Monocular Perception of Biological Motion - Detection and Labeling

Yang Song[†], Luis Goncalves[†], Enrico Di Bernardo[†] and Pietro Perona[†‡]

† California Institute of Technology, 136-93, Pasadena, CA 9112 5, USA

‡ Università di Padova, Italy

{yangs,luis,dibe,perona}@vision.caltech.edu

Abstract

Computer perception of biological motion is key to developing convenient and powerful human-computer interfaces. Successful body tracking algorithms have been developed; however, initialization is done by hand. We propose a method for detecting a moving human body and for labeling its parts automatically. It is based on maximizing the joint probability density function (PDF) of the position and velocity of the body parts. The PDF is estimated from training data. Dynamic programming is used for calculating efficiently the best global labeling on an approximation of the PDF. The computational cost is on the order of N^4 where N is the number of features detected.

We explore the performance of our method with experiments carried on a variety of periodic and non-periodic body motions viewed monocularly for a total of approximately 30,000 frames. Point-markers were strapped to the joints of the subject for facilitating image analysis. We find an average of 2.3% labeling error; the experiments also suggest a high degree of viewpoint-invariance.

1. Introduction

Being able to extract the position and motion of humans ('biological motion' in the literature of human vision) from images is very useful for human social interactions and is a most important technology for developing convenient and effective human-computer interfaces. Our visual system has developed a very strong ability in perceiving biological motion, even from monocular low resolution noisy data, e.g. NTSC television.

A striking demonstration of the capabilities of the human visual system is provided by the experiments of Johansson [8]. In his experiments, Johansson demonstrated that biological motion may be accurately perceived even from very poor data. We postulate that this is true because the degrees of freedom of the problem are highly constrained, both by the kinematics and dynamics of the body, and, more importantly, by the fact that humans move in stereotypical and predictable ways. *It is our belief that defining and estimating perceptual models of human motion is the key to automating biological motion perception.*

Much progress has been made recently in tracking the human body [12, 11, 6, 2, 7, 4] under a number of conditions: static background, periodic motion, stereo-scopic vision, hand-initialization. In all these schemes the body is segmented into parts which are independently tracked – the final estimate is obtained by enforcing the body's kinematic constraints and simple statistical models of motion (e.g. first order random walks). No use is made of dynamic and/or perceptual models of body motion. We believe that in order to achieve self-initializing trackers that will work against unmodelled backgrounds in the presence of general motions of the human body, models of how the body 'tends to move' have to be used.

In this paper we address the problem of defining and estimating a perceptual model of biological motion and use it for detecting the human body and labeling it in monocular image sequences. By 'labeling' we mean assigning to each region in the image a label that corresponds to the body part (shoulder, elbow etc) that is imaged in that region. We choose not to address the issue of detecting and classifying pictorial features that are associated to the body parts – for the time being this has been sufficiently explored by [1, 9, 5, 10]. Therefore our experimental setup is identical to Johansson's experiments: we suppose that a number of markers are attached to the body of an actor. At every frame we need to attach labels to a subset of the features (some may be caused by noise; some may have been missed). We approach the problem as a learning problem: we observe the subject moving about in order to estimate a model of his/her stereotypical motions. This model, which we formulate as the joint probability density function (PDF) of the position and motion of the body, is used to select the best labeling.

2. Approach and Notation

We choose to characterize body pose and motion by the joint probability density of the position and velocity of its parts. Our goal is to interpret monocular image sequences, hence we use part position and velocity in the image plane (Figure 1). In our Johansson scenario each part appears as a single dot (marker) in the image plane. Therefore its iden-
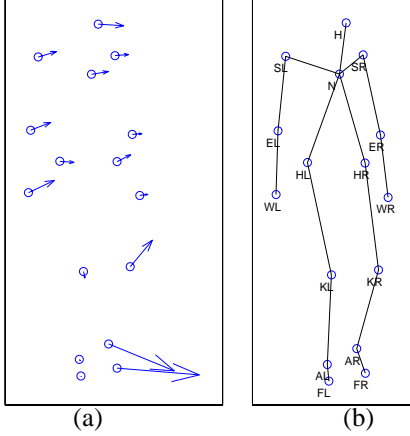
Figure 1: **The labeling problem:** Given the position and velocity of body parts in the image plane (a), we use a probabilistic model to assign the correct labels to the body parts (b). 'L' and 'R' in label names indicate left and right. H:head, N:neck, S:shoulder, E:elbow, W:wrist, H:hip, K:knee, A:ankle and F:foot.

tity is not revealed by cues other than its relative position and velocity.

Let $V = \{v_1, \ldots, v_m\}$ be the set of candidate markers in an image. Let $\mathcal{L} = \{L_1, L_2, \ldots, L_N\}$ be the set of N labels representing body parts such as head, neck, left elbow, etc. Since candidate markers can be wrongly detected, and some body parts may be missing due to occlusion, m is not always equal to N. If we assume that there are no missing points, then $N \leq m$. Therefore, the labeling problem is to find the mapping $f : \mathcal{L} \longrightarrow V$ such that $f(L_i) \neq f(L_j)$ for $i \neq j$ and the $Prob\{body\ part\ L_i\ is\ in\ marker\ f(L_i), 1 \leq i \leq N\}$ is maximized.

