model is no longer adequate. Our system allows a camera model to exist in multiple "levels of detail." All tracking calculations are performed with a procedural non-linear camera projection, user-level camera actions[1] are performed by a slightly less complicated model, and certain functions — silhouette edge determination, screen projections of three-dimensional models, et cetera — are the realm of an even simpler camera interface. This allows for the easy incorporation of new camera models when necessary, and keeps response times quick for the interactive visualizer.

We used the Tsai camera model[17] for all camera calculations involved with tracking and two-dimensional visualization. Unlike a tradition linear camera model, the Tsai model accounts for lens distortion and common defects in charge-coupled device manufacturing. The actual center of projection is often *not* at the center of the CCD array, and without a robust camera model, tracking in three dimensions — especially with multiple cameras — would be prohibitively difficult.

[1]Only the three-dimensional visualizer allows for viewer motion

1.4 Bayesian Inference

Given a series of images $\vec{I_t} = \{I_1, I_2, \cdots, I_t\}$, we want to determine the model parameterization ϕ_t at time t. We would like to sample from the distribution $p(\phi_t|\vec{I_t})$, which is known as the *posterior* distribution.

We would like to reformulate $p(\phi_t|\vec{I_t})$ to incorporate the temporal prior distribution $p(\phi_t|\phi_{t-1})$ and an image likelihood $p(I_t|\phi_t)$. We make a first-order Markov assumption and arrive at the following:

$$p(\phi_t|\vec{I_t}) = \int p(\phi_t, \phi_{t-1}|I_t, \vec{I}_{t-1}) \quad d\phi_{t-1}$$

Bayes' rule, in general, states that

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

We apply Bayes' rule to the integrand, yielding

$$p(\phi_t, \phi_{t-1}|I_t, \vec{I}_{t-1}) = \frac{p(I_t, \vec{I}_{t-1}|\phi_t, \phi_{t-1})p(\phi_t, \phi_{t-1})}{p(I_t, \vec{I}_{t-1})}$$

We observe the appearance of the temporal prior $p(\phi_t|\phi_{t-1})$. We transform the above and apply Bayes' rule to $p(\vec{I}_{t-1}|\phi_{t-1})$, eventually yielding

$$p(\phi_t|\vec{I_t}) = \kappa p(I_t|\phi_t) \int p(\phi_t|\phi_{t-1})p(\phi_{t-1}|\vec{I}_{t-1})d\phi_{t-1}$$

A complete proof is given in Sidenbladh[14]. Many of the terms in the above relation have individual significance.

- $p(\phi_t|\vec{I_t})$ is the *posterior* distribution. In words, the posterior represents the probability of model configurations given the input image sequence $\vec{I_t}$.

- $p(I_t|\phi_t)$ is the *likelihood* distribution. We carefully examine the likelihood term later in this document. Essentially, given a model parameterization ϕ_t, the likelihood specifies the probability that one would see the image I_t.

- $p(\phi_t|\phi_{t-1})$ is the *temporal prior* distribution. The prior distribution "predicts" a subsequent model configuration ϕ_t conditioned on a previous model configuration ϕ_{t-1}. The prior distribution used in our implementation is basic, and

there is much room for improvement here.[2]

- $p(\phi_{t-1}|\vec{I}_{t-1})$ is the posterior at time $t-1$. This expands recursively until $t=0$, at which point the model configuration is a known quantity.

1.5 Particle Filtering

We want to model the posterior, $p(\phi_t|\vec{I}_t)$. We note that, in the relation above, the posterior for time t is dependent on the posterior for $t-1$. The recursive definition stops at time t_0, as the initial configuration is given to us. We model the posterior distribution as a particle set[6], and run CONDENSATION[8], beginning at time t_0.

Given a set of k particles, we associate a parameterization ϕ_t^s with the sth particle at time t. We also store a weight at each particle which indicates its importance in the larger distribution. These weights are normalized such that the weights of all particles sum to 1.

For each time step t, we perform the following operations:

1. Take k samples $\left\{\phi_{t-1}^1, \cdots, \phi_{t-1}^k\right\}$ from the posterior distribution at time $t-1$. (To sample from the particle set, we pick a random number between 0 and 1, then find the particle whose weight occupies that space in the cumulative particle weight distribution)

2. Propagate each particle through the prior distribution to generate k particles $\left\{\phi_t^1, \cdots, \phi_t^k\right\}$.

3. Propagate these k particles through the likelihood $p(I_t|\phi_t)$.

4. Assign a non-normalized weight to each particle based upon its image likelihood and the probability of the prior propagation.

5. Normalize all weights and repeat this process for time $t+1$.

This is the essence of Isard and Blake's CONDENSATION algorithm, cited above.

[2]Sidenbladh, Black, and Fleet present a more intelligent prior distribution in their ECCV2000 paper[16]

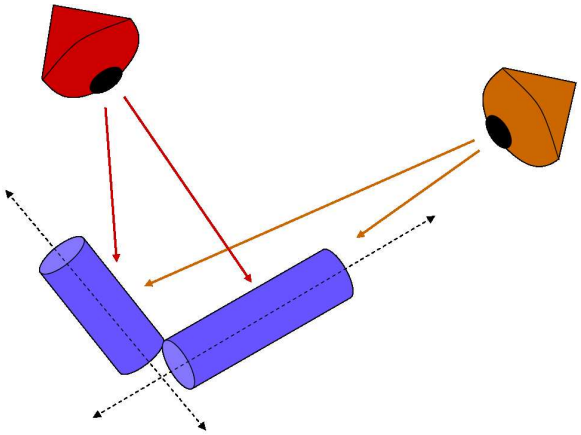


Figure 1.4: **The advantages of multiple cameras:** With multiple cameras, we are more likely to find a good view of each limb in a body parameterization.

1.6 Tracking with Multiple Cameras

In monocular tracking, depth ambiguities are among the most prominent difficulties. With only one camera, any motion perpendicular to the film plane will be difficult to detect. Since we are given multiple calibrated cameras, many depth ambiguities can be resolved.[9][5] However, in some cases, only one camera has a non-occluded view of a limb, and in such circumstances we are reduced to the monocular case.

We have established that multiple cameras increase the likelihood of observing limb motion in the film plane. Additionally, our likelihood determination will be more accurate when we view limbs from an amenable angle. When tracking, we boost the relative importance of limbs which are most visible from any given camera. This topic is addressed in more detail in Section 3.2.1.

Chapter 2

Taking Image Measurements

The Bayesian approach to human motion estimation places a heavy burden upon the likelihood term, $p(I_t|\phi_t)$. Without a robust likelihood model, tracking will break down quickly; when images correlate with a candidate body parameterization (which is stored in a particle), it is essential that the likelihood model reward such a parameterization. Moreover, the likelihood model must penalize a body parameterization that is implausible with respect to an image of the subject at the appropriate moment.

The posterior distribution — at least from the standpoint of particle filtering — should exhibit the following characteristics:

- Local maxima should appear for parameterizations which project a maximal number of limbs onto limb-like structures in the image
- The likelihood for the entire model should not penalize too harshly for a single unlikely limb. The limb can be occluded by an object extrinsic to the model itself (like a tree or a desk), or turned away from the camera.
- If most but not all of the model is aligned properly, the particle should not be given an excessively low likelihood; it should just be measurably less likely than a better parameterization with greater image correlation.
- As parameterizations gradually vary from the ideal to utter misalignment, the associated likelihoods should *gradually* tend towards a minimum. If the transition is too sudden, the particle filter will not naturally converge towards the ideal parameterizations and will instead have to find such parameterizations

through stochastic good fortune. In a search space as large as the human body, this is not practical.

2.1 Measurement Functions

Any likelihood model clearly must consider the image I_t in order to estimate $p(I_t|\phi_t)$. Our likelihood model creates dynamic mappings based upon simple scalar measurements extracted from an image or images. The likelihood model can be broken down into two distinct steps:

1. Use an image and a model configuration to make a *measurement*.
2. Use such a measurement, conditioned on past measurements, to determine a non-normalized log likelihood.

We factor out the second step by making certain assumptions about the measurements attained in the first step. Namely, we assume the following:

- All measurements are real numbers between 0 and 1.
- A larger measurement value indicates a greater likelihood for the given sample.

We reformulate the likelihood $p(I_t|\phi_t)$ in terms of a measurement function $m(I_t)$ as $p_m(m(I_t, \phi_t)|\vec{m}_k)$. We define the vector $\vec{m}_k$ as the last k measurements from the measurement function m. We define $\vec{m} = \vec{m}_\infty$ In most respects these measurements behave like non-normalized likelihoods, but it is more productive to think of them as energy values. In Section 3.1 we discuss the process of likelihood determination given a set of outputs from a measurement function. First, we must expand our discussion of the measurement functions themselves.

2.1.1 The Appropriate Scale of a Measurement Function

In order to determine the final probability $p(I_t|\phi_t)$, we attempt to find independent subsets of I_t — call them $I_{t1}, I_{t2}, \cdots, I_{tk}$ — and compute

$$p(I_t|\phi_t) = p(I_{t1} \cup I_{t2} \cup \cdots \cup I_{tk}|\phi_t) = p(I_{t1}|\phi_t) \cdot p(I_{t2}|\phi_t) \cdots p(I_{tk}|\phi_t) \quad .$$

In Sidenbladh[14], the "independent" likelihood terms each consider only a pixel in the original I_t; however, for most measurement criteria, the likelihoods of neighboring pixels are not independent. When measuring oriented edge detection, for instance, a strong response from one pixel on the edge of a proposed limb location is certainly dependent upon the edge likelihoods of other pixels incident to the proposed limb silhouette. Consequently, merely taking the product of the per-pixel probabilities is not sufficient.

In fact, it is more correct to assume the opposite: namely, that all the samples are *completely* dependent. In this case, we must take the kth root of the composite formulation of p above:

$$p(I_t|\phi_t) = p(I_{t1} \cup I_{t2} \cup \cdots \cup I_{tk}|\phi_t) = (p(I_{t1}|\phi_t) \cdot p(I_{t2}|\phi_t) \cdots p(I_{tk}|\phi_t))^{\frac{1}{k}}$$

This is a better approximation than our first attempt, but ultimately it also is theoretically incorrect and ineffective in practice; a measurement at one place on a projected limb does not necessarily predict all other measurements for that limb projection. So, if we are to break I_t into pixel-level subsets, we can neither assume complete independence nor complete dependence. The exact probabilistic interdependence between the individual likelihoods $p(I_{t1}|\phi_t), \cdots, p(I_{tk}|\phi_t)$ cannot be feasibly computed.

We step around this problem by recording our discrete measurements at the scale of individual limbs, and not at the level of individual pixels. This, too, is not theoretically ideal, as individual limb likelihoods are also neither entirely independent nor entirely dependent; however, this formulation is an improvement. The number of probabilistic approximations made is significantly smaller, and as such this approach may have greater theoretical viability. The prohibitively large number of incorrect dependency estimations while evaluating likelihoods per-pixel results in an eccentric posterior distribution for the particle set. Sections 3.2.3 and 4.2 address this topic in greater detail.

One could potentially take measurements at the scale of a single particle; in this case, we would be examining only the trivial subset of I_t, so there would be no presumption of dependence or independence. However, it is impractical to model this mapping while only considering the low-dimensional output of a measurement function (which in turn considers the high-dimensional space of images and body

configurations). This approach would require measurement criteria far more advanced and stateful than those presented in this document.

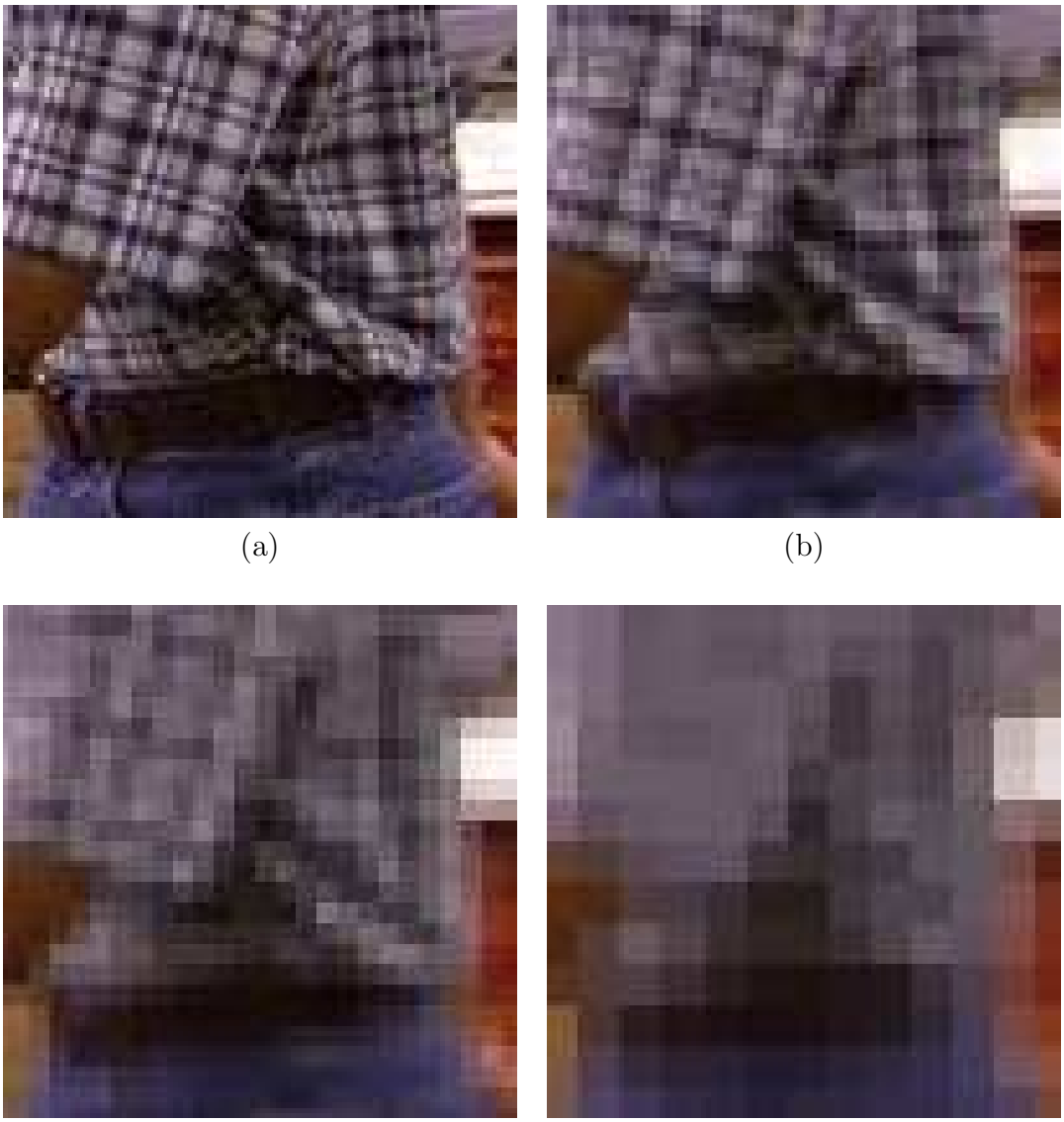
2.1.2 Image Pyramids

All input frames are treated as "image pyramids." Instead of merely storing the given raster data, the images build self-representations at various scales. Each successive scale has half the linear resolution of the previous scale and represents a Gaussian blur of its lower neighbor in the pyramid. Many measurement functions generate more meaningful results at higher pyramid levels, as the successively "lower" low-pass filtering helps to compensate for insufficiencies in the body model or, to a lesser extent, the camera projection. (See Figure 2.1)

2.1.3 Occlusions

Many limbs in any given body configuration will be either partially or entirely occluded by other limbs or objects extrinsic to the model (tables, chairs, trees, et cetera). It is difficult to determine the appropriate "measurement" for such limbs; an occluded object has an undefined appearance. The model as a whole must still have a likelihood, so we cannot simply discard such cases. Assigning a measurement of 0 (or any other constant) would also be incorrect, as a hidden limb is not necessarily in the wrong place — such an action would favor limbs that are visible over limbs that are not, though the visible versions may not line up well with image features.

