Estimating Human Body Configurations using Shape Context Matching

Greg Mori and Jitendra Malik
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720
{mori,malik}@cs.berkeley.edu

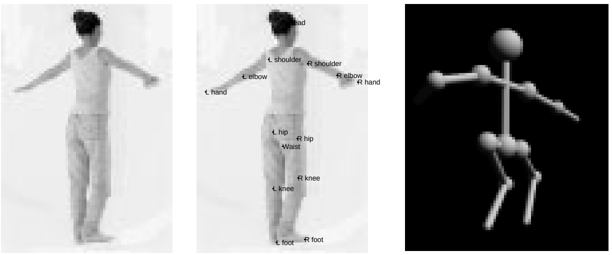
Abstract

The problem we consider in this paper is to take a single two-dimensional image containing a human body, locate the joint positions, and use these to estimate the body configuration and pose in three-dimensional space. The basic approach is to store a number of exemplar 2D views of the human body in a variety of different configurations and viewpoints with respect to the camera. On each of these stored views, the locations of the body joints (left elbow, right knee etc) are manually marked and labelled for future use. The test shape is then matched to each stored view, using the technique of shape context matching. Assuming that there is a stored view sufficiently similar in configuration and pose, the correspondence process will succeed. The locations of the body joints are then transferred from the exemplar view to the test shape. Given the joint locations, the 3D body configuration and pose are then estimated. We present results of our method on a corpus of human pose data.

1 Introduction

As indicated in Figure 1, the problem we consider in this paper is to take a single two-dimensional image containing a human body, locate the joint positions, and use these to estimate the body configuration and pose in three-dimensional space. Variants include the case of multiple cameras viewing the same human, tracking the body configuration and pose over time from video input, or analogous problems for other articulated objects such as hands, animals or robots. A robust, accurate solution would facilitate many different practical applications–e.g. see Table 1 in Gavrila's survey paper[10]. From the perspective of computer vision theory, this problem offers an opportunity to explore a number of different tradeoffs in –the role of low level vs. high level cues, static vs. dynamic information, 2D vs. 3D analysis, etc. in a concrete setting where it is relatively easy to quantify success or failure.

There has been considerable previous work on this problem [10]. Broadly speaking, it can be categorized into two major classes. The first set of approaches use a 3D model for estimating the positions of articulated objects. Pioneering work was done by O'Rourke and Badler [18], Hogg[11] and Yamamoto and Koshikawa [25]. Rehg and Kanade [19] track very high DOF articulated objects such as hands. Bregler and Malik [5] use optical flow measurements from a video sequence to track joint angles of a 3D model of a human, using the product of exponentials representation for the kinematic chain. Kakadiaris and Metaxas[15] use multiple cameras and match occluding contours with projections from a deformable 3D model. Gavrila and Davis [9] is another 3D model based tracking approach, as is the work of Rohr [20] for tracking walking pedestrians. It should be noted that pretty much all the tracking methods require a hand-initialized first video frame.



(a) (b) (c)

Figure 1: The goal of this work. (a) Input image. (b) Automatically extracted keypoints. (c) 3D rendering of estimated body configuration. In this paper we present a method to go from (a) to (b) to (c).

The second broad class of approaches does not explicitly work with a 3D model, rather 2D models trained directly from example images are used. There are several variations on this theme. Baumberg and Hogg[1] use active shape models to track pedestrians. Wren et al. [24] track people as a set of colored blobs. Morris and Rehg [17] describe a 2D scaled prismatic model for human body registration. Ioffe and Forsyth [12] perform low-level processing to obtain candidate body parts and then use a mixture of trees to infer likely configurations. Song et al. [21] use a similar technique involving feature points and inference on a tree model. Toyama and Blake [23] use 2D exemplars to track people in video sequences. Brand [4] learns a probability distribution over pose and velocity configurations of the moving body and uses it to infer paths in this space.

In this paper we consider the most basic version of the problem–estimating the 3D body configuration based on a single uncalibrated 2D image. The basic idea is to store a number of exemplar 2D views of the human body in a variety of different configurations and viewpoints with respect to the camera. On each of these stored views, the locations of the body joints (left elbow, right knee etc) are manually marked and labelled for future use. The test shape is then matched to each stored view, using the shape context matching technique of Belongie, Malik and Puzicha [3]. This technique is based on representing a shape by a set of sample points from the external and internal contours of an object, found using an edge detector. Assuming that there is a stored view "sufficiently" similar in configuration and pose, the correspondence process will succeed. The locations of the body joints are then "transferred" from the exemplar view to the test shape. Given the joint locations, the 3D body configuration and pose are estimated using Taylor's algorithm [22].

