Estimating Human Body Configurations using Shape Context Matching

Greg Mori and Jitendra Malik

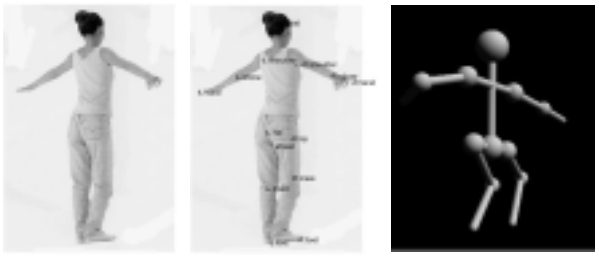
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720
{mori,malik}@cs.berkeley.edu

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 in conjunction with a kinematic chain-based deformation model. 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 can apply this technique to video by treating each frame independently – tracking just becomes repeated recognition! We present results on a variety of datasets.

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[1]. From the perspective of computer vision theory, this problem offers an opportunity to explore a number of different tradeoffs – 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 [1]. 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.



(a) (b) (c)

Fig. 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).

Pioneering work was done by O'Rourke and Badler [2], Hogg[3] and Yamamoto and Koshikawa [4]. Rehg and Kanade [5] track very high DOF articulated objects such as hands. Bregler and Malik [6] 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[7] use multiple cameras and match occluding contours with projections from a deformable 3D model. Gavrila and Davis [8] is another 3D model based tracking approach, as is the work of Rohr [9] for tracking walking pedestrians. It should be noted that pretty much all the tracking methods require a hand-initialized first video frame.

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[10] use active shape models to track pedestrians. Wren et al. [11] track people as a set of colored blobs. Morris and Rehg [12] describe a 2D scaled prismatic model for human body registration. Ioffe and Forsyth [13] perform low-level processing to obtain candidate body parts and then use a mixture of trees to infer likely configurations. Song et al. [14] use a similar technique involving feature points and inference on a tree model. Toyama and Blake [15] use 2D exemplars to track people in video sequences. Brand [16] learns a probability distribution over pose and velocity configurations of the moving body and uses it to infer paths in this space. Carlsson [17, 18] uses *order structure* to compare exemplar shapes with test images. In our previous work [19] we used shape context matching to localize keypoints directly.

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. This is the only user input

required in our method. The test shape is then matched to each stored view, using the shape context matching technique of Belongie, Malik and Puzicha [20, 21]. 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 the algorithm of Taylor [22].

The structure of the paper is as follows. In section 2 we describe the correspondence process mentioned above. Section 3 provides details on a parts-based extension to our keypoint estimation method. We describe the 3D estimation algorithm in section 4. We show experimental results in section 5. Finally, we conclude in section 6.

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, we estimate the 2D image positions of 14 *keypoints* (hands, elbows, shoulders, hips, knees, feet, head and waist) on the image by deformable matching to a set of stored exemplars that have hand-labelled keypoint locations. These estimated keypoints can then be used to construct an estimate of the 3D body configuration in the test image.

2.1 Deformable Matching using Shape Contexts

Given an exemplar (with labelled keypoints) and a test image, we cast the problem of keypoint estimation in the test image as one of deformable matching. We attempt to deform the exemplar (along with its keypoints) into the shape of the test image. Along with the deformation, we compute a matching score to measure similarity between the deformed exemplar and the test image.

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

We first perform Canny edge detection [23] 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 2 for examples of sample points.

The deformable matching process consists of three steps. Given sample points on the exemplar and test image:

1. Obtain correspondences between exemplar and test image sample points

(a) (b) (c) (d)

Fig. 2. Iterations of Deformable matching. Column (a) shows sample points from the two figures to be matched. The bottom figure (exemplar) in (a) is deformed into the shape of the top figure (test image). Columns (b,c,d) show successive iterations of deformable matching. The top row shows the correspondences obtained through the shape context matching. The bottom row shows the deformed exemplar figure at each step. The right arm of the exemplar is deformed to match the right arm of the test image. The left thigh of the figure is also correctly deformed. However, the left lower leg is not deformed properly, due to a failure in the correspondence procedure.

