Interactive Video-Based Motion Capture for Character Animation

JANE WILHELMS and ALLEN VAN GELDER
Computer Science Department, University of California, Santa Cruz
Santa Cruz, CA 95064
email: wilhelms,avg@cs.ucsc.edu

ABSTRACT
We combine techniques from interactive computer graphics and automated computer vision to extract articulated body movement from video. Active contours and techniques from image processing are used to identify features and establish feature relationships from frame to frame. A three-dimensional hierarchical model is associated with active contours, which then kinematically "pull" the model along with them. Motion capture is iterative and interactive, with users adjusting the extraction as needed. The technique is demonstrated using the motion of a horse, though it is equally appropriate for human figure animation.

KEY WORDS
Character Animation, Computer Vision, Motion Capture.

1 Introduction

The animation of real and fantastic humans and animals presents unique and difficult challenges because of the complexity of their motion and our expectations of their behavior. Such animation has many practical applications in entertainment, ergonomics, and biomechanical simulation. Believable virtual characters must move freely and in balance, with timing consistent with the real world. Keyframing techniques are laborious and require artistic talent and training. Physical simulation suffers from major control problems. Studio motion capture, at present, is most realistic, but expensive and environmentally restricted.

We are particularly interested in the natural and unrestricted movement of animals (including humans) in outdoor environments, such as horses cantering or children playing in the surf. Such motion cannot be easily captured in a studio, but can be easily recorded in the form of video. Unfortunately, video images only provide two-dimensional projections of the motion. (Though multiple, high-speed cameras could simplify the problem, we are interested in the most general case, of a single camera.) In this paper, we discuss our research in extracting three-dimensional, articulated motion from such two-dimensional recordings.

Our approach combines semi-automated methods from computer vision (to extract and track features), with partly interactive approaches from computer graphics (matching a model with image features). Our contribution is not so much a new algorithm, but a new way of combining techniques from different fields to solve a particular problem. A significant new contribution is our method for automatically adjusting the virtual model to track the image sequence. We concentrate on the horse as an example creature, with some results shown for a human. Our web site `www.cse.ucsc.edu/~wilhelms/fauna` provides more images and animations.

2 Background

The computer vision methods most applicable to our work are model-based techniques for extracting articulated body motion from monocular image sequences (an excellent survey paper is by Aggarwal [1]). Image features are tracked by an articulated *model* that may be created interactively or generated automatically. The model tracks by adjusting its geometry to minimize the error between it and image features. We track features and associate them with the model using *active contours*, or *snakes* [3, 5]. Active contours are sets of connected control points associated with features (usually edges) in the underlying image. They can both automatically track features in a sequence of images, and be easily manipulated by the user with geometric transformations. This makes them very appropriate for our combined vision-and-graphics approach.

3 The 3D Generic Model

Our horse model is a three-dimensional hierarchy with 83 *segments* connected by joints [8]. (Geometric bones, muscles and skin can be associated with the hierarchy, but this subject is not relevant to this paper.) We created a "generic" horse by fitting segments to anatomical diagrams [4]. As a first step in the video extraction procedure, the generic horse is interactively adjusted to the horse in the video. Each segment has a *default geometric transformation* that places it on its parent segment (or in the world, for the root segment), and a *state geometric transformation* that transforms it relative to the default. The Z-axis of the local segment frame is the longitudinal axis.

State transformations can be restricted by joint limits, either using simple Euler angle limits or (usually more appropriate) reach-cone limits [10]. Figure 1 illustrates reach-cone limits on the generic horse. Because it is unwieldy to manipulate a structure with 83 joints, we group linear chains of joints into *supersegments*, so that they can be manipulated together using inverse kinematics or by simply distributing weighted transformations among joints [6].

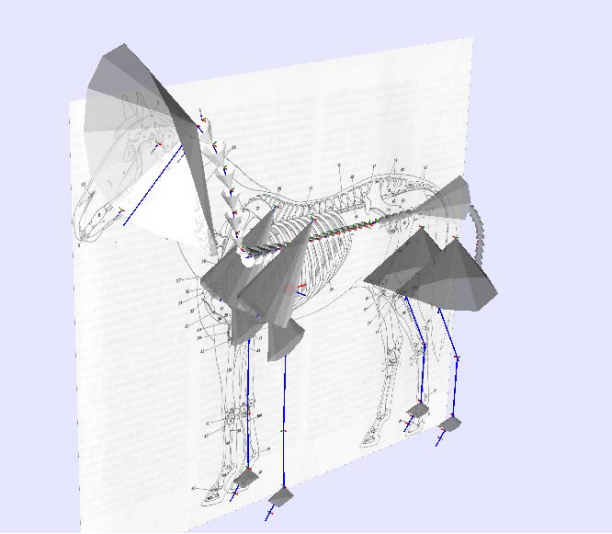

Figure 1. Generic model with joint limits.

