Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2009)
E. Grinspun and J. Hodgins (Editors)

Leveraging the Talent of Hand Animators to Create Three-Dimensional Animation

Eakta Jain[1], Yaser Sheikh[1], Jessica Hodgins[1,2]

[1]Carnegie Mellon University
[2]Disney Research, Pittsburgh

Abstract

The skills required to create compelling three-dimensional animation using computer software are quite different from those required to create compelling hand animation with pencil and paper. The three-dimensional medium has several advantages over the traditional medium—it is easy to relight the scene, render it from different viewpoints, and add physical simulations. In this work, we propose a method to leverage the talent of traditionally trained hand animators to create three-dimensional animation of human motion, while allowing them to work in the medium that is familiar to them. The input to our algorithm is a set of hand-animated frames. Our key insight is to use motion capture data as a source of domain knowledge and 'lift' the two-dimensional animation to three dimensions, while maintaining the unique style of the input animation. A motion capture clip is projected to two dimensions. First, time alignment is done to match the timing of the hand-drawn frames and then, the limbs are aligned to better match the pose in the hand-drawn frames. Finally the motion is reconstructed in three dimensions. We demonstrate our algorithm on a variety of hand animated motion sequences on different characters, including ballet, a stylized sneaky walk, and a sequence of jumping jacks.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Animation from Motion/Video Data, Animation of Articulated Figures, Believable Motion, Computer Vision for Animation]:

1. Introduction

Character animation is the art of making an inanimate character move as if it possesses a unique personality [JT95]. This art has its origins in the traditional hand-drawn style [McC14], though there is rich variety in the tools and techniques used today—sketching figures, manipulating puppets, and posing computer-generated characters. Each style of animation requires that the artist develop a different set of skills to create a visually compelling endproduct [Las94]. Animators who create traditionally animated movies such as *Snow White and the Seven Dwarfs* and *Princess Mononoke* are trained in the medium of pencil and paper. The magic of their art is intimately connected to the two-dimensional (2D) medium; there is little automatic transference of their skill to computer graphics software, with its sliders and inverse kinematics.

Although not intuitive to traditionally trained artists, the three-dimensional(3D) computer-generated medium has several advantages—it is easier to move the camera around, vary lighting, create shadows, and add physical simulations. Many of the animated movies released in the past decade have been fully computer-generated features [Wik09]. In this paper, we propose a method to leverage the unique talent of

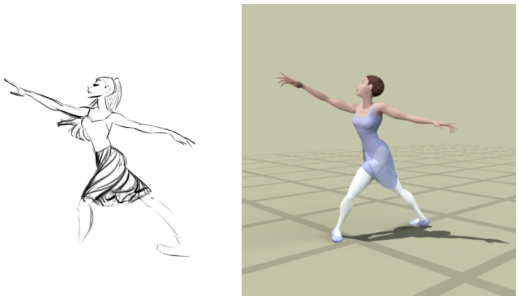

Figure 1: *A frame from a hand-drawn animation and the corresponding three-dimensional frame.*

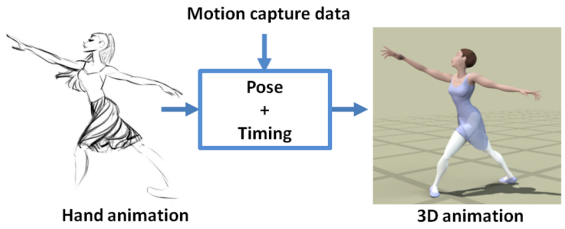

Figure 2: *Input: Hand animation. Output: 3D animation. Our key insight is to use motion capture data as domain information. We generate 3D animation that matches the style elements (pose and timing) of the hand animation.*

traditional animators to create 3D animation while allowing them to continue to work with pencil and paper.

The 2D animation workflow starts with an animator who sketches the character's movement at 6-12 frames per second. A cleanup/inbetweening artist cleans up the animator's lines and draws in the remaining frames to generate 24 frames for every second of animation. Our goal is to lift the 2D animation to three dimensions on a computer generated character (Figure 1) at this stage in the pipeline. This workflow is quite different from the typical 3D computer generated animation pipeline, which starts with modeling a character, rigging it for intuitive control, and finally animating it— all within a 3D computer graphics software package.

The problem of lifting 2D animation to 3D can be framed as a 3D reconstruction problem [Kan81, LC85] because information about the third dimension exists as a mental model in the artist's head, but is not directly represented in the drawing. Reconstruction is made more challenging because the parameters underlying the artist's drawings can change from frame to frame. The size and shape of the character's limbs may change intentionally to convey information about movement (e.g. tension), or unintentionally, as a result of imprecision or noise in the drawings. The camera exists in the artist's head, and parameters such as field of view may vary throughout a sequence. In addition, hand animation and computer-generated animation may be perceived differently by viewers—for example, John Lasseter hypothesizes that holding a pose for several frames is acceptable for a hand animated sequence, but makes the motion look lifeless in 3D [Las94].

Most importantly, in our experience, creating compelling three-dimensional animation is more than minimizing an image distance metric [BSB*07], [SB03]. We want to capture the style of the artist's hand animation, and by style we mean the nuances of pose and timing that give a character its personality. There are many components that go into an artist's unique style: squash and stretch, anticipation, follow through and exaggeration [JT95]. We observe that some of these style elements are contained in the poses of the animated character. In this paper, we focus on creating 3D animation which captures (1) the pose of the character in each frame, and (2) the timing of the series of frames. We do not capture significant model changes such as squash and stretch.