2. Estimate deformation of exemplar
3. Apply deformation to exemplar sample points

We perform a small number (4 in experiments) of iterations of this process to match an exemplar to a test image. Figure 2 illustrates this process.

Sample Point Correspondences

In the correspondence phase, 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 [20, 21] 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.

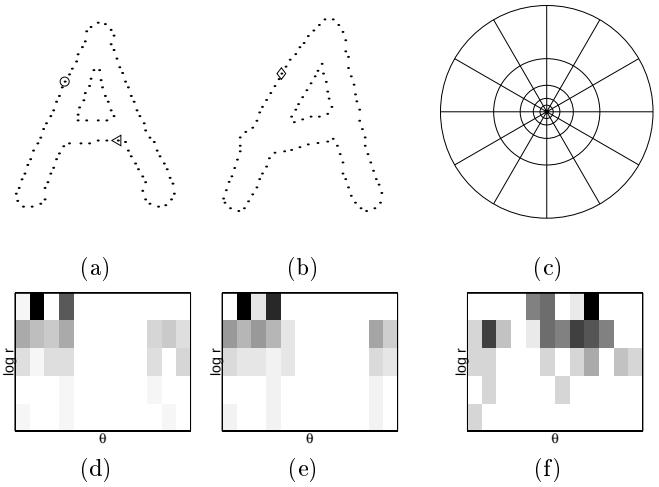


Fig. 3. Shape contexts. (a,b) Sampled edge points of two shapes. (c) Diagram of log-polar histogram bins used in computing the shape contexts. We use 5 bins for $\log r$ and 12 bins for θ. (d-f) Example shape contexts for reference samples marked by $\circ, \diamond, \triangleleft$ in (a,b). Each shape 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.

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,

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

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. 3(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 [20, 21], 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) [24]. 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. 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 [25] to find the optimal matching between the sample points of the two bodies.

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.

Deformation Model

Belongie et al. [20, 21] used thin plate splines as a deformation model. However, it is not appropriate here, as human figures deform in a more structured manner. We use a 2D kinematic chain as our deformation model. The kinematic chain has 9 segments: a torso, upper and lower arms, upper and lower legs (see Figure 4). Our kinematic chain allows translation of the torso, and 2D rotation of the limbs around the shoulders, elbows, hips and knees. This is a simple representation for deformations of a figure in 2D. It only allows in-plane rotations, ignoring the effects of perspective projection as well as out of plane rotations. However, this deformation model is only used in the matching process and is sufficient to capture the small deformations of an exemplar.

(a) (b) (c)

Fig. 4. The deformation model. (a) Underlying kinematic chain. (b) Automatic assignment of sample points to kinematic chain segments on an exemplar. Each different symbol denotes a different chain segment. (c) Sample points deformed using the kinematic chain.

In order to estimate a deformation or deform a body's sample points, we must know to which kinematic chain segment each sample point belongs. On

the exemplars we have hand-labelled keypoints – we use these to automatically assign the hundreds of sample points to segments.

Since we know the segment $S(p_i)$ that each exemplar sample point p_i belongs to, given correspondences $\{(p_i, p_i\prime)\}$ we can estimate a deformation D of the points $\{p_i\}$. Our deformation process starts at the torso. We find the best least squares fit estimate of the translation for the sample points on the torso.

$$D_{torso} = \hat{T} = \arg\min_T \sum_{p_i, S(p_i)=torso} \|T(p_i) - p_i\prime\|^2$$

$$\hat{T} = \frac{1}{N} \sum_{p_i, S(p_i)=torso} (p_i\prime - p_i), \quad \text{where} \quad N = \#\{p_i : S(p_i) = torso\}$$

Subsequent segments along the kinematic chain have rotational joints. We again obtain the best least squares estimates, this time for the rotations of these joints. Given previous deformation D_{prev} along the chain up to this segment, we estimate D_{limb_j} as the best rotation around the joint location c_j:

$$P_j = \{p_i : S(p_i) = limb_j\}$$

$$D_{limb_j} = R_{\hat{\theta}, c_j} = \arg\min_{R_{\theta}, c_j} \sum_{p_i \in P_j} \|R_{\theta, c_j}(D_{prev} \cdot p_i) - p_i\prime\|^2$$

