Chapter 2

Related Work

2.1 Realism in Computer Animation Techniques

Realistic animation of human figures has been an area of interest of animators and researchers in recent years. Potential applications can be found in entertainment, sports training, medicine, video games and education.

There are several ways to increase the realism of computer-generated animations, and in particular, computer generated postures. One way is geometric, rendering or modeling-wise leading to a natural looking character. Another way is provided by kinematics and dynamics characteristics: collision detection, physical laws. A high-level behavioral approach can also provide realistic attributes to animations, including sensations as happiness, sadness, etc.

This section presents a review concerning the contribution to the realism of several computer animation techniques.

2.1.1 Key-Frame Techniques

Key-frame techniques are the most known techniques to generate animations. A set of static frames is constructed, then, each frame is interpolated and the resulting animation appears dynamically over time. This technique has a low level of abstraction in the sense that the animator has to work directly with joint angles. But specifying animation in terms of joint angles requires a high involvement by the animator. The principal drawback is the tedious task that implies the design of key frames in order to produce the desired effect [Las87]. Therefore, the realism of an animation is usually proportional to the amount of effort spent by the animator.

This technique is the main method used in film production. High frame display rate are needed in order to obtain nice animations. Due to the cost required to generate each key-frame, other techniques as Forward or Inverse Kinematics are used to generate in-betweening frames. Several researches have used cubic splines controlled using dynamic constraints [Pin88] [Ara93] or kinematic constraints [Ste85] [Liu95] in order to have more realistic inbetweenings. Although the key-frames may be biomechanically and physically reasonable, the intermediate frames resulted from interpolation may not be.

In key-frame animations, the animator often develops a fine intuition for the behavior of physical phenomena. Well known effects in key-frame animations are squash, stretch and shrink.

2.1.2 Motion Capture and Motion Editing Techniques

Motion capture techniques are based on the placement of electromagnetic sensors or optical markers on the subject to records its motion over time. The results are realistic since joint values of the subject doing the motion are directly applied to the virtual human.

Motion capture is typically used in video game production. A drawback of this technique, however, is the cost of the hardware required to perform the capture. In addition, captured data need some manipulation to eliminate noise recorded with the signal.

Amaya et al. designed a model to generate “emotional” animation from “neutral” human motion [Ama96]. They used techniques from signal processing. They analyzed and extracted neutral motion and emotional differences, like angry or sad, from motion data. The components extracted were speed and spatial amplitude.

Several researchers have centered their studies in retargeting motion capture data for variable environments and for variable character dimension. Gleicher retargeted motion to new characters using specific features of the original motion as constraints [Gle98]. He utilized a space-time method to eliminate the spatial jerkiness of frame based retargeting techniques by anticipating constraints for the whole motion sequence. This method provides smooth motion data generation.

A transformation on captured data, covering data of emotion and stylistic characteristics of the motion is another interesting research line. Unuma interpolated existing motions using fourier series expansions [Unu95]. He achieved smooth transition between motions. For example, the transition from “a walk” to “a run” was done in a smooth and realistic way. Emotional characteristic of motion were extracted to create “tired” variations of a captured walking motion. However, those methods are based on kinematics and simulate more psychological aspects than physiological assessments of the human body.

Another study proposed to adapt existing motion data to fit arbitrary situations. The best fit motion was selected from a database and adapted for the current motion [Lam96].

Lee modified motions applying constraints [Lee99]. The constraints were satisfied for each frame and the resulting data points were interpolated with a multi-level B-spline fitting technique. It was used to modify walking postures.

A research also used signal-processing techniques on motion capture data for editing a motion style [Bru95]. A filtering algorithm made it possible to separate motion capture data into signals of regularly varying frequencies. For example, jittering motion was separated from steps motion due to their different frequencies.

Komura gave added insight into motion capture and realism proposing a method to use the musculoskeletal system of the human body for editing and retargeting human motion. He used a motion-capture device [Kom01b]. An interesting issue is that motions are applied to different bodies, by changing the size of bones and the volume of muscles. He changed segment parameters using the model defined in [Jen89]. Two kicking motions and a gait motion were first captured and then new motion was obtained by changing muscle parameters and dynamic parameters.

