Human Motion from Active Contours

Jane Wilhelms, Allen Van Gelder, Leon Atkinson-Derman, Alison Luo
Computer Science Dept., University of California, Santa Cruz, CA 95064
wilhelms,avg,ljderman,alison@cse.ucsc.edu

Abstract

We describe an approach for extracting three-dimensional articulated motion from unrestricted monocular video sequences. We combine feature extraction methods based on active contours with interactive adjustment. An articulated model is interactively aligned with the image in selected anchor frames. Active contour points are anchored to model segments in these frames. Occluded points are detected using object geometry and do not participate in edge tracking. Model joints are automatically adjusted in other frames to align with active contour points. The combination of interactive and automatic adjustment allows extraction of arbitrarily complex movements.

1 Introduction

Although the last twenty years have seen many interesting computer graphics techniques for high-level, constrained control of articulated body motion, it is still not possible to reliably generate truly realistic human and animal motion. Trained animators can do this using key-framing, but they are few in number and expensive. A successful alternative is *motion capture*, where the subject motion is measured and duplicated generally in a studio setting. Motion capture generally occurs in a studio setting where either multiple cameras or body-fixed sensors are used.

However, videos record human and animal motion of greater variety than is encountered in studios. Children playing in the surf, deer leaping, and cats hunting are examples of motion not amenable to extraction by multiple cameras or wearing special apparatus. Occlusion, rapid movement, out-of-plane rotations, and ambiguities hamper automatic approaches. We are exploring methods that combine computer vision techniques with interactive manipulation to extract arbitrary motion from video.

Our long-term goal is to develop a motion library that encodes movement so that it can be reused in different environments and with different creatures. Motions that we extract from video are approximate, but can provide a starting point for such encodings.

2 Background

The computer vision methods most applicable to our work are model-based techniques for extracting articulated body motion from monocular image sequences. The review of Aggarwal *et.al* is an excellent summary of applicable research [1]. Low-level image processing techniques may be used to prepare the images. While features take various forms, *active contours*, or *snakes* [5, 11] are the basis for our work. The model may be a stick-figure or be volumetric [4, 6, 15]. It may be interactively created by the user [2], or generated automatically during extraction [10]. Tracking is achieved by minimizing the error between the model and image [7, 16]. Matching a 2D model with the image may be followed by a separate 3D reconstruction [13], or the two may be combined, with joint and image constraints treated together [8]. Results using optical flow and probabilistic methods are encouraging [9, 14].

3 Method

We use *active contours* [11] as a feature representation that can both automatically react to the image and be easily manipulated by the user. Image processing techniques such as blurring, edge detection, and intensity mapping can be used initially to clarify image content. The user then designs active contours in selected frames; those in intervening frames can be created automatically or with user aid. The 3D model is interactively positioned in selected anchor frames and contour vertices are anchored to the model there. Kinematic adjustment automatically brings the model into alignment with contours in other frames.

3.1 Definitions

We start with a three-dimensional articulated model created interactively using our *zoo* creature modeling and animation software [3, 12, 17]. Articulations are *joints*, and body regions between joints are *segments*. Our standard human model has 70 individual segments, though not all need be involved in the extraction. Segments can be grouped into super-segments, [12], allowing many contiguous segments to be controlled by a few parameters. Each segment is defined in a local coordinate system relative to its parent segment, with the root segment defined in the world. The root segment has six degrees of freedom (rotations and translations) but others only rotate.

Images are digitized from video and texture-mapped onto a movable plane. The world Z-axis is perpendicular to the image plane.

We call a sequence of active contour points that act together a *fauna snake* or *fsnake* to indicate its existence both within the image and as part of the model. Many fsnakes may act on one model. They generally exist in multiple frames; there is a one-to-one relationship between points in different frames.

$\mathbf{P_{i,j}}$ refers to the i-th contour point in the j-th frame. The frame subscript is omitted when it is clear by context. Vectors are given in boldface and scalars in italics. Some calculations are done with three-dimensional vectors, while others use a projection of the vectors onto the image plane. The superscript **pw** (e.g., $\mathbf{P^{pw}_{i,j}}$) indicates a world-space vector that is projected so that its Z-coordinate is zero. Vector components are written as $[V_x, V_y]$.

