Modeling Relaxed Hand Shape for Character Animation

Michael Neff and Hans-Peter Seidel

MPI Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany,
{neff|hpseidel}@mpi-inf.mpg.de,
WWW home page: http://www.mpi-inf.mpg.de/~neff

Abstract. We present a technique for modeling the deformations that occur to hand pose under the influence of gravity when the hand is kept in a relaxed state. A dynamic model of the hand is built using Proportional-Derivative controllers as a first order approximation to muscles. A process for tuning the model to match the relaxed hand shape of subjects is discussed. Once the model is tuned, it can be used to sample the space of all possible arm orientations and samples of wrist and finger angles are taken. From these samples, a kinematic model of passive hand deformation is built. Either the tuned dynamic model or the kinematic model can be used to generate final animations. These techniques increase the realism of gesture animation, where the character often maintains a relaxed hand.

1 Introduction

People will often allow their hands and wrists to relax without exerting active control. When this occurs, the angles of the wrist and fingers will vary due to the influence of gravity as the arm moves. These subtle variations add important realism to an animated character. This is particularly important in gesture animation, where sometimes the hand is actively adjusted to a particular, meaningful pose while other times it is simply left relaxed as the arm moves.

In this work, we model this relaxed, passive variation in wrist and finger angles. Our approach is to first build and tune a dynamic model of the hand. This model is then used to automatically generate a large quantity of sample data, from which a kinematic model is built. Either the tuned dynamic model or the generated kinematic model can be used in the generation of final animations, depending on the requirements of the application.

This work makes the following contributions:

- A simple, low-cost tuning method that offers sufficient accuracy for generating natural animations,
- an approach for building a kinematic model from dynamic simulation,
- a model of relaxed hand shape that can improve the realism of animations.

2 Previous Work

The human hand has been a significant focus of research in the computer graphics community, but to our knowledge, no one has focused on passive hand shape nor used a dynamic model to build a kinematic hand model.

Much graphics hand research has focused on the problem of grasping. Rijpkema and Girard present a knowledge-based approach to grasp planning [1]. Sanso and Thalmann present a system that decides on an appropriate grasp type for a given task and uses forward and inverse kinematics (IK) to solve for hand shape [2]. Pollard and Zordan use motion capture data in building a physics-based controller for grasping [3]. In a related problem, ElKoura and Singh model finger coordination for guitar playing, using motion capture data to model joint correlations and introducing a multiple kinematic chain IK routine [4].

Physics-based simulation has been a recent research focus. Sibille et al. model bone movement and soft tissue deformation. They represent muscles as angular springs and minimize potential energy in order to solve for the joint equilibrium points [5]. Albrecht et al. present a system in which *pseudo* muscles are used to move bones and *geometric* muscles are used to deform skin tissue [6]. Tsang et al. present a physical simulation system in which the activation of individual muscles can be calculated and muscle bundles can be visually separated for anatomical study [7]. Pollard and Zordan employ a Proportional Derivative controller system, similar to the one employed here [3].

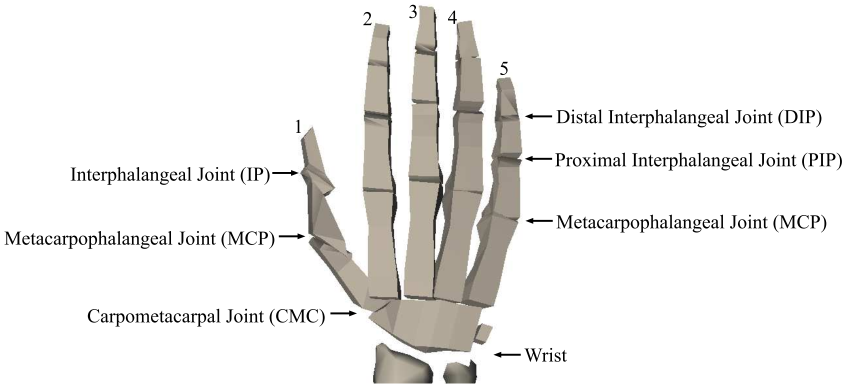
Research has also focused on improving other aspects of hand models. Lin et al. model constraints on hand motion, including joint limits and inter-joint constraints, such as the DIP angle being 2/3 the PIP angle (see Figure 1 for acronyms)[8]. Braido and Zhang conducted an experimental study to measure joint coordination patterns of fingers during grasping and flexion of individual fingers [9]. We do not explicitly model joint correlations. McDonald et al. focus on improving the articulation of hand models and apply their work to animating American Sign Language [10]. Kry et al. present a method for compactly representing skin deformations that can then be animated using graphics hardware [11]. Kurihara and Miyata use CT scans to generate a high quality model of skin deformations [12].