2.1.3 Inverse Kinematics

Techniques using kinematics solutions are Forward Kinematics and Inverse Kinematics. Forward Kinematics allows an animator to define the joint angles for a new posture.

However, it is not easy to estimate the exact joint angles needed to place an articulated structure to a predefined position. Instead of this, using Inverse Kinematics an animator specifies the desired position for an end-effector, and then an algorithm computes the joint angles needed. Inverse Kinematics technique gives a high level of abstraction for specifying postures due to the fact that the user avoids explicit control over all joint values.

The variety of ways in which our limbs move in relation to each other makes it hard to predict the appropriate movements when realistic results are required. Several researchers have worked in order to improve results obtained with this technique. Early studies on Inverse Kinematics were made for manipulators control in the robotics field [Pau81] [Cra82].

Phillip and Badler provided a mechanism for interactively controlling character's postures by specifying various geometric constraints such as end effector position and orientation [Phi91] [Bad93] [Phi90].

Lee designed a system to generate postures depending on the strength model of the figure [Lee90]. A visualization system of human strength was developed. The system displayed joint torque and the force exertable by the end effector by means of colored rods. The strength data were stored in a database, therefore it was not possible to calculate torque or force for arbitrary positions [Wei92].

Zhao looked in depth numeric and optimization techniques for satisfaction of constraints [Zha96b].

A research improved the realism of Inverse Kinematics generated postures controlling the balance of the character (See Figure 2-1). The term Inverse Kinetics was defined to comprise the combination of joint kinematics and mass data [Mas96] [Bou97].

The possibility of integrating different tasks with associated priorities also contributed to improve the natural look of generated postures. An efficient algorithm designed to manage multiple tasks (or constraints) made it possible to manipulate complex characters in an interactive way [Bae01]. In the field of robotics, Siciliano presented a recursive formulation for overlapping constraints of multiple end effectors [Sic91].

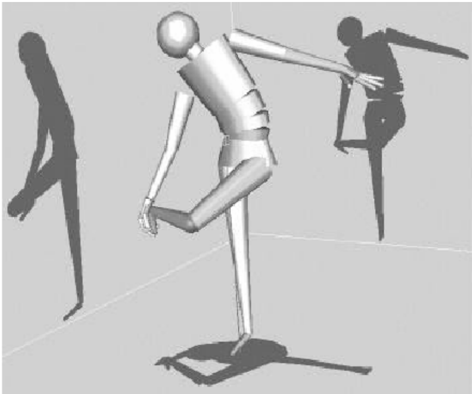

Figure 2-1. Character with the center of mass controlled (from [Mas96])

Another study developed an Inverse Kinematics algorithm for generating humanly natural looking arm positions [Kog94]. The algorithm was based on results from neurophysiology research [Soe89a] [Soe89b]. These results stated that an arm posture is determined by a simple sensorimotor transformation model. In this model, given the wrist position, the arm angles that specify the arm posture can be computed. However, the sensorimotor transformation is an approximation and it will be necessary to use the pseudoinverse Jacobian iteration to adjust joint angles. Another algorithm avoided the pseudoinverse calculation. It was also based in similar neurophysiologic data but instead of using arm angles directly, used the swivel angle computed from these arm angles. Swivel angle measures a rotation of the elbow around the shoulder-to-wrist axis [Lof97].

Komura proposed a method to use the musculoskeletal human body model for Inverse Kinematics so that animator's work is reduced to create natural human animation [Kom00]. Unnatural postures could be avoided using muscle dynamics. Muscle dynamics was researched in previous studies of this author.

A research from Tolani and Badler, more than generate realistic joint trajectories, investigated all kinematically feasible solutions [Tol00]. Due to the intrinsic problem of Jacobian methods that are unstable near a singularity, the author proposed a pure analytical approach or a hybrid analytical-numerical when necessary. The system separates the articulated structure in several kinematics chains for which an analytical approach is applicable. When Inverse Kinematics is underconstrained, a set of parameters are utilized so that the user explores several solutions merely changing the value of these parameters.

2.1.4 Dynamic Techniques

Dynamic techniques aim at creating animations relying in the laws of physics. These techniques give a great realism to the generated movement. A complete dynamic simulation will take into account gravity, friction, collision and external forces acting in the system. The drawback is the computational cost and the complexity of the dynamic control. That is, it is not easy to know the internal torque required to move a virtual human in a natural way. In this type of techniques, the challenge of researchers is to obtain realistic motion with low computation times.

