# Real-Time Character Animation Techniques

**Thanh Giang Robert Mooney Christopher Peters Carol O'Sullivan**

**Image Synthesis Group Trinity College Dublin**

**Email: Thanh.Giang@cs.tcd.ie, Robert.Mooney@cs.tcd.ie,**
**Christopher.Peters@cs.tcd.ie, Carol.OSullivan@cs.tcd.ie**

---

## Abstract

*This report reviews the literature in the field of computer animation and robotics to illustrate the numerous approaches to simulating, controlling and displaying the realistic, real-time animation of virtual characters. Critical analyses of the various high-level approaches are conducted, with a view to identifying key areas of interest for future research in the area of "Real-Time Character Animation Techniques"*

## 1. Introduction

This paper reviews the various techniques that are present in the field of computer animation. The review concentrates particularly on those techniques and problems involved in the production of realistic human synthesis and motion.

Section 2 of this paper deals with the layered modelling approach for virtual character construction. Various deformation techniques such as *joint dependent local deformations, non-linear global deformations, free form deformations, implicit surfaces and mass-spring systems* are discussed in this section. The problems and advantages associated with each are discussed and particular attention is paid to those methods that allow for real-time animation.

Section 3 presents the different techniques that are used within computer animation. The first part of this section gives a brief discussion into the background of basic, more popular animation techniques such as that of *keyframing* and *motion capture*. The section then proceeds to introduce more advanced techniques such as the use of *inverse dynamics* and *constraint based* approaches to animation and motion control. The last section of the paper discusses conclusions towards the feasibility of some of these techniques and ideas with regards to their potential uses for the production of realistic human characters. Also, areas of possible future research are considered.

## 2. Layered Modelling

In 1989 Chadwick et al. [CHA89] emphasised the efficiency of a layered approach for creating complex articulated models. Using this approach the animator divides the simulation model into a number of discrete layers, each with its own physical and geometric properties. The animator specifies the various constraint relationships between the layers and can then control the global motion from a high level.

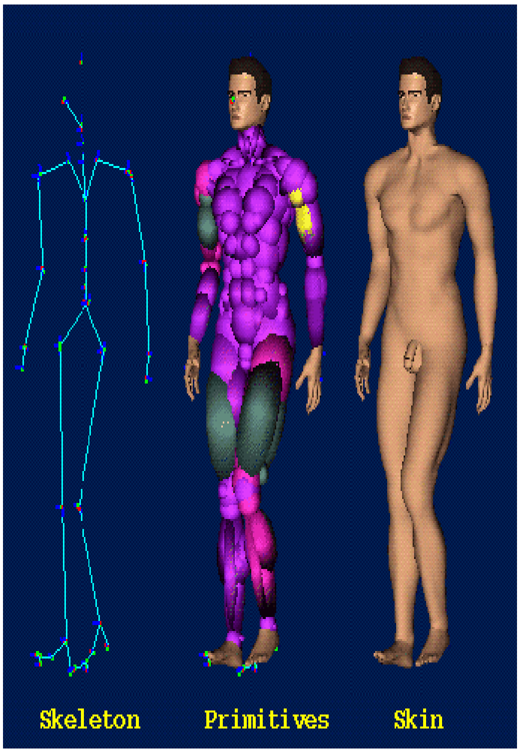


***Figure 1 Layered Character Construction***
**http://ligwww.epfl.ch/~shen/bb.html**

A 3-layered character construction is shown in figure 1. The skeletal layer is constructed by defining a hierarchy of primitives and joints, each with a number of degrees of freedom. Next the muscle/fatty tissue layer is applied by attaching geometric primitives to the underlying skeleton. Finally a mesh skin is wrapped around the character and rendered to give a more realistic appearance. When the appropriate constraints are applied between the layers only the skeletal layer need be scripted in order to achieve convincing motion.

## 2.1 Model Representation

Two central questions that need to be posed when beginning any character animation are 1) how many layers will be required to represent the character's structure and 2) will the character model be geometric, physically based or a hybrid representation ?