Three problems face us at this point: (a) What is the structure for the probability/likelihood function to be maximized? (b) How do we estimate its parameters? (c) How do we address the combinatorial search problem of finding the optimal labeling? Problems (a) and (c) need to be addressed together: the structure of the probability density function must be such that it allows efficient optimization.

A brute force solution to the optimization problem is to search exhaustively among all $(m)_N \stackrel{\text{def}}{=} m * (m - 1) * \cdots * (m - N + 1)$ possible f's and find the best one. The search cost is exponential with respect to N. Assume $N = m = 16$ (this is the case without missing points and wrong detections), then the number of possible mappings is 2×10^{13} which is computationally prohibitive.

It is useful to notice that the body is a kinematic chain: the wrist is connected to the body indirectly via the elbow and the shoulder. It is a reasonable approximation to assume that the position and the motion of the wrist are, therefore, independent of the position and velocity of the rest of the body once the position and velocity of elbow and shoulder are known. This intuition may be generalized to the whole body: once the position and velocity of a set S of body parts is known, the behavior of the body parts above and below (left and right) of S is independent. This approximation of course needs to be validated experimentally.

Our intuition on how to decompose the problem may be expressed in the language of probability: consider the joint probability function of 5 random variables $P(ABCDE)$. It may be expressed as $P(ABCDE) = P(ABC)P(D|ABC)P(E|ABCD)$. If these random variables are conditionally independent as described in the graph of Figure 3, then

$$P(ABCDE) = P(ABC)P(D|BC)P(E|CD) \qquad (1)$$

Thus, if the body parts can satisfy the appropriate conditional independence conditions, we can express the joint probability density of the pose and velocity of all parts as the product of conditional probability densities of n-tuplets. This approximation makes the optimization step computationally efficient as will be discussed below.

What is the best decomposition for the human body? What is a reasonable size n of the groups of body parts? We hope to make n as small as possible to minimize the cost of the optimization. But as n gets smaller, conditional independence may not be a reasonable approximation any longer. There is a tradeoff between computational cost and algorithm performance. In this paper we use models with $n = 3$ as described in Figure 2. Optimization on triangulated graphs such as these may be efficiently performed using Dynamic Programming [13].

Estimation of the conditional probability densities from training data and the dynamic programming algorithm are described in the next section.

3. Algorithms

We characterize each triplet $\{L_i, L_j, L_k\} \subset \mathcal{L}$ (corresponding to one triangle in Figure 2) with a 10-dimensional feature vector

$$X = (v_{ix}, v_{jx}, v_{kx}, v_{iy}, v_{jy}, v_{ky}, p_{ix}, p_{kx}, p_{iy}, p_{ky}) \qquad (2)$$

The first three dimensions of X are the x-direction (horizontal) velocity of body parts (L_i, L_j, L_k), the next three are the velocity in the y-direction (vertical), and the last four dimensions are the positions of body parts L_i and L_k relative to L_j. If the data are normalized for scale, it would be reasonable and convenient to assume that X is jointly Gaussian-distributed. Then

$$
\begin{aligned}
p(L_i, L_j, L_k) &= p(X) \\
&= \frac{\exp[-\frac{1}{2}(X - \bar{X})^T \Sigma^{-1}(X - \bar{X})]}{(2\pi)^{d/2}|\Sigma|^{1/2}} \\
&= \frac{\exp[-\frac{1}{2}\sum_i \frac{Y_i^2}{\lambda_i}]}{(2\pi)^{d/2}\prod_i \lambda_i^{1/2}} \qquad (3)
\end{aligned}
$$

where $\bar{X}$ is the mean value of X; Σ is the covariance matrix of X; d is the dimensionality of X ($d = 10$ here); $Y = \Phi^T(X - \bar{X})$; Φ is the unitary eigenvector matrix of Σ; and $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_{10})$ is the corresponding diagonal matrix of eigenvalues [3]. Φ and Λ can be obtained

from singular value decomposition (SVD) of the training set. After the joint probability is computed, the conditional one can be obtained accordingly:

$$p(L_i|L_j, L_k) = \frac{p(L_i, L_j, L_k)}{p(L_j, L_k)} \qquad (4)$$

where $p(L_j, L_k)$ can be obtained by estimating the joint probability of the vector $\{v_{jx}, v_{kx}, v_{jy}, v_{ky}, p_{kx}, p_{ky}\}$.

Suppose there are m markers available for a frame, then for each triangle in Figure 2, we can compute the (conditional) probabilities of all possible $(m)_3$ combinations of markers and rank them. The bigger the probability is, the more likely they are the right markers for the triangle.

If out of all the possible combinations the one with the correct markers always produces the highest probability, the labeling problem can be solved easily by picking the highest ranked combination for each triangle individually. In practice, since the data are noisy and we only have available an approximation of the true probability density functions, this will not work. In fact, since all triangles share at least one edge (and thus two vertices) with at least one other triangle, picking the top combination for each triangle individually won't even produce a consistent set of labels. What is needed is an algorithm that will search through all the legal labelings and find the one that maximizes the global joint probability. By the decomposition in equation (1), we know that dynamic programming can be used to solve this problem efficiently. The key condition for using dynamic programming is that the problem exhibits optimal substructure, namely, if equation (1) holds, then

$$\max_{A,B,C,D,E} P(ABCDE) = \max_{A,B,C} (P(ABC)$$
$$\cdot \max_D(P(D|BC) \cdot \max_E P(E|CD)))$$

If we take the probability as the cost function, a dynamic programming method similar to that described in [13] can be used, which requires the triangulated body graph to be decomposable. If all the cliques in a graph are of size three, then the decomposable property means that there always exists a free vertex to delete and the remaining subgraph is again a collection of triangles until only one triangle is left. A vertex is free when it is only contained in one triangle. Figure 2 shows two decomposable graphs of the whole body, along with the order of successive elimination of cliques.