The structure of the paper is as follows. In section 2 we elaborate on the estimation approach described above. We show experimental results in section 3. We discuss the issue of models versus exemplars in section 4. Finally, we conclude in section 5.

2 Estimation Method

In this section we provide the details of the configuration estimation method proposed above. We first obtain a set of boundary sample points from the image. Next, given a set of exemplars extracted from a training set (method for obtaining exemplars is outlined in Appendix A), we find the best match among the exemplars. We use this match, along with correspondences between boundary points on the test image and the exemplar, to estimate the 2D image positions of 14 *keypoints* (hands, elbows, shoulders, hips, knees, feet, head and waist) on the test image. These keypoints can then be used to construct an estimate of the 3D body configuration in the test image.

2.1 Matching using Shape Contexts

In our approach, a shape is represented by a discrete set $\mathcal{P} = \{p_1, \ldots, p_n\}$, $p_i \in \mathbb{R}^2$, of n points sampled from the internal or external contours on the shape.

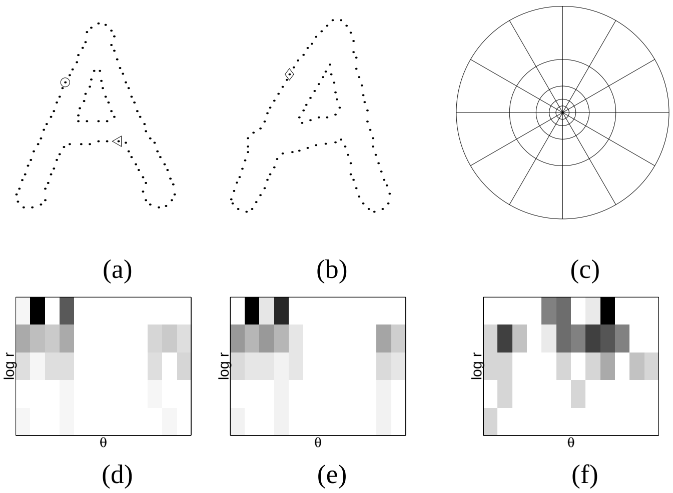


Figure 2: Shape contexts. (a,b) Sampled edge points of two shapes. (c) Diagram of log-polar histogram bins used in computing the shape contexts. We use 5 bins for $\log r$ and 12 bins for θ. (d-f) Example shape contexts for reference samples marked by $\circ, \diamond, \triangleleft$ in (a,b). Each shape context is a log-polar histogram of the coordinates of the rest of the point set measured using the reference point as the origin. (Dark=large value.) Note the visual similarity of the shape contexts for $\circ$ and $\diamond$, which were computed for relatively similar points on the two shapes. By contrast, the shape context for $\triangleleft$ is quite different.

We first perform Canny edge detection [6] on the image to obtain a set of edge pixels on the contours of the body. We then sample some number of points (around 300 in our experiments) from these edge pixels to use as the sample points for the body. Note that this process will give us not only external, but also internal contours of the body shape. The internal contours are essential for estimating configurations of self-occluding bodies. See Figure 4 (a) for examples of sample points.

For each point p_i on a given shape, we want to find the "best" matching point q_j on another shape. This is a correspondence problem similar to that in stereopsis. Experience there suggests that matching is easier if one uses a rich local descriptor. Rich descriptors reduce the ambiguity in matching.

The *shape context* was introduced in [3] to play such a role in shape matching. Consider the set of vectors originating from a point to all other sample points on a shape. These vectors express the configuration of the entire shape relative to the reference point. Obviously, this set of $n-1$ vectors is a rich description, since as n gets large, the representation of the shape becomes exact.

The full set of vectors as a shape descriptor is much too detailed since shapes and their sampled representation may vary from one instance to another in a category. The *distribution* over relative positions is a more robust and compact, yet highly discriminative descriptor. For a point p_i on the shape, compute a coarse histogram h_i of the relative coordinates of the remaining $n-1$ points,

$$h_i(k) = \# \{ q \neq p_i \ : \ (q - p_i) \in \text{bin}(k) \} \quad .$$

This histogram is defined to be the *shape context* of p_i. The descriptor should be more sensitive to differences in nearby pixels, which suggests the use of a log-polar coordinate system. An example is shown in Fig. 2(c). Note that the scale of the bins for $\log r$ is chosen adaptively, on a per shape basis. This makes the shape context feature invariant to scaling.