The idea of using physical simulation to generate a kinematic model has been applied previously by Yu and Terzopoulos who created a kinematic model of biomechanically based fish motion from a spring-mass-damper model of the fish [13].

Two previous works are particularly relevant to this paper. Zordan and Pollard [3] present a similar physical hand model, but whereas they actively compensate for torques induced by gravity, we make use of these torques to deform our model. Neff and Fiume [14] model tension and relaxation, including gravity based deformations, also using a PD-based control strategy, but they do not explicitly deal with hands, present a tuning methodology or generate a kinematic model from their simulations.

3 Dynamic Model of the Hand

We use a rigid-body hand model consisting of 23 Degrees of Freedom (DOFs), as shown in Figure 1. Each of the PIP and DIP joints of the fingers and thumb IP and MCP have one DOF for flexion. The MCP joints of the fingers each have an additional DOF to support abduction/adduction (spreading of the fingers). The CMC joint of the thumb has three DOFs. The wrist has two DOFs allowing the hand to be moved up and down and side-to-side. Axial rotation of the whole hand is accomplished by a rotational DOF associated with the forearm, which is not part of this model.



1
2
3
4
5
Distal Interphalangeal Joint (DIP)
Proximal Interphalangeal Joint (PIP)
Interphalangeal Joint (IP)
Metacarpophalangeal Joint (MCP)
Metacarpophalangeal Joint (MCP)
Carpometacarpal Joint (CMC)
Wrist

Fig. 1. Hand model illustrating the joints and finger numbering used in our system.

We use a forward-simulation approach to dynamics, in which at each time step, torques are applied to each DOF and the resulting accelerations are twice integrated to update the hand's position. Our torques are generated by a simple Proportional-Derivative (PD) controller, common in the literature. PD control can be thought of as a spring and damper arranged in parallel and the control law is given as:

$$\tau = k_s(\theta_{desired} - \theta) - k_d\dot{\theta}, \tag{1}$$

where τ is the resulting torque, θ is the current angle of the DOF, $\theta_{desired}$ is the desired angle of the DOF, $\dot{\theta}$ is the velocity of the DOF, and k_s and k_d are the spring and damper gains respectively. Essentially, the controller generates torque in order to minimize the error between the desired value and the actual value of the DOF. Note though, that even at steady-state, there will normally be some error between the actual and desired value of the DOF in the presence of external forces such as gravity. Intuition for this can be developed by considering a mass hanging from a spring. At steady-state, the spring will deviate from its rest length due to the force of gravity pulling on the mass. Other approaches compensate for this error [14, 3], but exploiting this "error" to generate natural hand deformations is the central idea of this work. We use SD/Fast [15] to generate the equations of motion for our simulation.

Joint limits proved important to prevent unnatural looking backwards bend of the fingers when the palm was facing up. Limits were maintained by adding additional limit terms to the PD-controller, a technique similar to that used in [3]:

$$\tau = k_s(\theta_{desired} - \theta) + c_0 k_{lim}(\theta_{low} - \theta) + c_1 k_{lim}(\theta_{high} - \theta) - k_d\dot{\theta}, \tag{2}$$

where k_{lim} is the limit gain, many times higher than the normal gain, θ_{low} and θ_{high} are the low and high limits respectively and

$$c_0 = \begin{cases} 1 \text{ if } \theta_{low} - \theta > 0 \\ 0 \text{ otherwise} \end{cases} \quad c_1 = \begin{cases} 1 \text{ if } \theta_{high} - \theta < 0 \\ 0 \text{ otherwise .} \end{cases} \tag{3}$$

3.1 Tuning the Model

A proportional-derivative controller has three free parameters: the proportional gain, or stiffness k_s; the damping gain k_d; and the set point $\theta_{desired}$. We tune each of these parameters to match observed passive hand deformations. The set point is used to define the desired rest posture of the hand. This is the pose the hand would assume without the influence of gravity. The proportional gains are adjusted to determine how far the joints move from this rest pose under the influence of gravity. The damping gain is tuned to control the duration of relaxation movements under the influence of gravity. Each of these tuning processes will be explained in detail.