If the decomposed body graph is decomposable and the corresponding conditional independence holds, then,

$$p(L_1, L_2, \cdots, L_N) = \prod_{t=1}^{N-3} p(l_t|a_t b_t) p(l_{N-2} l_{N-1} l_N) \qquad (5)$$

where $(l_1, l_2, \ldots, l_{N-3})$ are the body parts to be deleted in order in stage $t = 1, 2, \ldots, N-3$; (l_{N-2}, l_{N-1}, l_N) is the triangle in the final stage $T = N - 2$; and $(a_t b_t)$ are the two vertices connected to l_t when l_t is deleted. Let

$$\Psi_t(l_t, a_t, b_t) = -\log p(l_t|a_t b_t), \; for \; 1 \le t \le T-1 \quad (6)$$
$$\Psi_t(l_t, a_t, b_t) = -\log p(l_{N-2}, l_{N-1}, l_N), \; for \; t = T \quad (7)$$

be the cost function associate with each triangle. The dynamic programming algorithm can be described as follows:
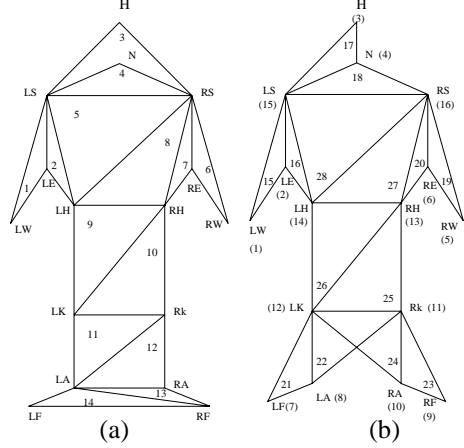

Figure 2: **Two decompositions of the human body into triangles.** The label names are the same as in Figure 1. The numbers inside triangles give the index of triangles used in the experiments. In (a) they are also the order in which the vertices are deleted. In (b) the numbers in brackets show the order.

Stage 1: for every pair (a_1, b_1),

Compute $\quad \Psi_1(l_1, a_1, b_1) \quad$ for all possible l_1,

Store $\quad \begin{cases} l^*_{1[a_1,b_1]} \\ \Psi_1(l^*_{1[a_1,b_1]}, a_1, b_1) \end{cases}$

where $\quad l^*_{1[a_1,b_1]} \quad$ minimizes $\quad \Psi_1(l_1, a_1, b_1)$

Stage t, $2 \le t \le T$: for every pair (a_t, b_t),

Compute $\quad \Psi_t(l_t, a_t, b_t) \quad$ for all possible l_t

Compute the total cost so far (till stage t):

– Let $\quad T_t(l_t, a_t, b_t) = \Psi_t(l_t, a_t, b_t) \quad$ the total cost so far

– If edge $\quad (l_t, a_t) \quad$ is contained in a previous stage and τ is the latest such stage, add the cost $T_\tau(l^*_{\tau[l_t,a_t]}, l_t, a_t)$ (or $T_\tau(l^*_{\tau[a_t,l_t]}, a_t, l_t)$ if the edge was reversed) to $T_t(l_t, a_t, b_t)$

– Likewise, add the cost of the latest previous stage containing edges (l_t, b_t) and edges (a_t, b_t) to $T_t(l_t, a_t, b_t)$

Store $\quad \begin{cases} l^*_{t[a_t,b_t]} \\ T_t(l^*_{t[a_t,b_t]}, a_t, b_t) \end{cases}$

where $\quad l^*_{t[a_t,b_t]} \quad$ minimizes $\quad T_t(l_t, a_t, b_t)$

When stage T calculation is complete, $T_T(l^*_{T[a_T,b_T]}, a_T, b_T)$ includes the value of each Ψ_t, $1 \le t \le T$, exactly once. Since the Ψ_t's are the logs of conditional (and joint) probabilities, then

$$T_T(l^*_{T[a_T,b_T]}, a_T, b_T)$$
$$= -\log(\text{joint probability density of the entire graph})$$

Thus picking the pair (a_T, b_T) that minimizes T_T automatically maximizes the joint probability.

Stage 1: for all valid choices of markers for C and D (C=1..5, D=1..5, $C \neq D$)
    for all valid choices of markers for E, (E=1..5, $E \neq C$, $E \neq D$)

    compute $\Psi_1(E, C, D)$

    Store $\begin{cases} E^*_{[C,D]} & E^*_{[C,D]} \text{ is best choice of E} \\ \Psi_1(E^*_{[C,D]}, C, D) & \text{for each choice of } C \text{ and } D \end{cases}$