As in [3], we use χ^2 distances between shape contexts as a matching cost between sample points.

We would like a correspondence between sample points on the two shapes that enforces the uniqueness of matches. This leads us to formulate our matching of a test body to an exemplar body as an assignment problem (also known as the weighted bipartite matching problem) [7]. We find an optimal assignment between sample points on the test body and those on the exemplar.

To this end we construct a bipartite graph (Figure 3). The nodes on one side represent sample points on the test body, on the other side the sample points on the exemplar. Edge weights between nodes in this bipartite graph represent the costs of matching sample points. Similar sample points will have a low matching cost, dissimilar ones will have a high matching cost. ϵ-cost outlier nodes are added to the graph to account for occluded points and noise - sample points missing from a shape can be assigned to be outliers for some small cost. We use the assignment problem solver in [14] to find the optimal matching between the sample points of the two bodies.

We compare the test body to all of the exemplars from our training set. The exemplar with the lowest total matching cost is chosen for use in keypoint estimation.

Note that the output of more specific filters, such as face or hand detectors, could easily be incorporated into this framework. The matching cost between sample points can be measured in many ways.

2.2 Locating Keypoints

The next step is to estimate the 2D image positions of the 14 keypoints (hands, elbows, shoulders, hips, knees, feet, head, waist) on the test body. From the solution to the assignment problem in section 2.1 we have correspondences between sample points (not keypoints) on the test body and the closest exemplar body. In addition, each exemplar from the training set has user-clicked (x, y) locations of keypoints. We would like to use these correspondences and exemplar keypoint locations to estimate the keypoint positions on the test body.

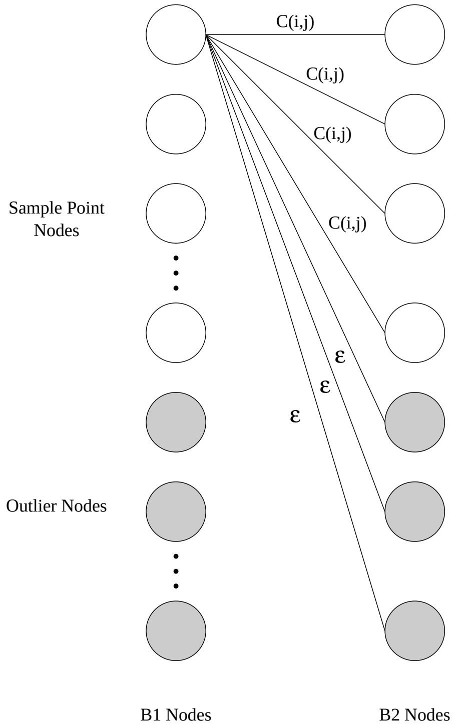


Figure 3: The bipartite graph used to match sample points of two bodies. Only the edges from the first node are shown for clarity. Each node from $B1$ is connected to every node from $B2$. In addition, ϵ-cost outlier nodes are added to either side. These outlier nodes allow us to deal with missing sample points between figures (arising from occlusion and noise).

We use a simple method for this estimation process (Figure 4). For each keypoint k_e^i on the exemplar we want to estimate its position on the test body k_t^i. We select a set of sample points P_e^i from the exemplar as a support for this keypoint. The solution A_{opt} to the assignment problem gives a corresponding set of points $P_t^i = A_{opt}(P_e^i)$ on the test body. We estimate the best (in the least-squares sense) transformation $T^i : R^2 \rightarrow R^2$ that takes P_e to P_t. We then apply this transformation to the keypoint: $k_t^i = T^i(k_e^i)$.

In our experiments, the support for a keypoint is defined to be all sample points within a disc of some small radius. The transformation T is simply a translation. One could alternatively use affine or rigid transformations.

2.3 Estimating 3D Configuration

We use Taylor's method in [22] to estimate the 3D configuration of a body given the keypoint position estimates. Taylor's method works on a single 2D image, taken with an uncalibrated camera.