The best angle of rotation $\hat{\theta}$ for this joint is then found as:

$$\hat{\theta} = \arg\min_{\theta} \sum_{p_i \in P_j} (D_{prev} \cdot p_i - c_j) R_{\theta} (c_j - p_i\prime)$$

$$\hat{\theta} = \arctan \frac{\sum_i q_{ix} q_{iy}\prime - \sum_i q_{iy} q_{ix}\prime}{\sum_i q_{ix} q_{ix}\prime + \sum_i q_{iy} q_{iy}\prime}$$

$$\text{where} \quad q_i = D_{prev} \cdot p_i - c_j \quad \text{and} \quad q_i\prime = p_i\prime - c_j$$

Steps 2 and 3 in our deformable matching framework are performed in this manner. We estimate deformations for each segment of our kinematic chain model, and apply them to the sample points belonging to each segment.

We have now provided a method for estimating a set of keypoints using a single exemplar, along with an associated score (the sum of shape context matching costs for the optimal assignment). The simplest method for choosing the best keypoint configuration in a test image is to find the exemplar with the best score, and use the keypoints predicted using its deformation as the estimated configuration. However, with this simple method there are concerns about the number of exemplars needed for a general matching framework. In the following section we will address this by combining matching results for multiple exemplars.

3 Using Part Exemplars

Given a set of exemplars, we can choose to match either entire exemplars or parts, such as limbs, to a test image. The advantage of a parts-based approach that

matches limbs is that of compositionality, which saves us from an exponential explosion in the required number of exemplars. Consider the case of a person walking while holding a briefcase in one hand. If we already have exemplars for a walking motion, and a single exemplar for holding an object in the hand, we can combine these exemplars to produce correct matching results. However, if we were forced to use entire exemplars, we would require a different "holding object and walking" exemplar for each portion of the walk cycle. Using part exemplars prevents the total number of exemplars from growing to an unwieldy size. As long as we can ensure that the composition of part exemplars yields an anatomically correct configuration we will benefit from this reduced number of exemplars.

The matching process is identical to that presented in the preceding section. For each exemplar, we deform it to the shape of the test image. However, instead of assigning a total score for an exemplar, we give a separate score for each limb on the exemplar. This is done by simply summing the shape context costs for sample points on a limb.

With N exemplars we now have N estimates for the location of each of the 6 "limbs" (arms, legs, head, waist). We have to find the "best" combination of these estimates. It is not sufficient to simply choose each limb independently as the one with the best score. There would be nothing to prevent us from violating underlying anatomical constraints. For example, the left leg could be found hovering across the image disjoint from the rest of the body. We need to enforce the *consistency* of the final configuration.

Consider again the case of using part exemplars to match the figure of a person walking while holding a briefcase. Given a match for the arm grasping the briefcase, and matches for the rest of the body, we know that there are constraints on the distance between the shoulder of the grasping arm and the rest of the body. Motivated by this, the measure of consistency we use is the 2D image distance between the bases $b(l_i)$ (shoulder for the arms, hip for the legs) of limbs l_i. We form a tree structure by connecting the arms and the waist to the head, and the legs to the waist. For each of the 5 links in this tree, we compute the N^2 2D image distances between all pairs of bases of limbs from the different exemplars. We now make use of the fact that each whole exemplar on its own is consistent. For a pair of limbs (l_i^1, l_j^2) – limb 1 from exemplar i and limb 2 from exemplar j, we compare the distance d_{ij}^{12} between the bases of the limbs with the same distances using limbs 1 and 2 on the same exemplar. We define the consistency cost of using this pair of limbs (l_i^1, l_j^2) together in matching a test image as the average of the two differences:

$$d_{ij}^{12} = \|b(l_i^1) - b(l_j^2)\|$$

$$Con_{ij}^{12} = \frac{|d_{ij}^{12} - d_{ii}^{12}| + |d_{ij}^{12} - d_{jj}^{12}|}{2} \quad (1)$$