Stage 2: for all valid choices of B and C
    for all valid choices of D

    compute $\Psi_2(D, B, C)$

    let $T_2(D, B, C) = \Psi_2(D, B, C) + \Psi_1(E^*_{[C,D]}, C, D)$

    Store $\begin{cases} D^*_{[B,C]} \\ T_2(D^*_{[B,C]}, B, C) \end{cases}$

Stage 3: for all valid choices of A and B
    for all valid choices of C

    compute $\Psi_3(C, A, B)$

    let $T_3(C, A, B) = \Psi_3(C, A, B) + T_2(D^*_{[B,C]}, B, C)$
    $= \Psi_3(C, A, B) + \Psi_2(D^*_{[B,C]}, B, C) + \Psi_1(E^*_{[C,D^*_{[B,C]}]}, C, D^*_{[B,C]})$
    $= -\log(p(A, B, C)) - \log(p(D^*_{[B,C]}|B,C)) - \log(p(E^*_{[C,D^*_{[B,C]}]}|C, D^*_{[B,C]}))$
    $= -\log(p(A, B, C, D^*_{[B,C]}, E^*_{[C,D^*_{[B,C]}]}))$

Thus chosing A, B, and C that minimizes T_3 finds the solution which maximizes the joint probability of the entire graph.
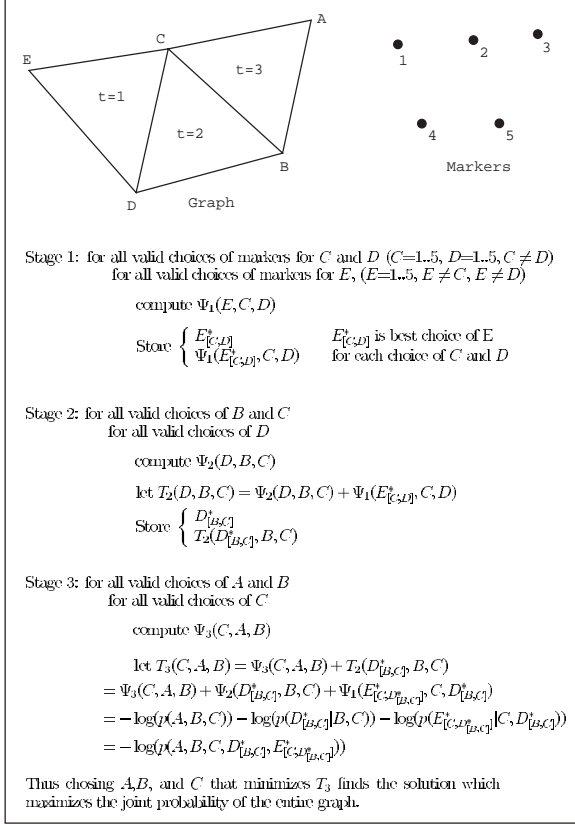
Figure 3: **An example of dynamic programming algorithm applied to a simple graph**

The best labeling can now be found tracing back through each stage: the best (a_T, b_T) determines l^*_T, then the latest previous stages with edge respectively (l^*_T, a_T), (l^*_T, b_T), and/or (a_T, b_T) determine more labels and so forth.

A simple example of this algorithm is shown in Figure 3.

The above algorithm is computationally efficient. Assume N is the number of body part labels and m is the number of candidate markers, then the total number of stages is $T = N - 2$ and in each stage the computation cost is $\mathcal{O}(m^3)$. Thus, the complexity of the whole algorithm is on the order of $N * m^3$.

4. Experiments

We trained our model and tested the performance of the algorithm on data obtained filming a subject moving freely in 3D; 16 light bulbs were strapped to the main joints of the subject's body. In order to obtain ground-truth the data were first acquired, reconstructed and labeled in 3D using a 4-camera motion capture system operating at a rate of 60 samples/sec. Since our goal is to detect and label the body directly in the camera image plane, a generic camera view was simulated by orthographic projection of the 3D marker coordinates. In the following sections we will indicate with viewing angle the azimuth viewing angle: a value of 0 degrees will correspond to a right-side view, a
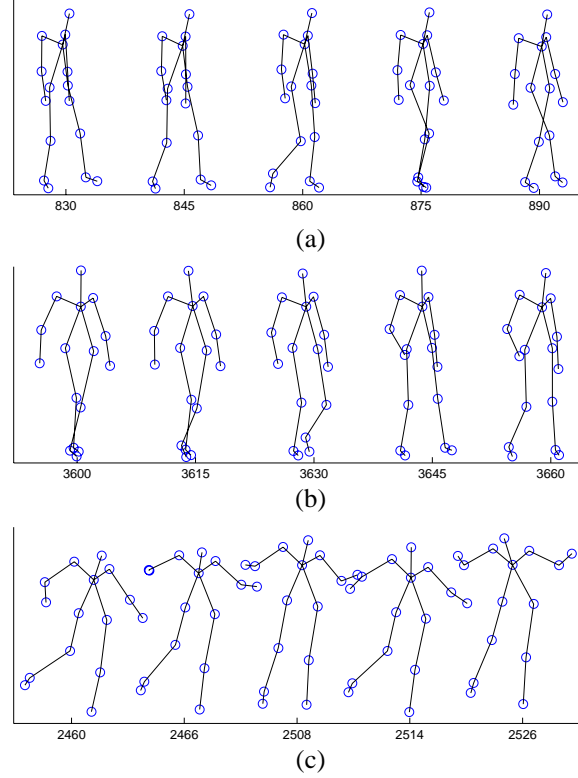

(a)

(b)

(c)

Figure 4: **Sample frames** for the **(a)** walking sequence W3; **(b)** happy walking sequence HW; **(c)** dancing sequence DA. The nubmers on the horizontal axes are the frame numbers.

value of 90 to a frontal view of the subject. Five sequences were acquired each around 2 minutes long. In the next sections they will be referred as follows: Sequences W1 (7000 frames), W2 (7000 frames): relaxed walking forward and backwards along almost straight paths (with ± 20 degree deviations in heading); W3 (5305 frames): relaxed walking, with the subject turning around now and then (Figure 4(a)); Sequence HW (5210 frames): walking in a happy mood, moving the head, arms, hips more actively (Figure 4(b)); Sequence DA (3497 frames): dancing and jumping (Figure 4(c)), with the subject moving his legs and arms freely and much faster than in the previous four sequences. Given that the data were acquired from the same subject and that orthographic projection was used to simulate a camera view, our data were already normalized in scale. The velocity of each candidate marker was obtained by substracting the positions in two consecutive frames.