It assumes that we know:

1. the image coordinates of keypoints (u, v)
2. the relative lengths l of body segments connecting these keypoints
3. a labelling of "closer endpoint" for each of these body segments
4. that we are using a scaled orthographic projection model for the camera

We can then solve for the 3D configuration of the body $\{(X_i, Y_i, Z_i) : i \in keypoints\}$ up to some ambiguity in scale s. The method considers the foreshortening of each body segment to construct the estimate of body configuration. For each pair of body segment endpoints, we have the following equations:

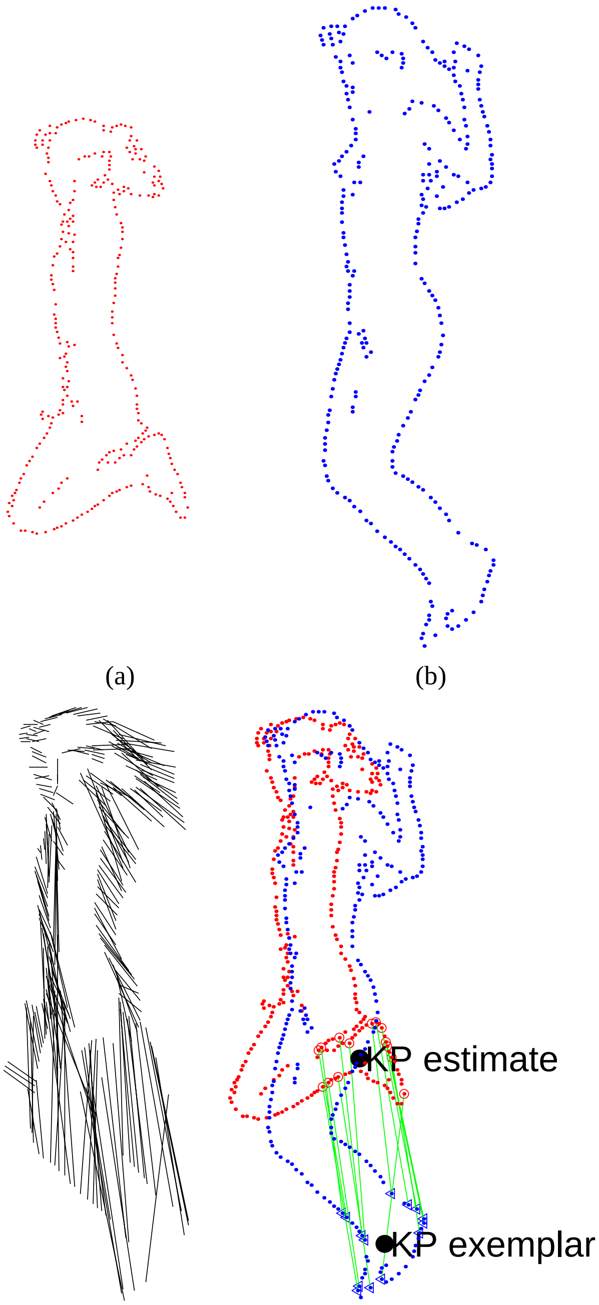
$$\begin{aligned}
l^2 &= (X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2 \\
(u_1 - u_2) &= s(X_1 - X_2) \\
(v_1 - v_2) &= s(Y_1 - Y_2) \\
dZ &= (Z_1 - Z_2) \\
\Longrightarrow dZ &= \sqrt{l^2 - ((u_1 - u_2)^2 + (v_1 - v_2)^2)/s^2}
\end{aligned}$$

To estimate the configuration of a body, we first fix one keypoint as the reference point and then compute the positions of the others with respect to the reference point. Since we are using a scaled orthographic projection model the X and Y coordinates are known up to the scale s. All that remains is to compute relative depths of endpoints dZ. We compute the amount of foreshortening, and use the user-supplied "closer endpoint" labels from the closest matching exemplar to solve for the relative depths.

Moreover, Taylor notes that the minimum scale s_{min} can be estimated from the fact that dZ cannot be complex.

$$s \geq \frac{\sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2}}{l}$$

This minimum value is a good estimate for the scale since one of the body segments is often perpendicular to the viewing direction.



(a) (b)

(a) (b)

Figure 4: Locating keypoints. (a) Sample points on exemplar and test body, with lines showing correspondences. (b) Support for estimating transformation of right foot, along with estimate of position.