Figure 1 illustrates the process we describe by showing a two-segment model being matched to an elliptical image figure. In the top diagram, the user positioned the model segments (represented by their labeled coordinate axes) on the image figure and created a three-point fsnake nearby. After automatic anchoring, each contour point of the fsnake is attached to a nearby segment longitudinal axis and this position is stored as an anchor $\mathbf{A}$ in the local segment frame. In an anchor frame, each contour point $\mathbf{P_{i,j}} = \mathbf{A_{i,j}}$.

In the middle diagram showing frame 1, the image figure has moved and the contour points have tracked it, so that they are no longer aligned with their respective anchors. In the lowest diagram, still frame 1, the model has been automatically transformed to align the virtual anchors and contour points. Segment a (the root) has translated and rotated, and segment b has rotated. The following sections describe how this process occurs.

3.2 Creating the Fauna Snakes

The user initially positions the fsnakes in selected frames by picking points on the image. Fsnakes in intervening frames can be created by copying from another frame, by interpolating between fsnakes already created, or by extrapolating positions in two previous frames as an estimate of velocity. An fsnake may be treated as rigid (e.g., representing a single limb) or non-rigid. Rigid fsnakes can be interpolated as a rigid body using translation and rotation, while with non-rigid fsnakes, points are interpolated individually. The user can also adjust an fsnake in any frame either as a rigid body or by moving individual points.

Within a frame, contour points can be *snapped* into position based upon a weighted combination of internal forces (*length* and *angle*) and external forces (*intensity, hue* and *gravity*). The user can control the weighting. The intensity force may be calculated all along an edge, while the others are just calculated at contour points. During snapping, contour points move at most one pixel per iteration, then forces are recalculated.

Contour points in certain frames are special, in that information such as hue, edge length, and angles between edges, are stored and used to move contour points in other frames. Further details are available [3].

3.2.1 Intensity, Hue and Gravity Forces

The intensity force is generally the most important, as it drives contour points toward an edge. The intensity $e(x,y)$ is a bilinear function of the intensities $e_{i,j}$ at the four image

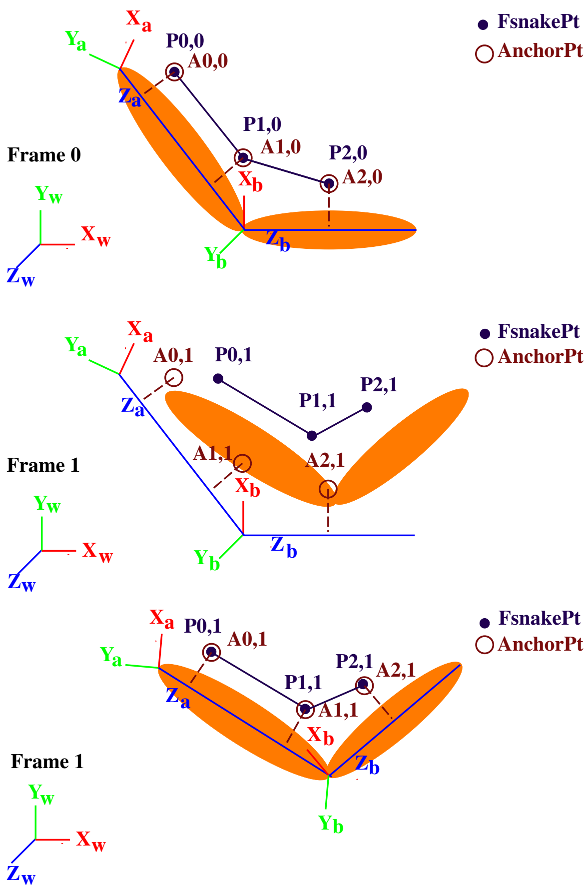

Figure 1. A two-segment model tracks a figure in an image sequence. See Section 3.1.

pixels surrounding the point (x, y). For x and y between 0 and 1:

$$\begin{aligned} e(x,y) &= (1-y)((1-x)e_{0,0} + xe_{1,0}) \\ &\quad + y((1-x)e_{0,1} + xe_{1,1}) \end{aligned}$$

The intensity force is the gradient, $\mathbf{F}_{int}(x, y) = \nabla_{x,y} e(x, y)$.