Among the above five sequences, walking sequences W1 and W2 are the relatively simple ones, so W1 and W2 were first used to test the validity of the probability model and the performance of two possible body decompositions (Figure 2). Since the heading direction of W1 and W2 was roughly along a line, performance under changing viewing angles was also investigated. Then experiments were conducted using W3, HW and DA to see how the model worked for more violent and non-periodic motions.

4.1. Detection of individual triangles

In this section, the performance of the probabilistic model for individual triangles is examined. In the training phase, the joint Gaussian parameters (mean and covariance) for each triangle in Figure 2 were estimated from walking sequence W1. In the test phase, for each frame in W2, each triangle probability was evaluated for all possible combinations of markers ($16 \times 15 \times 14$ different combinations). Ideally, the correct combination of markers should produce the highest probability for each respective triangle. Otherwise, an error occured. Figure 5(a) shows how well each triangle's joint probability model detects the correct set of markers. Figure 5(b) shows a similar result for the conditional probability densities of triangles, where for each triangle conditional probability density $p(L_i|L_j, L_k)$, we computed the probability of L_i for all the possible markers (14 choices), given the correct choice of markers for L_j and L_k. Figure 5 shows that the Gaussian model is very good for most triangles (in the joint case, if a triangle is chosen randomly, then the chance of getting the correct one is 3×10^{-4} and the probability models do much better than that).
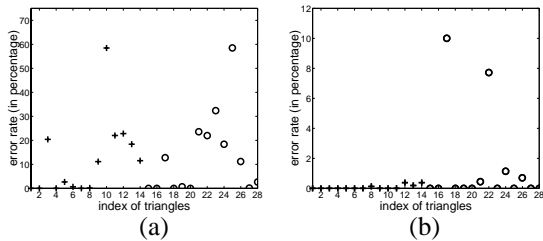

Figure 5: **Local model error rates** (percentage of frames for which the correct choice of markers did not maximize each individual triangle probability). Triangle indices are those of the two graph models of Figure 2. '+': results for decomposition Figure 2(a); 'o': results for decomposition Figure 2(b). **(a)** joint probability model **(b)** conditional probability model

It is not surprising that the performance of some triplets is much worse than others. The worst triangles in Figure 5(a) are those with left and right knees, which makes sense because the two knees are so close in some frames that it is even hard for human eyes to distinguish between them. Therefore, it is also hard for the probability model to make the correct choice.

Further investigation of the behavior of the triangle probabilities revealed that, for frames in which the correct choice of markers did not maximize a triangle probability, that probability was nevertheless quite close to the maximal value. Figure 6 shows the ratio of the probabilities of the correct choice over the maximizing choice for the two worst behaving triangles, over the set of frames where the errors occured. Figure6(a) shows the ratio of the joint probability distribution for triangle 10 (consisting of right hip, left knee, and right knee, as in figure 2(a)). Figure 6(b) shows the ratio of the conditional probability distribution for triangle 17 (head, neck, and left shoulder). Although these two triangles had the highest error rates, the correct marker com-
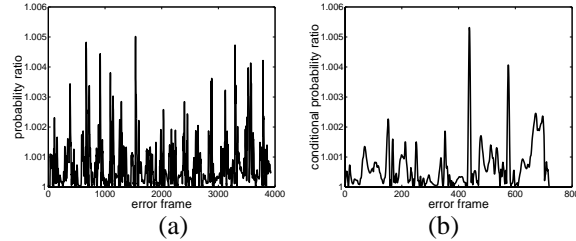

Figure 6: **probability ratio (correct markers vs. the solution with the highest probability when an error happens.)** The horizontal axis is the index of frames where error happens. **(a)** joint probability ratio for triangle 10 or 25 (RH, LK, RK) **(b)** conditional probability ratio for triangle 17 (H, N, LS)

bination was always very close to being the highest ranking, always less than a factor of 1.006 away. This is a good indication that the individual triangle probability models encode the distribution quite well.

4.2. Performance of different body graphs

We did experiments on the two decompositions in Figure 2. The training sequence W1 and the test sequence W2 were under the same viewing angle: 45 degrees, which is between the side view and the front view. Table 1 shows the results. The *frame-by-frame error* is the percentage of frames in which errors occurred, and *label-by-label error* is the percentage of markers wrongly labeled out of all the markers in all the testing frames. Label-by-label error is smaller than frame-by-frame error because an error in a frame does not mean all the markers are wrongly labeled.

decomposition model	(a)	(b)
frame-by-frame error	0.27%	13.13%
label-by-label error	0.06%	1.61%

Table 1: **Error rate using the models in Figure 2**

The performance of the algorithm using the decomposition of Figure 2(a) is almost perfect and much better than that of (b), which is consistent with our expectation (by Figure 5, the local performance of decomposition Figure 2(a) is better than that of Figure 2(b)). We used the better model in the rest of the experiments.