Forward Dynamics computes joint kinematics given force and torque data. It is used for realistic animation of passive articulated structures, for example a robot control system. On the contrary, Inverse Dynamics consists on finding the forces or efforts necessary to reach a defined movement. The movement is defined by joint angle, velocity and acceleration.

An early study considered classical mechanics when frames were computed [Bro88]. The resulted motion appeared more natural that generated by interpolation but it was restricted to linear dynamic system such as trucks and aircrafts.

Hahn proposed a physically-based method for realistic animation of rigid bodies [Hah88]. The motion depends on properties such as mass, elasticity, friction, etc.

Hodgins made dynamic simulation of human motion (running, cycling and vaulting) [Hod95] (See Figure 2-2). She used state machines that describe each specific motion. Later, she adapted existing controllers for running and cycling behaviors to figures of variable dimensions [Hod97].

A motion specification technique for quadrupeds used footprints [Tor98]. Each footprint is specified by timing and orientation data. The footprints define, in an implicit way, the motion trajectory. In a second stage, trajectory is optimized to accommodate a simplified dynamic model.

Bruderlin designed the system named Keyframe-Less Animation of Walking for the creation of walking sequences [Bru89]. Forward velocity, step length and step frequency are provided. He combined dynamic motion control with goal-directed motion control. Then, less parameters were needed to define a motion compared to keyframing.

Systems based on dynamics have proven to be appropriate for specific motions such as running and jumping. Even to generate those motions it is needed a control algorithm specific for each behavior. As our work aims at providing a general approach for simulate fatigued motions, this type of techniques are not the most adequate for our purposes.

Last years an effort has been made in order to enlarge the group of movements contemplated by physically based methods. A research proposed a framework that facilitates the exchange and composition of controllers. The system is capable of generating a variety of human-like motor behaviors [Fal01a] [Fal01b]. It defines two types of controllers, individual and supervisor. Individual controllers have specialized control knowledge. The method uses individual controllers to define pre-conditions, post-conditions, and expected performance that characterizes a specific motion. The supervisor controller sets as active controller the one with higher priority from among a set of individual controllers.

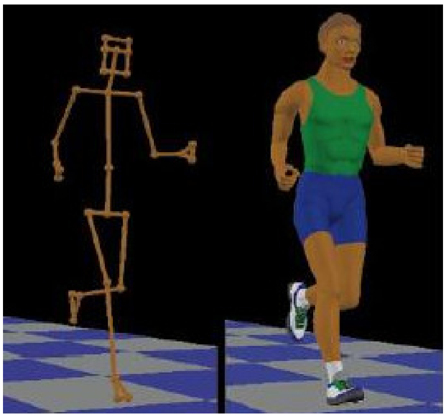

Figure 2-2. Dynamics (from [Hod95])

2.1.5 Task Level Techniques

Task level abstraction is a convenient means of specifying motion for certain applications. This animation technique provides the animator with abstraction beyond the low level of kinematics and dynamic techniques.

Korein made an early attempt to work with task level control [Kor82]. He addressed the problem in terms of effective Inverse Kinematics and interpolation algorithms for a six-link manipulator in 2D.

Lee developed a system to compute the trajectory of the arm when lifting objects of variable weight [Lee90]. A strength model was used to calculate a biomechanically realistic trajectory of joint angles and end effector positions from initial to final state. However, the sophistication of the biomechanical model compromises its general application to arbitrary tasks.

Zeltzer designed a hierarchical task oriented animation system in which the low-level walking motions were implemented kinematically, based on measured human data [Zel82]. The movement was obtained by "motor control programs" (MCPs). MCPs are finite-state machines for executing a particular class of movements (walk, run). MCPs are in the highest abstraction level and their states invoke a programs, in a lower level of abstraction, named "local motor programs" (LMPs). In this system the animator looses artistic control due to an automatic motion synthesis. Another research discussed animation abstraction in terms of a three-level hierarchy: guided, animator and task levels [Zel85]. Levison and Geib implemented a system that translates high level directives into task level commands [Lev94] [Gei94].