The consistency cost Con_{ii} for using limbs from the same exemplar across a tree link is zero. As the configuration begins to deviate from the consistent exemplars, Con_{ij} increases. We define the total cost of a configuration

$c \in \{1, 2, \ldots, N\}^6$ as the weighted sum of consistency and shape context limb scores:

$$Score(c) = (1 - w_{con}) \sum_{i=1}^{6} SC_{c(i)} + w_{con} \sum_{links:(l_u, l_v)} Con^{uv}_{c(u)c(v)}$$

There are N^6 possible combinations of limbs from the N exemplars. However, we can find the optimal configuration in $O(5N^2)$ time using a dynamic programming algorithm along the tree structure[1].

Moreover, an extension to our algorithm can produce the top K matches for a given test image. Preserving the ambiguity in this form, instead of making an instant choice, is particularly advantageous for tracking applications, where temporal consistency can be used as an additional filter.

```
PRE-PROCESSING:
    % Compute shape contexts for exemplars.
    % Click 14 keypoint locations on each exemplar.
MATCHING:
    % Compute shape contexts for test image I.
    foreach exemplar E_i
        [Limbs_i, Scores_i] = match(E_i, I)
    foreach link L_w = (l^u, l^v)
        foreach exemplar E_i
            foreach exemplar E_j
                D(i, j, w) = Con_ij^uv  % See equation (1)
    Sols = DYNAMIC-SOLVE (Limbs, Scores, D, waist)
    Sol = min(Sols)
DYNAMIC-SOLVE(Limbs, Unary, Binary, root):
    foreach link L_w = (root, next)
        Sols_rec(w) = DYNAMIC-SOLVE(Limbs, Unary, Binary, next)
    foreach exemplar E_i
        % Find best set of nodes from Sols_rec, while using E_i as root.
        ŝ = best(Sols_rec, E_i, Binary(root), Unary(root))
        Sols(i) = ŝ
    return Sols
```

4 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

[1] Dynamic programming on trees is a well-studied problem in the theory community. For an early example of such methods in the computer vision literature, refer to [26]

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.

5 Results

5.1 Pose File

We applied our method to the human figures in Eguchi's pose file collection [27]. 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 2 levels of clothing (casual and skirt).

We selected 18 training images (two from each clothing-action pair), to be used as exemplars. These exemplars were then used to match with 36 different test images (four from each clothing-action pair).

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. Figure 5 shows some example 3D configurations that were obtained using our method.

Figure 6 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. In tracking applications, these ambiguities could be resolved by:

1. for each frame returning the K best solutions instead of just the best one
2. exploiting temporal consistency by using an HMM or other dynamic model

5.2 Speed Skating

We also applied our method to a sequence of video frames of an Olympic speed skater (Figure 7). We chose 5 frames for use as exemplars, upon which we hand-labelled keypoint locations. We then applied our method for configuration estimation to a sequence of 20 frames. Each frame was processed independently – no dynamics were used, and no temporal consistency was enforced!

6 Conclusion

In this paper we have presented a simple, yet apparently effective, approach to estimating human body configurations in 3D. Our method first matches using 2D exemplars, then estimates keypoint locations, and finally uses these keypoints in a model-based algorithm for determining 3D body configuration. Our method requires minimal user input, only the locations of 14 keypoints on each exemplar.

