

Human Limb Delineation and Joint Position Recovery Using Localized Boundary Models

Chris McIntosh, Ghassan Hamarneh
Medical Image Analysis Lab
School of Computing Science
Simon Fraser University, Burnaby, BC
{cmcintos,hamarneh}@cs.sfu.ca

Greg Mori
Vision and Media Lab
School of Computing Science
Simon Fraser University, Burnaby, BC
mori@cs.sfu.ca

Abstract

We outline the development of a self-initializing kinematic tracker that automatically discovers its part appearance models from a video sequence. Through its unique combination of an existing global joint estimation technique and a robust physical deformation based local search method, the tracker is demonstrated as a novel approach to recovering 2D human joint locations and limb outlines from video sequences. Appearance models are discovered and employed through a novel use of the deformable organisms framework which we have extended to the temporal domain. Quantitative and qualitative results for a set of five test videos are provided. The results demonstrate an overall improvement in tracking performance and that the method is relatively insensitive to initialization, an important consideration in gradient descent-style search algorithms.

1 Introduction

The problem we address is that of recovering human joint positions from video sequences. Once extracted, body configurations and joint positions can be used for a variety of applications including motion capture, activity recognition, and automatic movement/kinetic analysis of professional athletes. The interested reader is referred to [2, 12] for a detailed review of techniques for solving the problem, as well as the applications of its solution.

Our solution is similar to those based on rectangular part tree models or cardboard people [6]. However, we employ a type of active contour (deformable organisms) to latch to the limb boundaries, thus obtaining more accurate appearance models than one would with a rectangular approximation, and enabling the use of a limb shape prior [1].

Deformable organisms, unlike traditional active contour models [7, 17] do not rely on the minimization of

a single energy functional, but rather minimize a variety of energy functionals at various image locations and model-fitting stages, based on dynamically-chosen and anatomically-driven behavioral methods. Previously, however, deformable organisms have been restricted to static images [4]. We extend deformable organisms to dynamic images, and automatically optimize each deformable organism for the detection of a particular body part influenced by both its own fit and that of its neighbors. The result is a robust technique for recovering joint positions and delineating limb boundaries. The solution takes form as a set of deformable organisms, with ‘brains’ that control, receive, and filter image-derived sensory data and execute behavioral routines in order to drive deformations designed to conform to local body parts in the video sequence.

In addition, we use single frame detections to bootstrap models of person appearance in video sequences. This strategy is inspired by the work of Ramanan and Forsyth, [15] who find typical poses to build rectangular limb appearance models. By using deformable boundaries, in contrast to fixed rectangular models, we are able to model shape, boundary, and intensity information as opposed to intensity information only. Sigal et al. [18] also use a related approach based on bottom-up part detectors. However, their method is not applied to monocular image data. Similarly, Zhang et al. [22] localize a piecewise-linear limb boundary model using Sequential Monte Carlo to perform inference. Though they also perform a form of local search, they initialize via part-detectors and guide the search in a bottom-up fashion. In contrast, our method uses a whole-body (top-down using exemplars) method to initialize a local search. Furthermore, the local search procedure itself, deformable organisms, uses a top-down, knowledge driven model-fitting strategy to complement traditionally bottom-up, data-driven deformable models.

The main contributions of this paper are: (i) the creation of a self-initializing kinematic tracker which automatically

discovers part appearance models from a video sequence; (ii) the unique use of a global (whole body) optimization approach to the initialization problems that typically plague deformable models; and (iii) the extension of deformable organisms to dynamic data. The strengths of our method are: (i) its relative insensitivity to initialization; (ii) its use of deformable models for the development of a shape model of each limb; and (iii) a strong approximation of its internal intensity and texture distributions.

In this paper we provide a brief review of deformable organisms (Section 1.1), with an overview of their application to joint recovery in 1.2. Section 2 provides details of the method, with details of how pre-process the images, automatically initialize obtain the limb models, and how we use the limb models to perform tracking. Qualitative and quantitative results are provided in section 3. We conclude in section 4.