For the hue force $\mathbf{F}_{hue}$, the $e_{i,j}$ values above are substituted with the difference between the corner hue values and the initial hue at a contour point when it is first created. The hue force drives the contour point toward the most similar hue. Gravity forces $\mathbf{F}_{gravity}$ encourage the contour to go in a particular user-specified direction.

3.2.2 Length- and Angle-Preserving Forces

These forces preserve the original contour shape. The angle-preserving force $\mathbf{F}_{ang}$ for the angle between three contour points $\mathbf{P_{i-1}}, \mathbf{P_i}$ and $\mathbf{P_{i+1}}$ is found by taking the difference α between the angle when the contour was first created and that found at present. Half the force is applied as a torque to one edge and half to the other.

$$\begin{aligned} \mathbf{F}_{ang}(\mathbf{P_{i-1}}) &= 0.25\alpha[-(\mathbf{P_i} - \mathbf{P_{i-1}})_y, (\mathbf{P_i} - \mathbf{P_{i-1}})_x)] \\ \mathbf{F}_{ang}(\mathbf{P_{i+1}}) &= 0.25\alpha[-(\mathbf{P_{i+1}} - \mathbf{P_i})_y, (\mathbf{P_{i+1}} - \mathbf{P_i})_x)] \\ \mathbf{F}_{ang}(\mathbf{P_i}) &= -0.25\alpha[-(\mathbf{P_i} - \mathbf{P_{i-1}})_y, (\mathbf{P_i} - \mathbf{P_{i-1}})_x)] \\ &\quad -0.25\alpha[-(\mathbf{P_{i+1}} - \mathbf{P_i})_y, (\mathbf{P_{i+1}} - \mathbf{P_i})_x)] \end{aligned}$$

The length-preserving force $\mathbf{F}_{length}$ is the difference between the initial edge length (in the initial frame) and the present length, applied in equal and opposite directions at the contour points defining an edge. It may maintain absolute length, or relative length between neighboring edges.

3.3 Model to Image Alignment

The initial alignment is done interactively. First the user must find a correct segment size by adjusting the model to the image in appropriate frames. Next the model is positioned in *model key frames* by translating the root segment and rotating joints. The use of super-segments and inverse kinematics simplifies the task [12].

Model key frames are selected because they clearly show a position, or because they define important changes in motion, such as the beginning and end of an out-of-plane trunk rotation. The initial position of the model in other frames can be an interpolation of the position in these model key frames. This is useful, because the interpolated motion includes rotations about longitudinal segment axes and out of the image plane. These motions are hard to track.

3.4 Anchoring Fauna Snakes

Contour points are anchored, in *anchor frames*, either to the segment to which they are nearest, or to a designated segment. If anchoring is to any segment, the nearest distance between a contour point and the projection of the segment longitudinal axis onto the image plane is found. The anchor point **A** receives the X and Y coordinates of its contour point, but the Z value is that of the nearest location on the longitudinal axis of the anchor segment. The anchor is transformed into the local coordinate system of the anchor segment, and it moves with the segment in all future frames, until a new anchor frame in encountered.

While fsnake point **P** and anchor point **A** originally project to the same position on the image plane in the anchor frame, their behavior in other frames is different (see Figure 1). **P** is fundamentally a world space position and may change in other frames to track the image figure. **A**, however, is fundamentally a fixed local position in the anchor segment coordinate frame. Its world space position changes when the model segments translate and rotate.

The model must be appropriately aligned with the image figure in anchor frames. It is useful to re-anchor whenever the fsnake is failing to produce the desired motion in the model.

3.5 Automatically Repositioning the Model

All that came before is to make possible the automatic adjustment of the model in future frames so that the error between contour points and their anchors is minimized by some criteria. Our approach is a *kinematic* one, based only on relative positions; although it is natural to speak of adjustments as due to forces and torques, they are virtual.