Such limbs and points should be treated as if they were neither particularly likely nor unlikely, but completely average and unremarkable; this way, a limb will not be rewarded or penalized if partially or fully occluded. We elaborate on this in Section 3.1.4; to summarize, we take the median value of recent outputs for the given measurement function over the given image set.



(a) (b)

(c) (d)

Figure 2.1: **Images at Multiple Scales:** In (a) we see the original source image at the lowest "pyramid level". In (b), (c), and (d), we see successively higher pyramid levels. In this work, we never consider pyramid levels coarser than that shown in (d).

2.2 Specific Measurement Criteria

Now that we have established the role of the measurement function in the likelihood formulation, we will examine several measurement criteria in depth. As discussed in Section 2.1, the likelihood term $p(I_t|\phi_t)$ can be rewritten as $p(m(I_t, \phi_t)|\vec{m}_k)$, where $m(I_t, \phi_t)$ is a measurement function. (The likelihood $p(m|\vec{m}_k)$ is addressed more carefully in Section 3) In our case, m must be a mapping to a scalar from the space of parameterizations ϕ and images I. It would be reasonable to define the codomain of m as a low-dimensional vector space (as mentioned in the previous subsection), though for the sake of simplicity we do not do so.

As mentioned previously, all measurements are taken per-limb. Each limb is parameterized by some [potentially trivial] subset of ϕ_t and a number of static parameters. These static parameters — length, non-uniform scaling factors, radii, fixed translational offsets, et cetera — define the general appearance of the limb, and the dynamic parameters ϕ_t ordain its position and orientation in the world. Each image I_t is associated with a specific camera, and the respective camera parameters are used to project the limb into the image plane. Because the limbs are modeled as tapered cylinders, the silhouette along the body of the limb appears as a pair of projected line segments.

Each measurement function is given both the image and the limb location in image space. Information about occlusions is also included in the limb description, as many points on the limb may not be visible due to self-occlusions in the model. Some measurement functions consider only points along the side silhouettes of the limb, and others consider a larger collection of points on the interior of the limb projection.

2.2.1 Oriented Edge Detection

Given the 2-dimensional projected shape of a limb, we can look for edges in the image corresponding to the lateral edges of the limb in question. This is perhaps the most obvious measurement, and one of the first utilized in model-based human motion tracking.[12] We perform this task at multiple pyramid levels for a number of samples distributed evenly along the limb projection silhouette.

In particular, we employ a "steered" edge detector. This routine examines the

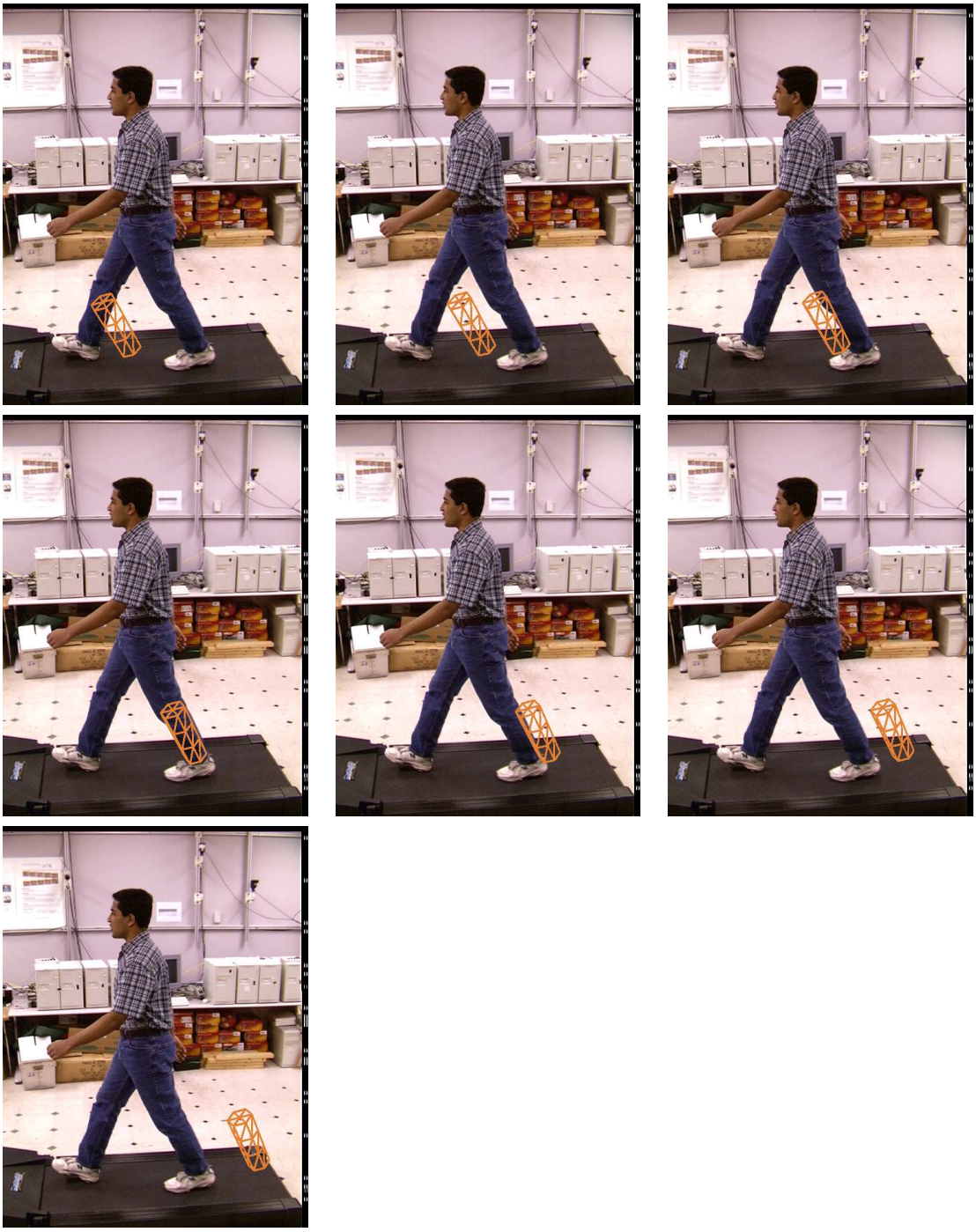


Figure 2.2: **Trial measurement locations:** The positions shown here correspond to the domain of the measurement function plots. An ideal measurement function would show a peak near frame 4. (This corresponds to domain value 240) For an example of such a plot, see Figure 2.3

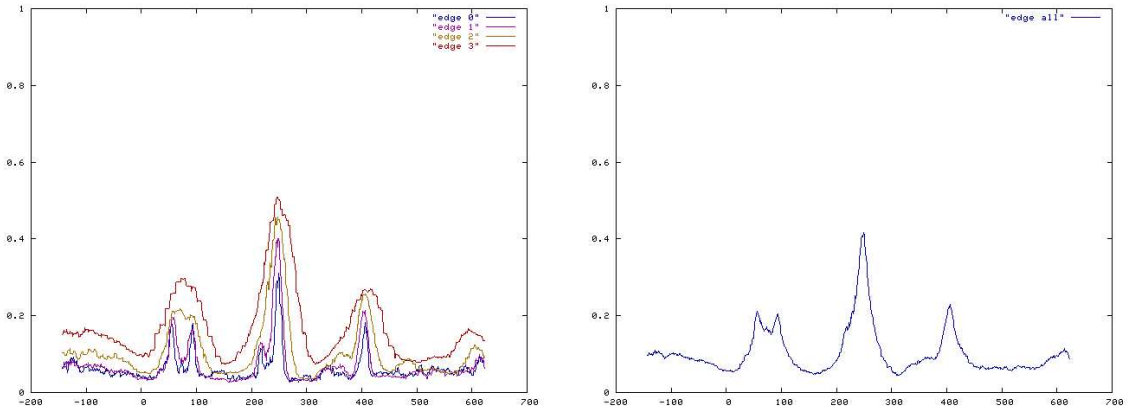


Figure 2.3: **Oriented edge measurement results:** Both plots correspond to oriented edge measurement values along the path illustrated in Figure 2.2. The plot on the left demonstrates the variation in response strength and character across pyramid scales. The plot on the right combines measurements from all pyramid levels. The peak at the center is the actual limb location in the image, and the false peaks at either side are the result of a "half-detection" when the right edge of the projected limb lay on the left edge of the actual limb (and vice versa)

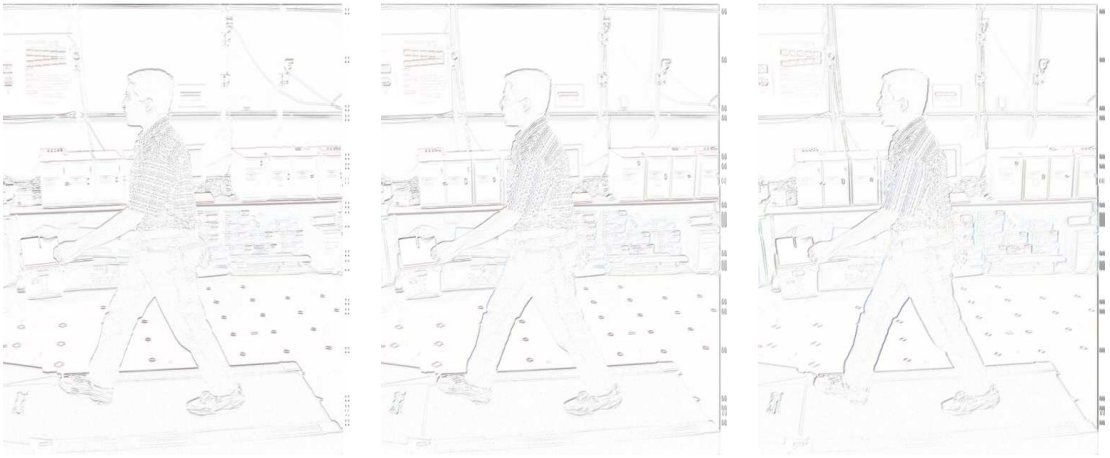
gradient of the image with respect to an edge angle θ. Specifically, for a pyramid level σ and the subsequent image I^σ, we designate its first partial derivatives of image brightness as I_x^σ and I_y^σ. (See Figure 2.4) Given an orientation θ, we can determine the steered edge measurement g at a position (x, y) in the image using the following formula:[1]

$$g(x, y, \theta, I^\sigma) = \sin(\theta) I_x^\sigma(x, y) - \cos(\theta) I_y^\sigma(x, y)$$

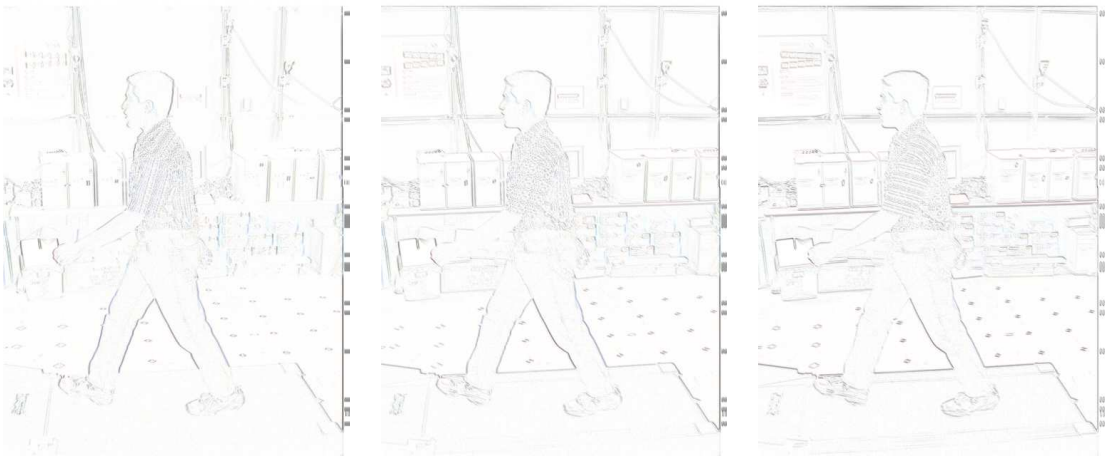
We do not take a fixed number of samples per limb, but take a fixed number of samples per unit length along the lateral silhouette edges of the limb. (See Figure 2.5) We take the absolute value of each computed sub-measurement g along a limb, then divide by the number of samples taken.

In general, the edge detection is more effective at the higher pyramid levels. Because adjacent pixels are not as similar near the top of image pyramid, edge responses are larger. Additionally, limb representations are rarely bordered by perfectly straight

[1]When using images with multiple color channels, we apply $g(x, y, I^\sigma)$ to all channels and choose the channel component with the maximal absolute value



(a) (b) (c)



(d) (e) (f)

Figure 2.4: **Oriented edge detection:** Images (a) through (f) demonstrate oriented edge measurements at the lowest pyramid level given an identical source image and a changing parameter θ. (The pixel intensities are negated for printability) Specifically, $\theta = \left\{0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}\right\}$ in the images (a) through (f) respectively.

(a) (b)

Figure 2.5: **The constancy of measurement frequency in one-dimensional image space:** Both (a) and (b) show the actual pixel locations of the individual oriented edge sub-measurements for a limb parameterized further and closer to the camera respectively. The number of samples taken in the image per unit distance is constant.

lines in the actual image, and thus the coarser image scales better model the uncertainty of appearance in the models.

The final edge measurement, for a limb, is a function of the limb projection L, the image I, and its first partial derivatives of image brightness. κ represents the maximum possible value of r, and division by $c\kappa$ guarantees that the final measurement value lies between 0 and 1. We define the measurement function as follows:

MEASURE-EDGE(L, θ, I)

1 $\theta \leftarrow$ The angle of the limb silhouette edge
2 $a \leftarrow 0$
3 $c \leftarrow 0$
4 **for** $\sigma \leftarrow \{1, 2, 3, 4\}$
5 **do for** Sample points (x, y) on lateral silhouette of L
6 **do if** (x, y) is not occluded
7 **then** $c \leftarrow c + 1$
8 $a \leftarrow a + g(x, y, I^{\sigma})$
9 **return** $\frac{a}{c\kappa}$

Note that c will equal zero if (and only if) all sampled points are occluded. In this case, the median measurement is assigned to this limb. (More detail can be found in Section 3.1.4)

Ignoring any occluded points, this algorithm reduces a summation over n positions $(x, y) \in \text{silhouette}(L)$:

$$\textsc{Measure-Edge}(L, \theta, I) \approx \frac{1}{4n\kappa} \sum_{\sigma=1}^{4} \sum_{(x,y)\in \text{sil}(L)} \sin(\theta) I_x^\sigma(x, y) - \cos(\theta) I_y^\sigma(x, y)$$

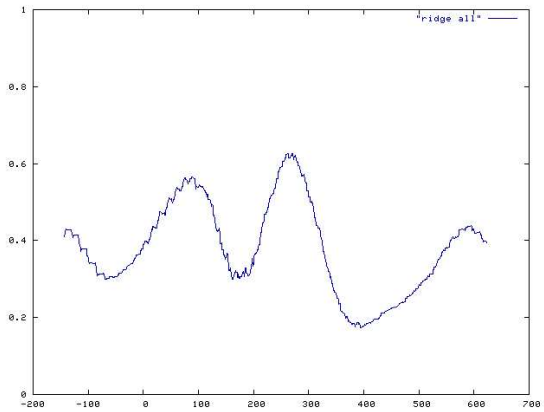


Figure 2.6: **Oriented ridge measurement results:** The plot corresponds to oriented ridge measurement values gathered along the path illustrated in Figure 2.2. Because the oriented ridge measurement is scale-specific, there are not separate plots for each pyramid level. As seen in the oriented edge measurement plot, there are notable false positives on either side of the actual leg in the image.

2.2.2 Oriented Ridge Detection

Where edge detection examines the first partial derivatives of an image, ridge detection examines the second partial derivatives. "Ridges" are bumps in the brightness function of the image. We want a way to detect such bumps of the appropriate scale and orientation for any given limb in our image. The ridge measurement is only valid for a single scale. There are two steps: finding the appropriate scale, then taking the oriented second derivative.