3 Results

We applied our method to the human figures in Eguchi's pose file collection [8]. This book contains a thorough collection of artist reference photographs of one model performing typical actions. The book depicts actions such as skipping, jumping, crawling, and walking. Each action is photographed from about 8 camera angles each, and performed in 3 levels of clothing (nude, casual, and skirt).

We selected 48 training images (two from each clothing-action pair), and derived 20 exemplars from this training set. These 20 exemplars were then used to match with 48 different test images (again two from each class).

The positions of 14 keypoints (hands, elbows, shoulders, feet, knees, hips, waist and head) were manually labelled on each training image. We automatically locate the 2D positions of these keypoints in each of our test images, and then estimate a 3D configuration. Figures 5,6, and 7 show some example 3D configurations that were obtained using our method.

Figure 8 shows distributions of error in the location of the 2D keypoints. Ground truth was obtained through user-clicking. The large errors (especially for hands and feet) can be attributed to ambiguities between left and right limbs. Discrimination between left and right limbs in a 2D image requires more complex reasoning.

4 Discussion

We have presented a hybrid exemplar and model-based approach to estimating human body configurations. Our method matches using 2D exemplars, estimates keypoint locations, and then uses these keypoints in a model-based algorithm for determining 3D body configuration. We believe that this hybrid approach has advantages over strictly model or exemplar-based approaches.

3D model-based matching is difficult. An articulated model of a person has a high number of degrees of freedom. Typical approaches involve projecting the model into the image, computing an error term, and minimizing this error. This minimization is problematic – there are many local minima, and a gradient descent procedure will have difficulty finding a correct pose match. In contrast, our exemplar-based matching process is fast (assignment problems can be solved in $O(N^3)$ time for N sample points), and guaranteed to find the global optimum. There could be some concerns over the number of exemplars needed to deal with a wide range of poses. However, previous work [3, 16] suggests that our shape context based matching will be successful in scaling to handle more classes of activities and variation in human appearance.

While 2D exemplars make solving the correspondence problem easy, a good case can be made for inferring 3D pose parameters as features to be used in dynamic models for tracking and activity recognition. There appear to be two advantages: (1) the space of pose parameters has reduced dimensionality compared to the image measurement space and (2) acquiring models of activities as trajectories in pose space makes it unnecessary for us to learn these models separately for various camera viewpoints as would be necessary in a purely view-based approach.

5 Conclusion

In this paper we have presented a simple, yet apparently effective, approach to estimating human body configurations in 3D, based on matching with multiple 2D exemplars. There are several obvious directions for future work

1. The 2D shape matching could make use of additional attributes such as distances from labelled features such as faces or hands, orientation of edge elements etc.

2. The 2D matching process can be iterated with deformation to better align the exemplar with test shape. In previous shape context work, the thin plate spline was used as the deformation model and found to considerably improve both recognition accuracy as well as reduce the number of exemplar views that need to be stored. For human body matching, the 3D kinematic model is an obvious choice. Other alternatives might be learned 2D deformation models along the lines of e.g. Jojic et al.[13].

3. When video data is available, then estimation can benefit from temporal context. Human dynamic models are most naturally expressed in joint angle space, and our hybrid framework provides a natural way to incorporate this information in the 3d configuration estimation stage.

A Obtaining Exemplars

We derive a set of exemplars from a training set using the k-medoids algorithm [2]. We compute pairwise distances between training bodies using the method outlined in section 2.1. These distances are fed into the k-medoids algorithm to produce clusters of similar training bodies. The k-medoids algorithm is an analog of the k-means algorithm in which one has only pairwise distances between points, and no underlying metric space. The cluster centers are constrained to be actual elements of the cluster. We use these cluster centers as our exemplars.

We would like a relatively small set of exemplars that adequately covers the variation in body configurations. The k-medoids algorithm gives us just that, clustering similar configurations to be represented by a single exemplar.

References

[1] A. Baumberg and D. Hogg. Learning flexible models from image sequences. *Lecture Notes in Computer Science*, 800:299–308, 1994.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, November 2000.

[3] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Eighth IEEE International Conference on Computer Vision*, volume 1, pages 454–461, Vancouver, Canada, July 2001.

[4] M. Brand. Shadow puppetry. *Proc. 7th Int. Conf. Computer Vision*, pages 1237–1244, 1999.

[5] C. Bregler and J. Malik. Tracking people with twists and exponential maps. *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 8–15, 1998.

[6] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8:679–698, 1986.