The input to our algorithm is a sequence of hand animated frames (Figure 2). Our key insight is to leverage motion capture data as a source of domain knowledge to aid in resolving ambiguities and noise in the 2D drawings. We show that 3D animation can be created by modifying motion capture animation with style elements from the hand-drawn frames. We modify the timing of the motion capture poses to match the hand-drawn poses via dynamic time warping. We introduce a translation and scale invariant pose descriptor to capture pose information. We present results on a variety of hand animations including a ballet sequence, a stylized sneaky walk and a jumping jacks sequence. We also evaluate our algorithm quantitatively using a motion capture sequence as input.

2. Related Work

The graphics community has recognized that the medium of pencil and paper (or stylus and tablet) is intuitive in many respects, for example, to rapidly sketch character motion [TBvdP04], or key poses [DAC*03]. In Section 2.1, we discuss the work that is most closely related to ours in that the 3D pose is being modified or reconstructed based on an artist's 2D input. The computer vision community is also interested in reconstructing articulated 3D pose from images for such applications as marker-less motion capture, activity recognition and generation of novel viewpoints. In Section 2.2, we discuss representative works that assume a known skeleton, and use motion capture data as domain knowledge in reconstruction.

2.1. 3D Animation from a Hand Animator's Input

Animation researchers have attempted to use input from traditionally trained artists to make motion capture animation more expressive [LGXS03]. Li and colleagues start with a motion capture animation, and ask an artist to redraw certain key frames to make them more expressive. A user alters the pose and limb lengths of the skeleton in the corresponding motion capture frame to match the drawing as closely as possible. The algorithm then warps the mesh to match the artist's drawing and provides a seamless transition to previous and subsequent frames. Our work attempts to exchange the human-in-the-loop for a more automatic method that sets the skeletal poses for an entire hand-animated sequence.

Davis and colleagues generate all possible 3D poses underlying a given hand-drawn stick figure [DAC*03]. They design a user interface to present these poses in order of desirability. The onus of picking the 'right' 3D pose rests on the user of the interface. Our work complements this work as we generate only one 3D pose for each input drawing—the onus is on our method to select the 'right' 3D pose so that the final animation captures the style of the original hand-animation.

In [BLCD02], Bregler and colleagues track a 2D animated character and retarget the animation onto new characters in 2D and 3D. Their method requires that the animator provide key poses, which their algorithm interpolates and warps. In the case of retargetting to a new 3D character, the animator will have to generate key poses in 3D, most likely using a commercial animation package. We build upon this previous work to be able to generate all the skeletal poses without requiring that the artist interact with 3D software.

2.2. 3D Human Pose Reconstruction from Images

We note that much research has been devoted to studying the problem of reconstructing 3D pose from 2D image data. An extensive survey of techniques can be found in [MG01] and [Gav99].

A large body of work assumes a known 3D skeleton and tries to find the skeletal parameters that minimize an image distance metric [LC85, SB01, Tay00, DCR01, ST03, Smi07]. Notably, Taylor [Tay00], DiFranco [DCR01] and Sminchisescu [Smi07] track 3D human pose using monocular input data. DiFranco and colleagues [DCR01] assume that camera parameters and 2D correspondences are known for every frame to be tracked, and determine the 3D poses that minimize the image projection error under kinematic and dynamic constraints. Taylor [Tay00] does not assume that the input images were acquired using a calibrated camera, but asks a user to choose the desired pose from the set of all possible solutions, for every input image. Sminchisescu [ST03] estimates articulated human motion using image features like optical flow, motion boundaries and edge energies. In general, images provide a rich set of features that can be used for tracking robustly and handling noise [SB03]—our inputs are line drawings, which lack this richness.

Even though reconstruction of human poses has been done without human motion models [BM98, DCR01, CTMS03], recent reconstruction techniques have benefited from learning probabilistic models of human motion—either from motion capture data [HLF99, Bra99, SBF00, SBS02, AT04, AT06, UFF06], or, from 3D range scan data [BSB*07]. In particular, Sidenbladh, Black and Sigal [SBS02] observe that for high dimensional data (such as human motion) it is easier to match to segments in a huge database, than it is to model the database. They generate a 3D pose by picking a sample from the database in such a way that the previously generated poses are a good match to the sample's previous history. This 3D pose can be used as a prior for image-based tracking as in [SB01].

The most important difference between our work and the existing body of research is that the vision community has input that has a true 3D interpretation. In contrast, the skill of an artist lies in how cleverly they can convey the style of the motion, and their techniques may include modifying perspective, altering lengths of limbs and curving bones—as a result, there is likely no correct 3D interpretation of the hand-drawn frames that fits precisely to a hierarchical skeleton. Vision researchers are interested in tracking human figures in a reliable and automated manner, whereas we are interested in capturing the essence of a 2D character's movement. In addition, in the presence of noise, there is a trade-off between tracking precisely and maintaining naturalness of movement. We choose naturalness and smoothness of motion over precise tracking when faced with this trade-off.