The rest pose of the hand will vary from subject to subject and is likely a parameter animators will want to control. Mount et al. [16] measured the neutral postures of astronauts in space, giving an indication of rest pose in the absence of gravity. They found flexion varied from 21 to 60 degrees across six subjects, further indicating the potential need for customization. We apply a simple measurement approach to a test subject in order to determine a sample rest pose. We ask the subject to relax his hand and hold it so that the plane of finger flexion is horizontal, thus minimizing deflection due to gravity. We photograph the hand and estimate joint angles from this. The angles used in our experiment are included in the appendix and these define the $\theta_{desired}$ parameters for each of the PD controllers. This process can be repeated with different values to model different hand behaviour, as desired by the animator.

The gain values, k_s, are determined so that the hand will have a desired shape under the influence of gravity when in a particular orientation. At steady state, the PD control law can be rewritten as:

$$k_s = \frac{T}{(\theta_{desired} - \theta)} \tag{4}$$

where $\theta_{desired}$ is the rest pose defined above and T is the torque acting on the joint due to the force of gravity pulling on the limb and each limb lower down in the kinematic chain. For a given pose, T can be calculated, so if we can define the vector of θ values that define an observed pose, we can solve for k_s.

The hand orientation in which we define this vector of θ values must have two properties: it must allow the theta values to be easily determined and there must be a significant torque due to gravity acting on the measured joints in order to obtain meaningful k_s values. We use three different orientations. For the fingers, we hang the hand straight down, allowing gravity to partially straighten the fingers. For the wrist forward rotation, we hold the forearm horizontal with the palm down and allow the hand to sag downwards. This is also used for the downward rotation of the thumb CMC joint. Similarly, we simply rotate the palm to vertical with the arm horizontal for the sideways wrist rotation and the remaining thumb DOFs. The angles in these poses are once again determined by photographs and measurements of our test subject and are summarized in the appendix. The MCP DOFs related to abduction/adduction were not tuned as their deformations were considered to be too small to be worth modeling.

The third free parameter, k_d, is used to control the timing of motion under the influence of gravity. Like moving in molasses, as k_d increases, it will slow the movement. k_d values that are too small will allow the joints to move too quickly and oscillate too much. We define k_d as a factor of the k_s value for the corresponding joint and solve for three different factors: one for each wrist DOF, and one for the thumb and fingers. These factors are determined by matching animation timing to video of the subject. For the two wrist orientations, we ask the subject to hold his wrist straight and then instantly release tension so that the wrist falls to its relaxed orientation, as per above. For the fingers, we ask him to curl them into a loose fist, and then instantly release tension so that again, they relax to the default orientations above. These tests were performed multiple times and ten samples of each were chosen in which the movement appeared natural. The mean times for each movement are summarized in Table 1. Since it is difficult to precisely judge the start and stop frame of a motion, there will be some error in these values. Precise measurements do not appear to be needed, however, for the animation application. The damping factors were calculated by recreating the scenarios under simulation and adjusting the damping factor to match the original timing. The three damping factors ranged from five to seven ($k_d \approx k_s * 5$ for our low proportional gains).

Body Part	Time to Relaxed Pose	Standard Deviation
Wrist Y	0.48	0.06
Wrist Z	0.41	0.06
Fingers	0.38	0.13

Table 1. Mean relaxation times.

4 Building a Kinematic Model

The goal of our kinematic model is to automatically set the values for the wrist and finger DOFs based on the world orientation of the forearm. The model is built by using dynamic simulation to sample the space of possible forearm orientations. We only attempt to capture the static pose deformation caused by gravity, not dynamic deformations resulting from inertial effects.

The orientation of the forearm in world-space can be captured by two parameters, which we refer to as *inclination* and *rotation*. Inclination is the angle of the arm relative to the horizontal plane; essentially its latitude on a sphere. Rotation refers to the amount of twist around the axis of the arm. Note that the rotation of the arm around the vertical axis can be ignored as the effect of gravity will not change under this transformation.

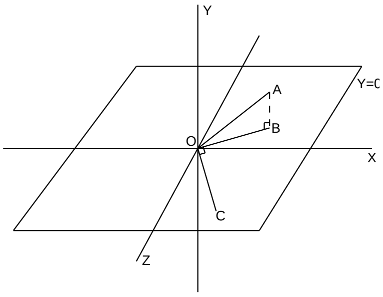


Fig. 2. Vector $\vec{OA}$ represents the arm orientation. The angle between $\vec{OA}$ and $\vec{OB}$ defines *inclination*. The cross product of $\vec{OA}$ and $\vec{OC}$ is used in determing arm *rotation*.