2.1.6 Behavioral Techniques

These techniques provide realism to computer generated animations in the sense that autonomous intelligent behavior is given to characters. The goal of this type of techniques is to obtain free virtual actors in unpredictable virtual environments. Usually, character's behavior depends on two factors: animator specification and its own behavioral model.

Haumann presented a computer animation system in which a complex movement is the result of a simulation of simple behavior rules between several actors [Hau88]. His goal was to provide not only physical behavior, but also other behaviors that express social or personal aspects. Thalmann proposed virtual actors solving problems in game playing and obstacle avoidance. A model for learning and forgetting behaviors was implemented [Tha99] (See Figure 2-3).

Figure 2-3. Autonomous Virtual Humans (from [Tha99])

A system integrating behavioral, motivational and motor-level commands was proposed by Blumberg [Blu95]. Another system allowed the animator to direct virtual actors with high-level primitives at critical points in the story [Mat99].

A research went beyond providing the possibility of characters discovering new behaviors [Fun99]. The author studied cognitive modeling which governs how knowledge is acquired, referenced and internally represented.

Badler proposed an agent-based architecture with two levels [Bad97]. The lower, the sensor-control-activer (SCA) produces local and adaptative movements. Its behaviors are "looking", "walking", etc. The high level control is called PaT-Nets, parallel transition networks, allowing the expression of more elaborated behaviors patterns than the basics.

Goldberg designed a system based on scripts allowing the creation of autonomous synthetic actors [Gol97]. Autonomy is given to the system by a means of scripts having several different conditional executions. Actors can also have more special personal characteristic and preferences, establishing their own personality.

High-level parameters could give personality to the motion of a virtual human. Allbeck and Unuma tried to influence agent behaviors basing on state of mind or character's personality [All02] [Unu95].

Maes presented a general approach for model behavior [Mae95]. Her work has been a source of inspiration for other behavior's researchers in computer animation. Her approach utilizes a perception-action concept as opposed to other approaches using planning. This system is provided with an action selection algorithm, based on the current situation and current goals. This system is dynamic in a changing environment during the execution.

2.2 Animation Based on Physiological Parameters

The human body is able to move with the help of more than 650 muscles. Each bone has at least two muscles attached for this purpose. Joints in the human body have limited ability to create movement or sustain a posture; this is due to the force exertable by muscles.

Several factors influence the muscular system so that it can exert more or less force. A fatigued or injured person exerts less force than a trained one. Therefore, the animation of the human body taking into account physiological factors would produce more natural results. This subsection deals with researches that include musculoskeletal models or physiological factors as fatigue to improve the realism of resulted motion or generated postures.

2.2.1 Minimum Muscle Action

Komura proposed a human animation system to enable the interpolation of arbitrary postures using the optimal control theory [Kom97]. Optimal control consists in choosing from amongst all admissible control variables u(t), the one that takes the dynamical system from an initial state at time t_0, $x(t_o)$, to x(T) at some terminal time T, in such a way as to achieve a maximum or minimum of an objective function. He proposed a function to minimize muscle action.

He used two main sources of data. Friedrich and Brand for biomechanical data [Fri89] [Bra82] and Hill type muscle model for muscle dynamics [Hil38]. Biomechanical data include muscle parameters like muscle origin and insertion data, maximum muscle force, fiber length, tendon slack length, tendon elasticity, etc. Figure 2-4 depicts the well known Hill's muscle model where CE represents the contractile element, that is the muscle fibers. The connective tissue is represented by a parallel element, PE, and the muscle tendon is represented by a serial element, SE.

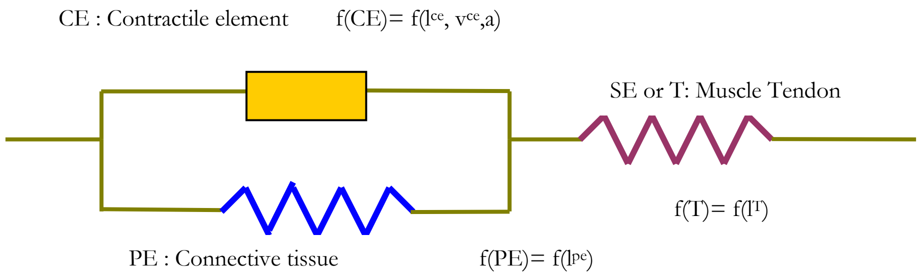

Figure 2-4. Hill's Muscle model

The force delivered by PE and T depends only on their length but the force delivered by the contractile element depends on three factors, length, velocity of contraction and activation level. When a muscle exerts its maximum force, the muscle activation level is considered 1. On the contrary, when the muscle exerts its minimum force, the muscle activation level is 0. Other muscle models derive from Hill's model [Win90b].