Sidenbladh[14] determined the following formula to compute the image scale factor s for a limb of pixel width w:

$$s = \textsc{Max}(0, -24.0 + 4.45 \cdot w)$$

We then determine the appropriate 0-indexed pyramid level σ according to the following formula:

$$\sigma = \log_4(3 \cdot s + 1)$$

Now, given a pyramid level σ and thus the image I^σ, we designate its second partial derivatives of image brightness as I^σ_{xx}, I^σ_{xy}, and I^σ_{yy}. Given an orientation θ, we can determine the steered ridge measurement r at a position (x, y) in the image

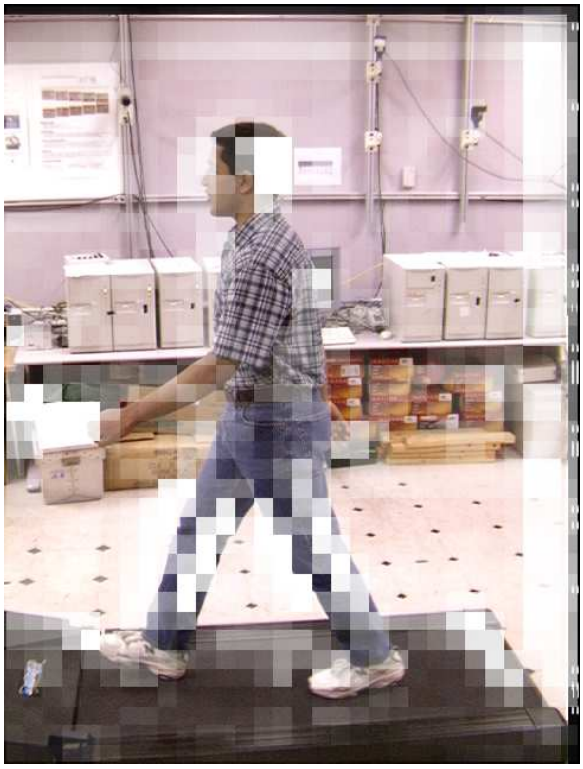


Figure 2.7: **Oriented ridge detection:** Here we see a source image with the oriented ridge response superimposed in white. The measurement is parameterized by the left calf. The calf responds well, as we would hope. Though it is difficult to see, there are also strong false-positive measurements in the background at either side of the limb.

using the following formula:[2]

$$g(x, y, \theta, I^\sigma) = \left|(\sin^2(\theta))I^\sigma_{xx}(x, y) + (\cos^2(\theta))I^\sigma_{yy}(x, y) - 2\sin(\theta)\cos(\theta)I^\sigma_{xy}\right|$$

In Sidenbladh[14] and Lindeberg[10], the ridge measurement includes an analogous subtractive term oriented at $\theta + \pi/2$. In theory, this term prevents the ridge response from misidentifying blobs as ridges in the image. Unfortunately, many legitimate ridges are also weakened or eliminated in the process, and as such we do not employ the subtractive term.

As with background subtraction and template matching (which are discussed in subsequent sections), we take a fixed number of samples per unit area in the image.

[2]When using images with multiple color channels, we apply $g(x, y, I^\sigma)$ to all channels and choose the channel component with the maximal absolute value

(See Figure 2.9) Thus, limbs which appear larger in projection will be sampled more carefully. κ represents the maximum possible value of r; dividing through by $c\kappa$ ensures that the final measurement value lies between 0 and 1. So, given a limb L of width w, an image I, and its second partial derivatives of image brightness, we perform the ridge measurement as follows:

MEASURE-RIDGE(L, θ, I)

1 $\theta \leftarrow$ The angle of the limb silhouette edge
2 $a \leftarrow 0$
3 $c \leftarrow 0$
4 $\sigma \leftarrow \log_4 (3 \cdot (\text{MAX}(0, -24 + 4.45 \cdot w)))$
5 **for** Sample points (x, y) uniformly across L
6 **do if** (x, y) is not occluded
7 **then** $c \leftarrow c + 1$
8 $a \leftarrow a + r(x, y, I^\sigma)$
9 **return** $\frac{a}{c\kappa}$

As with the oriented edge measurement function, the median measurement is assigned if $c = 0$ at procedure termination. This algorithm also reduces to a summation over n positions $(x, y) \in \text{silhouette}(L)$ when we ignore occlusions:

$$\text{MEASURE-RIDGE}(L, \theta, I) \approx \frac{1}{4n\kappa} \sum_{(x,y)\in \text{sil}(L)} g(x, y, \theta, I^\sigma)$$

where σ is defined as above.

2.2.3 Background Subtraction

For stationary cameras, we can use the static elements in a scene to help us determine what is foreground and what is background in a given image. Since the visible portions of the model will necessarily be in the foreground, background subtraction is an excellent way to prune the parameter space. Results improve with the incorporation of background subtraction.[3]

For our test data, background images — images where the subject is not present — were provided. Such images could be generated automatically given enough footage of a scene; for each pixel, one could compute the median intensity value over time and consider this to be the background. In either case, we can test whether any pixel in the source image differs significantly from the corresponding pixel in the background image. The background subtraction criterion presumes that the areas of difference between the source and background images correspond with some portion of the model projection. Given that presumption, a background image can easily rule out many model configurations. (Figure 2.8 shows the results of per-pixel background subtraction)

The measurement process is straightforward. A series of locations are chosen uniformly[3] across the limb projection. At each sample location, the brightness value of the image at that location is compared to the respective brightness value of the background. If these values lie within an experimentally-discovered threshold of each other, then an accumulation value is incremented by a value i inversely proportional to the number of samples taken at non-occluded positions. (i is, specifically, the reciprocal of the number of samples taken. This guarantees that the measurement function will not generate numbers larger than 1, as per the constraints outlined above)

Where the oriented edge measurement function only examined pixels incident to the projected silhouette edges of the test limb, the background subtraction measurement function examines pixels *inside* the [convex] silhouette. In fact, the edge pixels are of the least meaning, as this is where the tapered-cylindrical limb model is weakest.

As with the oriented ridge criterion, the background subtraction measurement

[3]The process is outlined in greater detail in Section 2.2.4

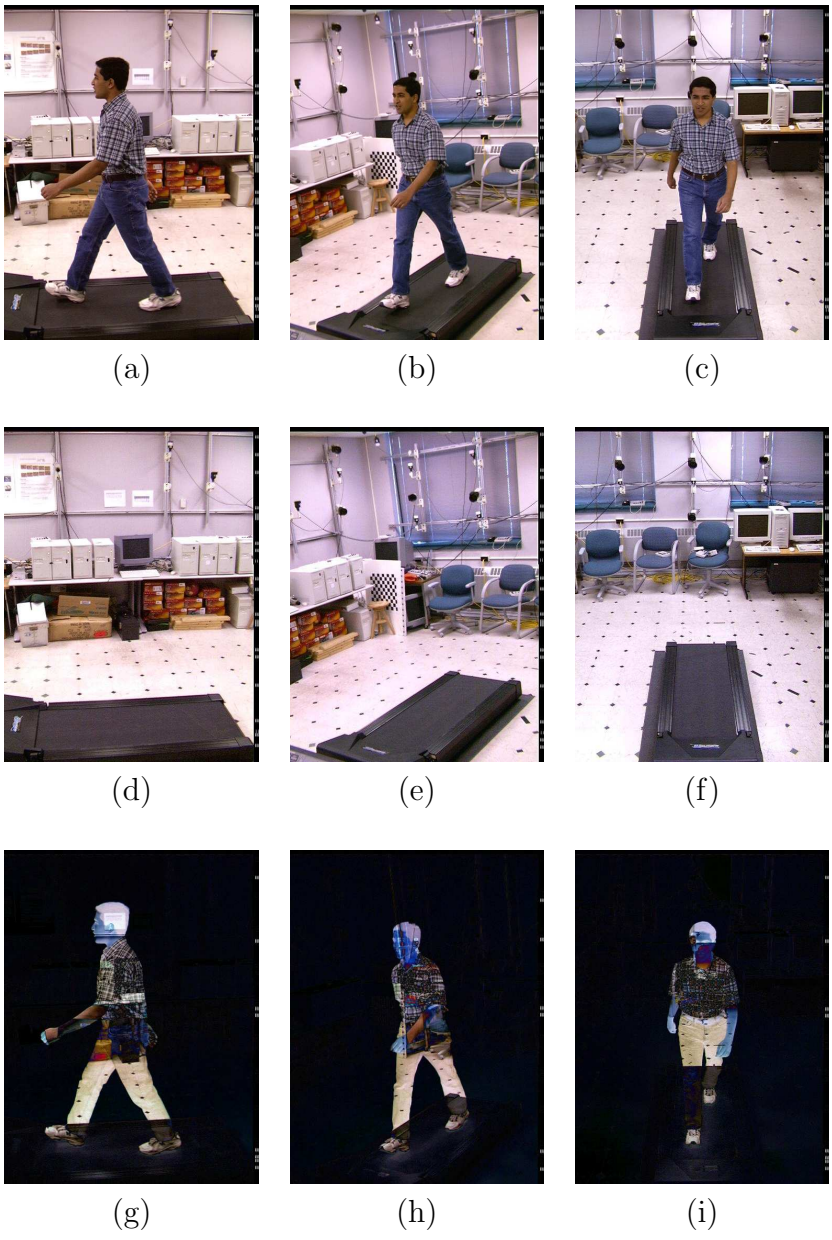


Figure 2.8: **Background subtraction per camera:** Images (a), (b), and (c) show the first of many input images from each of the three cameras used in this dataset. Images (d), (e), and (f) show the [provided] background images from these three cameras, and (g), (h), and (i) represent the negated difference between (a)/(b)/(c) and (d)/(e)/(f) respectively. Note that portions of the background-subtracted images — like the right calf in (i) — appear nearly as white as the background proper.

(a) (b) (c)

Figure 2.9: **The constancy of measurement frequency in two-dimensional image space:** Images (a), (b), and (c) show the actual pixel locations of the individual background subtraction sub-measurements for a limb parameterized further and then increasingly closer to the camera. The number of samples taken in the image per unit area is constant.

function takes a number of samples proportional to the projected area of the given limb L. (See Figure 2.9) In other words, limbs with smaller projection areas using a given camera will have fewer terms in their measurement summation, and thus have a smaller computational footprint. Likewise, larger projection areas earn more computational resources and thus a more accurate estimation given the greater input fidelity.

Specifically, given a limb L, an image I, a background image B, and a threshold ν, the background subtraction measurement is defined as follows:

MEASURE-BGSUBTR(L, I, B)

1 $a \leftarrow 0$
2 $c \leftarrow 0$
3 **for** $\sigma \leftarrow \{1, 2, 3, 4\}$
4 **do for** Measured points (x, y) on body of L
5 **do if** (x, y) is not occluded
6 **then** $c \leftarrow c + 1$
7 **if** $|I^{\sigma}(x, y) - B^{\sigma}(x, y)| > \nu$
8 **then** $a \leftarrow a + 1$
9 **return** $\frac{a}{c}$

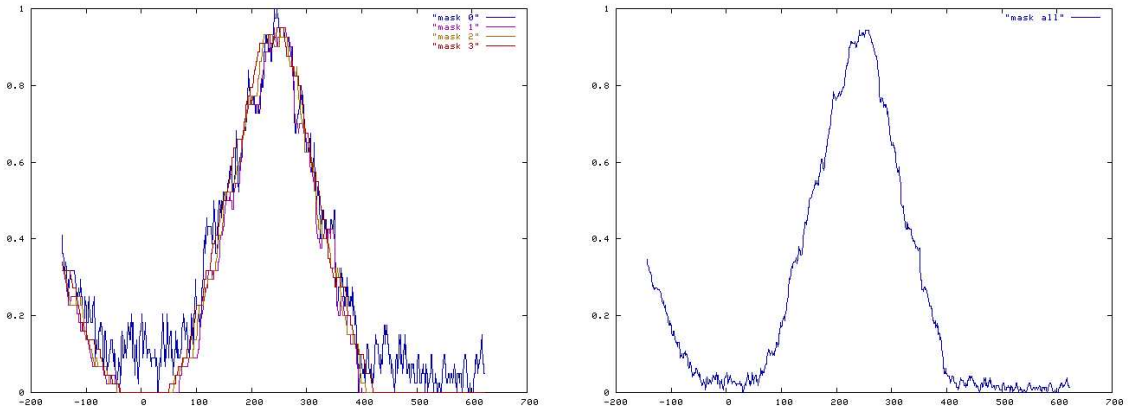


Figure 2.10: **Background subtraction measurement results:** Both plots correspond to background subtraction measurement values along the path illustrated in Figure 2.2. The plot on the left demonstrates the variation in response strength and character across pyramid scales. The noisy results at the lowest pyramid level are due to threshold contention. The plot on the right combines measurements from all pyramid levels. There is a noteworthy false positive as the measurement finds the left leg despite the incoherence of orientation.

As we have seen before, c will equal zero if (and only if) all sampled points are occluded. In this case, the median measurement is assigned to this limb.

Small variations in the CCD responses over time add noise to the images. The low-pass filtering that takes place in the image pyramid largely compensates for this, so it is important to consider all pyramid levels.

2.2.4 Template Matching

Like the background subtraction measurement function, the template matching measurement function examines points within the limb boundary, and not on the silhouette as in the edge and ridge measurements. Template matching is unique in that it examines two image/limb pairs: one at the current moment in time, and one from the first capture frame.

The template matching measurement assesses whether a given limb projection correlates with the same limb in the first frame; this initial image of the limb acts as a template, hence "template matching."[2] The computation is done at multiple scales, as small variations in shading and texture are less problematic at the higher pyramid levels.

Specifically, given an image at time t and an image at time t_0, we determine two limb projections, L_t and L_{t_0}. We parameterize both 2D limb projections in terms of scalars a and b, $0 < a, b < 1$. A pair of two-dimensional basis vectors $(\vec{u}_t, \vec{v}_t)$ and $(\vec{u}_{t_0}, \vec{v}_{t_0})$[4] are constructed for both limbs; $\vec{u}_t$ runs along L_t's silhouette edge, and $\vec{v}_t$ runs between the points a fraction a along L_t's two silhouette edges. $(\vec{u}_{t_0}, \vec{v}_{t_0})$ are defined similarly. We find points $(x_t, y_t) = a \cdot \vec{u}_t + b \cdot \vec{v}_t$ and $(x_{t_0}, y_{t_0}) = a \cdot \vec{u}_{t_0} + b \cdot \vec{v}_{t_0}$. A single component of the template matching measurement function at position (x_t, y_t) within L_t and (x_{t_0}, y_{t_0}) within L_{t_0} can be expressed as follows:

$$b(x_t, y_t, x_{t_0}, y_{t_0}, I_t, I_{t_0}) = 1 - \frac{|I_t(x_t, y_t) - I_{t_0}(x_{t_0}, y_{t_0})|}{\kappa}$$

The scaling factor κ is defined to be twice the maximal possible pixel intensity value; dividing by $c\kappa$ (below) guarantees that the final measurement lies between 0 and 1. The "1−" preserves the property of increasing likelihood: we want to reward limb configurations that show the least change between $I_{t_0}(x_{t_0}, y_{t_0})$ and $I_t(x_t, y_t)$; hence, a matching template.

For images I_t and I_{t_0}, limbs L_t and L_{t_0}, and the κ specified above, the complete template matching measurement function is defined as follows:

MEASURE-TEMPLATE-MATCHING$(L_t, L_{t_0}, I_t, I_{t_0})$

1 $a \leftarrow 0$

[4] In actuality, $\vec{u}_t$ and $\vec{v}_t$ are paired with an origin point P_t to define the projected limb coordinate system. For the sake of notational clarity, P_t will be elided in this discussion.