[7] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

[8] H. Eguchi. *Moving Pose 1223*. Bijutsu Shuppan-sha, 1995.

[9] D. Gavrila and L. Davis. 3d model-based tracking of humans in action: A multi-view approach. *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 73–80, 1996.

[10] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.

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

[12] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *Proc. 8th Int. Conf. Computer Vision*, volume 1, pages 690–695, 2001.

[13] N. Jojic, P. Simard, B. Frey, and D. Heckerman. Separating appearance from deformation. *Proc. 8th Int. Conf. Computer Vision*, 2:288–294, 2001.

[14] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.

[15] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion. *IEEE Trans. PAMI*, 22(12):1453–1459, 2000.

[16] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. To appear at CVPR 2001.

[17] D. Morris and J. Rehg. Singularity analysis for articulated object tracking. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 289–296, 1998.

[18] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. PAMI*, 2(6):522–536, 1980.

[19] J. M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. *Lecture Notes in Computer Science*, 800:35–46, 1994.

[20] K. Rohr. Incremental recognition of pedestrians from image sequences. In *CVPR93*, pages 8–13, 1993.

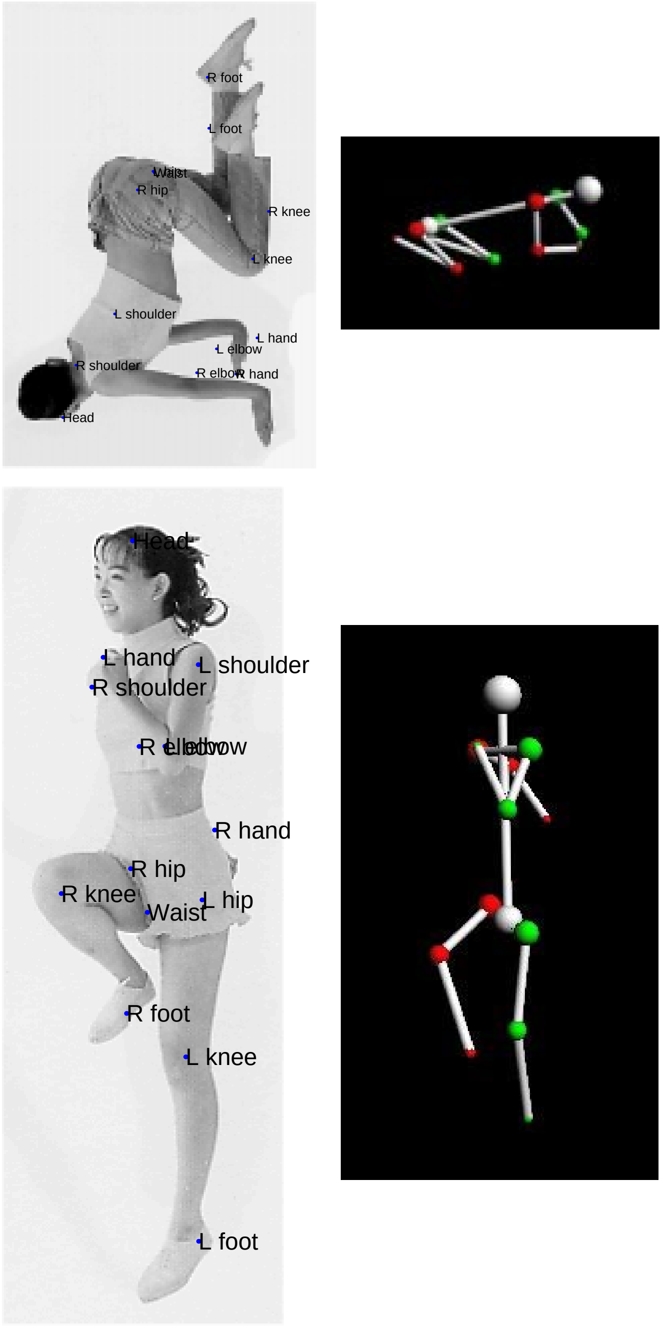
[21] Y. Song, L. Goncalves, and P. Perona. Monocular perception of biological motion - clutter and partial occlusion. In *Proc. 6th Europ. Conf. Comput. Vision*, 2000.

[22] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding (CVIU)*, 80:349–363, 2000.