Figure 2-5 describes the steps followed to generate a motion minimizing the muscle action. Inverse Dynamic was used to calculate the force and torque made at each joint. However, several muscles can control each joint so that to obtain muscles forces from joint torque is not immediate. Then, previous result on muscles force prediction is used to obtain individual muscles forces from torques. This result used numerical optimization to do the prediction.

Crowninshield predicted muscle force minimizing a specified criterion [Cro81]. Crowninshield's method was modified to take into account the dynamics of the musculotendon in the muscle force prediction. That is, the elastic forces that muscles exert from a determined length [Rie99] [Est95]. The definition of the optimization criteria differentiates passive force and contractile force. The sum of each muscle contractile force divided by the average of muscle cross-section area (PCSA) is minimized.

Once muscle force is obtained, muscular activation can be calculated as: $a = f(f^T, l^{MT}, V^{MT})$. When a is obtained, an optimal control process generates trajectories trying to minimize a criteria, in this case, the minimum muscle action:

$$J = \int_{t0}^{tf} \sum_i a_i^2 + \sum_i w(a)_i dt$$

where a_i represents the activation level of muscle *i* calculated as explained above and $w(a)_i$ is a penalty function used in the case that a_i is out its range from 0 to 1.

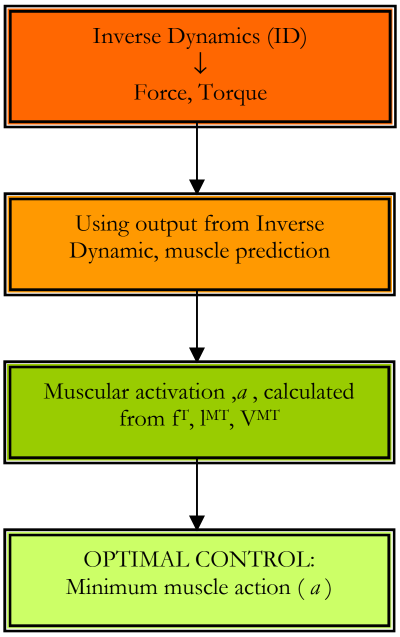

Figure 2-5. Steps followed to minimize muscle action

In short, Komura used a muscle-attached model to create human motions. To create such model, data of the muscle attachment geometry and muscle specific parameters are necessary. Compared to the research described previously, which focus in a detailed description of individual muscles, we focus on calculations at antagonist muscle group level, not at individual muscle level.

2.2.2 Musculoskeletal & Fatigue in the Lower Leg

A posterior study of Komura combined Delp's musculoskeletal model and Giat's fatigue model to deal with realistic character animations [Kom99].

A detailed description of Delp's musculoskeletal model can be found in [Del90]. To summarize, this model includes a 7-segment leg, the attachment sites of 43 muscles on each leg and physiological parameters such as the length of tendons, range of joint angles, etc. The lower half of the body is composed of the pelvis, femur, tibia, patella, talus, calcaneous and toes in each leg. The joints are either a 3-DOF gimbal joint (i.e. hip) or a 1-DOF (i.e. knee). Muscle data of the lower extremity contain a dimensionless force-length curve for tendon, dimensionless force-length curves for muscle, a dimensionless force-velocity curve for muscle and definitions of forty-three lower limb muscles. The definition of each muscle contains a list of coordinates that describe its line of action and parameters (peak isometric force, optimal fiber length, tendon slack length and pennation angle) needed to compute isometric muscle force.