The user has discretion over the kind of change that an fsnake can cause to the model, and also over what parts of the model are affected by it. Each contour point may cause a *translation*, an *image-plane rotation*, and/or an *out-of-plane rotation*. Each segment of the body may be affected by: (1) only contour points anchored to that segment; (2) by contour points anchored to the chain of joints to which it belongs (the super-segment); or (3) by any contour points distal to that segment in the model hierarchy. The latter two cause an *inverse kinematic* adjustment to a chain of segments, where a chain of joints is repositioned.

Because distal segment positions are affected by proximal changes, adjustments must be applied recursively from the root of the model outwards. Translations are applied first; then, for each segment, in-plane rotation followed by out-of-plane rotation.

3.5.1 Translation

Translational fauna snakes apply a position change to the model root segment parallel to the image plane. The translation is calculated using the difference between the projected positions in world space of each contour point $\mathbf{P_i^{pw}}$ and its anchor $\mathbf{A_i^{pw}}$. The total translation change of the root segment from n translate fsnake points in frame j is:

$$\mathbf{F_j} = \sum_{i=1}^{n} \frac{\mathbf{P_{i,j}^{pw}} - \mathbf{A_{i,j}^{pw}}}{n}$$

3.5.2 Image-Plane Rotation

Image plane rotations occur around the Z-axis of world space, perpendicular to the image plane. (Rotations are transformed into local segments space for application.) Some set of n contour points (Section 3.5) exert virtual torques on each segment. The torque due to contour point $\mathbf{P}_i$ on segment b around its origin $\mathbf{O_b}$ depends on the angle θ_i between two vectors projected onto the image plane: the vector from the segment origin to the anchor point in world space, which is $\mathbf{A_i^{pw}} - \mathbf{O_b^{pw}}$, and the vector from the segment origin to the contour point in world space, which is $\mathbf{P_i^{pw}} - \mathbf{O_b^{pw}}$ (see Figure 2). The angle θ_i is weighted by the relative squared distance of the projected anchor $\mathbf{A_i^{pw}}$ from $\mathbf{O_b^{pw}}$. Letting r_i denote the actual projected distance, the actual angular change θ_b applied to segment b due to n contour points is then:

$$\theta_b = \frac{\sum_{i=1}^{n} \theta_i r_i^2}{\sum_{i=1}^{n} r_i^2}$$

The axis of rotation is the world Z-axis. However, because rotations must be applied in the local segment space, the world Z-axis must be transformed to local segment space before the rotation is applied.

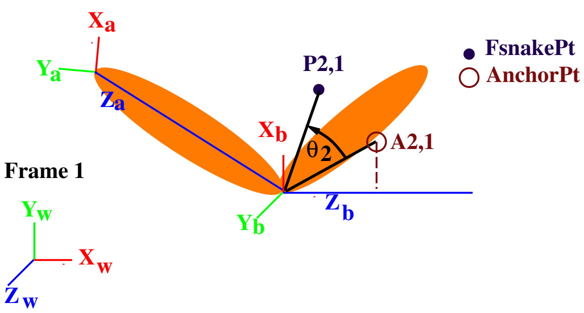

Figure 2. θ_2 **is the image plane correction angle due to** $\mathbf{P_{2,1}}$ **and** $\mathbf{A_{2,1}}$**.**

3.5.3 Out-of-Plane Rotation

If part of the image figure is rotating out of the image plane, projected contour points tracking it will move closer or farther from the point about which rotation occurs. This is mimicked in the model by rotating the appropriate segment about an axis perpendicular to both the segment longitudinal axis and the image plane normal. This rotation will not affect the projected direction of the longitudinal segment axis.