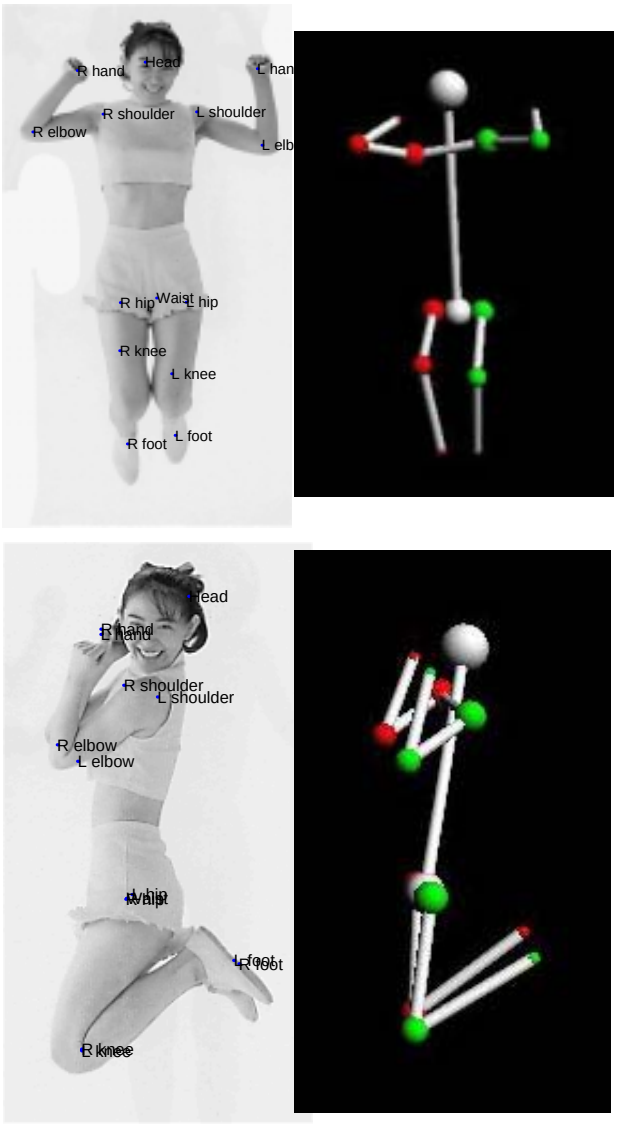
There are several obvious directions for future work:

1. The 2D shape matching could make use of additional attributes such as distances from labelled features such as faces or hands.
2. When video data are available, then estimation can benefit from temporal context. Human dynamic models are most naturally expressed in joint angle space, and our framework provides a natural way to incorporate this information in the 3D configuration estimation stage.

Acknowledgements
This research was supported by Office of Naval Research grant no. N00014-01-1-0890, as part of the MURI program.

References

1. Gavrila, D.M.: The visual analysis of human movement: A survey. Computer Vision and Image Understanding: CVIU **73** (1999) 82–98
2. O'Rourke, J., Badler, N.: Model-based image analysis of human motion using constraint propagation. IEEE Trans. PAMI **2** (1980) 522–536



(a) (b)