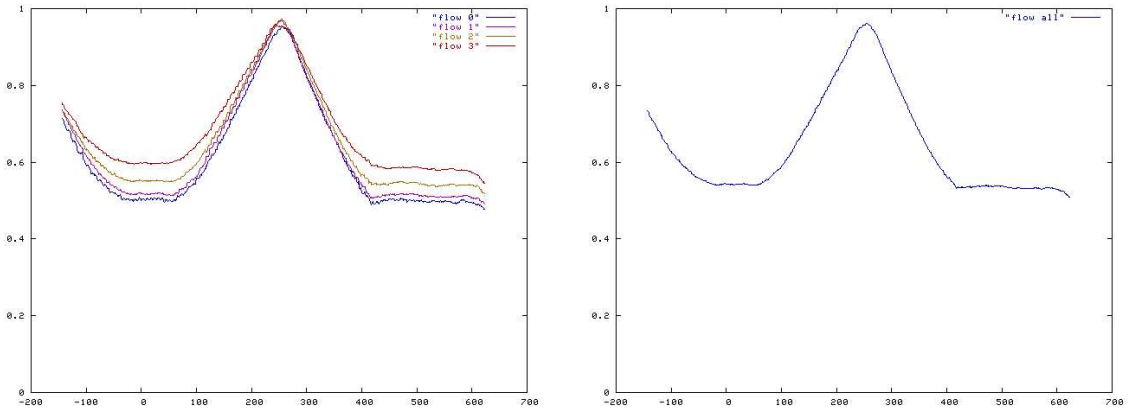


Figure 2.11: **Template matching measurement results:** Both plots correspond to template matching measurement values along the path illustrated in Figure 2.2. The t_0 frame is shown in Figure 2.2, and L_{t_0} is positioned over the right calf of the subject. The t frame (the second frame) appears many times in Chapter 4. The plot on the left demonstrates the variation in response strength and character across pyramid scales. The plot on the right combines measurements from all pyramid levels. As with the background subtraction measurement function, there is a noteworthy false positive as the measurement finds the left leg despite the incoherence of orientation.

2 $c \leftarrow 0$
3 **for** $\sigma \leftarrow \{1, 2, 3, 4\}$
4 **do for** Measured points (x_t, y_t) and (x_{t_0}, y_{t_0}) on body of L_t and L_{t_0}
5 **do if** (x_t, y_t) and (x_{t_0}, y_{t_0}) are not occluded
6 **then** $c \leftarrow c + 1$
7 $a \leftarrow a + b(x_t, y_t, x_{t_0}, y_{t_0}, I_t^\sigma, I_{t_0}^\sigma)$
8 **return** $\frac{a}{c}$

Again, if $c = 0$ at procedure termination, the median measurement value is used instead. Ignoring occlusions, for n position pairs in I_t and I_{t_0}, this measurement can be approximated by a simpler summation.

$$\textsc{Measure-Template-Matching}(L_t, L_{t_0}, I_t, I_{t_0}) \approx \frac{1}{4n} \sum_{\sigma=1}^{4} \sum_{(x_t, y_t, x_{t_0}, y_{t_0}) \in L_t, L_{t_0}} b\left(x_t, y_t, x_{t_0}, y_{t_0}, I_t^\sigma, I_{t_0}^\sigma\right)$$

2.3 Measurement Tradeoffs

All of the measurement criteria described thus far have relative strengths and weaknesses. The oriented edge and ridge measurement functions do not easily distinguish between limbs and other objects with straight sides, and the other measurement functions do not check limb orientation. In general, we want implausible model configurations to be cleanly and routinely rejected by at least one measurement model.

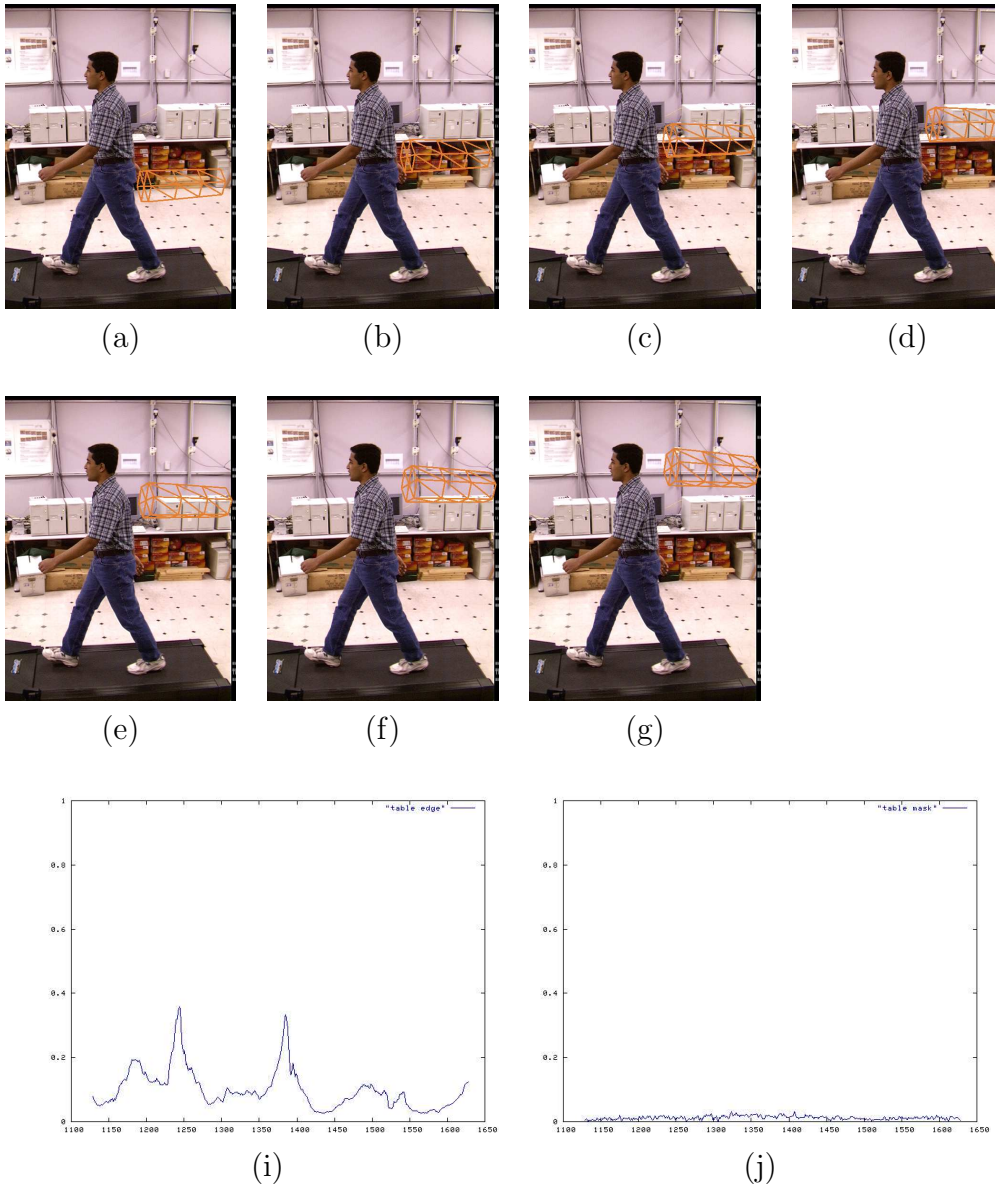


Figure 2.12: **Countertop misidentification:** The plot domain is illustrated by the limb positions in images (a) through (h). The oriented edge measurement function generated (i), and the background subtraction measurement function generated (j). There are prominent peaks in (i) due to the strong edge on the countertop. The background subtraction measurement function recognizes that these image areas are nearly identical to the background, and thus does not make a similar mistake. This illustrates the deductive capacity of background subtraction.

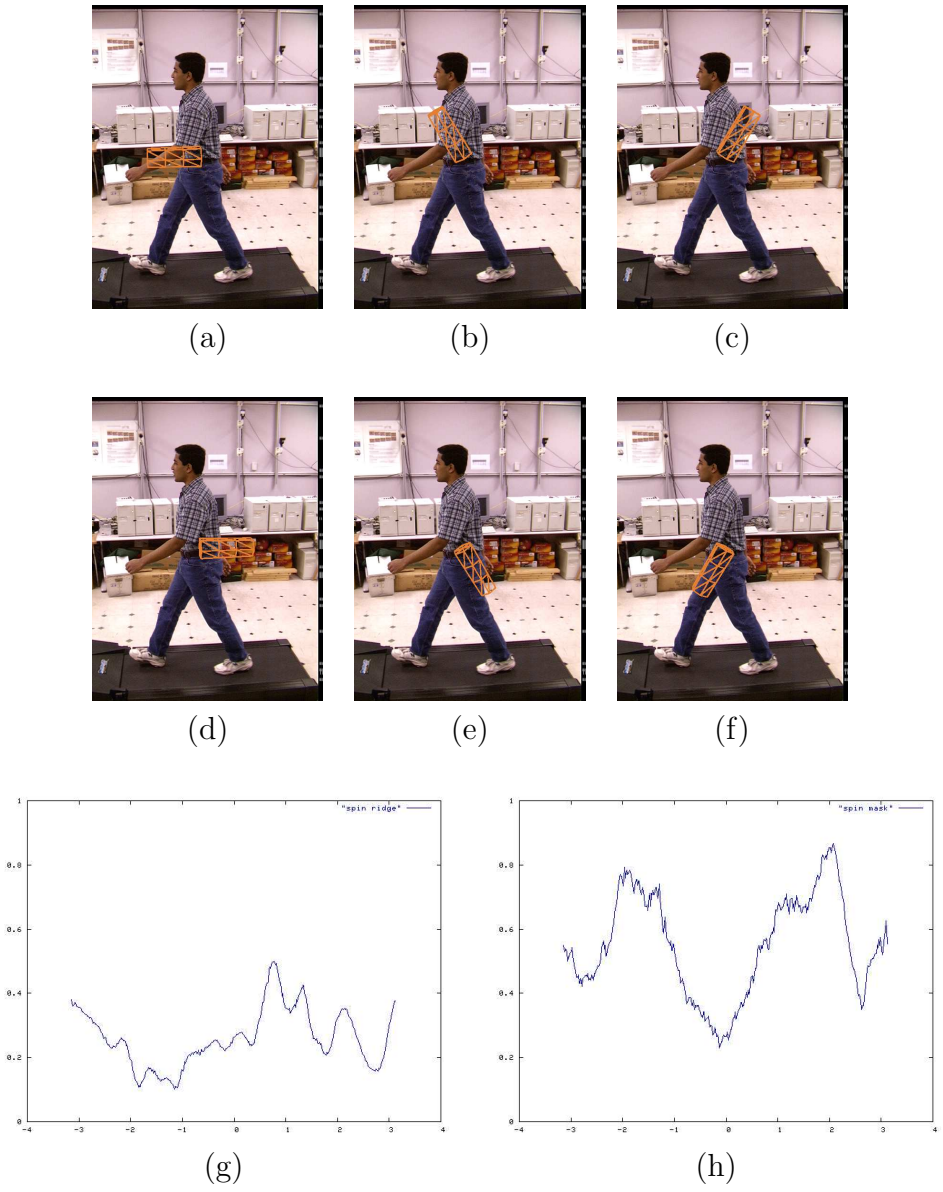


Figure 2.13: **Orientation misidentification:** The plot domain is illustrated by the limb positions in images (a) through (f). The oriented ridge measurement function generated (g), and the background subtraction measurement function generated (h). Because the spinning limb is positioned — at least partially — within the model torso, there are many prominent peaks in (h). However, since the limb is oriented incorrectly, the ridge measurement function does not respond significantly.

Chapter 3

Dynamic Likelihood Determination

In the previous section, we examined the notion of a generic measurement function based on some heuristic criterion. We can think of these measurements as energy values, but not as probabilities.[1] In order to eventually measure the relative likelihood of a given particle, we must devise a method to map from the raw measurements to probabilities. This was discussed at the beginning of Chapter 2; specifically, we want to define $p(m(I_t, \phi_t)|\vec{m}_k)$ for each distinct measurement function m.

In Sidenbladh[15], the likelihood was trained using hand-marked limb positions in a variety of test images. This technique requires significant user intervention, and in the attempt to model all images with one distribution, input sequences with unusual statistical properties (very bright, very dark, et cetera) will often yield inaccurate likelihood information. Partly for these reasons, and partly due to the lack of training data for the newly devised measurement functions, these trained likelihoods were not used in this work. With appropriate training data and an adequately automated marking process, the trained likelihoods would be a desirable addition to the tracking framework.

[1] In fairness, they do resemble probabilities in many ways: they must take on values between 0 and 1, and the higher values are generally more probable. However, no larger presumptions are made about the relative likelihood associated with a given measurement

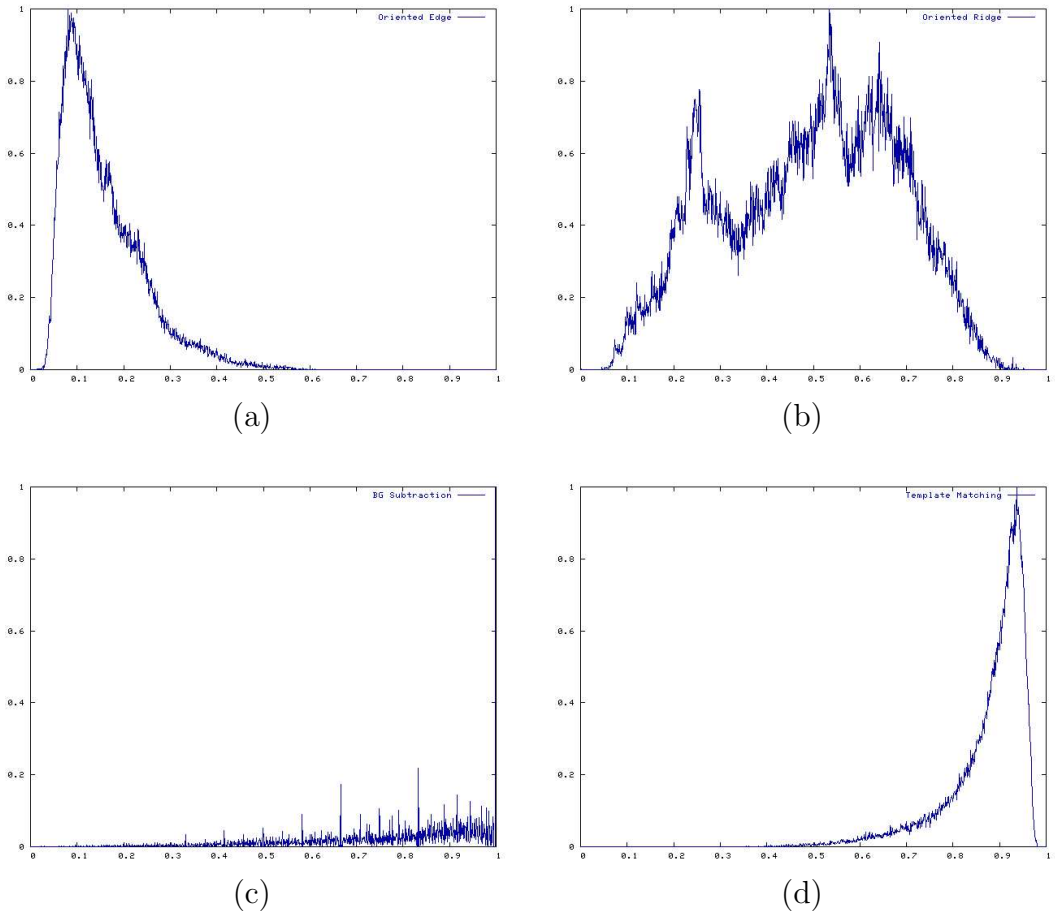


Figure 3.1: **Measurement function histograms:** Plots (a), (b), (c), and (d) represent the measurement function histograms for oriented edge detection, oriented ridge detection, background subtraction, and template matching respectively. These histograms were constructed after tracking three frames with 1000 particles.

3.1 Building a Distribution

As measurements are extracted from an image sequence, we build histograms from the return values of each measurement function. (See Figure 3.1) The codomain of the measurement functions — namely the range [0,1] — is broken into a number of histogram buckets. With each measurement, the appropriate histogram bucket counter is incremented by one. Over time, this builds an "energy distribution," which we must use to construct an appropriate likelihood mapping.

It is tempting to hand-craft a [potentially] intricate procedural solution to this

problem. A complicated piecewise likelihood mapping could be devised. However, it is certainly simpler, more efficient, and debatable more effective to build a Gaussian mapping over the measurement codomain. In order to define the Gaussian, we must, of course, determine the mean and variance μ and σ.