4 Image Feature Tracking

On the vision side, we next need to identify those pixels in each video frame that represent the *feature* of interest, i.e., the horse. Once this feature is found in a single frame, it must be *tracked* in other frames. Because of the unrestricted nature of our video sequences, we depart from the concentration on fully automatic techniques which characterizes many computer vision approaches. We place no constraints on camera parameters, camera motion, feature motion, feature characteristics, lighting, or background. We accept the necessity of considerable user interaction to guide the feature extraction process. However, we provide the user with many techniques, interactive and automatic, to simplify the process.

Active contours (also called *snakes*) [5] are our feature representation. In our case, these are two-dimensional polylines that snap to and track image features. The user creates them in one or more frames so that they lie near a recognizable image feature (such as the outline of a leg). The contours are copied to succeeding frames, and adjusted automatically by examining the pixels near the snake so that they follow the same feature.

Active contours are appropriate for our problem because their simple geometry is easily amenable to user interaction. The user creates them with the mouse, by tracing the desired features. During the tracking process, the user can intervene at any time to adjust the snakes (either single vertices or as a rigid body) when automated alignment fails. The contour geometry is also easily associated with the hierarchical model, in the final extraction phase (Section 5).

Low-level image processing methods, such as hue and intensity manipulation, smoothing, and edge detection can also be applied to images to clarify features (Figure 2).

Figure 2. Left: the original image; right after conversion to grey-scale and edge detection.

4.1 Active Contours

We found active contours (or snakes) [5] desirable because they: (1) identify image features; (2) track automatically; (3) can be easily manipulated where tracking fails; and (4) can drive the animation of the three-dimensional model. There are generally several active contours in each frame, each associated with a particular feature of interest in the image, usually a part of the body of animal being tracked. Figure 3 shows the active contours used to extract the motion of the foal walking.

An active contour for one frame can be used to automatically generate new versions of itself in other frames in three ways: (1) by simply copying it from another frame; (2) by extrapolating the control points from two previous frames, as if the velocity were constant; and (3) by interpolating between the positions of control points in specified frames before or after the new given frame. Once added to the new frame, several tracking heuristics are applied.

Tracking can be problematic when motion is fast (features are many pixels away in the next frame) and when features aren't consistent (features are occluded or fade away). Active contours can also be subject to "creeping" along image edges. Thus, the user can at any point interactively adjust either the whole contour (rotating and translating it as if it were rigid) or adjust individual points that sneak away from their features.

Given one or more initial positions where the contours are appropriately positioned relative to image features, the contours must automatically adjust themselves to track these features. To do this, virtual *forces* are applied to the contours. These forces may be categorized as *internal forces*, which represent a contour's interaction with itself; *external forces* which arise from the contour's interaction with the image; and *gravity forces*, which pull the contour in a particular direction. The heuristics used to determine virtual forces are under development and not very satisfactory at present, and considerable human interactive is involved. Additional details are given elsewhere [2, 11].

Figure 3. Active contours for foal walking: top, the creation frame; bottom, a later frame.

5 Extracting 3D Animation

Finally, the two-dimensional active contours that track features in the video frames are associated with the three-dimensional model, and "pull" it into new positions, creating an animation. As this is a very under-constrained problem, considerable user input is involved. This is an iterative process, and proximal body parts are generally captured before distal ones. Filtering can be applied to the motion curves thus created, to reduce noise. The steps are:

1. In a preliminary step, the user interactively adjusts the model geometry to fit the subject in the video, so that it has the appropriate size and proportions.

2. The user then aligns the model with image features in *model key frames* where their relationship is clear. Interpolation is used between the key frames to create an initial approximate animation.

3. The active contours are automatically anchored to the model in *anchor frames* (Section 5.1), where the model is properly aligned. (Generally model key frames are also anchor frames.)