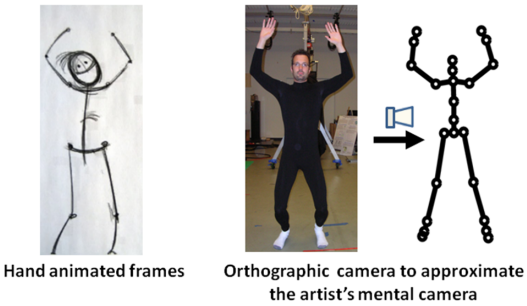

Figure 3: *3D motion capture poses are projected to 2D using a user-specified camera.*

3. Approach

The input to our algorithm is a sequence of hand-drawn images (Figure 3). A user identifies virtual markers on the animated figure's joints in every input frame, denoted $\tilde{\mathbf{x}}_h$. They also specify an orthographic camera $\mathbf{R}$ that approximates the imaginary camera used by the artist to create their drawings. Both these inputs can be reasonably integrated into the 2D animation workflow because the cleanup artist already touches up every frame of the animation when it reaches their station. In practice, it takes less than a minute per frame to mark out the virtual markers, and a couple of minutes per sequence to specify the camera using a simple GUI. We also assume that a 3D model has been constructed and rigged with a simple skeleton that represents a good approximation to the hand-drawn figure.

We take advantage of motion capture as a source of data (see, for example, *mocap.cs.cmu.edu*) and assume that a motion capture segment, $\mathbf{X}_{3D_{mocap}}$, is given that is a reasonable match to the hand animated sequence (Figure 3). The timing of this motion capture segment depends on the style of the actor who was motion captured, and is bound by the laws of the physical world (e.g. gravity). In contrast, the artist uses timing to communicate emotion or tempo, and may choose not to respect physical laws [Bab09]. The first step in our method is to warp the motion capture segment to align with the timing of the hand animation (Figure 4). The second step is to align to the pose intended by the artist in each frame of their drawing (Figure 5). We now describe each of these steps in more detail.

3.1. Description of Skeleton

The complete skeleton for a 3D character is parametrized by the three rotation values for M joints and the 3D world position of the root. The pose at a time t is denoted by $\mathbf{X}_{ja}$, which is a $(3M+3)\times 1$ vector.

For the purpose of lifting the 2D pose to three dimensions, we use a simplified parametrization—the 3D world positions of N virtual markers. Then, the 3D pose at a given time t is

$$\mathbf{X}_{3D} = \left[\mathbf{X}_1{}^T, \mathbf{X}_2{}^T, \ldots, \mathbf{X}_N{}^T\right]^T. \quad (1)$$

where each $\mathbf{X}_i \in \mathcal{R}^3$ is the 3D position of a virtual marker, and $\mathbf{X}_{3D}$ is a $3N\times 1$ vector. These marker positions contain information about two of the three joint angles for each joint. We compute the third joint angle, yaw rotation about the bone of the limb segment, separately to obtain the complete pose $\mathbf{X}_{ja}$, which can be sent to a commercial software package for rendering (Section 3.6).

In our examples, N is either 19 or 24, and M is either 20 or 21. This character model is hierarchical in that every joint is defined relative to its parent in the hierarchical chain. An example limb hierarchy would be *pelvis - upperback - lowerback - right clavicle - right humerus - right radius - right hand*. We define the hierarchical chains in terms of both joint angles and corresponding virtual markers.

3.2. Pose Descriptor

We introduce a pose descriptor to extract translation and scale invariant information about a 2D pose—the intuition behind this descriptor is that a character can be in the same pose at different locations, and two characters can have the same pose even if their relative limb lengths are different. We are inspired by shape contexts [BMP02], which are intended to be descriptors for matching two input shapes. Our goal is the inverse—to generate a 3D pose to match the descriptor extracted from the input hand drawing.

For a given 2D pose, the descriptor starts at the root (which is the pelvis) and travels every hierarchical limb chain. For every link in the chain, we determine the position vector of the child marker in a coordinate frame fixed to its parent. As illustrated in Figure 5, the position vector for the wrist would be calculated with respect to a coordinate frame fixed to the elbow. The reference orientation for this coordinate frame can be absolute (i.e. oriented along the x-axis of the world coordinate frame), or relative (i.e. oriented along the corresponding limb, in this case, *right radius*). The pose descriptor $\mathbf{P}$ for a given pose would be the vector of polar angles for the position vectors of K virtual markers of the skeletal model,

$$\mathbf{P} = [\theta_1, \theta_2, ..., \theta_K]^T \quad (2)$$

where K is the number of limbs that are needed to characterize the pose. For all our examples, we use $K = 8$ *(left and right humerus, radius, femur and tibia)* .

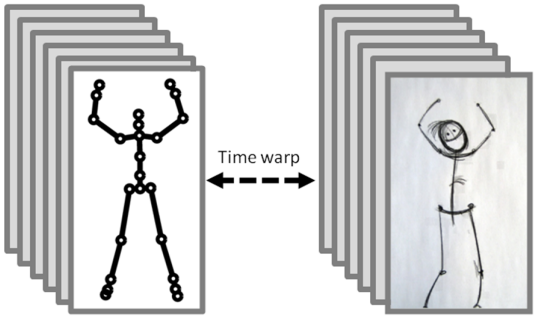

Figure 4: *The projected motion capture segment is retimed to match the timing of the hand animated frames.*

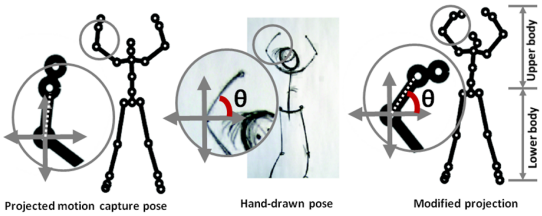

Figure 5: *The pose descriptor consists of in-the-image-plane angles for every limb segment. The limb segments of the projected motion capture pose are modified to match the pose descriptor for the hand drawn pose via planar rotation.*

3.3. Temporal Alignment

Given the sequence of hand-drawn poses $\tilde{\mathbf{x}}_h$, we use the dynamic time warp algorithm to re-time the motion capture segment $\mathbf{X}_{3Dmocap}$. This algorithm aligns two variable length time series, subject to monotonicity in the time dimension. The optimal alignment minimizes the total distance between warped elements in the two series [SC90], [Ell03].