Along with Delp's model, Giat's model was used [Gia93]. Giat proposed a mathematical model for the fatigue and recovery phases of paraplegic's quadriceps muscle. He studied the mechanical and the metabolic profiles of fatigue of a paralyzed muscle under activation by FES (Functional Electrical Stimulation). Fatigue is characterized by peripherical and central aspects, paraplegic's muscles lack of the central one. Since paraplegic patients are isolated from VC (Voluntary Control), the system mechanically analyzed is simpler than others where VC is included therefore allowing the calculation of muscle force from the measured joint torque.

Records of the time variations of the muscle's metabolites obtained from MRS (Magnetic Resonance Spectroscopy) were used to calculate the intracellular pH level in the muscle and this latter parameter was incorporated in a musculotendon model. Several studies had indicated that, despite the atrophy that characterizes these muscles, the metabolic state at rest in these muscles which have been treated by long-term low frequency electrical stimulation was very close to that of normal muscle [Lev93].

The goal was to construct a function that represents the intracellular pH level at any stage of intermittent FES. The analysis was based on three functions experimentally obtained by Levy: pH decrease during stimulation, pH increase during recovery and the relation between the pH and force levels in the quadriceps during the stimulation.

The curves describing the pH history and the force history as well as the pH versus force were all derived from MRS studies. Figure 2-6 summarizes data obtained.

The incorporation of fatigue and recovery, as a function of the pH level, in the expression of the force exerted by the contractile element defined in Hill's muscle model is as follows:

$f(CE) = f(l_{ce}, v_{ce}, a) \cdot f^N_{pH}$

where l_{ce} is the length of the contractile element, v_{ce} represents the contraction velocity, *a* is the activation level and f^N_{pH} is obtained by normalizing the force at time t by the force obtained at the beginning of the experiment.

The fatigue model was used to determine the maximal amount of force exertable by the muscles. The fatigue phase, produced when a muscle exerts a large amount of force, causes the intracellular pH level inside the muscle to decrease. Then, the maximum amount of force exertable by the contractile element decreases. Recovery phase is produced when the muscle is not used, then the pH level incresases and the exertable force increases.

- Least-square curve fitting of the pH decay during fatigue with respect to time t:
  $pH^F(t) = c_1 - c_2 \tanh[c_3 (t - c_4)]$, where c_1, c_2, c_3, c_4 are constant parameters.
- Least-square curve fitting of the pH increase during recovery:
  $pH^R(t) = d_1 + d_2 \tanh[d_3 (t - d_4)]$, where d_1, d_2, d_3, d_4 are constant parameters.
- Concatenation of pH^F and pH^R to form a fluctuation function pH (a,t), where pH is expressed as a function of activation level and time. The complete derivation can be found in [Gia96].
- Least square curve fitting of the force output with respect to the corresponding pH level:
  $f_{pH}(pH) = d_5 [1 - e^{d_6 (pH - d_7)}]$, where d_5, d_6, d_7, are constant parameters.
- f_{pH} normalized respect to the force at time t_0:
  $f^N_{pH}(pH(t)) = f_{pH}(pH(t)) / f_{pH}(pH(t_0))$

Figure 2-6. Derivation of the fatigue/recovery model

It is important to note that the relationship between the activation level and the pH level is not included in the model. Then, a threshold of 0.5 of activation level is established to switch between fatigue and recovery.

Unlike Giat's pH-based fatigue model, our model of fatigue is mainly based on ergonomic studies which use the maximum holding time that a posture can be maintained and the current holding time [Man86]. Maximum holding time is defined as a function of current torque normalized by maximum voluntary contraction (muscular strength).

2.2.3 Minimizing other Physical Values

Bioengineering researchers have studied methods to simulate human motion by minimizing some physical values. Tamar and Neviller tried to minimize the jerk [Tam85]. Uno and Kawato did the same but for torque change. The criterion defined was:

$C = \int_{t_0}^{t_f} \sum_{i=1}^{2} (\frac{d\tau_i}{dt})^2 dt$, where τ_i represents the torque of joint i, t_0 is the time at which the movement starts and t_f is the time at which the movement ends.

Kawato also worked on the minimization of muscle signal change [Kaw92] [Kaw90]. Uno studied the minimization of muscle force change [Uno89a] [Uno89b]. Another study has solved the redundancy of Inverse Kinematics problems using a criterion of minimum muscle-force change [Kom01].