Calculations are done in world space, but applied in local segment space. To simplify the mathematics, we effectively translate the segment to the world origin and rotate around the world Z-axis to place the longitudinal segment axis in the world $Y = 0$ plane; call this rotation angle α. After computing and performing the out-of-plane rotation, we invert the rotation by α and the translation. In the new frame rotated by α, the out-of-plane rotation is simply a rotation around world Y by some angle ϕ and the error for each contour point is simply the difference in X-value between the point and its anchor. Let $\mathbf{P}_i^{\alpha w}$ and $\mathbf{A}_i^{\alpha w}$ be the positions of the contour point i and its anchor after rotation by α. Let $\mathbf{A}_i^{\alpha w}(\phi)$ be the anchor position after a further rotation by ϕ around the Y-axis. Subscripts of x, y, and z denote the X, Y, and Z components of vectors. Thus $A_{ix}^{\alpha w}(\phi) = A_{ix}^{\alpha w} \cos\phi + A_{iz}^{\alpha w} \sin\phi$.

To find the solution angle ϕ, we (approximately) minimize the squared error E, which is the sum over i of the squared distance in X between the i-th contour point and its anchor, in the rotated frame. We (approximately) find the ϕ for which $\partial E/\partial\phi = 0$.

$$E = \sum_i \left(P_{ix}^{\alpha w} - A_{ix}^{\alpha w}(\phi)\right)^2$$

$$\frac{\partial E}{\partial \phi} = \sum_i 2\left(A_{ix}^{\alpha w}(\phi) - P_{ix}^{\alpha w}\right) \frac{\partial A_{ix}^{\alpha w}(\phi)}{\partial \phi}$$

Dropping higher-order terms in $\tan\phi$, the solution is

$$\tan\phi = \frac{\sum_i A_{iz}^{\alpha w}(P_{ix}^{\alpha w} - A_{ix}^{\alpha w})}{\sum_i \left(A_{ix}^{\alpha w}(P_{ix}^{\alpha w} - A_{ix}^{\alpha w}) + (A_{iz}^{\alpha w})^2\right)}$$

3.6 Occlusion and Camera Motion

A virtual anatomy can be rendered with our model [17] and used to estimate when contour points become occluded. (So far, we have only tested this with ellipsoidal approximations to limbs.) Because contour anchor points have a world Z value determined by their anchor segment, they become hidden when another part of the anatomy is rendered in front. By giving the anchors a known unique color and checking the color at the projected anchor position in the frame buffer, visibility can be detected. When the anchor is invisible, forces are not calculated for the contour

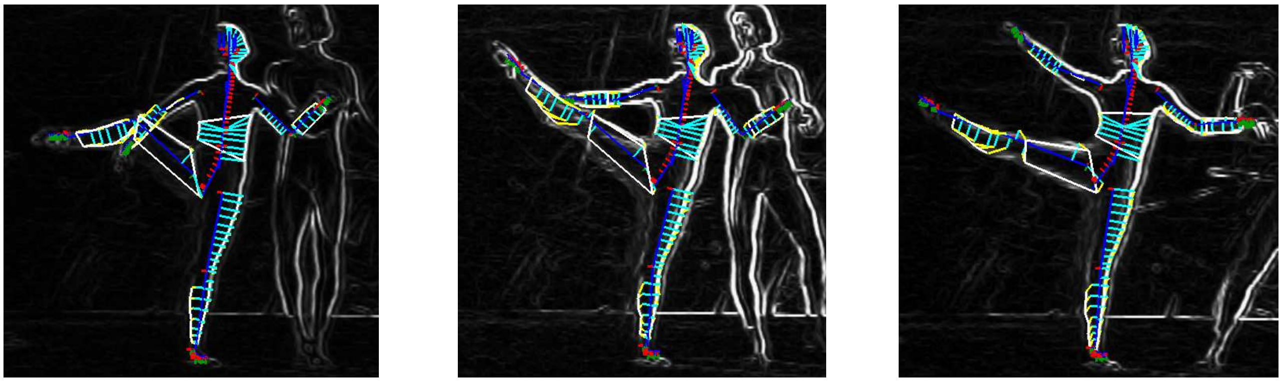

Figure 3. Three Frames from Dance Figure.

point, though it is still influenced by the neighboring points. Although contour points and their anchors do not project to the same position, they are generally close enough that this approach improves tracking.