1.1 Deformable Organisms

Artificial Life (AL) modeling concepts have been successfully applied to the problem of producing life like computer animations of biological organisms [19]. In 1994 ‘Artificial Fishes’ were introduced as a prime example, under which each fish is controlled by a brain that utilizes a cognitive decision process. It is capable of making these decisions based on stimulus obtained from its surrounding environment, and carries out the decision through learned behavioral methods.

In 2001 Hamarneh *et al* applied AL modeling concepts to the creation of geometrically based deformable organisms (Figure 1) for medical image analysis [4]. They created ‘self-aware’ organisms capable of searching for anatomical structures within 2D images by following a schedule of geometrically constrained deformations. In later work they replaced the geometrical deformation layer with that of a physically-based deformation layer [5], and extended the approach to 3D [11, 10]. These deformable organisms were capable of rejecting local minima through the exploration of a localized parameter space, and the use of anatomical knowledge.

This approach is carried out through a layered design, where higher layers elicit control over the lower layers. Progressing from bottom to top: The geometrical layer is responsible for the shape of the organism; the physical layer incorporates both internal and external physical forces engineered to manipulate its geometry; the behavioral layer contains the organism’s many possible actions, and issues the actual commands to the muscle actuator systems; and the cognitive layer is responsible for making decisions based upon sensory input, current goals, and a pre-stored schedule which may involve interaction with other organisms (Figure 1-b). For example, the deformable organism assigned

to segmenting the bicep, elicits control over the one segmenting the forearm, thereby maintaining connectivity and reducing the parameter search space.

We adopt deformable organisms because they provide a modular framework for modeling the geometrical construction of shape models, implementing the model deformations, scheduling behaviors, obtaining local image statistics, and making decisions (e.g .what/when to deform, what/when to sense). Consequently, they enable the direct implementation of high level intuitive ideas not easily provided by classical energy-minimizing deformable models or region growers.

1.2 Deformable Organisms for Joint Position Recovery

In this paper we demonstrate physically-based deformable organisms for the recovery of human joint positions and delineation of limb boundaries in video. What follows is an overview of how deformable organisms are applied to the joint position recovery problem.

Upon recovery of the joint position estimations (section 1.3), a part model recovery organism is placed in each frame between successive pairs of joints (bicep, forearm, thigh, calve) and at each end point (hand, foot). Each of these organisms optimize their respective search schedules (Section 2.2) in order to obtain limb appearance and shape models which can be used to drive the search in other frames. We refer to the combined appearance and shape model as a limb model (LM). Hence, each person yields 12 LMs (Figure 1-a). Given the LMs, new part finding organisms are placed in each frame with the goal of locating the part described by their respective LM (Section 2.3). Each of these organisms in turn optimizes its schedule, yielding a refined estimate of the joint positions in each frame. During this phase the search is hierarchical in nature, whereunder the limb organisms are anatomically (kinematically) connected, and, as such, moving one moves those connected to it.

In summary, we use the global view of shape contexts to initialize a natural and hierarchical person model that uses active contours to build localized shape and appearance models of each limb (Algorithm 1). Thus addressing the initialization problems that typically plague deformable models, and gaining the benefit of limb-specific energy functionals instead of a ‘one size fits all’ approach.

1.3 Initializing the Tracker

The deformable organisms are designed to carry out localized searches and consequently require another method (known as a *sensor* in deformable organism terminology) to process the image for a likely human position and configuration. A method with a global view is chosen to emphasize