4. Finally, the active contours "pull" the model into position in the remaining frames (Section 5.2).

5.1 Anchoring Active Contours to the Model

In an anchor frame, each active contour vertex is associated with a particular nearby model segment. Distance is calculated between the two-dimensional projection of the longitudinal axis of a segment and the i-th contour vertex $\mathbf{P_i}$. The projection is onto the image plane, which is the world-space XY plane. When the nearest point is found in two dimensions, each active contour vertex is given a Z-value equal to that of the near point on the segment axis. Each vertex is then converted to the local coordinate frame of the anchoring segment, establishing a relationship between the world space features represented by active contour points, and the three-dimensional model. The local representation of an active contour vertex $\mathbf{P_i}$ is the anchor $\mathbf{A_i}$. The user can determine whether all segments, a particular supersegment, or a particular segment is examined to find anchors, because the nearest segment in the body is not always the one most desirable for anchoring.

The world space active contour vertex $\mathbf{P_i}$ and its local segment representation $\mathbf{A_i}$ project to the same point in an *anchor frame*; they generally do not in other frames. The active contours in other, non-anchor frames have moved to follow the image features. The geometric relationship between the active contour anchor vertices and their vertices on other frames is used to adjust the model to follow the moving features (Figure 4).

5.2 Automatically Repositioning the Model

To reposition the model in a non-anchor frame, the anchor representations of active contour points (which are stored in the local segment coordinate system) are converted to world space and projected onto the image plane. Then model segment geometry is automatically adjusted to minimize the distance between the active contours for this frame and the projected anchors. The user controls the type of geometric adjustment that is used, by designating the type of "pull" any active contour applies. The discussion here is necessarily brief due to space considerations; for a longer version, see the technical report [9].

A *translation* contour moves the model root segment parallel to the image plane, using the average difference between the 2D positions in projected world space (**pw**) of each contour point $\mathbf{P_i^{pw}}$ and its anchor $\mathbf{A_i^{pw}}$.

To compute a rotational adjustment for a given segment, the world-space positions $\mathbf{P_i}$ of each active contour vertex affecting the segment, and the corresponding local anchor positions $\mathbf{A_i}$, are converted to the local coordinate frame of the segment being adjusted. These are then projected to a plane perpendicular to the desired axis of rotation. The virtual torque is dependent on the angle θ between these two projected vectors, weighted by the relative squared distance of the projected anchor from the segment frame origin r_i. The actual angular change θ_s applied to segment s due to n active contour vertices is

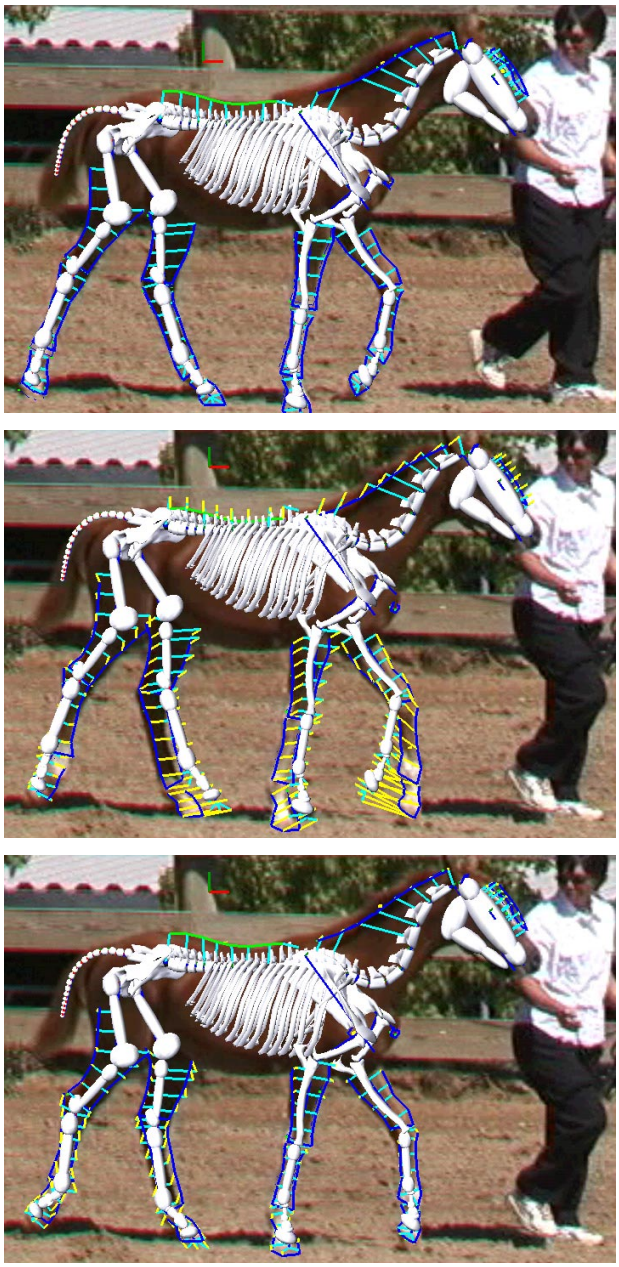

Figure 4. Adjusting the model to match the active contours: top, the anchor frame; middle, the next frame before model adjustment; bottom, after adjustment. Active contours are shown in dark blue or green; association with segments in cyan; and displacement between contour points and anchors in yellow.