The process for calculating the two arm values is shown in Figure 2. The vector $\vec{OA}$ defines the current forearm orientation in world coordinates. This vector is projected onto the plane $Y = 0$ to yield the vector $\vec{OB}$. The angle between these vectors is the *inclination*. A vector $\vec{OC}$ is calculated which is perpendicular to $\vec{OB}$ and lies in the $Y = 0$ plane. Taking the cross-product of $\vec{OA}$ and $\vec{OC}$ yields a vector $\vec{OD}$ that is perpendicular to $\vec{OA}$ and lies in the plane containing the points O, A, B. We define a vector in local forearm coordinates that is perpendicular to the forearm and points straight up in the forearm local frame. This up direction is then converted to world coordinates. The angle between this vector and $\vec{OD}$ defines the *rotation* of the arm.

4.1 Sampling Process

For ease of look up, we would like to store our data points on a grid (2D array) where the coordinates of the grid correspond to arm inclination and rotation. This is facilitated by a sampling process that computes one column of this grid on each iteration. We begin with the arm hanging down, held back behind the character and bring it forward and up, bending the elbow, until the hand is about head height and the forearm is again leaning backwards. This gives a complete sampling of the inclinations of interest. This movement is done over 8s to minimize inertial effects. Each iteration is completed with a specified world space arm rotation. The actual forearm axial rotation is varied at each time step in order to achieve this desired orientation. We have tried spacing the rotation sampling by two and six degrees, with no noticeable reduction in visual quality in the reconstructed motion. Note that since we are using PD controllers with reasonable stiffness to control the arm orientation, there will be some error between the actual orientation and the desired orientation. This error is less than our sampling spacing and we simply store the actual orientation and use this in our reconstruction. Each sample consists of the actual inclination and rotation along with values for the 23 DOFs defining the hand and wrist pose. The sample file used to build our model is available online via the first author's home page. It would be difficult, if not impossible, to complete such a controlled sampling procedure using motion capture and a human actor.

4.2 Reconstruction

Once the data has been collected, there are several ways in which it can be used for kinematic animation. The data is quite smooth, so 2D functions could be fit to each DOF to limit storage or the data could be queried for hand poses at pre-set keyframes. We use a very simple reconstruction process: At each time step, we calculate the current orientation of the arm. We then use this value as a pointer into our samples to determine the values for the 23 DOFs associated with the hand.

As stated above, the samples are stored in a grid indexed by inclination and orientation values. For a forearm orientation input, we find the four surrounding grid points. Bilinear interpolation of these vectors is performed to determine the final DOF values. Note that the interpolation is based on the actual inclination and rotation values stored with each sample, not their grid indices. This process is very fast and can be used in interactive applications where the hand pose is automatically updated as the animator moves the character's arm.

5 Results and Conclusion

Figure 3 shows frames from a kinematic animation sequence without and with the relaxed hand model applied. The more natural hand and wrist posture with the relaxed hand model is clearly evident. The video accompanying the submission includes this sequence and also a dynamic sequence using the tuned values for our model developed with the above procedure. The dynamic sequence shows subtle inertial effects that are missing in the kinematic sequence, however, when the hand is brought up to head height at a reasonable speed, the fully relaxed wrist appears slightly too loose to provide a realistic motion in this area given the inertial effects. This is not surprising, as it is rare for a person to have their wrist fully relaxed in this position. The relaxed hand model represents a baseline for hand stiffness values and they should be increased for less loose movements.

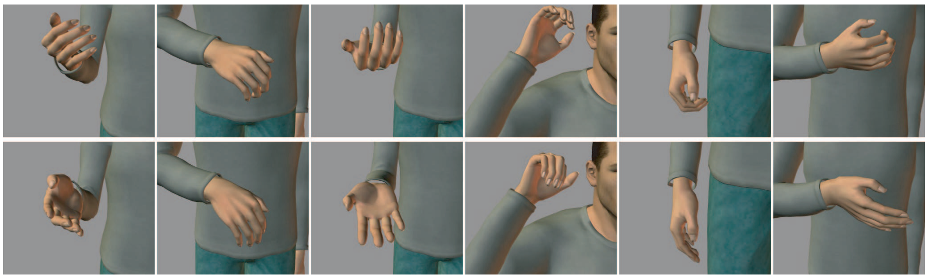


Fig. 3. The top row shows frames from an animation sequence in which the hand is held at its rest pose. The bottom row shows frames from the same kinematic animation sequence, but with our passive hand model applied.

There are other effects that would be interesting to model, in particular, correlations between finger joints and the impact of wrist movement on finger deflection; for instance, a severe downward movement of the wrist will cause some straightening of the fingers as the tendons are stretched over the wrist. It would be possible to model this effect by varying the set point of the controller if a model for the correct amount of variation was available.