3.1.1 Finding the Mean

We remember that the "best" possible measurement for any measurement function will be given a value of 1. We thus infer that the likelihood must be maximized for any measurement $m = 1$. μ must be greater than or equal to all measurements; otherwise, the greatest measurement will not have the greatest probability. For the set of all measurements $M = \{m_1, m_2, \cdots, m_k\}$, the mean μ is defined as

$$\mu \geq \textsc{Max}(M)$$

The variance σ is less obvious. Variance determination will be discussed in Section 3.1.3.

3.1.2 False Positives and False Negatives

As demonstrated earlier, none of the measurements are perfect. With the eventual motivation of variance determination for the measurement-specific Gaussian distributions, it would be illuminating to address the topic of measurement failure.

Measurement functions return two forms of "bad data": they may either return a high value for a tested limb configuration that is not, in fact, projected onto the desired limb in the image, or the measurement function may return a low value when a tested limb configuration does, in fact, project onto or near the proper limb in the image. As one might intuit, we denote these two failure case classes as "false positives" and "false negatives" respectively.

Both false positives and false negatives beget undesirable tracking results. If we use a diverse set of measurement criteria, it is unlikely that all measurement functions will return a false positive for a bogus limb parameterization. Thus, false positives in one measurement function will likely be "recognized" as such by at least one of the other measurement functions. If any one measurement fails, the given particle

can (and will) be penalized by multiplication with a low probability for the given measurement.

That said, many of the strongest responses for some measurements lie on table edges or improperly rotated limbs. (See Section 2.3) Thus, false positives often result in disproportionately high likelihoods for "bad" particles. When the next set of particles are generated, these mistakes are "propagated"[2] forward at the expense of better parameterizations. False negatives are also damaging for self-evident reasons; if a measurement function is not able to recognize a valid limb parameterization, then tracking suffers.

The choice of variance for our likelihood mapping will be a tradeoff between false positives and false negatives. If our variance is too large, there will be little probabilistic difference between valid and invalid parameterizations. If our variance is too small, many good particles will be thrown away.

3.1.3 Finding the Variance

Variance determination is difficult to generalize. With the edge measurement function, we want to discard virtually all particles with measurements below a certain threshold. However, with the background subtraction, the top half of the measurement histogram is worth keeping; in that case, we need a comparatively larger variance.

We could specify a static variance for each measurement function. This would be an acceptable solution in many ways. However, such a scheme would be inadequate in certain situations:

- Many measurement functions will return results at different scales for different input image sets. For instance, edges will be less pronounced in images that are generally darker.

- If or when tracking breaks down in a sequence, we would like the variance to grow and the likelihood mapping to be more forgiving in an effort to sample

[2]The mistakes are actually made when sampling the the posterior, which is not a proper propagation.

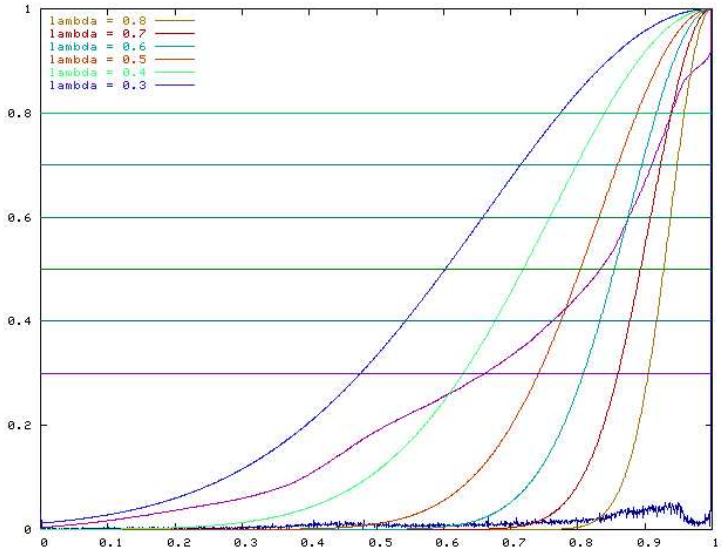


Figure 3.2: **Variance with respect to λ:** The non-decreasing function (shown in both plots) represents the cumulative plot of the measurement histogram values. The relationship between λ and σ can be seen in the plot above. The successively smaller Gaussian distributions correspond to λ values of 0.3, 0.4, 0.5, 0.6, 0.7, and 0.8. The horizontal lines corresponding to each λ value intersect the cumulative plot directly above the domain value σ measurement units from the mean $\mu = 1.0$.

more broadly from the posterior particle distribution. A fixed variance makes this an impossibility.

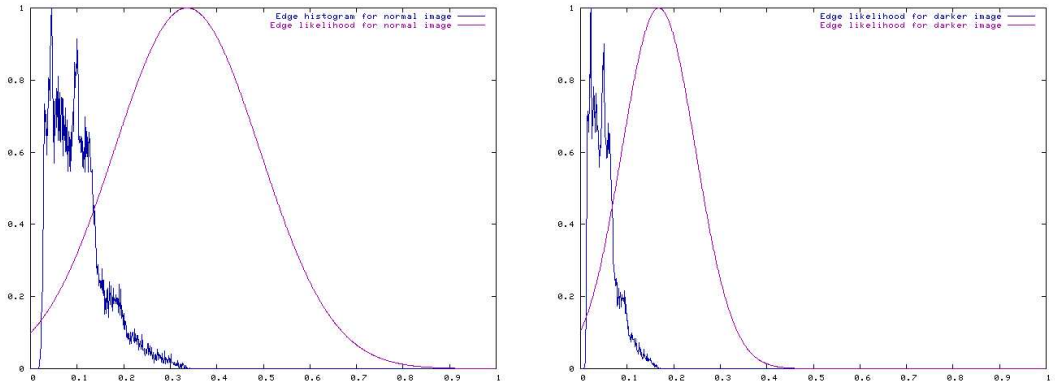
Still, there must be some user-dependent parameter to indicate the relative size of variances from distinct measurement likelihood mappings. In this work, we define the variance σ such that the top $(1 - \lambda)$ fraction of the measurements lie within σ of the mean μ. Thus, if $m_{\max}$ is the maximal measurement and m_λ is the smallest measurement such that a fraction λ of the measurements are smaller, the variance σ is defined such that

$$\sigma = m_{\max} - m_\lambda$$

So, for background subtraction, λ is relatively small, and for edge detection λ is relatively large. (See Figure 3.2) The heuristic λ has a significant effect on the eventual quality of tracking results. If set too low, the particle filter will waste particles on unlikely configurations; if set too high, the particle distribution becomes *eccentric* and tracking is consequently fickle. Using a particle filter to model the posterior, we define an *eccentric* distribution such that a relatively small number of particles hold a relatively large portion of the weight across all particles. An eccentric posterior expresses less information about the actual posterior, and is usually inaccurate. Finding an appropriate value for λ is an experimental process — our results can be found in Section 4.2. As we shall see, an appropriate choice of λ enables robust tracking through heterogenous data sets.

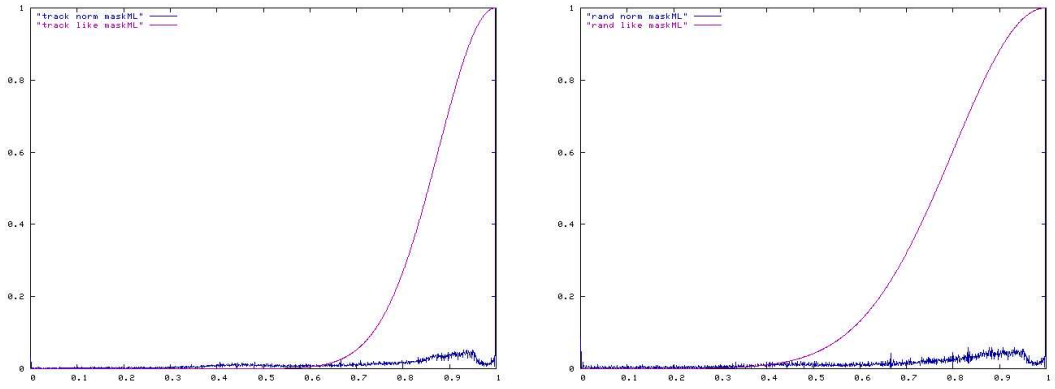
Adjusting to Changes in Image Statistics

The above method for variance determination is robust to certain changes in image statistics. In many situations, images from a single sequence will become darker, lighter, or noisier over time; in *most* situations, images taken from distinct tracking sequences will have remarkably different statistics. A dynamic mapping with a fixed variance would not be an effective method in this case; as images get darker, for instance, one would hope that the variance for edge detection would constrict automatically. (See Figure 3.3)



(a) (b)

Figure 3.3: **Oriented edge variance with respect to image brightness:** The domain in these plots are measurement values. The jagged plots are oriented edge measurement histograms, and the Gaussians are likelihoods — conditioned on the measurement histograms — for given measurement values. The plot in (a) shows the likelihood mapping for the oriented edge measurement given our normal test sequence as input. The plot in (b) shows the same distribution when the input images have half of their original brightness. The mean and variance dynamically adjust to the change in image statistics; the edges in the darker image will be less pronounced in the measurement function, but the oriented edge likelihood will not be affected.



(a) (b)

Figure 3.4: **Variance of likelihood mapping with respect to tracking fidelity:** Plot (a) shows the background subtraction measurement histogram and likelihood Gaussian for normal tracking. In (b), we propagate through the prior 5 times before taking the next set of measurements. This results in less accurate particle and limb proposals. The histogram is consequently more chaotic, and the variance of the likelihood mapping dynamically expands to compensate for the compromised tracking efficacy.

Adjusting to Changes in Tracking Efficacy

We would like our generated mappings to accommodate poor tracking results. If the tracking degrades or the prior distribution $p(\phi_t|\phi_{t-1})$ is unable to accurately predict body position, the measurement functions will generally return lower values. We would like to see the likelihood mappings broaden to increase the search space in this scenario.

To test this, we repeatedly propagate through the prior distribution for a test particle, thereby magnifying any inaccurate predictions in the prior. After a fixed number of samples from the prior, we take a measurement and add it to our histogram. We find that the variance of the given measurement's likelihood mapping increases with respect to "normal" sampling. Thus, as tracking fidelity suffers, the mapping adjust accordingly. (See Figure 3.4)

3.1.4 Details of the Measurement Likelihoods

To summarize, given a single measurement function m we can determine a likelihood (a log likelihood, more precisely) for that measurement by evaluating a Gaussian parameterized on past measurements and a heuristic λ. This mapping is defined with respect to a specific measurement function, but adjusts dynamically to changing image and tracking conditions.

In order to make this system efficient, we only recalculate the mean and variance periodically. The actual likelihood mappings are built upon the last k samples. Smaller k values allow for shorter periods of adjustment to changes in image statistics, et cetera, though the measurement histogram will not be well populated if k is too small.

If a measured limb is entirely occluded, no measurement can be faithfully extracted from the input image sequence. Given this case of total occlusion, we do not know if the limb is likely or unlikely. We assume that the measurement for the given limb is the median of all measurements taken thus far for the given measurement function.

$$m_{\text{occluded}}(L) = \textsc{Median}(\{\text{All previous measurements from } m\})$$

After assigning this measurement value, we evaluate the measurement likelihood

mapping as per usual. In the absence of other information we must guarantee that occluded limbs neither increase nor decrease the relative likelihood for their particle; assigning the median measurement is the theoretically appropriate mechanism for maintaining the relative likelihood of model configurations (i.e., particles) with and without occluded limbs.

3.2 Building the Distribution Across All Particles

We have shown how to compute a [non-normalized] likelihood for one measurement. This value is specific to a single measurement function, a single limb, a single camera, and a single particle. However, there are multiple distinct measurement functions, limbs, cameras, and particles. In Sidenbladh[14], log probabilities are accumulated for each particle and then normalized across the distribution afterwards. Sidenbladh also restricted input to monocular sequences and the scale of the individual measurements was smaller.[3] We instead form a particle distribution for each measurement, normalize those distributions, then combine to form a single cumulative distribution across all particles which we then normalize one last time. After that final normalization, we can sample from the cumulative distribution in the particle filter. Simply adding the probabilities together is not effective due to the differences in the Gaussian distributions for the distinct measurement likelihood mappings.

3.2.1 One Particle, One Measurement Function

We will approach this problem from the top down. We must calculate the likelihood — or, more specifically, the log likelihood — of each particle for each measurement function. We begin by outlining the "naive approach," and then go on to describe heuristics that generally improve tracking results.

[3] Measurements in this work are taken per limb, but in Sidenbladh[14] they are taken far more frequently; each examined point in the image had a one-to-one correspondence with an accumulated log probability. This distinction is discussed more carefully in Section 2.1.1

The Naive Approach

One could iterate over all limbs and accumulate the log likelihoods. The final log likelihood l for an image I, the set of limbs $B = \{b_1, b_2, \cdots, b_k\}$, and measurement function m would thus be

$$\log\left(l\left(I, B, m\right)\right) = \sum_{i=1}^{k} \log\left(\rho\left(\mu(m), \sigma(m)\right)\left(I, b_i\right)\right)$$

This does, in fact, work, but we can improve our results significantly by adding two relative limb weighting heuristics.

Manual Limb Weights

Some limbs are easier to detect than others. Namely, the outermost extremities are the easiest to spot. Additionally, if the calves, forearms, and head are in the right position, we expect that the torso, thighs, and upper arms are also approximately correct. Thus, we track with non-uniform limb weights. (See Figure 3.1)

We can formulate the original model likelihood, for the time being, as a product:

$$l\left(I, B, m\right) = \prod_{i=1}^{k} \rho\left(\mu(m), \sigma(m)\right)\left(I, b_i\right)$$

We then define a set of exponents $E_l = \{e_{l1}, e_{l2}, \cdots, e_{lk}\}$ such that

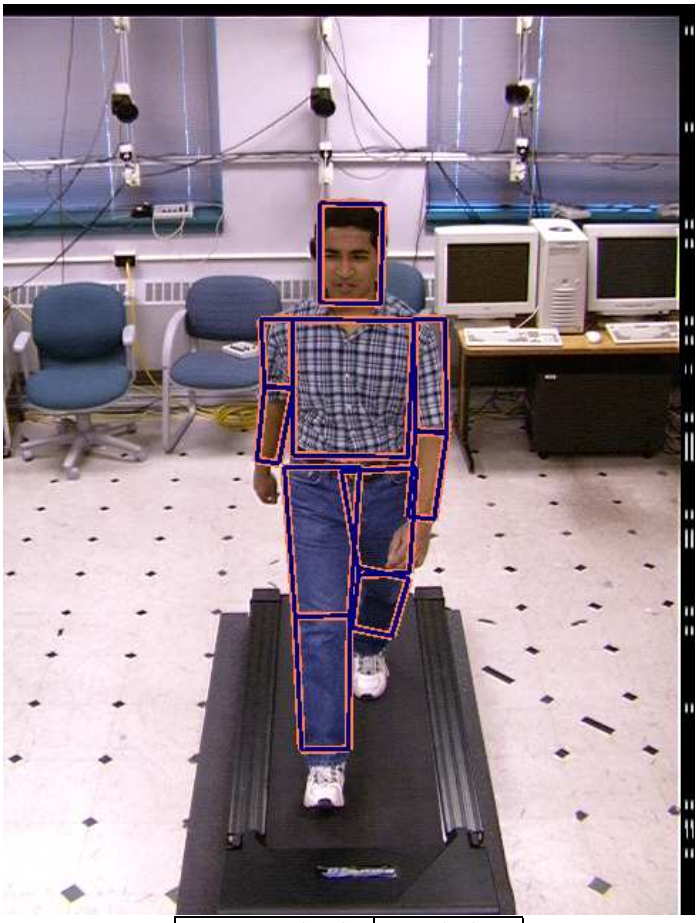
$$1 = \sum_{i=1}^{k} \frac{e_{li}}{k}$$

$$k = \sum_{i=1}^{k} e_{li}$$

We reformulate the original model likelihood as

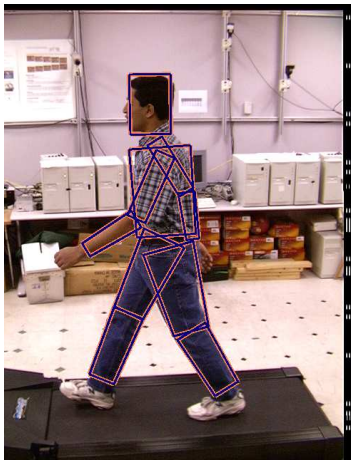
$$l\left(I, B, m\right) = \prod_{i=1}^{k} \left(\rho\left(\mu(m), \sigma(m)\right)\left(I, b_i\right)\right)^{e_{li}}$$

And observe that the original formulation is a special case, where $e_{li} = 1$ for all $1 \leq i \leq k$. This new formulation allows the user to specify the relative importance of various limbs. For instance, the head is crucial to successful tracking, whereas the torso is nearly impossible to track effectively with our measurement models; thus, we could assign a high value to e_{head} and a low value to e_{torso}. In practice, this heuristic substantially improves tracking results.