then:

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

For *image-plane rotations*, which are most accurately tracked, the desired axis of rotation is perpendicular to the image plane. For Euler axis rotations, it would be the appropriate X, Y, Z-axes. *Out-of-plane* rotations are more difficult to track, and are discussed elsewhere [9].

The motion extraction is considerably helped, when the user has positioned the model in a few selected *model key frames*, so that the interpolated position of the model is already reasonably close, in three dimensions, to a good solution. The combination of human and computer produces results more successfully and faster than either alone. The use of joint limits to constrain automatic repositioning also helps restrict motion to reasonable values.

Any segment may be affected by: (1) only active contour vertices anchored to that segment; (2) by any active contour vertex anchored to the chain of segments (supersegment) to which the segment belongs; or (3) by any contour points distal to the segment. For example, the left front leg in Figure 3 has snakes that are individually applied to one segment, while the right has a foot snake that uses inverse kinematics.

6 Results

We have experimented with a variety of video sequences, both humans and animals. Further results, including animations, can be seen on our web site: `www.cse.ucsc.edu/~wilhelms/fauna`.

The horse examples shown here are of a year-old Arabian foal. We first adjusted the generic horse geometry for this breed and age, using captured video of the foal walking. As we had video both from the side and front, we could easily adjust the geometry in three-dimensions.

We captured the foal walking parallel to the image plane largely using active contours, with some user adjustment. Such motion is relatively easy to capture, the motion being largely two-dimensional. Figures 3 and 4 show frames from one walk cycle.

Figure 6 shows a foal running free and jumping an obstacle. The speed of the motion made this difficult for automatic tracking; however, with user input, a convincing animation of a complex motion was extracted. Though the horse is moving at an angle to the image plane, the use of three model key frames to give a base three-dimensional motion simplified further adjustment.

Figure 5 shows a horse model including skin in a jumping pose from the animation extracted from the foal motion. Our approach for skin animation and mapping motion to new models is described elsewhere [7].

Figure 5. Adult horse model with skin, from an animation extracted for foal motion.

7 Discussion and Conclusions

We have found that automatic tracking with active contours is most successful for moderately slow motions and where the background is relatively simple. Here we challenged the method with complex, fast, non-planar motion against non-trivial backgrounds. An improved active contour algorithm, or an alternate feature tracking approach would be helpful. Segmenting and tracking moving objects is an active area in the computer vision community; we hope to avail ourselves of some of their newer methods. However, for our scenes, we found interactively adjusting the active contours easy and fast.

We similarly found that user supervision was necessary to extract reasonable three-dimensional motion from complicated motion out of the image plane. However, we were able to extract motion that gives the timing and "feel" of a motion, and is reasonably close geometrically.

The motion extracted needs to be further processed to remove camera motion, and refined for constraints such as feet placement on the ground.

Acknowledgements

This research was supported by NSF Grants CCR-9972464 and CDA-9724237. We wish to thank Leon Atkinson-Derman, Luo Hong, Maryann Simmons, and Mark Slater for their help with this project.

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

[3] Andrew Blake and Michael Isard. *Active Contours*. Springer-Verlag, 1998.

[4] Peter C. Goody. *Horse Anatomy: A Pictorial Approach to Equine Structure*. J.A. Allen, London, 1983.

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

[6] Jeff Lapierre and Jane Wilhelms. Matching anatomy to model for articulated body animation. In *Proceedings of 1999 IASTED Computer Graphics and Imaging Conference*, Palm Springs, Ca., 1999.

[7] Maryann Simmons, Jane Wilhelms, and Allen Van Gelder. Model-based reconstruction for creature animation. *ACM Symposium on Computer Animation)*, July 2002. To appear.

[8] Jane Wilhelms and Allen Van Gelder. Anatomically based modeling. In *Computer Graphics (ACM SIGGRAPH Proceedings)*, Aug. 1997.

[9] Jane Wilhelms and Allen Van Gelder. Combining vison and computer graphics for video motion capture. Technical report, UCSC, November 2001.