In summary, we have presented a simple method for tuning hand parameters that is inexpensive and appears to offer sufficient accuracy for computer animation. We have also built a kinematic model of passive hand shape that provides real-time performance and that significantly improves the naturalness of character animation.

References

1. Rijpkema, H., Girard, M.: Computer animation of knowledge-based human grasping. In: Computer Graphics (Proceedings of SIGGRAPH 91). Volume 25. (1991) 339–348
2. Sanso, R.M., Thalmann, D.: A hand control and automatic grasping system for synthetic actors. Computer Graphics Forum **13**(3) (1994) 167–177
3. Pollard, N.S., Zordan, V.B.: Physically based grasping control from example. In: 2005 ACM SIGGRAPH / Eurographics Symposium on Computer Animation. (2005) 311–318
4. ElKoura, G., Singh, K.: Handrix: animating the human hand. In: 2003 ACM SIGGRAPH / Eurographics Symposium on Computer Animation. (2003) 110–119
5. Sibille, L., Teschner, M., Srivastava, S., Latombe, J.C.: Interactive simulation of the human hand. In: CARS 2002. (2002) 7–12
6. Albrecht, I., Haber, J., Seidel, H.P.: Construction and animation of anatomically based human hand models. In: 2003 ACM SIGGRAPH / Eurographics Symposium on Computer Animation. (2003) 98–109

7. Tsang, W., Singh, K., Fiume, E.: Helping hand: An anatomically accurate inverse dynamics solution for unconstrained hand motion. In: 2005 ACM SIGGRAPH / Eurographics Symposium on Computer Animation. (2005) 319–328
8. Lin, J., Wu, Y., Huang, T.S.: Modeling the constraints of human hand motion. In: Proc. Workshop on Human Motion. (2000) 121–126
9. Peter Braido, X.Z.: Quantitative analysis of finger motion coordination in hand manipulative and gestic acts. ACM Transactions on Graphics **22** (2004) 661–678
10. McDonald, J., Toro, J., Alkoby, K., Berthiaume, A., Carter, R., Chomwong, P., Christopher, J., Davidson, M.J., Furst, J., Konie, B., Lancaster, G., Roychoudhuri, L., Sedgwick, E., Tomuro, N., Wolfe, R.: An improved articulated model of the human hand. The Visual Computer **17**(3) (2001) 158–166
11. Kry, P.G., James, D.L., Pai, D.K.: Eigenskin: Real time large deformation character skinning in hardware. In: ACM SIGGRAPH Symposium on Computer Animation. (2002) 153–160
12. Kurihara, T., Miyata, N.: Modeling deformable human hands from medical images. In: 2004 ACM SIGGRAPH / Eurographics Symposium on Computer Animation. (2004) 355–363
13. Yu, Q., Terzopoulos, D.: Synthetic motion capture: Implementing an interactive virtual marine world. The Visual Computer **15**(7/8) (1999) 377–394
14. Neff, M., Fiume, E.: Modeling tension and relaxation for computer animation. In: ACM SIGGRAPH Symposium on Computer Animation. (2002) 81–88
15. Hollars, M.G., Rosenthal, D.E., Sherman, M.A.: SD/FAST User's Manual. Symbolic Dynamics Inc. (1994)
16. Mount, F.E., Whitmore, M., Stealey, S.L.: Evaluation of neutral body posture on shuttle mission sts-57 (spacehab-1). Technical Report TM-2003-104805, NASA (2003)

Appendix

Finger	DOF	Angle	Deviation	DOF	Finger	Angle	Deviation
1	CMC Abduction	42	-2	1	CMC Flexion	28	+6
1	MCP Flexion	36	+2	1	IP Flexion	24	+2
2	MCP Abduction	1	N/A	2	MCP Flexion	40	-21
2	PIP Flexion	50	-30	2	DIP Flexion	35	-34
3	MCP Abduction	0	N/A	3	MCP Flexion	42	-21
3	PIP Flexion	50	-25	3	DIP Flexion	38	-35
4	MCP Abduction	-1.5	N/A	4	MCP Flexion	42	-21
4	PIP Flexion	50	-25	4	DIP Flexion	38	-35
5	MCP Abduction	-6	N/A	5	MCP Flexion	29	-14
5	PIP Flexion	42	-24	5	DIP Flexion	40	-28
-	Wrist Side	0	+29	-	Wrist Forward	0	+22

Table 2. Angles used to calibrate our dynamic hand model. The "Angle" column defines the rest pose of the hand. "Deviation" refers to the difference between the rest angle and the angle of the DOF that is used to calculate gain values with the specified arm orientation.