In order to define a distance metric to compare hand-drawn 2D poses with motion capture poses, we project each 3D pose $\mathbf{X}_{3Dmocap}$ using the orthographic camera $\mathbf{R}$ to get 2D poses $\tilde{\mathbf{x}}_{2D}$. We compute the pose descriptors $\mathbf{P}_h$ and $\mathbf{P}_{2D}$ for every pose $\tilde{\mathbf{x}}_h$ and $\tilde{\mathbf{x}}_{2D}$. The distance between a hand-drawn pose and a motion capture pose (either 3D, or its projection) is defined as the sum of the internal angle difference between the angles comprising the corresponding pose descriptors,

$$d(\mathbf{X}_{3Dmocap}, \tilde{\mathbf{x}}_h) = d(\tilde{\mathbf{x}}_{2D}, \tilde{\mathbf{x}}_h) = d(\mathbf{P}_{2D}, \mathbf{P}_h) \quad (3)$$

$$d(\mathbf{P}_{2D}, \mathbf{P}_h) = \sum_{i=1}^{K} f(\theta_{2Di}, \theta_{hi}) \quad (4)$$

where f returns the internal angle difference between θ_{2Di} and θ_{hi}.

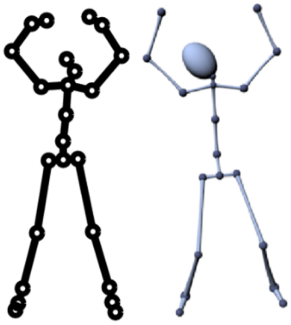

Figure 6: *The reconstructed 3D pose minimizes the error in projected marker positions. We determine the joint angles corresponding to the 3D marker positions, and estimate yaw from motion capture data.*

3.4. 2D Pose Alignment

Given the retimed motion capture segment, our goal is to generate 3D poses to match the pose descriptors for the corresponding artist drawn poses. Because the input provides information about only two dimensions of a 3D pose, we project the 3D motion captured poses $\mathbf{X}_{3D}$ to two dimensions with the user-defined camera $\mathbf{R}$. The projected motion capture poses are denoted $\tilde{\mathbf{x}}_{2D}$. We achieve our goal in two steps: first, modify the projected motion capture poses $\tilde{\mathbf{x}}_{2D}$ (Figure 5), and then, reconstruct the 3D marker positions from the modified projections (Figure 6).

We divide the character model into 'upper body', which consists of the hierarchical chains containing the two arms and the head, and 'lower body', which consists of the joint chains involving the legs. We make this distinction because limbs that are in contact with the ground need to be handled differently from limbs that are not in contact.

We start at the pelvis and work our way through each hierarchical chain of the upper body. Each limb segment of the projected motion capture pose is rotated in the image plane so that the in-plane polar angle is the same as the desired pose descriptor, that is, the corresponding polar angle in the hand drawn pose (Figure 5). The lower body is not modified in this step.

The modified pose is denoted $\mathbf{x}_{2D}$. Through this modification step, we attempt to extract the pose of the hand-drawn character while filtering out much of the noise and imprecision.

3.5. 3D Marker Reconstruction

Given the 2D pose $\mathbf{x}_{2D}$, we infer the missing third dimension. We find 3D virtual marker positions $\mathbf{X}^*_{3D}$ such that

$$\mathbf{X}^*_{3D} = \underset{\mathbf{X}_{3D}}{\operatorname{argmin}}(e_p + \lambda_1 e_l + \lambda_2 e_r) \tag{5}$$

where e_p is the projection error, e_l is the error in limb lengths, e_r is the motion capture regularization term, and λ_1 and λ_2 are the associated weights. Increasing λ_1 permits the limb lengths to change more (accounting for small amounts of squash and stretch in the artist drawn pose), and increasing λ_2 biases the 3D reconstruction towards the motion capture pose.

The first term e_p in the error function minimizes the sum of the squared distance between the projection of $\mathbf{X}_{3D}$ and the 2D marker positions that describe the pose $\mathbf{x}_{2D}$,

$$e_p = \|\mathbf{R}\mathbf{X}_{3D} - \mathbf{x}_{2D}\|. \tag{6}$$

Equation 6 is underconstrained because there are $3N$ unknowns and $2N$ constraints. Because we assume that the individual limbs of the 3D character model are rigid bodies, we enforce constant limb lengths through the second error term e_l. The positions of the two virtual markers on the ends of a given limb can be denoted by (x_i, y_i, z_i) and (x_j, y_j, z_j). Then, the length of the limb is

$$(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 = L^2 \tag{7}$$

where L is the length of the given limb. We can linearize Equation 7 using a Taylor series expansion around the best match motion capture pose and stacking the length equations for each limb to yield

$$e_l = \|\mathbf{A}_{limb}\mathbf{X}_{3D} - \mathbf{b}_{limb}\|. \tag{8}$$

where $\mathbf{A}_{limb}$ and $\mathbf{b}_{limb}$ are functions of the retimed motion capture pose and L.