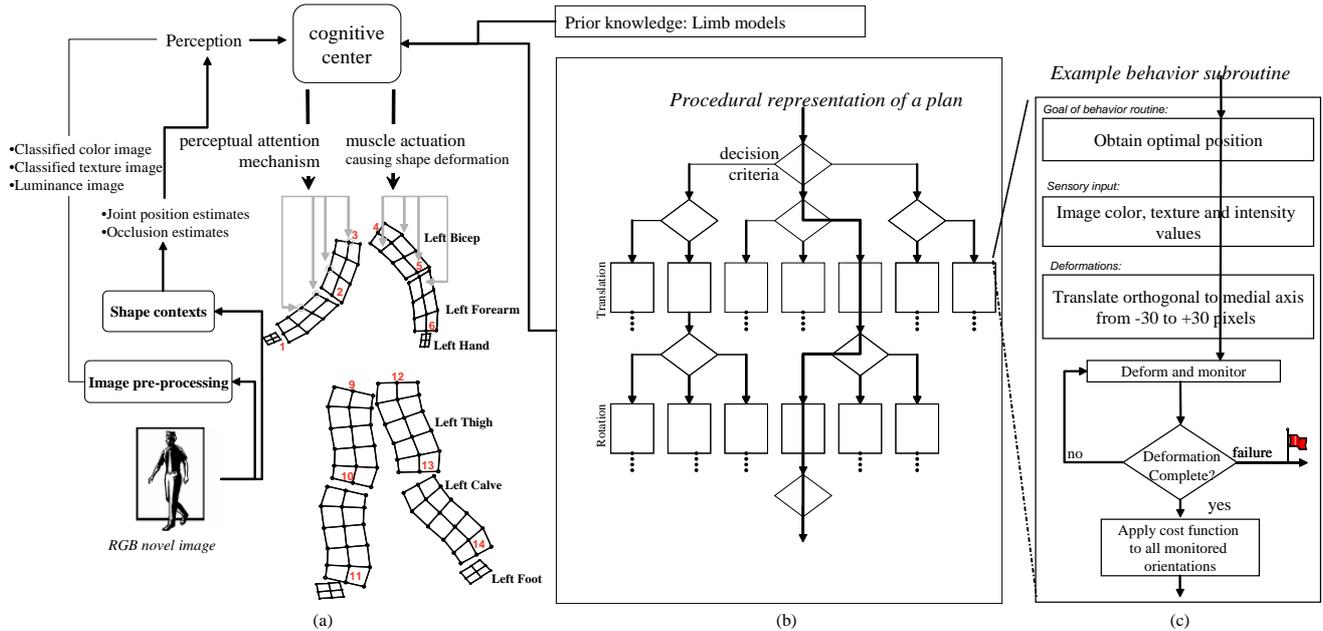


Figure 1. An example deformable organism for people tracking. (a) A pre-processed novel image, the shape context position estimates, a set of limb models, and a schedule are input into the cognitive center which then controls the shape deformations, which in turn search the local space. Muscle actuations elicit deformations, while sensors detect image features to relay to the cognitive center for future decisions. (b) Schedule used for refining joint position estimates (Adapted from [5]). Along each row are behaviors with different parameters. In our application LM finding organisms (Section 2.2) are designed to take a different path than limb finding organisms (Section 2.3). An example schedule operation is shown in (c), which performs a local search by translating the model and estimating the cost of each location.

Algorithm 1 Human Limb Delineation and Joint Position Recovery Using Localized Boundary Models

Require: A sequence of $|F|$ frames.

Ensure: Limb delineation and joint position recovery for all frames.

- 1: **for** $f = 1$ to $|F|$ **do**
- 2: Compute joint position estimates $E(f)$ using shape contexts (See reference [13])
- 3: **end for**
- 4: Obtain the LMs (see algorithm 2)
- 5: Use the LMs to refine the estimates (see algorithm 3)

finding a structure that is topologically human in the global body sense. That is to say, one with a likely configuration of the joints given its shape. The pose estimation technique of Mori and Malik [13], using exemplar-based matching with shape contexts, meets these requirements.