An algorithm was designed to convert a motion to a physiologically feasible one minimizing the additional external load needed to accomplish a task [Kom00]. He defined an optimization problem to convert an unfeasible motion into a feasible one. A motion is unfeasible when muscles forces are not enough to afford a motion and then supplementary external forces are needed. Therefore, the approach was to minimize the supplementary external forces needed to afford the motion. That is, $\min_{f,\tau_{ext}} |\tau_{ext}|^2$, where f is a vector storing the forces of each muscle $f = \left(f_1^t, ..., f_{n_m}^t\right)$, n_m represents the number of muscles and τ_{ext} is the supplementary torque that is applied when motion cannot be realized by the muscles alone. If the balance constraints are to be satisfied, an additional term is added in the optimization.

Another study used a musculoskeletal model to simulate jumpings of maximum height [Pan90]. Popovic achieved, reducing the degree of freedom of the knee, a limp motion from gait data [Pop99].

Witkin used a space-time approach to set physical constraints for a character [Wit88]. The problem was solved as a nonlinear constrained optimization in which an objective function is minimized to satisfy a set of constraints that have been placed on the character and the corresponding motion. The objective function described the energy expended in the performance of the motion.

Muscle strength is defined as the maximal force that can be generated by a muscle under isometric conditions. The most common technique used to assess strength involves the measurement of its physiological cross-sectional area through ultrasonic techniques [Ika68] or computed tomography [Wei60]. The cross-sectional area is multiplied by the maximal stress that muscle tissue can produce to give muscle strength.

Another research made experiments and results on strength were fitted by curves. In our research we have used data from Chaffin [Cha88]. As will be seen in the following section, Lee used muscle strength in movement optimization [Lee90].

A research has studied strength difference in male and female to adapt aircraft cockpits and redesign aviation life support equipment [Mey96].

Chen and Zeltzer developed a finite element model of muscle to simulate muscle forces and to visualize the deformations that muscles undergo during contraction [Che92]. Finite element methods are based on a continuum model and generate physically based results. The authors utilized polygonal data derived from MRI (Magnetic Resonance Imaging) scans or data digitized from anatomically accurate plastic models to represent muscles. However, this study has not been used for the control of the human body model in a dynamic environment.

2.3 Human Body Positioning. Optimization and Characterization.

Positioning is a difficult task when we deal with highly articulated figures such as the human body.

Girard and Maciejewski's system uses a combination of dynamics and kinematics to achieve the key positions. The user defines the path and type of the walk [Gir85].

Lee utilized strength, comfort and perceived exertion as heuristics for optimizing movement [Lee90]. A strength model was used as a criterion for control and path decisions. The system chose the paths of the tasks, not the animator, as happens in key-frame animations. In this system, muscular strength indicates the comfort region to accomplish a task. Comfort is given by the maximum torque ratio, defined as the current torque divided by the maximum torque. The author defined the expected level of difficulty in the achievement of a task as the perception of amount of strength required to accomplish it. Several motion strategies were defined: available strength, reducing moment, pull back and recoil and jerk. In the available torque strategy, the person tends to move the stronger joint. In the reducing moment, the person tries to increase the available torque. The pull back strategy consists in taking back a joint when it reaches a maximum strength level. The recoil and jerk strategy tries to reduce the forces needed to complete a task for active joints. In this approach, the completion of a task is not prioritized while enforcing a torque strategy.

Mas introduced the concept of Inverse Kinetics to improve the balance control for human posture optimization [Mas96]. Inverse Kinetics integrates the mass distribution information to embody the position control of the center of gravity of any articulated figure in single support.

A work on automatic posture generation predefined body postures storing them in a database. The relation between the body and object postures were incorporated into the algorithm through a database that stores natural looking postures recorded from real life. The database was designed to provide global information about the reachable space. With this data it is possible to make accurate estimations even for target objects that are far away from the hand position. The algorithm consists of three steps: classify the target object as one of the primitive objects, estimate the hand posture using the data of the nearest sub-volume and finally estimate the entire body posture using Inverse Kinematics starting from the stored body posture of the nearest sub-volume [Ayd98] (See Figure 2-7).