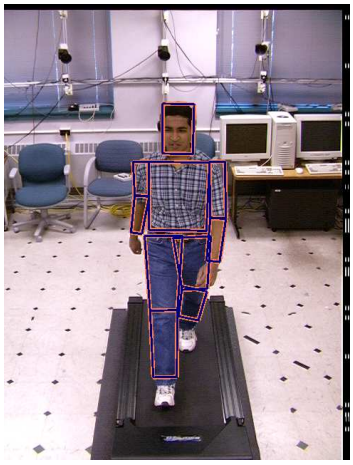
Limb	Weight
R. Thigh	0.392157
R. Calf	1.307190
L. Thigh	0.392157
L. Calf	1.307190
Torso	0.065359
R. Up. Arm	0.065359
R. Forearm	1.960784
L. Up. Arm	0.065359
L. Forearm	1.960784
Head	2.483660
Sum	10

Table 3.1: **A static limb weighting heuristic:** Each limb is assigned a static weight in the body model. These weights are then normalized such that the sum across all limbs is equal to the number of limbs in the model. The pre-normalized values were determined experimentally.



Limb	Weight
R. Thigh	1.055356
R. Calf	0.877345
L. Thigh	1.000584
L. Calf	0.665880
Torso	0.944177
R. Up. Arm	1.114166
R. Forearm	0.839794
L. Up. Arm	1.244135
L. Forearm	1.031666
Head	1.226896
Sum	10

Limb	Weight
R. Thigh	1.294253
R. Calf	1.434852
L. Thigh	0.871124
L. Calf	0.442202
Torso	1.093947
R. Up. Arm	0.723565
R. Forearm	0.910396
L. Up. Arm	1.114277
L. Forearm	0.768695
Head	1.346688
Sum	10



Limb	Weight
R. Thigh	1.756245
R. Calf	1.434011
L. Thigh	0.563740
L. Calf	0.254320
Torso	1.011449
R. Up. Arm	0.419396
R. Forearm	0.864819
L. Up. Arm	1.578187
L. Forearm	0.791437
Head	1.326396
Sum	10

Table 3.2: **A view-dependent limb weighting heuristic:** Each limb is assigned an initial weight of $1 - |\cos(\theta)|$, where θ represents the angle between the limb axis and the vector to the camera. The weights are then normalized such that the sum of all weights equals the number of limbs in the model.

View-Dependent Limb Weights

The manual limb weighting heuristic demonstrably enhances model likelihood determination. We can augment these static weighting coefficients with dynamic information about the camera-limb relationship. As the camera's view vector approaches the axis vector of the limb in world space, the reliability of any measurement information degrades significantly: the silhouette detection is inadequate for acute viewing angles. The limb model's faults are most apparent when viewed along the axis. Thus, we can generate and normalize a second limb weighting heuristic based on the dot product of the view vector and the limb axis. (See Figure 3.2)

As in the static case, we define a set of exponents $E_v = \{e_{v1}, e_{v2}, \cdots, e_{vk}\}$ such that, for limb axis v_{b_i} and camera viewing vector v_c,

$$e_{vi} = \frac{k}{\kappa}(1 - |v_{b_i} \cdot v_c|)$$

where

$$\kappa = \sum_{i=1}^{k} (1 - |v_{b_i} \cdot v_c|)$$

We can see that

$$\sum_{i=1} k e_{vi} = \frac{k}{\kappa} \sum_{i=1} k(1 - |v_{b_i} \cdot v_c|) = \frac{k}{\kappa} \cdot \kappa = k$$

and we reformulate the original model likelihood as

$$l\left(I, B, m\right) = \prod_{i=1}^{k} \left(\rho\left(\mu(m), \sigma(m)\right)\left(I, b_i\right)\right)^{e_{vi}}$$

If all limbs are viewed from the same incident angle, then this is identical to the original formulation. Otherwise, limbs which are expected to have the most accurate measurements are prioritized in the log likelihood summation. Likewise, any limb which is barely visible due to the view angle will be play an inconsequential role in the final likelihood. Kakadiaris[7] proposes a more advanced technique along these lines.

Merging Heuristics

We can easily merge these two heuristics. (See Figure 3.3) We define the set of exponents $E = \{e_1, e_2, \cdots, e_k\}$ such that

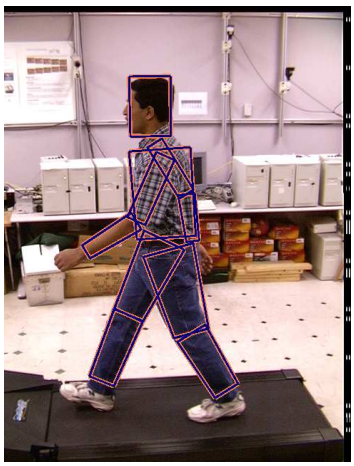
$$e_i = \frac{k}{\kappa} e_{li}(1 - |v_{b_i} \cdot v_c|)$$

where

$$\kappa = \sum_{i=1}^{k} e_{li}(1 - |v_{b_i} \cdot v_c|)$$

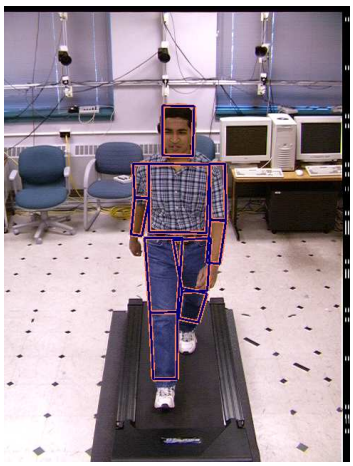
Again,

$$\sum_{i=1} k e_i = \frac{k}{\kappa} \sum_{i=1} k e_{li}(1 - |v_{b_i} \cdot v_c|) = k$$



Limb	Weight
R. Thigh	0.424211
R. Calf	1.175526
L. Thigh	0.402195
L. Calf	0.892192
Torso	0.063254
R. Up. Arm	0.074642
R. Forearm	1.687819
L. Up. Arm	0.083349
L. Forearm	2.073443
Head	3.123369
Sum	10

Limb	Weight
R. Thigh	0.500963
R. Calf	1.851279
L. Thigh	0.337183
L. Calf	0.570539
Torso	0.070572
R. Up. Arm	0.046678
R. Forearm	1.761920
L. Up. Arm	0.071883
L. Forearm	1.487681
Head	3.301302
Sum	10



Limb	Weight
R. Thigh	0.698835
R. Calf	1.902044
L. Thigh	0.224320
L. Calf	0.337326
Torso	0.067078
R. Up. Arm	0.027814
R. Forearm	1.720619
L. Up. Arm	0.104664
L. Forearm	1.574621
Head	3.342680
Sum	10

Table 3.3: **Merged heuristics for relative limb weighting:** The heuristics described in Sections 3.2.1 and 3.2.1 are combined into a merged heuristic. The static limb weights are multiplied by the geometric term $1 - |cos(\theta)|$, then normalized such that the sum of all weights equals the number of limbs in the model. This is the heuristic used when generating the results for this paper.

As before,

$$l\left(I, B, m\right) = \prod_{i=1}^{k} \left(\rho\left(\mu(m), \sigma(m)\right)\left(I, b_i\right)\right)^{e_i}$$

We return to the log domain for numerical stability, yielding

$$\log\left(l\left(I, B, m\right)\right) = \sum_{i=1}^{k} e_i \cdot \log\left(\rho\left(\mu(m), \sigma(m)\right)\left(I, b_i\right)\right)$$

3.2.2 Merging Likelihoods Across Measurement Functions

Because there is neither a guarantee nor even the slightest possibility that the non-normalized likelihoods sum to one — one cannot naively multiply the non-normalized particle likelihoods across all measurement functions. Instead, we must normalize the particle weight distributions for each measurement function individually, then merge them afterwards.

Thus, for any measurement function m, we have an initial distribution of n non-normalized log likelihoods (one for each particle). Before normalization, we must transform each likelihood $\log(\pi^i_{\text{orig}})$ in the measurement-specific distribution as follows:

$$\begin{aligned}
\text{LogMax} &= \textsc{Max}(\log(\pi^1_{\text{orig}}), \log(\pi^2_{\text{orig}}), \cdots, \log(\pi^n_{\text{orig}})) \\
\pi^i_{\text{trans}} &= e^{\text{LogMax} - \log(\pi^i_{\text{orig}})} \qquad \text{For all } 1 \leq i \leq n \\
\pi^i_{\text{norm}} &= \frac{\pi_{\text{trans}}}{\sum_{i=1}^{n} \pi_{\text{trans}}} \qquad \text{For all } 1 \leq i \leq n
\end{aligned}$$

We compute $\pi^{1,\cdots,n}_{\text{norm}}$ for each measurement distribution, then multiply the distributions together per-particle[4] and renormalize the cumulative distribution once again.

The cumulative particle distribution becomes more eccentric as more likelihoods are merged in. By multiplying the likelihoods, we presume that they are independent across measurement criteria; in many cases — namely for the edge and ridge likelihoods — this is not the case. It is thus sometimes more effective to only use either the edge or ridge measurement criteria. Plots and more discussion of eccentricity can be found in Section 4.2.

[4]This "multiplication" is done in the log domain for numerical stability

3.2.3 The Final Particle Distribution

After taking measurements, building measurement likelihood mappings, evaluating these distributions, building particle distributions, merging them, and normalizing, we have a distribution over the particles for the current moment t. Throughout the results chapter, we often refer to the "best" or most likely particles; these are, as one might expect, the particles with the greatest weight in the posterior distribution. With pre-trained likelihoods, one often ends up with an undesirable posterior distribution in which nearly all of the cumulative weight is shared between a small collection of particles within the larger particle set. (Section 4.2 contains plots of such eccentric distributions) The dynamic measurement mappings allow us to generate a final particle distribution with varied levels of eccentricity.

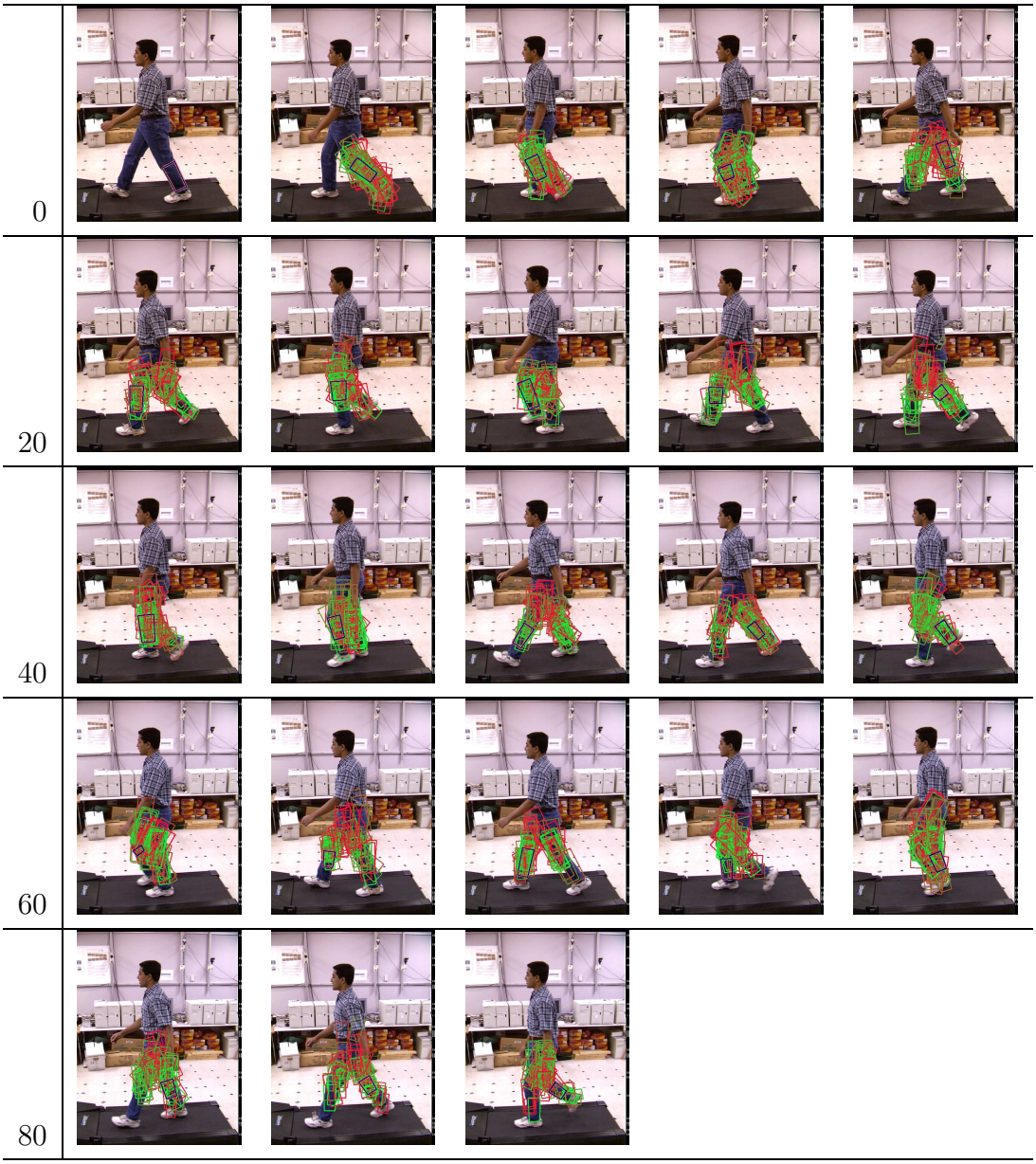
Chapter 4

Results

Having presented the theoretical framework and the details of our likelihood model, it is now appropriate to present our tracking results. The input sequence comes from the MoBo Database[13] at Carnegie Mellon University. There were many cameras in the room, but only three with accurate calibration information. The initial body configuration was specified manually. All camera input frames were synchronized.

Our results are presented in several sections. In Section 4.1, a single limb is tracked as an illustration of the particle filter. Section 4.2 demonstrates the effects of λ on tracking results and the eccentricity of the associated particle distributions. Section 4.3 illustrates the troubles involved with monocular tracking. Section 4.4 presents the experimental results of measurement model combinations on lower-body tracking; both the quality of the motion estimation and the eccentricity of the particle distributions change substantially with the incorporation of multiple measurement models. Finally, Sections 4.5.1 and 4.5.2 show our best efforts to estimate human motion given the techniques presented in this document.

4.1 Illustrating the Particle Filter



We try to track a single limb through this 90-frame sequence. The posterior distribution is modeled with 150 particles, and at each frame we display the n most probable limb configurations (as determined by our weighted particle set). In the first image array, $n = 75$ (the better half of the particles), and in the second, $n = 10$. The limb projection corresponding to the most likely particle is drawn in blue with an orange border. The other limbs are drawn in colors ranging from green to orange to red; green limbs are more likely and red limbs are less likely.

We only consider images from a single camera. The oriented edge and background

subtraction measurement models were enabled for this sequence.