[23] K. Toyama and A. Blake. Probabilistic exemplar-based tracking in a metric space. In *Proc. 8th Int. Conf. Computer Vision*, volume 2, pages 50–57, 2001.

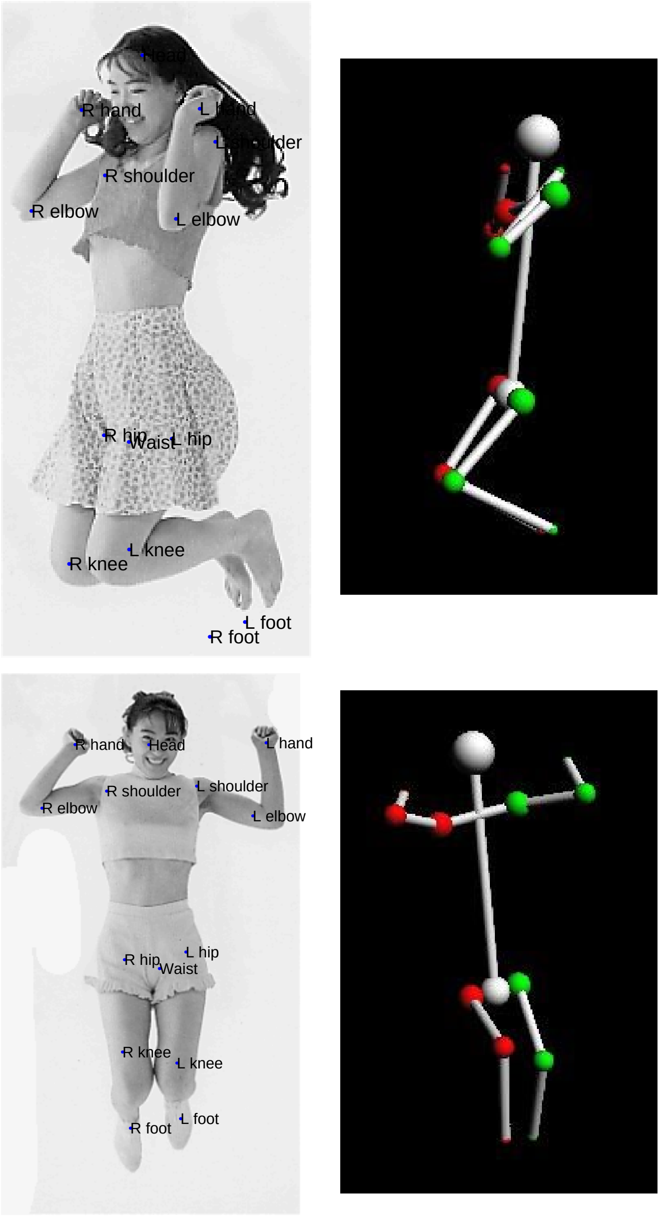
[24] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. PAMI*, 19(7):780–785, July 1997.

[25] M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 664–665, 1991.



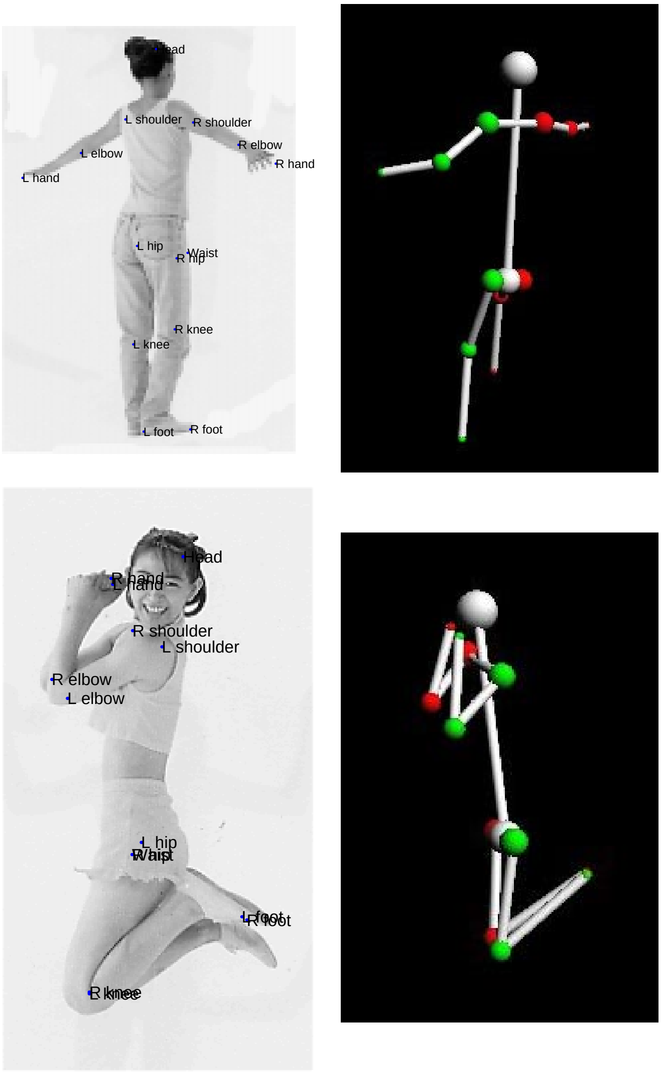
(a) (b)