In Section 3.4, we discussed how to modify the projection of the best match motion capture pose $\mathbf{X}_{3D}$. We use this 3D pose as a prior for the reconstruction from the modified projection $\mathbf{x}_{2D}$. The regularization term is

$$e_r = \|\mathbf{X}_{3D} - \mathbf{X}_{3Dmocap}\|. \tag{9}$$

The final objective function (Equation 5) can be rewritten in terms of these three error terms.

$$\mathbf{X}^*_{3D} = \underset{\mathbf{X}_{3D}}{\operatorname{argmin}} \|\mathbf{R}\mathbf{X}_{3D} - \mathbf{x}_{2D}\| + \lambda_1 \|\mathbf{A}_{limb}\mathbf{X}_{3D} - \mathbf{b}_{limb}\| + \lambda_2 \|\mathbf{X}_{3D} - \mathbf{X}_{3Dmocap}\|$$

We can formulate a linear least squares system to estimate the optimal $\mathbf{X}_{3D}$ that minimizes this objective function, in closed form,

$$\mathbf{W} \begin{bmatrix} \mathbf{R} \\ \mathbf{A}_{limb} \\ \mathbf{I} \end{bmatrix} \mathbf{X}_{3D} = \begin{bmatrix} \mathbf{x}_{2D} \\ \mathbf{b}_{limb} \\ \mathbf{X}_{3Dmocap} \end{bmatrix} \tag{10}$$

$$\mathbf{W}\mathbf{A}_{full}\mathbf{X}_{3D} = \mathbf{b}_{full}. \tag{11}$$

where $\mathbf{W}$ contains the weights of the various error terms.

The set of solutions $\mathbf{X}_{3D}$ lies in a low dimensional subspace of the set of all possible poses [SHP04]. Each pose

can be modeled as a linear combination of basis vectors,

$$\mathbf{X}_{3D} = \mu + \sum_{i=1}^{P} b_i c_i \quad (12)$$

$$= \mu + \mathbf{C}\mathbf{X}_b \quad (13)$$

where c_i is a basis vector, b_i is the corresponding weight, $\mathbf{C}$ is a $3N \times P$ matrix of basis vectors, $\mathbf{X}_b$ is a vector of weights for each of the P basis vectors, and μ is the mean. Principal Component Analysis finds the set of basis vectors that maximize variance in a lower dimensional representation. We perform PCA on an activity specific database to find the set of basis vectors for each example motion separately.

The weighted least squares system in Equation 10 can then be written as

$$\mathbf{W}\mathbf{A}_{full}(\mathbf{C}\mathbf{X}_b + \mu) = \mathbf{b}_{full}, \quad (14)$$

$$\mathbf{W}\mathbf{A}_{full}\mathbf{C}\mathbf{X}_b = \mathbf{b}_{full} - \mathbf{A}_{full}\mu, \quad (15)$$

$$\mathbf{W}\mathbf{A}\mathbf{X}_b = \mathbf{b} \quad (16)$$

We can find the least squares solution to Equation 16 and reproject $\mathbf{X}_b$ to get the 3D marker positions $\mathbf{X}_{3D}$. As this is a linear system, the solution is the global minimum, is numerically stable, and can be found in closed form.

3.6. Computation of Joint Angles: Roll, Pitch, Yaw

Every limb is described by the three joint angles (roll, pitch and yaw) relative to its parent limb. Because the root joint for our character model is the pelvis, we start by recovering the rotation of the pelvis with respect to the world coordinate frame via procrustes analysis (details are provided as supplementary material) [JCG04] and work our way down each hierarchical chain to generate the full pose $\mathbf{X}_{ja}$.

The marker positions $\mathbf{X}_{3D}$ give us two out of the three joint angles for a limb segment, roll and pitch. We compute yaw rotation from motion capture data.
The joint angles for our skeletal model (Section 3.1) describe the rotation of the limb segment in the *xyz* ordering—when we convert this description to the *zyx* ordering, θ_x and θ_y are functions of 3D marker positions (roll and pitch), and θ_z is the 'yaw' angle, which can not be computed from marker positions. We look up the rotation of the corresponding limb segment in the motion capture pose, and simply use that θ_z to complete the generated 3D pose.

4. Results

We demonstrate our method on four hand animations by two artists. The examples include ballet, a stylized sneaky walk, a jumping jacks sequence, and a 'happy flower' sequence. The ballet dancer is drawn to be a fully fleshed character, while the rest of the characters are drawn as stick figures. All our results are best seen as video.

Figures 7 and 9 are frames taken from the videos for the ballet sequence and the jumping jack sequence. The top row contains frames from the hand animated sequence, the middle row is the 3D animation generated by our method, and the bottom row is the time aligned motion capture segment. The skirt and ponytail on the ballet dancer, and the hair on the jumping cartoon were separately simulated. In Figure 7, we can see the head move from side to side in the 3D animation (as in the hand animation). We found that it was quite difficult for a human subject to perform a vigorous action like jumping jacks while bobbing their head. As a result, our motion capture database does not contain any jumping jacks sequence with significant side-to-side head movement. A comparison of the character's arms in the frames in Figure 7 also illustrates how our algorithm modifies the motion capture segment to better match the hand-drawn poses. In Figure 9, we see how the 3D animation generated by our method matches the ballet animation. Figure 8 shows sample frames from the 'happy flower' and 'sneaky walk' sequences. The feet of the flower are clamped to the ground as they were in the hand-drawn animation.