Mori and Malik estimate human body configurations through shape context-based matching of test images into a database of stored views. Given a single 2D test image, their method extracts both internal and external contours of the body shape via canny edge detection and uses shape contexts in conjunction with a kinematic chain-based deformation model to compute similarity between the test input and

the model database. In order to avoid intractability in the model database a part-based approach is applied wherein deformation scores are obtained for each limb on the model instead of only the model itself. A weighted sum of both Euclidean distance and shape context scores yields the final matching score for each limb. Finally, the matched model is used to infer joint locations on the input image. In this paper we have used their method to initialize our AL models.

2 Methods

In this section we provide details of the image preprocessing steps required to construct the LMs, the deformable organisms designed to obtain those models, and the deformable organisms designed to locate the respective limbs in new frames.

2.1 Sensors: Color and Texture Classified Images

In deformable organism frameworks, image features are captured and processed through sensors [4]. Throughout their execution cycles our deformable organisms require

Algorithm 2 Obtaining the LMs

Require: A sequence of $|F|$ frames.**Ensure:** A set of LMs, one for each limb visible in F .

```
1: Learn color and texture clusters (Section 2.1)
2: for  $f = 1$  to  $|F|$  do
3:   Compute gradients (Section 2.2)
4:   for Each limb (bicep, forearm, hand, thigh, calve, foot) do
5:     Initialize a rectangular, medial-axis based deformable organism at the locations provided by  $E(f)$  (Algorithm 1).
6:     Search: deform to image gradients while monitoring image statistics within the organism’s boundary
7:     Build a LM at the obtained position (color histogram, texture histogram, intensity variance, thickness, length)
8:   end for
9: end for
10: for each limb do
11:   Compute inter-organism texture histogram distance
12:   Sort the organisms by their average distance and average the top 5 to obtain the final LM
13: end for
```

sensory input in the form of color and texture classified images in order to construct their respective histogram models.

We first convert images to L^*a^*b space. Then, to avoid the sparsity and computational complexity related problems often associated with 3D color histograms, we use k-means to perform dimensionality reduction.

Texture classified images are obtained through a similar method [9]. RGB images are first converted to gray-scale and then convolved with the MR8 filter bank—a set of 38 L_1 normalized filters consisting of 18 odd-symmetric filters, 18 even-symmetric filters, and 2 radially-symmetric filters [20]. Then α filter response vectors in \mathbb{R}^{38} are passed into k-means in order to learn a set of texture descriptors, or ‘*textons*’, where α is the number of sampled filter responses per image (8500 in all results for this paper).

We equip each deformable organism with sensors designed to monitor the image intensity along with classified color and texture measurements located within their respective boundaries. Through these sensors the organism will estimate the goodness of its fit at each stage of the search phase. At the end of each behavior (Figure 1-c) it will then decide the correct location from the set of examined locations by maximizing equation 4, before proceeding to the next phase (block) of the search schedule.

2.2 Obtaining the Limb Models

In order to carry out the recovery process a part model, to be used over the entire video sequence, must be automatically obtained for each part: biceps, forearms, hands, thighs, calves, and feet. A set of such models are shown overlaid in figure 2. Each limb has a medial axis with flexible thickness springs connecting to the boundary nodes, which are in-turn connected by stiffer boundary springs designed to enforce a level of smoothness (Figure 1-a). During this phase the organisms have no prior-knowledge, and, consequently, trust heavily in the shape context by using a restricted search space just large enough to correct for small



Figure 2. Automatically obtained LMs for one person in the test set. Each image shows a particular LM overlaid with red nodes and black springs.

errors in the joint position estimations. The schedule for these organisms consists of two main steps, the first is a restricted local search using small translation (± 15 pixels in a direction orthogonal to the medial axis) and rotation deformations ($\pm 15^\circ$ around the medial axis) as defined in [5] which minimize variance in luminance, and the second is a gradient based deformation designed to conform to the limb (Algorithm 2).

In order to carry out the physics-based deformations [5], an image force must be chosen and applied to the model. Previously, deformable organisms have been restricted to scalar images. However, in this work we extend them to take advantage of color information as follows:

1. RGB input images are first converted to L^*a^*b color space.
2. A color gradient is computed as in [8]. Specifically,

Algorithm 3 Using the LMs to refine joint position estimates and segment boundaries

Require: A sequence of $|F|$ frames, and a set of LMs LM .