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

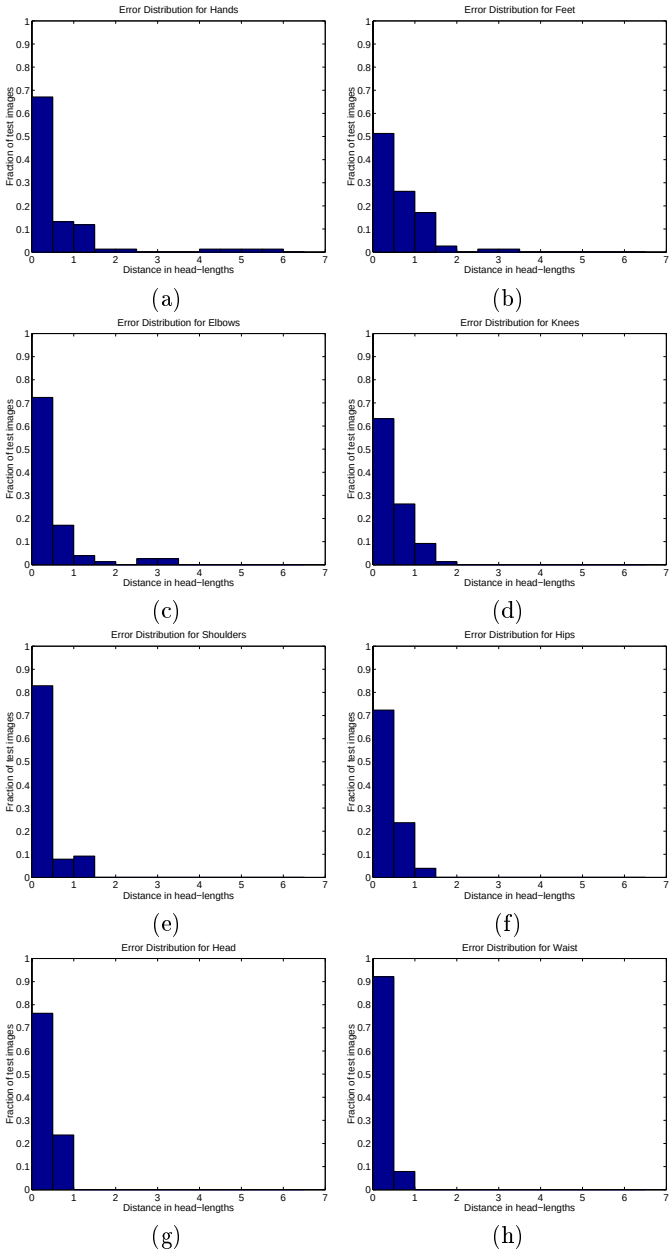
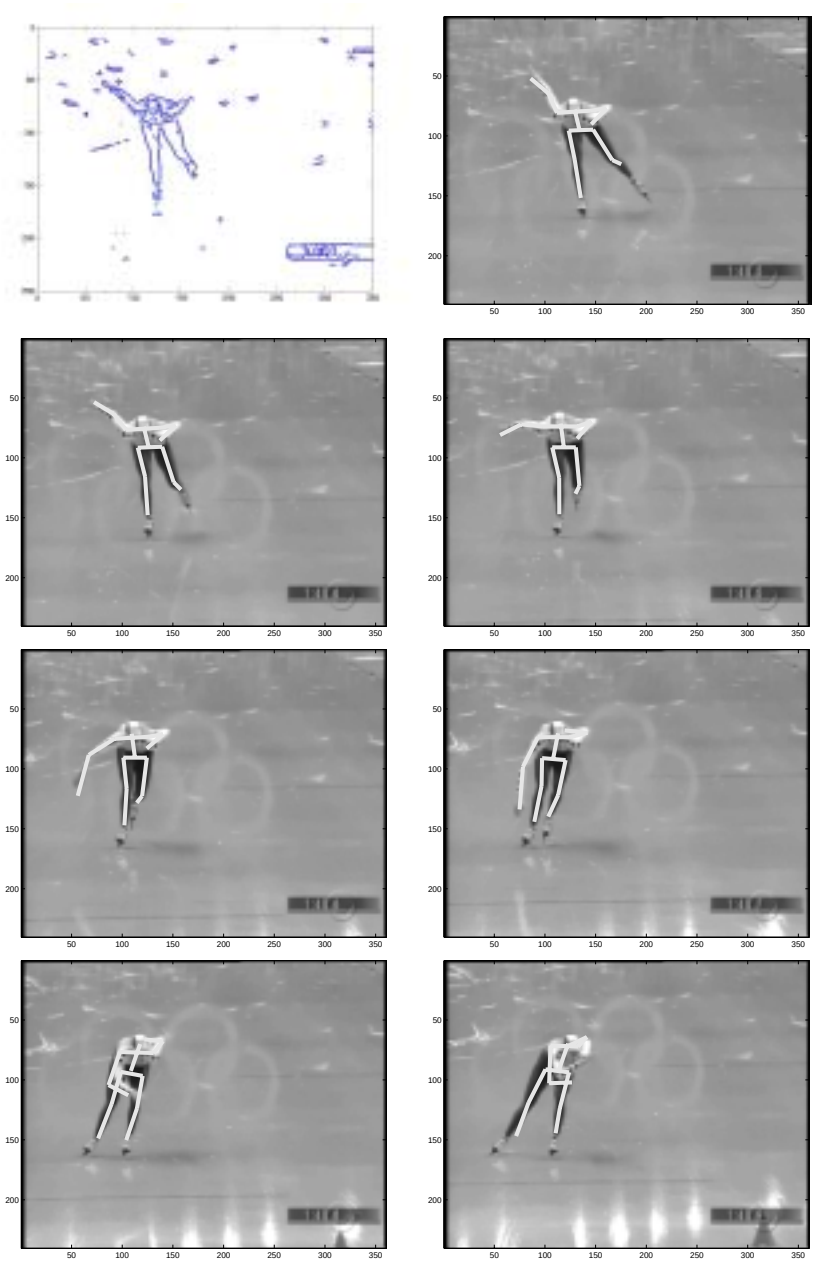