Our rudimentary approach to removing camera motion is to treat the world coordinate system as an independent segment. Contour points are anchored to this segment in the normal manner and changes in world space are produced. As movement of the world, in which the image plane and the human model exist, is the inverse of the movement of the camera, this motion can be tracked and removed from the human model. In terms of a motion library, other methods of separating the human motion from the world can be applied, such as calculating the position of a known stationary body part or calculating the distance traveled in a stride.

4 Results

We experimented on two sequences: a simple dance movement consisting of a largely planar motion, and the complicated cavorting about of a child in the surf. Color images and animations can be seen on our web site: `www.cse.ucsc.edu/~wilhelms/fauna`.

The dancer was nicely distinguishable against the simpler background using a combination of median filter, edge detection, greyscale, and intensity adjustment (see Figure 3). The lines perpendicular to the segment axes show the nearest points of contour point anchors to their anchor segments. Eleven fauna snakes were used, and automatic tracking worked fairly well except on the lower arms, whose edges were occluded or unclear in some frames. Only a single initial key position was used for the model. Only the neck fauna snake used inverse kinematics. Fsnakes were applied in 15 frames of 150 frames.

More challenging was the child in the surf (Figure 4). The background was too complex for accurate tracking in most frames, and the user had to guess what was happening when the boy went behind the man. Every other frame of the video was used, for a total of 64 frames. Four complete model positions were originally keyframed, and parts of the model were keyframed further when necessary. Only five fsnakes were used: one translated and oriented the trunk, two moved the left leg, one moved the right leg and one the left arm using inverse kinematics. The right arm was left as keyframed, and missed much of the complexity of the motion. The results shown, from model creation to stored animation, took one afternoon.

We find that when automatic tracking is problematic, creating a different fauna snake for each segment works best. In this case, the fsnake can be treated as rigid and adjusted easily when tracking becomes incorrect, a matter of a few seconds per frame. Further, the motion can be extracted gradually outward from the body root, adding distal fsnakes as needed. The motion extracted can be quite noisy, and additional filtering is necessary. We can apply a mean filter to the captured motion curves (position versus time) at any time during or after extraction.

5 Discussion and Conclusions

While image-plane rotations proved fast and reliable, out-of-plane rotations are more problematic, and need to be used with discretion, because the tendency of contour points to slide along an edge makes angle calculation inaccurate. Rotations along segment longitudinal axes are not explicitly dealt with at all, although keyframing the model in three dimensions produces a reasonable longitudinal rotation in in-between frames, and this remains after adjustment by the contours. No joint limits are applied; we will address that in future work.

We found it possible to catch a usable approximation of complex motion in a reasonable time. Thus we feel that a combination of computer vision and interactive methods can be used to extract articulated motion in cases where completely automatic methods fail.