Ensure: Refined joint positions and limb boundaries.

```
1: for  $f = 1$  to  $|F|$  do
2:   Obtain color texture classified images
3:   Compute gradient image (Section 2.2)
4:   Run shape contexts to obtain joint position estimates (See reference [13])
5:   for  $l = 1$  to  $|LM|$  do
6:     Create a new organism with its thickness, length and appearance model described by  $LM(l)$ 
7:     if  $f = 1$  then
8:       Use shape context result  $E(f)$  (Algorithm 1)
9:     else
10:      Use average of shape context result  $E(f)$  (Algorithm 1) and refined estimate from  $f - 1$ 
11:    end if
12:    Search: deform to image gradients while monitoring image statistics within the organism's boundary. (Algorithm 4).
13:  end for
14: end for
```

Algorithm 4 The local search algorithm

Require: A frame f , and LM l .

Ensure: Refined joint positions and limb boundaries.

```
1: While deforming to image gradient
2: Hierarchically translate horizontally, translate vertically, and rotate (Figure 1-c)
3: for top 5% of results rotate do
4:   Rotate lowest two limbs only (e.g. forearm, hand)
5: end for
6: Decide best location
7: Perform final hierarchical rotation
8: for top 5% of results do
9:   Rotate lowest two limbs only (e.g. forearm, hand)
10: end for
11: Perform final hierarchical translation
12: for top 5% of results rotate do
13:   Rotate lowest two limbs only (e.g. forearm, hand)
14: end for
15: Decide final location and store for use in next frame
```

each color channel is differentiated with respect to x and y , then stacked in a matrix D

$$D = \begin{bmatrix} \partial L / \partial x & \partial L / \partial y \\ \partial a / \partial x & \partial a / \partial y \\ \partial b / \partial x & \partial b / \partial y \end{bmatrix} \quad D^T D = \begin{bmatrix} p & t \\ t & q \end{bmatrix}$$
$$\lambda = \frac{1}{2} \left(p + q + \sqrt{(p + q)^2 - 4(pq - t^2)} \right) \quad (1)$$

The magnitude of the gradient M is then taken as λ , the largest eigenvalue of DD^T .

3. We then carry out a type of gradient diffusion process to obtain a gradient vector field that directly approximates the gradient in high magnitude areas and smoothly declines in lower valued regions using the method proposed in [21].

Once each limb is roughly segmented across the set of frames, a decision must be made as to which LM for each

part is optimal. It is important to note here that these LMs will then be used to segment the entire frame sequence (section 2.3). The organism's decision function minimizes the intra-class distance for each LM type (calve, forearm, bicep, etc.). First it calculates the texture histograms for each organism in the class. Then for each limb the intra-class distances amongst all the organisms of that limb type is calculated using earth movers distance (EMD). Finally, the top five models' thickness, length, histogram and intensity variance are averaged to remove minor fluctuations. We then build the LMs by calculating probability density functions (PDF) (normalized histograms) for both classified intensity and texture. As we will eventually calculate a log probability (Equation 4), we add $\epsilon = 0.01$ to each PDF to ensure that no zero entries are present. The number of bins in each histogram is set equal to the total number of classes, and hence hard binning is used.

2.3 Joint Position Refinement: Using the Limb Models

With a LM for each part obtained, a new set of deformable organisms is employed to find their respective parts in each new frame. Here the search is conducted using an average between the last frames recovered joint position and the current frame's shape contexts as input. Since these organisms possess much greater certainties of their targets then their model-finding counterparts (Section 2.2) larger translation (± 20 pixels in a direction orthogonal to the medial axis) and rotation deformations ($\pm 45^\circ$ around the medial axis) are performed to increase the search space. The amount of each deformation is chosen to reflect the relatively small variation in joint position between frames of walking people, and the range of error typically incurred by the shape contexts. The search phase is hierarchical [14], starting with larger parts (biceps, thighs) and working its way down to smaller ones (feet, hands).