Fig. 6. Distributions of error in 2D location of keypoints. (a) Hands, (b) Feet, (c) Elbows, (d) Knees, (e) Shoulders, (f) Hips, (g) Head, (h) Waist. Error (X-axis) is measured in terms of head-lengths (average length of head in 2D image). Y-axis shows fraction of keypoints in each bin. The average image size is 380 by 205 pixels. Large errors in positions are due to ambiguities regarding left and right limbs.



(a) (b)

Fig. 7. Frames of speed skater sequence. The top left image shows an example of sample points extracted using edge detection. The rest of the images show the estimated body configuration of every 3^{rd} frame of the sequence.

3. Hogg, D.: Model-based vision: A program to see a walking person. Image and Vision Computing **1** (1983) 5–20
4. Yamamoto, M., Koshikawa, K.: Human motion analysis based on a robot arm model. CVPR (1991) 664–665
5. Rehg, J., Kanade, T.: Visual tracking of high dof articulated structures: An application to human hand tracking. Proc. of 3rd ECCV **II** (1994) 35–46
6. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn. (1998) 8–15
7. Kakadiaris, I., Metaxas, D.: Model-based estimation of 3d human motion. IEEE Trans. PAMI **22** (2000) 1453–1459
8. Gavrila, D., Davis, L.: 3d model-based tracking of humans in action: A multi-view approach. IEEE Computer Society CVPR (1996) 73–80
9. Rohr, K.: Incremental recognition of pedestrians from image sequences. In: CVPR93. (1993) 8–13
10. Baumberg, A., Hogg, D.: Learning flexible models from image sequences. Lecture Notes in Computer Science **800** (1994) 299–308
11. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Trans. PAMI **19** (1997) 780–785
12. Morris, D., Rehg, J.: Singularity analysis for articulated object tracking. In: Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn. (1998) 289–296
13. Ioffe, S., Forsyth, D.: Human tracking with mixtures of trees. In: Proc. 8th Int. Conf. Computer Vision. Volume 1. (2001) 690–695
14. Song, Y., Goncalves, L., Perona, P.: Monocular perception of biological motion - clutter and partial occlusion. In: Proc. 6th Europ. Conf. Comput. Vision. (2000)
15. Toyama, K., Blake, A.: Probabilistic exemplar-based tracking in a metric space. In: Proc. 8th Int. Conf. Computer Vision. Volume 2. (2001) 50–57
16. Brand, M.: Shadow puppetry. Proc. 7th Int. Conf. Computer Vision (1999) 1237–1244
17. Carlsson, S.: Order structure, correspondence and shape based categories. In: Shape Contour and Grouping in Computer Vision. Springer LNCS 1681 (1999) 58–71
18. Carlsson, S., Sullivan, J.: Action recognition by shape matching to key frames. Workshop on Models versus Exemplars in Computer Vision at CVPR (2001)
19. Mori, G., Malik, J.: Estimating human body configurations using shape context matching. Workshop on Models versus Exemplars in Computer Vision at CVPR (2001)
20. Belongie, S., Malik, J., Puzicha, J.: Matching shapes. In: Eighth IEEE International Conference on Computer Vision. Volume 1., Vancouver, Canada (2001) 454–461
21. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. PAMI (2002) (in press).
22. Taylor, C.J.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. CVIU **80** (2000) 349–363
23. Canny, J.: A computational approach to edge detection. IEEE Trans. PAMI **8** (1986) 679–698
24. Papadimitriou, C., Stieglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Prentice Hall (1982)
25. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing **38** (1987) 325–340
26. Amit, Y., Kong, A.: Graphical templates for model registration. IEEE Trans. PAMI (1996)
27. Eguchi, H.: Moving Pose 1223. Bijutsu Shuppan-sha (1995)