[10] Jane Wilhelms and Allen Van Gelder. Efficient spherical joint limits with reach cones. Technical report, UCSC, April 2001. Available at `ftp://ftp.cse.ucsc.edu/pub/avg/jtl-tr.pdf`.

[11] Jane Wilhelms, Allen Van Gelder, Leon Atkinson-Derman, and Hong Luo. Human motion from active contours. In *IEEE Workshop on Human Motion*, Austin, TX, December 2000.

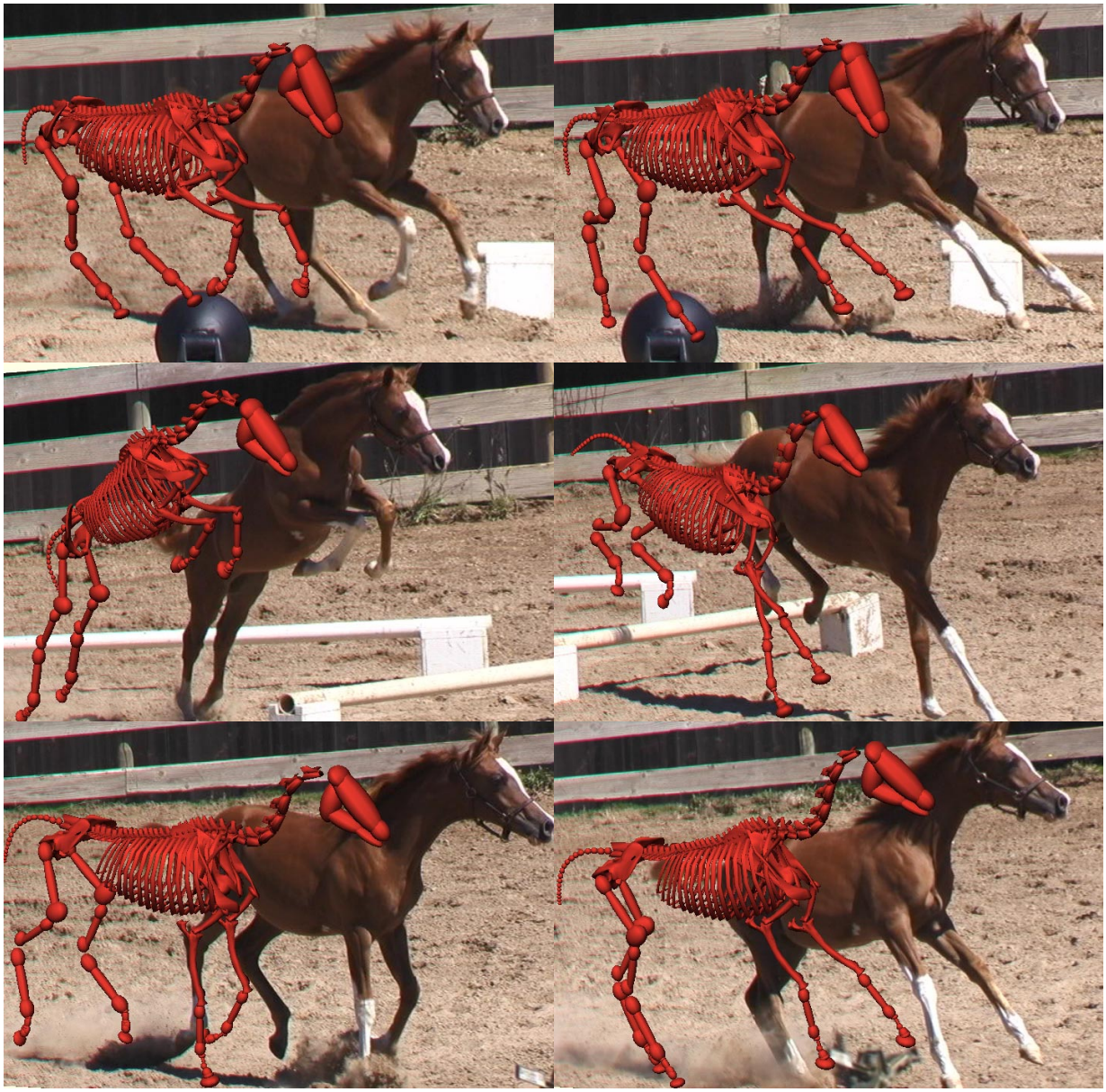

Figure 6. Foal jumping an obstacle, shown at an angle to the image plane.