During this process the organisms *monitor* distances between their obtained color, texture histograms and the model’s histograms. At each step the configuration with the least distance is chosen. We experimented with both EMD and χ^2 distance, before settling on a probabilistic approach (motivated by results shown in [16]):

$$\begin{aligned} \log(P(L|C)) &= \log\left(\frac{P(C|L)P(L)}{P(C)}\right) \propto \log\left(\prod_{i=1}^N P(C_i|L)\right) \\ &\propto \sum_{i=1}^N \log P(C_i|L) \end{aligned} \quad (2)$$

where $\log P(L|C)$ is the color cost term with L for a particular limb and C_i a particular color learned using k-means. $\log P(L|T)$ would be similarly defined, with T_i for a particular texton. i ranges from one to N as it sums over all pixels contained within the boundary of the organism being evaluated. Here we have taken $P(C)$ as a normalizing constant, assumed $P(L)$ to be a uniform distribution, and assumed individual colour/texture distributions to be independent given limb position. Essentially at each point during the search phase we measure the amount of evidence demonstrated by the pixels within the deformable organisms’s boundary. Additional cost terms based on intensity variance, and expected start/end point limb position and shape are given by

$$\begin{aligned} v &= \left| E_v - \sum_{i=1}^N (I(x_i, y_i) - \mu)^2 \right| \\ s &= \frac{1}{M} \sum_{j=1}^M |Et_j - Ot_j| \\ p_{start} &= |El_{start} - O_{start}| \\ p_{end} &= |El_{end} - O_{end}| \end{aligned} \quad (3)$$

where E_v is the expected intensity variance, I the luminance image of the current frame and μ its mean within the boundary of enclosed by the shape model. Et_j and Ot_j are the expected and measured thickness at medial nodes $1 \dots M$. El_{start} , El_{end} , O_{start} , and O_{end} are the expected/obtained position of the start/end medial node (e.g. top/bottom of bicep). In all cases the expected values are those represented by the LMs, except for El_{start} , and El_{end} which are equal to the initialized location of the model. Finally, the fit term we seek to maximize is a sum of all provided terms.

$$\begin{aligned} fit &= \alpha \log(P(L|C)) + \beta \log(P(L|T)) + \gamma v + \lambda s \\ &+ \chi p_{start} + \omega p_{end} \end{aligned} \quad (4)$$

where $\alpha, \beta, \gamma, \lambda, \chi,$ and ω are weighting constants whose sum must equal one. For all results in this paper their values are 0.14, 0.28, 0.105, 0.175, 0.09, and 0.21, reflecting what we found to be their relative levels of importance/reliability

by qualitatively examining the feature images. It is important to note that the final fitness is also hierarchical. For example, the fitness for a bicep is actually its own fitness plus the fitness of the best forearm it yields, plus the fitness of the best hand implied by that forearm (Algorithm 3).