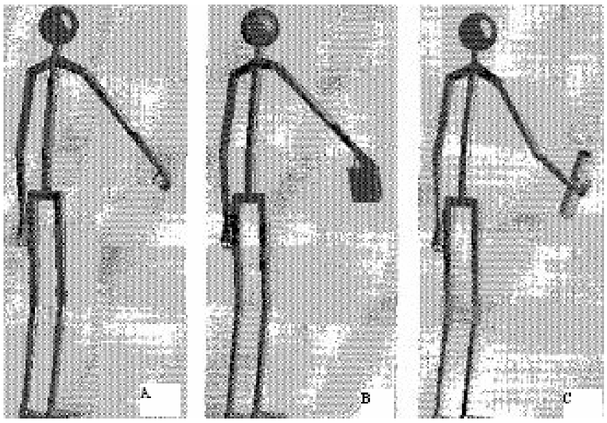

Figure 2-7. Stored (A) and estimated postures (B, C) from a standing character (from [Ayd98])

Posture prediction is also an important aspect of human movement simulation. Posture prediction represents algorithms that estimate human joint angles for a given posture. There are two approaches related to posture prediction. One uses empirical statistical models and another uses biomechanical models. Researchers that have used statistical models are Beck, Zhang [Bec92] [Zha96a] and Faraway [Far99].

Jung utilized a cost function for posture prediction [Jun94]. He proposed a cost function integrating discomfort of joints and the joint range availability.

Later on, Jung presented an analytical reach prediction algorithm using Denavit and Hartenberg (D-H) notation [Jun95] and developed a regression model to predict perceived discomfort [Jun96]. It should be noted that Jung's model was applied to simple articulated structures such as the arm.

Dysart designed three models and objective functions to predict postures of human in standing sagital lifting tasks [Dys96]. He utilized Inverse Kinematics to represent feasible postures and explored diverse criteria functions for selecting a final one.

Wang used a combination of methods such as rule-based empirical and optimization [Wan99]. He showed that the Inverse Kinematics problem is ill-posed due to the redundancy of the arm and designed an algorithm to tackle this problem. The algorithm handled the non-linearity of joint limits with no need of inverse matrix calculation. In this way, he avoided the stability and convergence problems that appear near a singularity of the

Jacobian.

Inverse Kinematics can be used in task feasibility studies in which a character has to simulate the performance of a reachability task. Inverse Kinematic is then used to determine reachable objects, in other words, what parts of space are reachable.

Abdel-Malek used the implicit function theorem to delineate singular surfaces that are on the boundary of the reachable workspace [Abd02]. This is done studying the rank deficiency of the Jacobian. Once it is determined if a point is reachable, he proposes to predict the posture using three cost functions: minimum displacement from the initial posture, potential energy and dexterity. The cost functions are used to drive the arm, not to optimize the best solution. To do it, he uses a genetic algorithm with an initial population given by joint values and joint limits.

The characterization of human limb postures is useful in various research fields: neurophysiology, rehabilitation, workplace design and sport performance. In addition, the determination of which particular pose a human chooses to perform a task is a valuable way to design workplaces.

Rancourt investigated the existence of a criterion by analyzing the consistency in upper limb pose of human subjects in a given task, in particular, in a drilling task [Ran00]. He introduced tests for comparing the relative orientation of markers using techniques of orientation statistics. Most experimental subjects had statistically different poses for the drilling locations that were kinematically equivalent. Statistical analysis of limb poses suggested that a posture, in a drilling task, would not be selected upon a selection criterion leading to a unique solution. As a result, evidence of selection criterion for pose in a drilling task was difficult to demonstrate.

2.4 Summary

In this chapter, we have reviewed the literature regarding computer animation, placing emphasis in their contributions to the realistic animation of virtual humans. We made a detailed review of researches that provide insight in the musculoskeletal system in order to exploit it in the generation of natural postures or motions. Section 2.1 described previous works of different computer animation techniques. Section 2.2 introduced related work based on the exploitation of physiological factors. Section 2.3 reviewed researches on positioning of the human body.

After the review of the literature, the justification for our approach is clear, a fatigue model for autonomous characters, without an a priori knowledge of the movement description, is missing. Therefore, we have focused our research on providing such a model with continuous update of fatigue and posture optimization. Our approach is applied in a slowly evolving context.