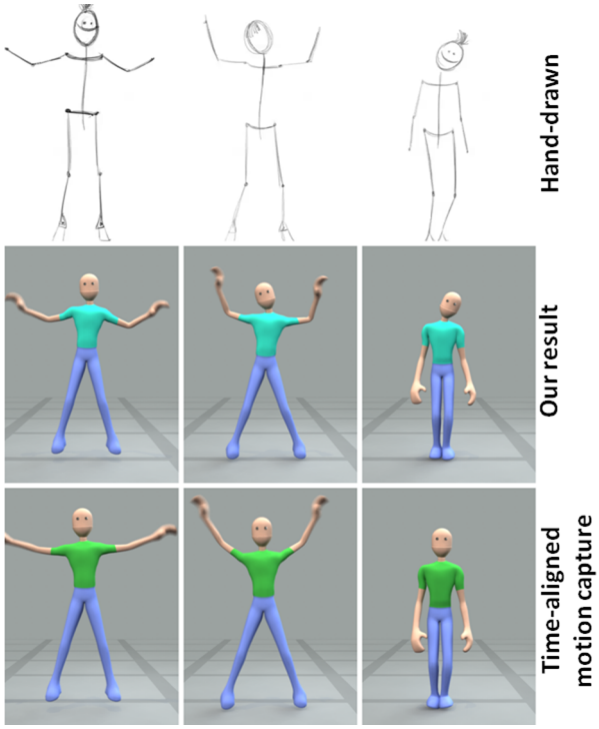

Figure 7: *Note that the head bobs from side to side in the 3D animation generated by our method, and the arms match the hand animation much more closely than the motion captured poses.*

The free parameters in our method are the weights of limb length constraints λ_1 and regularization λ_2, the number of principal components P and the length of the smoothing window w used to smooth the input markers $\tilde{\mathbf{x}}_h$. In practice, $\lambda_1 = 0.001$ and $\lambda_2 = 0.1$ work well for almost all the

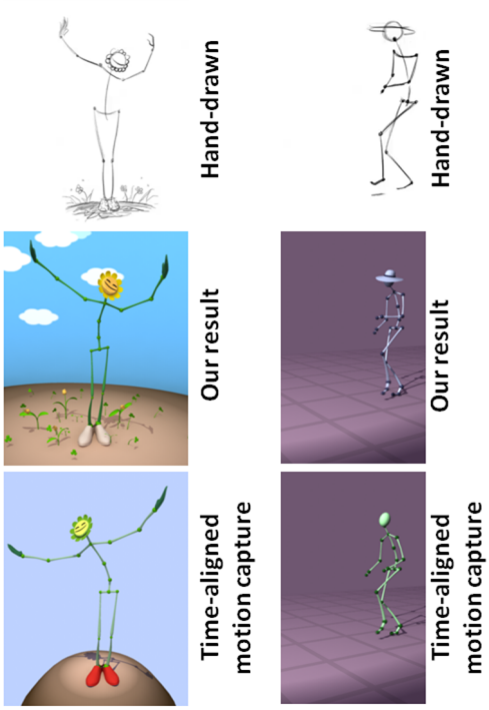

Figure 8: *Results on a happy flower and a stylized walk.*

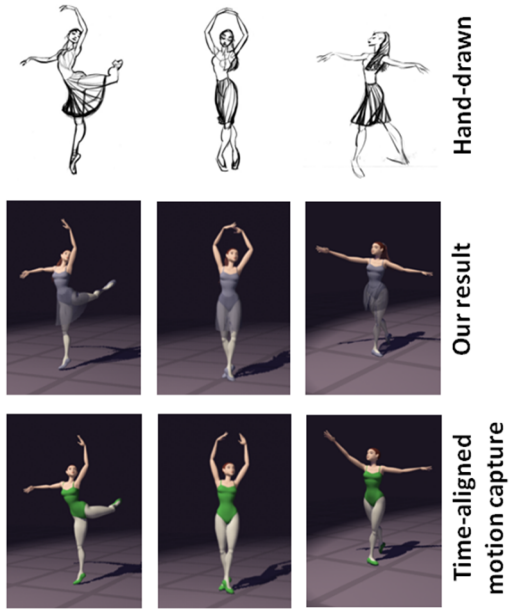

Figure 9: *The arms and torso better match the hand animation compared to the motion capture data.*

Table 1: *Reconstruction error when camera parameters are varied.*

Δ_θ (degrees)	Δ_ϕ (degrees)	mean rms error (meters)	max rms error (meters)
0	0	0.0045	0.0173
5	0	0.0046	0.0173
-5	0	0.0045	0.0174
10	0	0.0048	0.0179
-10	0	0.0047	0.0175
0	10	0.0050	0.0169
0	-10	0.0051	0.0180

Baseline is the error between ground truth marker positions and time aligned motion capture marker positions : mean rms error = 0.0062, max rms error = 0.0257

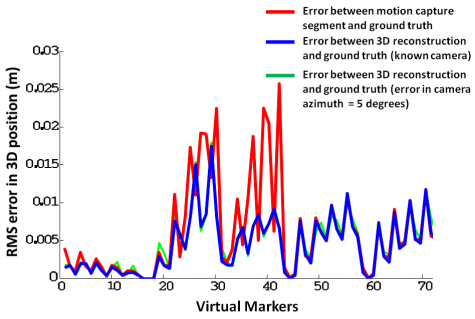

Figure 10: *Error in 3D marker position.*

results we generated. The pirouette sequence in the ballet example required higher regularization $\lambda_2 = 1$ because 2D information without any temporal coherence is not sufficient to disambiguate whether the arms are in front of the torso or behind it. We chose $P = 20$ for all the examples.