4.3. Viewpoint invariance

In the previous sections the viewing angle for training and for testing was the same. Here we explore the behavior of the method when the testing viewing angle is different from that used during training. Figure 7 shows the results of three such experiments where walking sequence W1 was used as the training set and W2 as the test set .

The solid line in Figure 7(a) shows the percentage of frames labeled correctly when the training was done at a viewing angle of 90 degrees (subject facing the camera) and the testing viewing angle was varied from 0 degrees (right
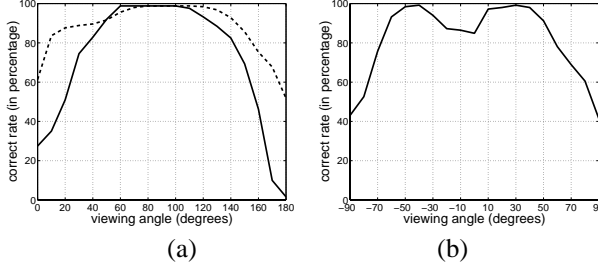
Figure 7: **Labeling performance as a function of viewing angle. (a)** Solid line : percentage of correctly labeled frames as a function of viewing angle, when the training was done at 90 degrees (frontal view). Dashed line: training was done by combining data from views at 30, 90, and 150 degrees. **(b)** Labeling performance when the training was done at 0 degrees (right side view of walker). The dip in performance near 0 degrees is due to the fact that from a side view orthographic projection without inter-body occlusions it is almost impossible to distinguish left and right.

side view) to 180 degrees (left side view) in increments of 10 degrees. When the viewing angle was between 60 to 120 degrees almost all frames were labeled correctly, thus showing that the probabilistic model learned at 90 degrees is insensitive to changes in viewpoint by up to 30 degrees.

The solid line in Figure 7(b) shows the results of a similar experiment where the training viewpoint was at 0 degrees (right side view) and the testing angle was varied from -90 degrees (back view) to 90 degrees (front view) in 10 degree increments. A noticeable dip in the performance centered around 0 degrees is visible in the plot. Inspection of the errors which occurred at these viewing angles revealed that they consisted solely of confusions between homologous left-right leg parts ; i.e., the two hips were sometimes confused, as were the knees, the ankles, and the feet. Considering that an orthographic projection of the 3-D data was used to create the 2-D views, this result is not surprising; given an orthographic side view of a person walking (with no intra-body occlusions) a person viewing the motion is unable to distinguish the left and right sides of the body. Thus, modulo this left-right ambiguity, the model learned at 0 degrees viewing angle is insensitive to changes in viewpoint of up to 40 degrees.

The dashed line in Figure 7(a) shows the results of an experiment to try to increase the invariance of the probabilistic model with respect to changes in viewpoint. The same 3-D training sequence was used to generate three 2-D data sequences with viewing angles at 30, 90, and 150 degrees. The three 2-D sequences were combined, and used all together to learn the probability density functions of the graph triangles. As shown in the plot, this procedure does in fact improve the labeling accuracy. At 0 degrees, the only errors were the above mentioned left-right ambiguity within the legs. Between 10 and 60 degrees, besides left-right errors, also the feet and ankles were confused. From 120 to 180 degrees, the errors once again consisted solely of swapped left and right body parts.

4.4. Performance with different motions

The previous sections show that for simple motions very good results can be achieved using the probabilistic model. Here we want to investigate how the method works for more general sets of motions. We did experiments on walk sequence W3, happy walking sequence HW and dancing sequence DA. Each sequence was divided into four segments for a total of twelve segments. To test a segment, frames from all the other eleven segments were used as the training set. The error rates for different sequences are obtained by averaging the results of the corresponding segments.

test set	ALL	W3	HW	DA
frame error	6.81%	3.02%	4.49%	15.95%
label error	0.69%	0.38%	0.50%	1.45%

Table 2: **Error rates for different sequences**. Frame error means *frame-by-frame error* and label error means *label-by-label error*. ALL: average over all three sequences; W3:walking sequence; HW: walking in happy mood; DA: dancing sequence

Table 2 shows the error rates for different sequences. The first column is the average result for all the three sequences, and the next three columns show the error rates for walking sequence W3, happy walking sequence HW and dancing sequence DA respectively. The results for walking sequence W3 and happy walking sequence HW are very good, with *frame-by-frame error* less than 5% and *label-by-label error* no more than 0.5%. It is not surprising that the error rates of dancing sequence are higher than the walking sequences because the motions in the dancing sequence are more random and agitated and therefore it is harder to model. Another possible reason is that the dancing sequence is shorter than the other sequences, so the motion of dancing has relatively less weight in the training set.
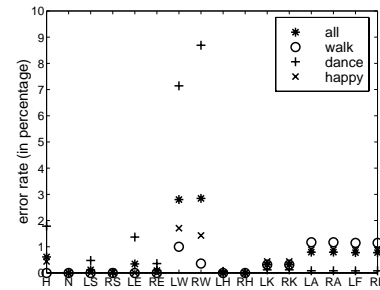

Figure 8: **Error rates for individual body parts.** 'L' and 'R' in label names indicate left and right. H:head, N:neck, S:shoulder, E:elbow, W:wrist, H:hip, K:knee, A:ankle and F:foot.