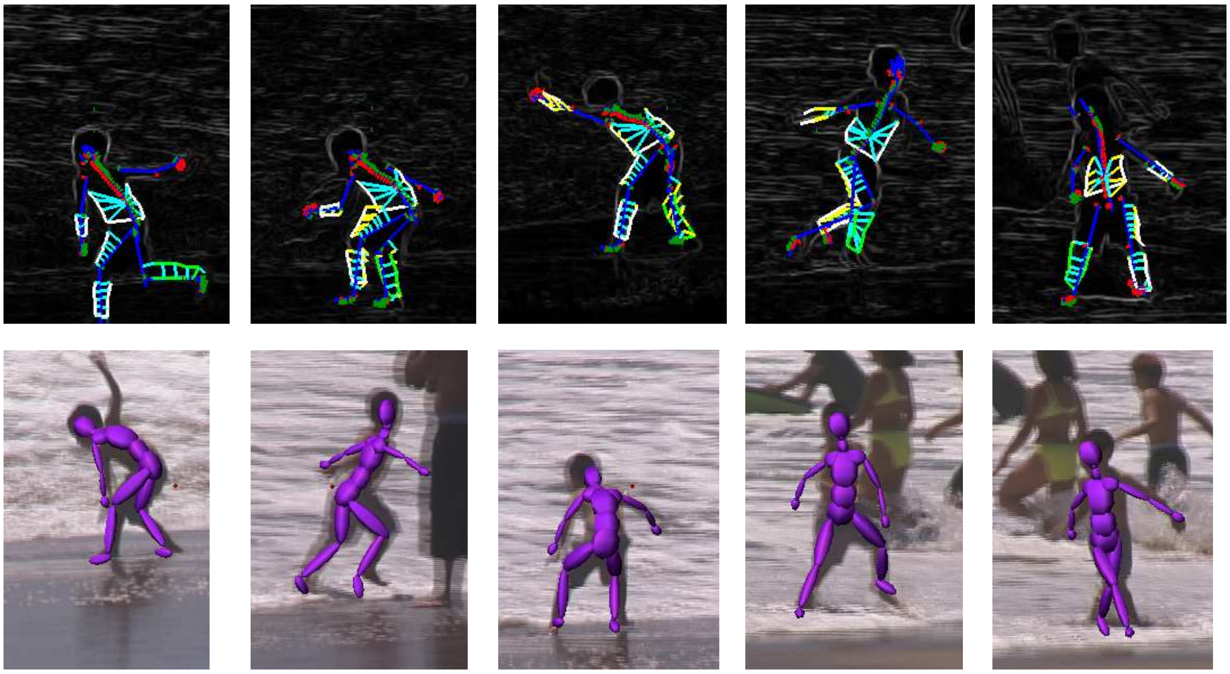

Figure 4. Boy in the surf: Top shows fsnakes, bottom ellipsoidal model. (Right arm not tracked.)

Acknowledgments: This research was supported by NSF Grants CCR-9503829, CCR-9972464 and CDA-9724237.

References

[1] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision and Image Understanding*, 70(2):142–156, 1998.

[2] K. Akita. Image sequence analysis of real world human motion. *Pattern Recognition*, 17(1):73–83, 1984.

[3] L. Atkinson-Derman. Tracking on the wild side – using active contours to track fauna in noisy image sequences. Master's thesis, UC Santa Cruz, CA 95064, June 2000.

[4] A. Bharatkumar, K. Daigle, M. Pandy, Q. Cai, and J. Aggarwal. Lower-limb kinematics of human walking with the medial axis transformation. *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 70–76, Austin, TX, Nov. 11–12 1994.

[5] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

[6] Z. Chen and H. Lee. Knowledge-guided visual perception of 3D human gait from single image sequence. *IEEE Trans. on Systems, Man, and Cybernetics*, 22(2):336–342, 1992.

[7] L. Goncalves, E. D. Bernardo, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3D. *Proc. IEEE Fifth Int'l Conference on Computer Vision*, pages 764–770, Cambridge, Mass., 1995.

[8] Y. Hel-Or and M. Werman. Constraint fusion for recognition and localization of articulated and constrained objects. *Int'l Journal of Computer Vision*, 19(1):5–28, July 1996.

[9] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int'l Journal of Computer Vision*, 29(1):5–28, 1998.

[10] I. Kakadiaris, D. Metaxas, and R. Bajcsy. 3D human body model acquisition from multiple views. *Proc. IEEE Workshop on Non-Rigid and Articulated Objects*, pages 618–623, Boston, MA, June 20–23 1995.

[11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int'l Journal of Computer Vision*, 1(4):321–331, 1988.

[12] J. Lapierre. Matching anatomy to model for articulated body animation. Master's thesis, UC Santa Cruz, CA, Dec. 1999.

[13] D. D. Morris and J. M. Rehg. Singularity analysis for articulated object tracking. *Proc. Computer Vision and Pattern Recognition*, pages 289–296, Santa Barbara, CA, June 23–25 1998.

[14] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7), 1991.

[15] F. Perales and J. Torres. A system for human motion matching between synthetic and real images based on a biomechanic graphical model. *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 83–88, 1994.

[16] J. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 16–22, 1994.

[17] J. Wilhelms and A. Van Gelder. Anatomically based modeling. *Proc. ACM SIGGRAPH*, Aug. 1997.