The first frame shows all particles set to the manually-determined initial parameterization. Throughout, we notice that the green particles generally correlate well with limbs or — at the very least — limb-like structures in the image. The background subtraction measurement criterion keeps limbs on the body, and the edge and ridge criteria weed out particles that are aligned poorly. The prior had to be adjusted slightly for this test, as there was insufficient variance in the prior for the positional velocity[1].

As the left leg occludes the right leg, the tracking breaks down as one might anticipate. By frame 10, there are highly probable particles on both legs. From here on, with some exceptions, we see the particle filter successfully track both legs simultaneously. The occlusion created ambiguity, but both legs are thus considered for the remainder of the monocular sequence.

It is worth noting that many of the limb configurations — and even the more probable limb configurations — may seem "shorter" than other limbs because they are turned away from the camera. These particles are poor matches in three dimensions, but they are plausible here due to the lack of depth information in a monocular setting.

Without the rest of the lower body to provide guidance, tracking a single calf is impractical without multiple cameras. As we will see later on, increasing the parameter space — perhaps surprisingly — improves tracking results by providing certain positional constraints on limbs.

[1]Variances for angular and positional velocity priors are distinct from one another. There are only 3 positional parameters, all of which pertain to the root of the model skeleton. In this case, the "root" (the knee) accelerates more rapidly through space than when tracking the full body.

4.2 Ideal λ Values

The λ heuristic was introduced in Section 3.1.3. The variance σ for a measurement likelihood $p(m|\vec{m})$ is inversely proportional to the choice of λ for that measurement. As mentioned earlier, there are two opposing ideals which determine our choice for λ:

1. We require that λ be large enough for roughly optimal tracking results when using a given measurement criterion.

2. Given the above precondition, we require that λ be as small as possible (in the interests of a "healthy" particle distribution).

To determine an appropriate λ value for each measurement function, we try to track the lower body using a single measurement function for likelihood determination. We track with multiple cameras, but for clarity the tracking results are presented from the perspective of only one camera. 1000 particles were used throughout.

4.2.1 Plotting Eccentricity

In this section we will present figures which graphically represent the eccentricity of a particle distribution. To create the plots, the following steps are taken:

1. A set $L = \{\pi_1, \pi_2, \cdots, \pi_n\}$ is built from the normalized particle likelihoods.

2. The set L is sorted to generate the new L_s, also of cardinality n.

3. For each of n items in L_s, a point $(\frac{k}{n}, \sum_{i=1}^{k} L_{s_i})$ is plotted, and a line drawn from the previous plotted point.

In other words, we make a cumulative plot of the particle distribution. Due to the sorting step, the rate of change for the cumulative plot must always be positive. Additionally, the points $(0, 0)$ and $(1, 1)$ will always appear in the plot. If all particles have equal likelihood, the plot will be linear. If the cumulative likelihood is distributed among only a small fraction of particles — in other words, if the distribution is eccentric — the plot will approach the point $(1, 0)$, though it can never reach this point since the normalized distribution must sum to 1. For an example of these plots, see Figure 4.1.

4.2.2 Oriented Edge Detection

The oriented edge results for $\lambda = \{0.6, 0.7, 0.8, 0.9, 0.95\}$ can be seen in Figure 4.1. Tracking was roughly comparable for $\lambda = 0.9$ and $\lambda = 0.95$, and was slightly worse when $\lambda = 0.8$. In the latter case, tracking was unable to recover after frame 32 because much of the weight in the particle set was essentially wasted on particles that were far from the legs. Tracking was ineffective for the $\lambda = 0.6$ and $\lambda = 0.7$ cases. Ultimately, we chose $\lambda = 0.9$ as per the heuristics outlined above in Section 4.2.

4.2.3 Oriented Ridge Detection

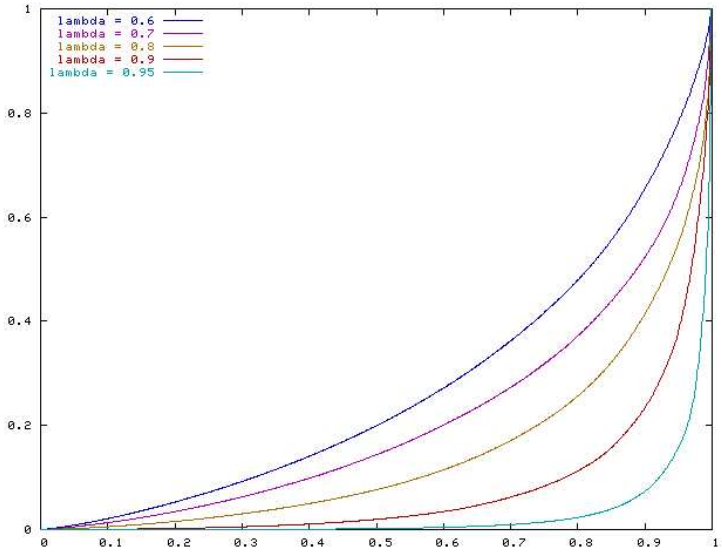
Again, we test the oriented ridge measurement for $\lambda = \{0.6, 0.7, 0.8, 0.9, 0.95\}$ can be seen in Figure 4.2. The tracking results are generally unimpressive for the ridge measurement, but they are more "precisely unimpressive" for $\lambda = \{0.9, 0.95\}$. In addition, the ridge response is prone to false positives, and this motivates the choice of a high λ value. Thus, we define $\lambda = 0.9$.

4.2.4 Background Subtraction

We test $\lambda = \{0.3, 0.4, 0.5, 0.6, 0.8\}$ for the background subtraction measurement. For $\lambda > 0.5$, we see an unacceptably eccentric particle set. Tracking is adequate for $\lambda = 0.4$ and slightly more precise for $\lambda = 0.5$. As such, we set $\lambda = 0.45$ for the background subtraction measurement.

4.2.5 Template Matching

The template matching results for $\lambda = \{0.3, 0.4, 0.5, 0.6, 0.7\}$ can be seen in Figure 4.4. The results are excellent across the board, though the posterior variance is smaller when $\lambda > 0.4$. We thus set $\lambda = 0.45$.



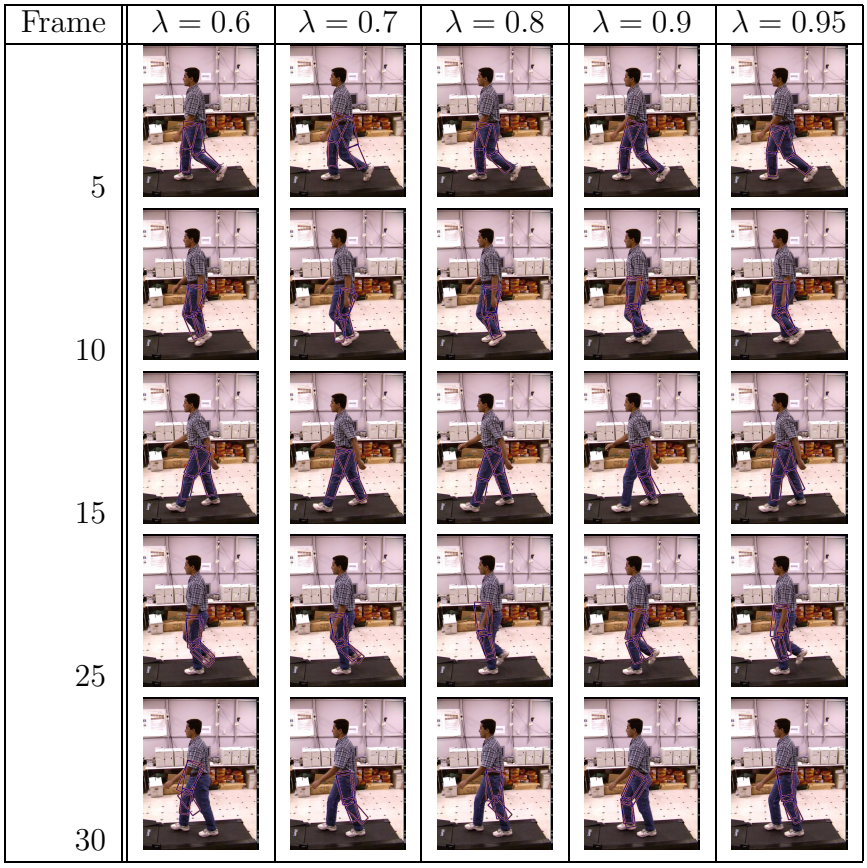
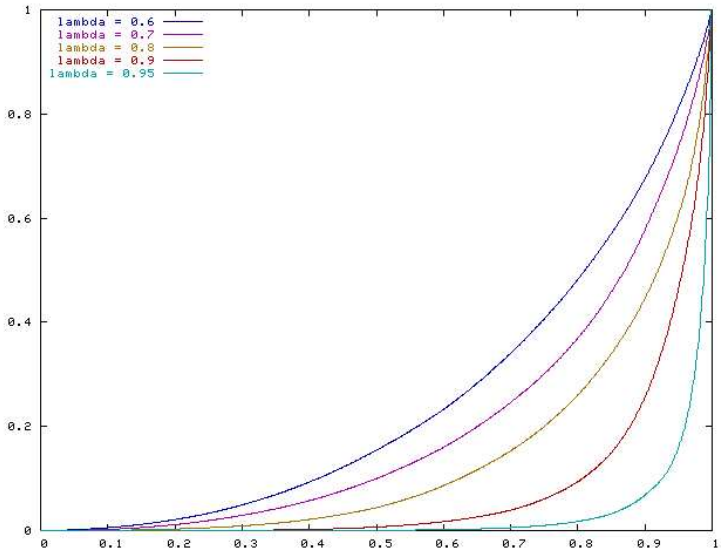
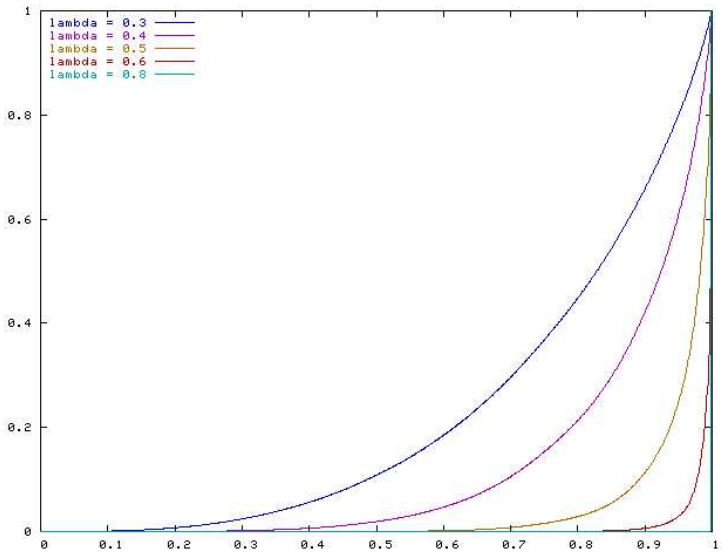


Figure 4.1: **Oriented edge tracking results with respect to λ:** The distributions (top) are plotted according to the specifications in Section 4.2.1. The tracking broke down shortly after frame 30 for $\lambda = 0.8$.



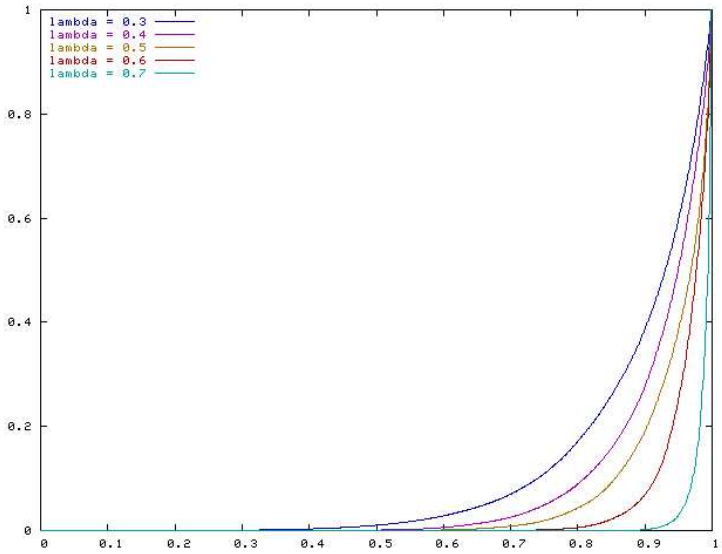
Frame	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 0.95$
7					
15					
23					
31					

Figure 4.2: **Oriented ridge tracking results with respect to λ:** The distributions (top) are plotted according to the specifications in Section 4.2.1.



Frame	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.8$
5					
15					
25					
32					
45					

Figure 4.3: **Background subtraction tracking results with respect to λ:** The distributions (top) are plotted according to the specifications in Section 4.2.1. The tracking results were generally equivalent for $\lambda > 0.4$, though the eccentricities for the higher λ values would prohibit robust tracking with multiple measurement models.



Frame	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.7$
3					
7					
11					
15					

Figure 4.4: **Template matching tracking results with respect to λ:** The distributions (top) are plotted according to the specifications in Section 4.2.1.

Frame:	5	10	15	20	25
Camera 1:					

Figure 4.5: **Tracking results when considering camera 1 exclusively.**

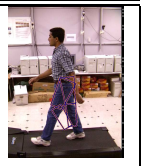
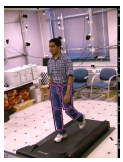
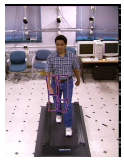
Frame:	3	6	9	14	15
Camera 1:					
Camera 2:					
Camera 3:					

Figure 4.6: **Tracking results when considering camera 2 exclusively:** By the end of the sequence, the model has veered off course with respect to cameras 1 and 3.

4.3 Monocular Tracking

In the sequences presented in this section, we use 1000 particles to track the lower body. The edge and background subtraction measurement criteria were used throughout. As one might expect, the results that appear reasonable from the primary camera are less coherent from the other viewing angles. Multi-camera results are available in Section 4.4; even though fewer particles were used in these multi-camera trials, there are substantial improvements in tracking accuracy.

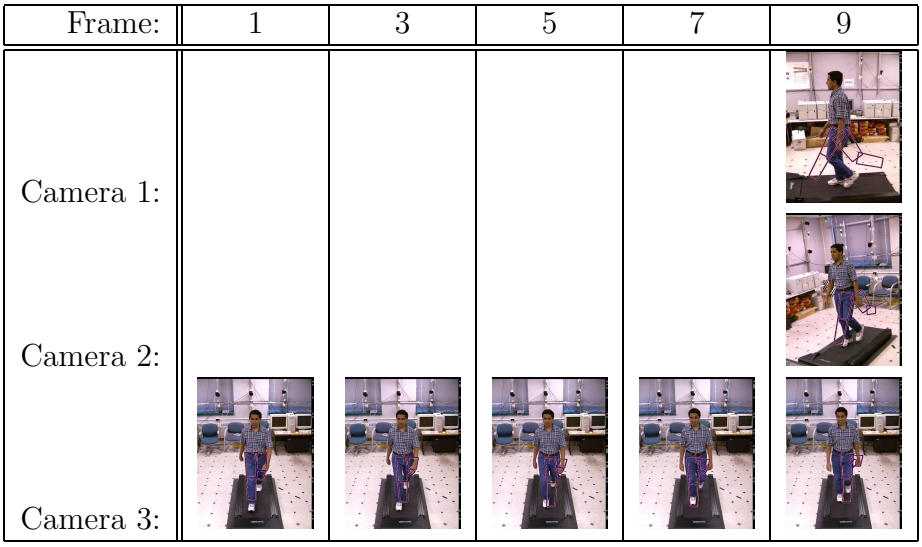


Figure 4.7: **Tracking results when considering camera 3 exclusively:** Camera 3 has little chance of tracking the legs in this sequence, as nearly all of the motion is perpendicular to the film plane. Tracking breaks down rapidly.

4.4 Combining Measurements

Frame:	3	6	9	12	15
E/R:					
E/BgS:					
E/T:					
R/BgS:					
R/T:					
BgS/T:					

Frame:	3	6	9	12	15
E/R/BgS:					
E/BgS/T:					
R/BgS/T:					
All:					

In this section we try to track the lower body with 500 particles and all 3 cameras. The second and third cameras are not shown here. The results (and the cumulative posterior distribution plots in Figure 4.8) are largely self-explanatory, but a few points are worthy of special emphasis:

- In the background subtraction and template matching combination, the tracking results are very good even though the posterior distribution is quite eccentric. The eccentricity is largely due to the dependency between the measurements and the breadth of the histograms; the tails of the likelihood Gaussian approach zero, and only with these two measurement criteria do the measurements often map to the low tails of the likelihood.