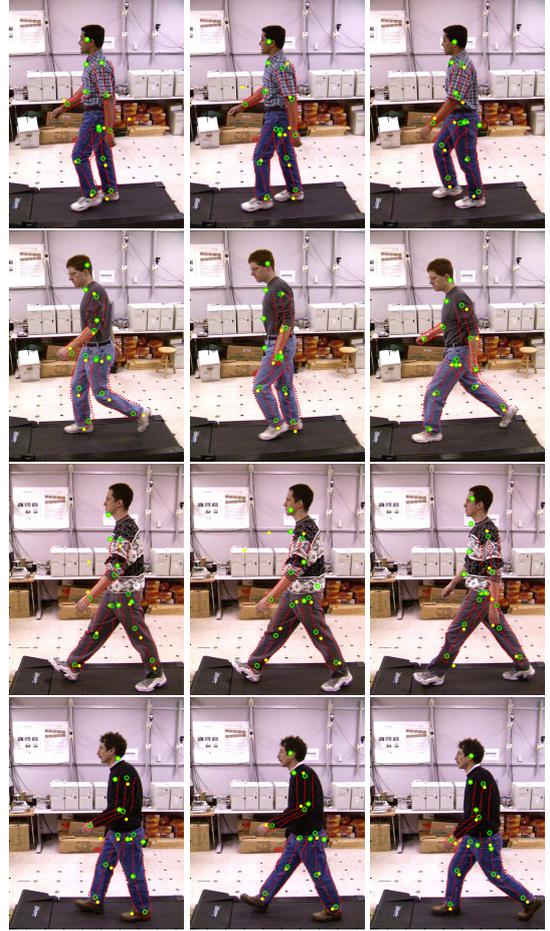


Figure 3. Results of our method on frames from three different videos. Boundary delineation is shown in red, with green circles indicating recovered joint positions, and yellow points for shape context based initialization positions. The first two images on the third row show a significant improvement over the shape contexts, while the 4th row demonstrates the methods ability to handle cases with an exceptionally unclear boundary between the arm and the torso.

3 Results

We provide quantitative and qualitative analysis of our approach through comparison made to the original shape context method on 5 video sequences each consisting of 30 frames of the CMU MoBo dataset [3]. We set $k = 80$ for

	1	2	4	5	6	9	10	11	12	13	14
04011	25.4± 22	17.4± 14	10.4± 6	10.8± 6	10.8± 5	15.1± 9	12.4± 9	14.4± 12	9.2± 7	10.0± 7	18.1± 14
04011	29.2± 21	17.0± 13	14.0± 7	17.0± 10	24.9± 16	16.8± 7	18.3± 10	18.8± 14	16.3± 8	13.9± 11	13.7± 10
04013	29.1± 27	18.7± 17	14.3± 9	20.2± 11	29.9± 10	20.9± 10	10.9± 8	16.3± 23	15.1± 6	8.1± 4	12.1± 8
04013	33.5± 28	20.0± 17	12.3± 13	19.0± 16	27.3± 21	14.8± 8	14.1± 8	19.2± 22	14.6± 6	11.7± 6	14.8± 21
04068	27.5± 25	19.1± 15	11.9± 5	17.9± 6	16.0± 9	14.5± 8	9.6± 8	11.7± 8	14.7± 7	7.5± 4	9.4± 5
04068	27.7± 24	19.8± 14	13.1± 9	24.1± 12	27.8± 37	15.1± 9	21.8± 29	24.5± 46	15.4± 10	20.1± 31	25.4± 51
04070	44.5± 34	25.7± 14	18.8± 10	15.5± 11	13.7± 13	20.8± 11	17.8± 9	23.5± 10	27.4± 11	33.0± 12	45.3± 13
04070	44.7± 34	23.6± 16	17.6± 11	30.7± 30	42.6± 41	24.4± 14	33.7± 30	45.6± 49	24.8± 12	32.9± 29	47.0± 52
04071	50.9± 39	14.0± 9	13.3± 6	20.1± 17	15.1± 8	23.5± 14	13.1± 6	14.7± 8	22.6± 11	12.1± 7	10.5± 6
04071	29.7± 31	13.4± 12	17.7± 9	21.0± 13	25.8± 24	16.1± 10	21.6± 23	22.4± 37	14.3± 10	18.6± 18	23.8± 39

Table 1. MoBo database sequence numbers versus joint position numbers. Each cell shows error obtained by our proposed method (top) over the shape contexts (bottom) error. Results are averaged across frames, and standard deviations are provided. Error is measured as the Euclidean distance in pixels between the estimated joint position and its correct location. Here joint positions are ordered (from 1 to 14) as: right hand, elbow, and shoulder. Left shoulder, elbow, and hand. Head, and waist. Right hip, knee and foot. Left hip, knee and foot. As illustrated in Figure 1