We also evaluate our method quantitatively by projecting a motion capture walk sequence to 2D using a known camera. The projected poses are the target poses ($\tilde{\mathbf{x}}_h$) for the evaluation. The results are graphed in Figure 10. The ticks on the x-axis represent the 3D positions of the virtual markers i.e. the elements of the vector $\mathbf{X}^*_{3D}$. Ticks 1 through 42 represent markers on the 'upper body', that is, the limb segments that were modified to match the pose descriptor of the hand-drawn pose. The red curve is the RMS error between the 3D marker positions of the best match motion capture segment and the ground truth marker positions, for a 100 frame segment. The blue curve is the RMS error between the 3D marker positions generated by our method (called 3D reconstruction), and the ground truth marker positions.

We also investigate how the error in specifying $\mathbf{R}$ affects reconstruction error. Because we assume a simple orthographic camera model, the camera can be parametrized by the azimuth and elevation angles. The green curve in Figure 10 plots the RMS error in marker position (over all N markers) when the azimuth is off by 5°. In Table 1, we report the mean and maximum RMS error in 3D marker positions

when the azimuth and elevation are off by $\pm 5^\circ$ and $\pm 10^\circ$. The baseline is the mean and maximum RMS error between the marker positions of the best match motion capture segment and ground truth (the red curve in Figure 10). For errors up to $\pm 10^\circ$, the error in the marker positions generated by our method is less than the baseline. This demonstrates that the pose descriptor is robust enough to tolerate a small mismatch between the user-specified camera and the artist's imaginary camera.

Our method takes approximately 2 minutes to generate joint angles for a 200 frame input sequence—it takes around 7 seconds to compute the pose descriptors for all the frames, and around 1 second to compute the 3D reconstruction using the least squares minimization. Most of the 2 minutes are spent in time alignment and in computing joint angles from marker positions.

5. Conclusion and Discussion

We have demonstrated a method that generates 3D animation of human motion from traditional hand-drawn frames. Our key insight is that domain information can help resolve ambiguities and handle the noise that is inherent in 2D hand drawings. Our approach extracts the style elements of the artist's drawings and transfers them to 3D. We use motion capture data as a starting point, and describe a pose descriptor to quantify the difference between a motion captured pose and a hand-drawn pose. By modifying the projected motion capture pose in 2D, and then lifting the modified pose to 3D, we create new 3D animation.

An assumption made early on in our method is that model changes such as the lengthening of limbs, or the curving of bones, are 'noise', not input data. The technique of "squash and stretch" is used to great effect by hand animators; our method generates the 3D poses for a rigid skeleton and does not address the significant changes in shape seen in traditional squash and stretch. A natural next step would be to incorporate this element into the 3D animations we generate. There can be many ways to compare human poses, for example, image measurements such as moments [Bra99], various filter responses [SB01], chamfer distances [BSB*07], or the basic bit-xor [CTMS03], all of which are global descriptors. Histograms of shape contexts [AT04, AT06] are comparatively 'quasi-local' in that they encode silhouette shape over angular and radial bins centered at regularly spaced points on the silhouette. These descriptors could be useful if we were comparing changes in the silhouette shapes of individual body parts and not only the orientations of the bones.

One limitation of this approach is that we are dependent in a few essential ways on the motion capture segment that we start with. Because foot plants are picked from the motion capture data, we cannot change the given motion capture walk to a wider stance walk in our results. Also, the yaw rotation for every limb is computed from motion capture data. So, if the database contains a hand wave with the palm facing forward, we cannot change that to have the palm face backward. Despite this, motion capture data provides a good starting point because it is a sequence of smooth, natural and coherent 3D poses. This sequence is not the 3D animation we desire because it is in the style of the actor who was captured, and the motion is fundamentally limited by the laws of physics. Animation, on the other hand, involves conveying the unique style of the character and the 'plausibly impossible' in the context of physics. Our method uses motion capture data as a starting point to create the desired 3D sequence— one which contains the same set of moves, but in the style and personality of the animated character. It would be interesting to think about how to extend this approach for motions that cannot be easily captured in a motion capture studio.

Another limitation is that the pose descriptor can only handle in-the-image-plane mismatch in limb angles—we have not included foreshortening in our definition. This limitation is perhaps not so crucial because the animator picks camera angles that show the motion off well as part of the animation process. We choose this pose descriptor over the sum of squared distances between virtual marker positions [HLF99, DCR01, Tay00] as it is a local descriptor and can be used to capture subtle differences in the angular orientation of individual body parts. Also, the properties of translation and scale invariance are well suited for comparing motion captured poses and hand-drawn poses because the actor and the character could occupy different parts of the image, and may have different body proportions.

The third limitation of our method is that we use a simple orthographic camera model instead of the general perspective form. This choice allows us to formulate the 3D reconstruction problem as a linear system. We can then solve the minimization in a single step by finding the least squares solution. This formulation influenced many of our design decisions, for instance, representing pose as the set of 3D marker positions instead of hierarchical joint angles.

Our work is an important step in leveraging the unique talent of traditional hand animators to create animations in the 3D medium. Many interesting interactions are possible in this scenario. For example, one can physically simulate the ballet dancer's skirt once we have the dancer's motion in 3D, and composite it onto the hand animation to reduce the burden on the hand animator. Or, allow a hand-drawn character to interact with physical simulations in the 3D environment, batting a ball, for example.

Acknowledgments

Thanks to Glen Keane and Tom LaBaff for providing the input animations, and to Justin Macey and Moshe Mahler for help with motion capture and video. Thanks also to Autodesk for their donation of the 3D animation and rendering package Maya.

References

[AT04] AGARWAL A., TRIGGS B.: Learning to track 3d human motion from silhouettes. In *ICML '04: 21st International Conference on Machine learning* (2004), ACM, p. 2.