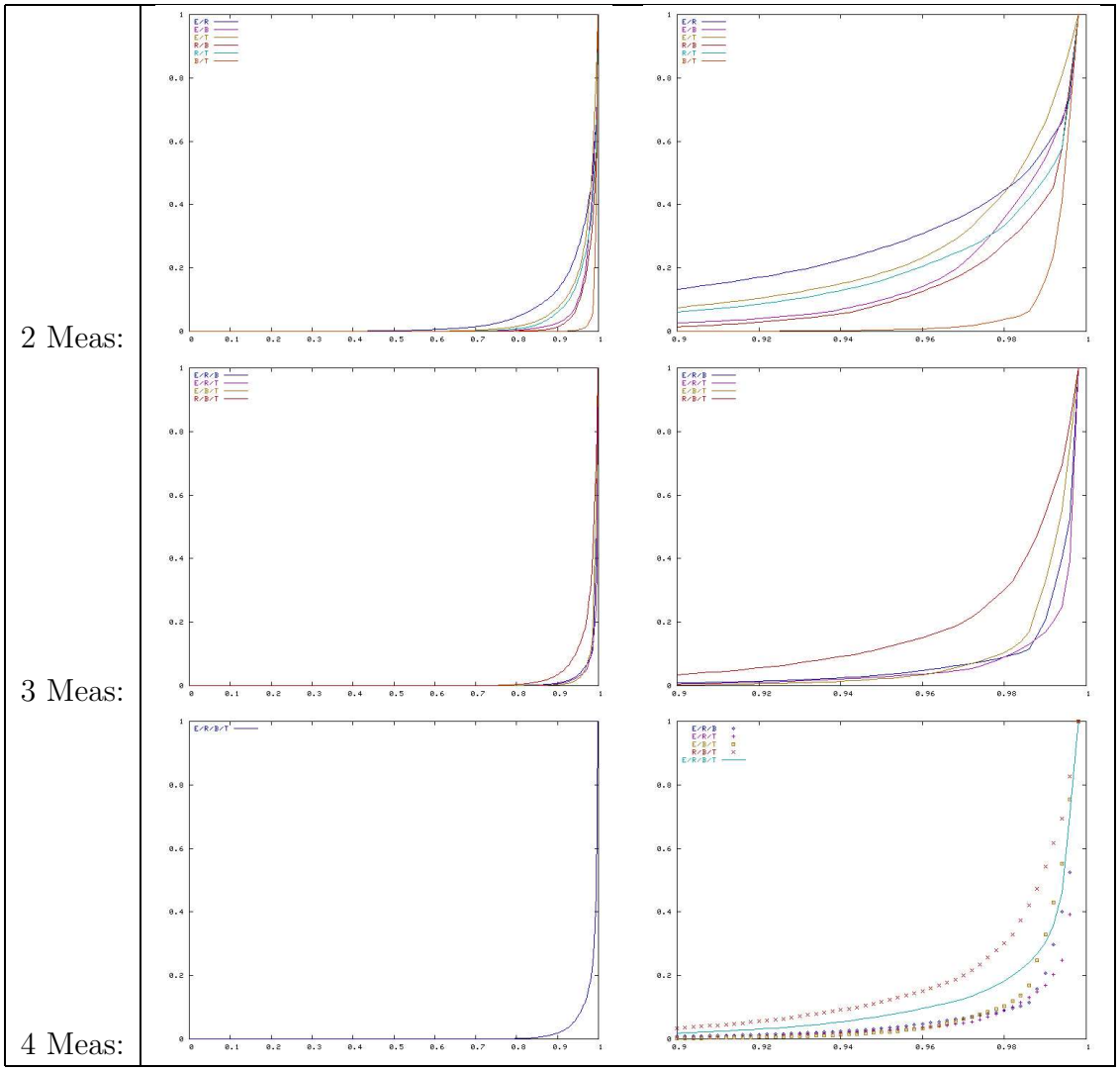
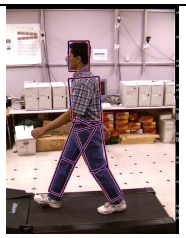
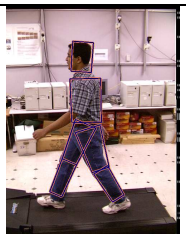
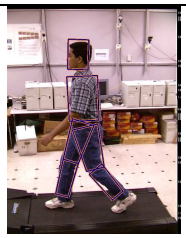
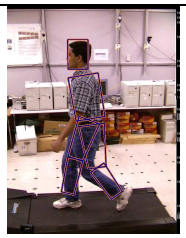
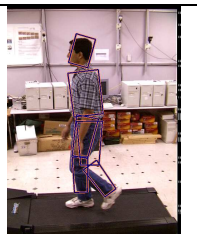


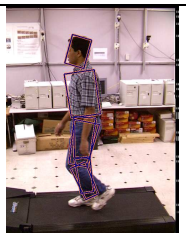
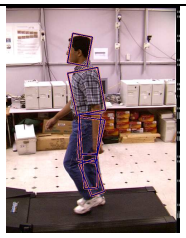
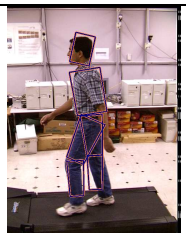
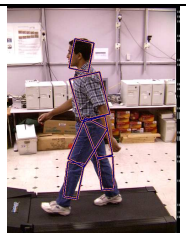
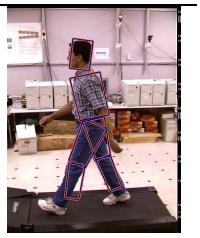
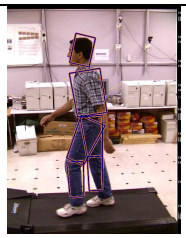
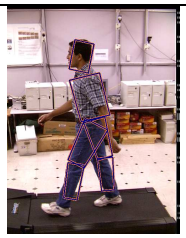
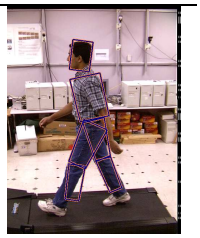
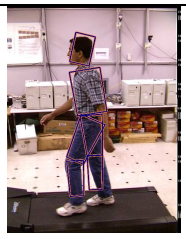
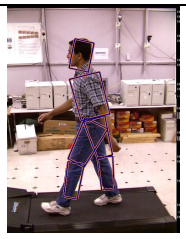
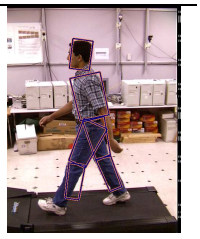
Figure 4.8: **The effect of measurement combinations on the posterior distribution:** The cumulative plots of the posterior distributions become more eccentric as additional measurement criteria are considered. Note that for measurements which are increasingly dependent probabilistically, the cumulative distribution is especially eccentric. (E.g., background subtraction and template matching) The magnified (right) cumulative plot for the posterior distribution after using 4 measurements is given context by the plots for 3 measurements.

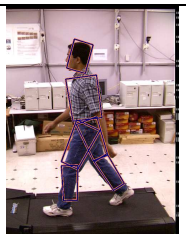
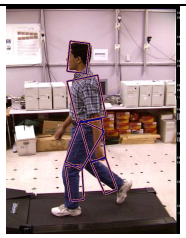
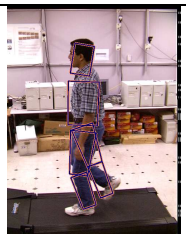
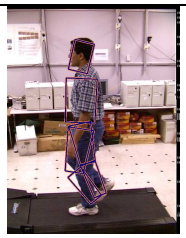
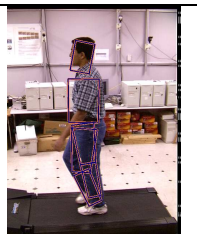
- The combination of all four measurement criteria performs notably worse than some of the 3-way combinations. This, too, is due to the increase in invalid dependency assumptions and extremely low tail values for some likelihood Gaussians.

4.5 Extended Tracking Results

4.5.1 Tracking Modulo Arms

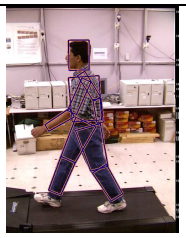
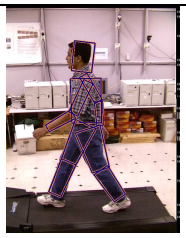
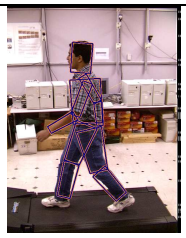
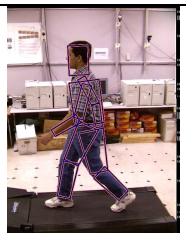
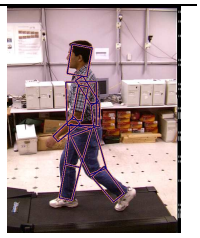
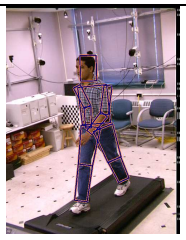
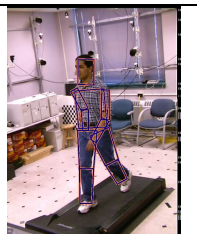
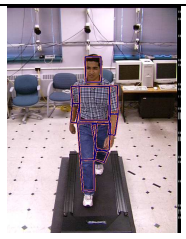
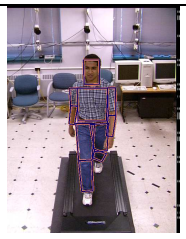
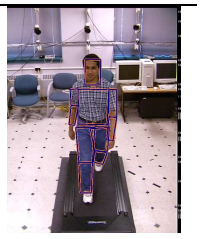
Frame:	0	2	4	6	8
Camera 1:					
Camera 2:					
Camera 3:					

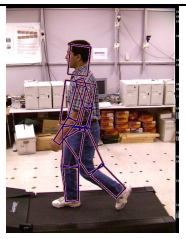
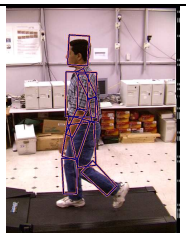
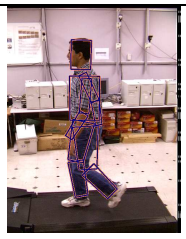
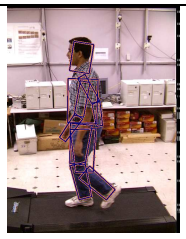
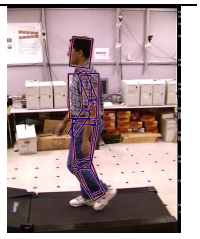
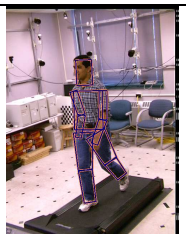
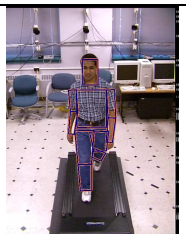
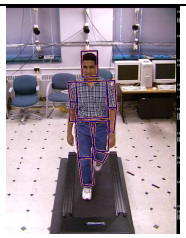
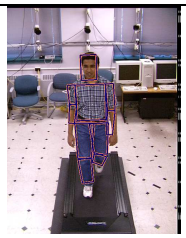
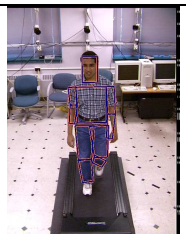
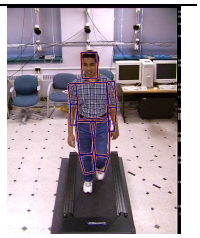
Frame:	10	12	14	16	18
Camera 1:					
Camera 2:					
Camera 3:					

Frame:	20	22	25	26	28
Camera 1:					

When tracking the lower body with the edge and background subtraction measurement criteria, the legs tend to creep up into the torso. We can address this problem by incorporating the template matching measurement, but we can also stabilize the legs by adding the torso and head to the model. The head constrains the model, and the legs cannot climb up the body as seen in parts of Section 4.4. It is interesting to note that, at least in this case, increasing the parameter search space actually improves tracking results substantially.

4.5.2 Full-Body Tracking

Frame:	0	1	2	3	4
Camera 1:					
Camera 2:					
Camera 3:					

Frame:	5	6	7	8	9
Camera 1:					
Camera 2:					
Camera 3:					

Ultimately, we are unable to track the full-body through the entire input sequence. Tracking breaks down once the subject's left arm occludes his torso, at which point the edges are harder to find. In addition, the subject's right arm is only visible from camera 3 for many frames. Consequently, its motion — which is perpendicular to camera 3's film plane — is not easily tracked. Given a fourth view from the subject's right side, tracking the full body would probably be much easier. Even after the arms are lost, the legs, torso, and head maintain adequate tracking. These results were generated using 12000 particles, but 5000 particles worked nearly as well.

Chapter 5

Conclusions

We have documented a system designed to track a human through video sequences with few input preconditions. We use a model-based approach based upon Bayesian inference, and approximate the posterior distribution with a particle set.

We found that the particle filter is less effective when the particle distribution becomes eccentric. To help guard against such a circumstance, we broke likelihood determination into two steps: one for a raw measurement, and one for a dynamic likelihood mapping. This mapping was parameterized by the heuristic λ. By experimentally optimizing λ, our particle distribution was well-behaved and we consequently achieved improved tracking results.

Additionally, we found that multiple cameras — as one would expect — stabilized tracking in three dimensions by disambiguating limb locations in the depth dimension for any one specific camera.

5.1 Future Work

This work focused primarily on the sub-problem of likelihood determination. As such, with minimal work, one could build many other tracking models[4] around the likelihood system proposed here. Specifically, belief propagation holds promise for human motion tracking: limb parameterizations are independent, and joints have greater flexibility due to a spring-like message passing framework.

None of the measurement functions were of significant procedural complexity.

Countless measurement criteria could be devised which would, on their own, return more reliable information than those presented in this work. In particular, we would like to incorporate a measurement based upon optical flow from frame to frame,[1][14] as well as a measurement designed to find median distances to strong properly-oriented edges for a given limb configuration.

Were we to take the median value for each pixel over the image sequence, we could dynamically determine the background images. With automatic background subtraction and multiple cameras, we should be able to significantly constrain the search space for initialization, even in input sets without provided background images. Using a body model with independent limb parameterizations, we could propose potential initial configurations for the body rather than hand-initialize the pose ourselves.

Bibliography

[1] C. Bregler and J. Malik. Tracking people with twists and exponential maps. 1998.

[2] T-J. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. volume 1, pages 239–245, 1999.

[3] J. Deutscher, B. North, B. Bascle, and A. Blake. Tracking through singularities and discontinuities by random sampling. volume 2, pages 1144–1149, 1999.

[4] D. M. Gavrila. The visual analysis of human movement: a survey. 73(1):82–98, 1999.

[5] D. M. Gavrila and L. S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. pages 73–80, 1996.

[6] N. Gordon. On-line filtering for nonlinear/non-gaussian state space models. In *European Conference on Simulation Methods in Econometrics*, 1996.

[7] D. Metaxas I.A. Kakadiaris. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. *Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pages 81–87, June 1996.

[8] M. Isard and A. Blake. Condensation – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[9] I. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. pages 81–87, 1996.

[10] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156.

[11] D. Morris and J. M. Rehg. Singularity analysis for articulated object tracking. pages 289–296, 1998.

[12] K. Rohr. Towards model-based recognition of human movements in image sequences. *Image Understanding*, 59(1):94–115, 1994.

[13] J. Shi and R. Gross. The CMU Motion of Body (MoBo) Database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University, June 2001.

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

[15] H. Sidenbladh and M. J. Black. Learning image statistics for Bayesian tracking. In *Int. Conf. on Computer Vision, to appear*, 2001.

[16] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. volume 2, pages 702–718, 2000.

[17] R. Y. Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL*, pages 364–374, 1986.