Figure 5: Example renderings. (a) Original image with located keypoints. (b) 3D rendering (green is left, red is right).



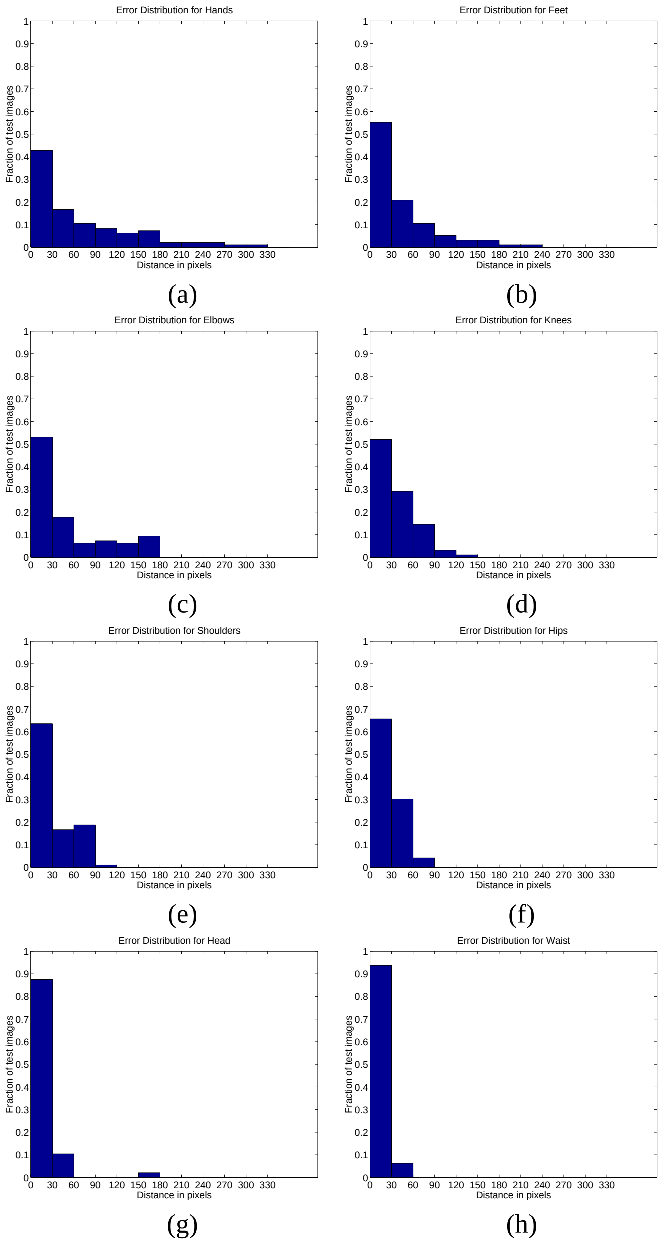
(a) (b)

Figure 6: Example renderings. (a) Original image with located keypoints. (b) 3D rendering (green is left, red is right).



(a) (b)

Figure 7: Example renderings. (a) Original image with located keypoints. (b) 3D rendering (green is left, red is right).



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 8: Distributions of error in 2D location of keypoints. (a) Hands, (b) Feet, (c) Elbows, (d) Knees, (e) Shoulders, (f) Hips, (g) Head, (h) Waist. Error (X-axis) is measured in terms of pixels. Y-axis shows fraction of keypoints in each bin. The average image size is 380 by 205 pixels. Large errors in positions are due to ambiguities regarding left and right limbs.