color and $k = 100$ for texture classification using k-means. We observed that 80 colors were sufficient to capture the amount of color variation present in our images, and estimated 10 textons per texture and at most 10 textures per person (skin, shirt, pants, shoes, hair, etc.). As shown in table 1 our method obtains significantly improved results over the initializations provided by shape contexts when measured against labeled ground truth data. Though it has a harder time with some of the right-side limbs. Specifically, the right forearm is troublesome since it is only visible for a few frames which gives the organism less time to build a model and fewer opportunities to improve on the shape context results. The right leg, however, despite heavy occlusion tracks remarkably well (see supplementary videos), as does the left bicep despite the lack of any strong border when it occludes the body (Bottom row figure 3). Qualitative results for a few frames of four of the segmented videos are shown in figure 3. Our methods robustness to initialization is demonstrated in figure 3’s third row. The 4th row demonstrates the methods ability to handle cases with an exceptionally unclear boundary between the arm and the torso. The algorithm takes approximately one to two minutes per frame in Matlab on a 3.0Ghz machine.

4 Conclusions

We have developed a self-initializing kinematic tracker which automatically discovers part appearance models from a video sequence. Application to the recovery of human joint positions and boundaries in video sequences of walking people was presented, through the unique combination of globally optimal (whole body) shape contexts with locally optimal (per limb) deformable organisms. With the

addition of a scale deformation the method could be readily extended to handle activities where the size of limbs changes over time. Our use of deformable models enables the development of a shape model of each limb, and a strong approximation of its internal intensity distributions. We have extended deformable organisms to the temporal domain, thereby creating dynamic deformable organisms. Finally we have shown our method is able to refine the initial joint position estimates obtained with the original shape context technique, and is relatively insensitive to initialization, an important consideration in gradient descent-style search algorithms.

References

- [1] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001.
- [2] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.
- [3] R. Gross and J. Shi. The cmu motion of body (mobo) database. *Tech. Report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University*, 2001.
- [4] G. Hamarneh, T. McInerney, and D. Terzopoulos. Deformable organisms for automatic medical image analysis. In *MICCAI*, pages 66–76, 2001.
- [5] G. Hamarneh and C. McIntosh. Physics-based deformable organisms for medical image analysis. *SPIE Medical Imaging*, 2005.
- [6] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. *FG ’96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, page 38, 1996.

- [7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.
- [8] H.-C. Lee and D. R. Cok. Detecting boundaries in a vector field. *IEEE Transactions on Signal Processing*, 39:1181–1194, 1991.
- [9] T. K. Leung and J. Malik. Contour continuity in region based image segmentation. In *ECCV (1)*, pages 544–559, 1998.
- [10] C. McIntosh and G. Hamarneh. Spinal crawlers: Deformable organisms for spinal cord segmentation and analysis. pages 808–815, 2006.
- [11] C. McIntosh and G. Hamarnerh. Vessel crawlers: 3d physically-based deformable organisms for vasulature segmentation and analysis. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1084 – 1091, 2006.
- [12] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.
- [13] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
- [14] S. M. Pizer, P. T. Fletcher, Y. Fridman, D. S. Fritsch, A. G. Gash, J. M. Glotzer, S. Joshi, A. Thall, and et al. Deformable m-reps for 3d medical image segmentation. *International Journal of Computer Vision*, 55:85–106, 2003.
- [15] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. *Computer Vision and Pattern Recognition*, 2005.
- [16] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.
- [17] J. A. Sethian. Level set methods; evolving interfaces in geometry, fluid mechanics. In *Computer Vision and Material Sciences*. Cambridge University Press, 1996.
- [18] L. Sigal, B. Sidharth, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. *CVPR*, 2004.
- [19] D. Terzopoulos, X. Tu, and R. Grzeszczuk. Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4):327–351, 1994.
- [20] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*, 2005.
- [21] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.
- [22] J. Zhang, J. Luo, R. Collins, and Y. Liu. Body localization in still images using hierarchical models and hybrid search. *Computer Vision and Pattern Recognition Conference*, 2006.