Figure 8 shows the error rates of each individual body part for each of the sequences. Notice that most errors occur at the left and right wrist (LW and RW) in the dancing sequence. This is because in the dancing sequence wrists are very close to hips in some frames, and the program wrongly took hips as wrists. The reason why the program wouldn't take wrists as hips is that hips have better motion constraints than wrists. In our decomposed body graph Figure 2(a),

both left and right hip (LH and RH) appear in five triangles, but the wrists (LW and RW) are only in one triangle each.

5. Dealing with missing body parts

The labeling method discussed so far assumed that all body parts were detected. However, when dealing with real images, this is not always true. As a first step towards being able to handle missing data, we extended our algorithm to the case where individual triplets may have up to one marker missing. This is done by adding a new special point v_0 (that represents a missing marker) to the set of candidate markers. The definition of joint and conditional probability density for the triplets is extended to include the case where one of the body parts is the missing marker v_0.

Consider the generic triplet (L_1, L_2, L_3). Let $\bar{q_m}$ denote $L_m\ missing$, and q_m denote $L_m\ present$ for $m = 1, 2, 3$. If none of the body parts is missing, then for $i \neq 0$, $j \neq 0$ and $k \neq 0$,

$$
\begin{aligned}
&p(L_1 = v_i, L_2 = v_j, L_3 = v_k, q_1, q_2, q_3) \\
=\ &p(L_1 = v_i, L_2 = v_j, L_3 = v_k | q_1, q_2, q_3)p(q_1, q_2, q_3)
\end{aligned}
$$

where $p(L_1 = v_i, L_2 = v_j, L_3 = v_k | q_1, q_2, q_3)$ is the 10-dimensional probability density function we used in previous sections and $p(q_1, q_2, q_3)$ is the prior probability of all the three body parts present, which can be learned through the training set.

If body part L_1 is missing, then for $i \neq 0$ and $j \neq 0$,

$$
\begin{aligned}
&p(L_1 = v_0, L_2 = v_i, L_3 = v_j, \bar{q_1}, q_2, q_3) \\
=\ &p(L_1 = v_0, L_2 = v_i, L_3 = v_j | \bar{q_1}, q_2, q_3)p(\bar{q_1}, q_2, q_3) \\
=\ &p(L_1 = v_0 | \bar{q_1})p(L_2 = v_i, L_3 = v_j | q_2, q_3)p(\bar{q_1}, q_2, q_3)
\end{aligned}
$$

The second equality can hold because L_1 is missing and therefore it is reasonable to assume that L_1 and other body parts are independent. In the above equation the prior probability $p(\bar{q_1}, q_2, q_3)$ and $p(L_2 = v_i, L_3 = v_j | q_2, q_3)$ can be obtained from the training set. And $p(L_1 = v_0 | \bar{q_1})$ can be estimated as a uniform density covering an ellipsoid in the 4-dimensional sub-space which describes the position and velocity of L_1 in the original 10-dimensional pdf space. The size of the ellipsoid is chosen to be such that, if the body part were present, it would be inside the ellipsoid some high percentage (say 99%) of the time. Therefore, the uniform density can be computed by scaling with an appropriate constant the inverse of the square root of the determinant of the corresponding 4-dimensional sub-matrix of the covariance matrix of the original distribution.

By the same idea, $p(L_1 = v_i, L_2 = v_0, L_3 = v_j, q_1, \bar{q_2}, q_3)$ or $p(L_1 = v_i, L_2 = v_j, L_3 = v_0, q_1, q_2, \bar{q_3})$ can be estimated. Similarly, the lower dimensional case can also be handled.

Two experiments were performed to test the extended algorithm. In the first experiment, the exact same data as in section 4.4 were used, so the results with the extended algorithm could be directly compared to the previous ones.

Table 3 shows the resulting error rates. The possibility that markers may be missing increases the error rate only slightly for the walking sequences, but dramatically for the dance sequence. The reason is that for the difficult cases such as some frames in the dancing sequence, even if the ground truth has the highest probability, the probability itself may not be high. So when the missing points are allowed, the configuration with missing points get higher probability than the ground truth.

test set	ALL	W3	HW	DA
frame error	15.75%	3.8%	8.89%	43.84%
label error	2.25%	0.48%	0.83%	7.04%

Table 3: **Error rates for different sequences (using the algorithm that allows for missing markers).** Frame error means *frame-by-frame error* and label error means *label-by-label error*. ALL: all the three sequences; W3:walking sequence; HW: happy walking sequence; DA: dancing sequence
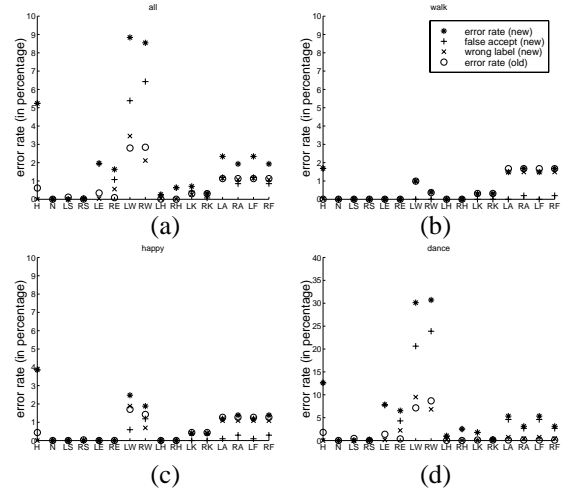