[AT06] AGARWAL A., TRIGGS B.: Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 1 (2006), 44–58.

[Bab09] BABBITT A.: "...animation follows the laws of physics - unless it is funnier otherwise.". Wikipedia, May 2009.

[BLCD02] BREGLER C., LOEB L., CHUANG E., DESHPANDE H.: Turning to the masters: Motion capturing cartoons. *ACM Transactions on Graphics 21*, 3 (July 2002), 399–407.

[BM98] BREGLER C., MALIK J.: Tracking people with twists and exponential maps. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 0* (1998), 8.

[BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 4 (2002), 509–522.

[Bra99] BRAND M.: Shadow puppetry. In *ICCV '99:International Conference on Computer Vision* (1999), IEEE Computer Society, p. 1237.

[BSB*07] BALAN A., SIGAL L., BLACK M., DAVIS J., HAUSSECKER H.: Detailed human shape and pose from images. In *2007 Conference on Computer Vision and Pattern Recognition (CVPR 2007)* (June 2007), pp. 1–8.

[CTMS03] CARRANZA J., THEOBALT C., MAGNOR M. A., SEIDEL H.-P.: Free-viewpoint video of human actors. *ACM Transactions on Graphics 22*, 3 (July 2003).

[DAC*03] DAVIS J., AGRAWALA M., CHUANG E., POPOVIĆ Z., SALESIN D. H.: A sketching interface for articulated figure animation. In *SCA '03:ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 320–328.

[DCR01] DIFRANCO D. E., CHAM T.-J., REHG J. M.: Reconstruction of 3-d figure motion from 2-d correspondences. In *2001 Conference on Computer Vision and Pattern Recognition (CVPR 2001)* (Dec. 2001), pp. 307–314.

[Ell03] ELLIS D.: Dynamic time warp (dtw) in matlab. Web resource, available: http://www.ee.columbia.edu/ dpwe/resources/matlab/dtw/, 2003.

[Gav99] GAVRILA D. M.: The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU 73*, 1 (1999), 82–98.

[HLF99] HOWE N. R., LEVENTON M. E., FREEMAN W. T.: Bayesian reconstruction of 3d human motion from single-camera video. In *Neural Information Processing Systems: NIPS* (1999), pp. 820–826.

[JCG04] JOHN C. GOWER G. B. D.: *Procrustes Problems*. Oxford University Press, USA, 2004.

[JT95] JOHNSTON O., THOMAS F.: *The Illusion of Life: Disney Animation*. Disney Editions; Rev Sub edition, 1995.

[Kan81] KANADE T.: Recovery of the three-dimensional shape of an object from a single view. *Artif. Intell. 17*, 1-3 (1981), 409–460.

[Las94] LASSETER J.: Tricks to animating characters with a computer. In *ACM SIGGRAPH Course Notes* (1994).

[LC85] LEE H.-J., CHEN Z.: Determination of 3d human body postures from a single view. *Computer Vision, Graphics, and Image Processing 30*, 2 (1985), 148 – 168.

[LGXS03] LI Y., GLEICHER M., XU Y.-Q., SHUM H.-Y.: Stylizing motion with drawings. In *SCA '03:ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), pp. 309–319.

[McC14] MCCAY W.: Gertie the dinosaur, 1914.

[MG01] MOESLUND T. B., GRANUM E.: A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU 81*, 3 (2001), 231–268.

[SB01] SIDENBLADH H., BLACK M. J.: Learning image statistics for bayesian tracking. In *ICCV '01:International Conference on Computer Vision* (2001), pp. 709–716.

[SB03] SIDENBLADH H., BLACK M. J.: Learning the statistics of people in images and video. *International Journal of Computer Vision 54*, 1-3 (2003), 181–207.

[SBF00] SIDENBLADH H., BLACK M. J., FLEET D. J.: Stochastic tracking of 3d human figures using 2d image motion. In *ECCV '00:6th European Conference on Computer Vision-Part II* (2000), Springer-Verlag, pp. 702–718.

[SBS02] SIDENBLADH H., BLACK M. J., SIGAL L.: Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV '02:7th European Conference on Computer Vision-Part I* (2002), Springer-Verlag, pp. 784–800.

[SC90] SAKOE H., CHIBA S.: Dynamic programming algorithm optimization for spoken word recognition. *Readings in speech recognition* (1990), 159–165.

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics 23*, 3 (Aug. 2004).

[Smi07] SMINCHISESCU C.: *Learning and Inference Algorithms for Monocular Perception. Applications to Visual Object Detection, Localization and Time Series Mod-*

els for 3D Human Motion Understanding. PhD thesis, University of Bonn, Faculty of Mathematics and Natural Sciences, 2007.

[ST03] SMINCHISESCU C., TRIGGS B.: Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research 22*, 6 (2003), 371–393.

[Tay00] TAYLOR C. J.: Reconstruction of articulated objects from point correspondences in a single image. In *2000 Conference on Computer Vision and Pattern Recognition (CVPR 2000)* (June 2000), pp. 677–685.

[TBvdP04] THORNE M., BURKE D., VAN DE PANNE M.: Motion doodles: an interface for sketching character motion. *ACM Transactions on Graphics 23*, 3 (Aug. 2004), 424–431.

[UFF06] URTASUN R., FLEET D. J., FUA P.: Temporal motion models for monocular and multiview 3d human body tracking. *Computer Vision and Image Understanding 104*, 2 (2006), 157–177.

[Wik09] WIKIPEDIA: List of animated feature-length films, 2009.