Figure 9: **Comparison of the extended algorithm (new: allowing missing points) with the original (old: without considering missing points).** The exact same sequences were used with the two algorithms (all markers present in all frames). The error rates of the new algorithm can be decomposed into **false accept** (missing marker chosen instead of correct real marker) and **wrong label** (the wrong real marker is chosen). Horizontal axis labels denote individual body parts. **(a)** ALL: average performance of all the three sequences **(b)** walking sequence W3 **(c)** happy walking sequence HW **(d)** dancing sequence DA

Figure 9 sheds more light on this result by decomposing the error rates into different categories on a label-by-label basis. Note that with the inclusion of missing markers, there are three types of errors that can occur – false accept: the missing marker is chosen but the real point is there; false reject: some marker is chosen but actually it's missing (can't occur in this experiment since all markers were always present); wrong label: an incorrect point is chosen instead of the ground truth. Notice that for the dancing sequence the majority of errors occur in the labeling of the wrists, with the most common type of error being that the wrists are deemed missing when in fact they are

test set	ALL	W3	HW	DA
frame error	16.32%	5.87%	9.63%	42.01%
label error	2.28%	0.7%	0.97%	6.61%

Table 4: **overall error rates for different sequences (using the algorithm allowing missing point, and all the sequences with at most one missing point).** Frame error means *frame-by-frame error* and label error means *label-by-label error*. ALL: all the three sequences; W3:walking sequence; HW: happy walking sequence; DA: dancing sequence

present. This error may arise as the combination of three effects. First, only one triangle in the graph models contains the wrists, whereas most other body parts are represented in multiple triangles. Second, in the dancing sequence the arms moved quite randomly in relation to the rest of the body (even with respect to the elbows), so that the estimate of the wrists' probability density is not as 'tight' as the one corresponding to other body parts. Finally, the dancing sequence was shorter than the other two sequences, so that it accounted for only approximately one-seventh of the training data.

The second experiment tested the performance of the algorithm when some body part was missing in the data. The program run 16 times on all the sequences, each time a different body part was removed. Table 4 shows the resulting average error rates including the case of all the markers present and the case of one marker removed. The results are not much different from the test case when no markers were removed.

6. Summary and Conclusion

We have built a perceptual model for solving the labeling problem based on finding the set of labels which maximizes the joint probability density function defined over the entire body. By decomposing the body into a decomposable graph composed of triplets according to the kinematic-chain structure of the human body, dynamic programming has been used to find the globally optimal solution in an efficient manner. For each triplet of the graph a Gaussian model of the mutual positions and velocities of the body parts is learned. When these PDFs are composed together, they define the joint probability density function of the entire body.

Experiments done on a frame-by-frame basis indicate that the learned probability models for triplets of body parts are reliable, although care needs to be taken in choosing the triplets. The method was tested on several types of motions and has an overall label-by-label error rate of 0.7% (without considering missing points), although very vigorous and random motions of wrists were not modeled as well as the rest of the body. The method is robust to point of view, having good performance with variations of viewpoint up to 30 to 40 degrees from that of training. An initial extension of the method able to handle the occurrence of missing markers (with up to one missing marker per triplet) also showed good labeling capability, with an overall label-by-label error rate of 2.3%.

Our model may be applied to body detection and labeling in markerless monocular image sequences by detecting image features and regions using the techniques described in [1, 9, 5, 10]. The position, motion and photometric characteristics of these features and regions would be inserted into the joint probability density function that describes the set of 'typical' body postures and motions. Extensions to the current model include training on a larger set of motions, testing different probability density functions that are more sophisticated than the Gaussian, dealing with different persons and different scales, extending viewpoint invariance to 360^0, enforcing certain constrains to reduce the computational cost and using a temporal model to further decrease labeling errors.

References

[1] A.Jepson and M.J.Black. Mixture models for optical flow computation. In *Proc. IEEE CVPR*, pages 760–761, 1993.

[2] E. D. Bernardo, L. Goncalves, and P. Perona. Monocular tracking of the human arm in 3d: Real-time implementation and experiments. In *International Conference on Pattern Recognition*, August 1996.

[3] B.Moghaddam and A.Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:696–709, 1997.

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

[5] C.Wren, A.Azarbayejani, T.Darrell, and A.Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997.

[6] L. Goncalves, E. D. Bernardo, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3d. In *Proc. 5th Int. Conf. Computer Vision*, pages 764–770, Cambridge, Mass, June 1995.

[7] I. Haritaoglu, D.Harwood, and L.Davis. Who, when, where, what: A real time system for detecting and tracking people. In *Proceedings of the Third Face and Gesture Recognition Conference*, pages 222–227, 1998.

[8] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[9] J.Shi and C.Tomasi. Good features to track. In *Proc. IEEE CVPR*, pages 593–600, 1994.

[10] M.Yang and N.Ahuja. Extracting gestural motion trajectories. In *International Conference on Face and Gesture Perception*, pages 10–15, Nara, Japan, April 1998.

[11] J. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Proceedings of the workshop on Motion of Non-Rigid and Articulated Bodies*, pages 16–24, November 1994.

[12] K. Rohr. Incremental recognition of pedestrians from image sequences. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 8–13, New York City, June, 1993.

[13] A. Y.Amit. Graphical templates for model registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:225–236, 1996.