The number of layers used for real-time character animation is generally constrained by the underlying hardware's computational ability. Where initially 2-layered (skeleton and muscle) and 3-layered (skeleton, muscle and skin) models prevailed, the ongoing improvement in processing performance has allowed more complex multi-layered models to be constructed (clothing, hair and fur). Traditionally, geometric models have been used for character representation in computer games; characters are represented using a hierarchy of spheres, ellipsoids and cylinders that maintain the same shape and look throughout the animation. While this technique generally offers efficiency and ease of implementation it does little to portray the character or the situation. Physically based models however, allow soft-objects to be represented. This is achieved by using the laws of Newtonian physics to control the shape and movement of the objects in the character model. Typically a physically based approach will endow the primitives in the character model with attributes such as velocity, acceleration and force in order to achieve perceptually realistic deformation and motion. It is by using these simple techniques that torque, friction and collision response can be modelled.

## 2.2 Deformation Techniques

By using the layered approach for character animation, the animator can rely on the skeletal layer to control the motion of the character. However, in order to achieve perceptually realistic movement, the higher layers must deform in accordance with the surrounding layers. For example, when a limb bends it would be desirable for the muscle on the limb's joint to bulge and for the skin around the joint to crease.

There are a number of techniques that can be used to deform objects in this way, the most popular of which will be examined in the following sections.

## 2.2.1 Joint Dependant Local Deformations (JLD)

This animation technique, which uses a polygonal mesh skin digitised from a sculpture, maps each vertex point to a particular point on the skeleton using JLD operators. These are specific local deformation operators that depend on the nature of the joints, and control the evolution of the surface. Each operator is applied to a specific section of the surface referred to as the domain of the operator. The value of each operator is obtained from the angular values of the set of joints which define it.

Thalmann et al. [THA88] used this technique to simulate the complex motion of the human hand. Although JLDs have been used to model creasing around joints as well as muscle bulging, no successful model has been constructed that allows for deformation due to external forces. A further drawback is that this method relies on data that is specific to a given joint, therefore it is unlikely that this approach could be used to generate a general character model.

## 2.2.2 Implicit Surfaces

An implicit surface can be described as the set of points P that satisfy the equation:

$$F(P) - Iso = 0$$

Where $F(P)$ is the implicit function and $Iso$ represents the threshold value at which the surface is defined. To illustrate this idea more clearly, consider the equation of a sphere:

$$x^2 + y^2 + z^2 - r^2 = 0$$

Here the implicit function $F(P) = x^2 + y^2 + z^2$ and $Iso = r^2$. It is evident that altering the value of $r^2$ will result in the sphere's surface changing position in 3D space. Since the implicit function always returns a 1D/scalar value it is sometimes referred to as a scalar field function. This eases calculations since the surface can be described independently of the space. A further advantage of the scalar result is that for any given $(x,y,z)$ $F(P) - Iso$ will return one of three results:

1. A negative result, indicating that the point is within the sphere.
2. A positive result, indicating that the point is outside the sphere.
3. A result of zero, indicating that the point is on the surface of the sphere.

From a character animation point of view, implicit surfaces offer a number of attractive qualities. Firstly their ability to represent continuous surfaces allows the virtual character to possess smooth joints, a quality which is not readily achievable using other methods. Furthermore implicit surfaces can be used to handle collision detection and response as well as unwanted blending between the primitives in the character hierarchy. While these are certainly desirable facilities in any character animation, implicit surfaces are not without their disadvantages.

***Figure 2 Implicit surface model of a hand***

Typically characters constructed from a hierarchy of implicit surfaces appear overly rounded *(Fig 2)* which detract from their visual realism. Another problem is that rendering them is difficult because there is no parametric co-ordinate system to serve as a basis for sampling. Although easily raytraced, polygonisation involves using an expensive sampling technique such as the marching cubes algorithm. Turner [TUR93] suggests that superquadrics (a particular family of implicit surfaces introduced by Alan Barr [BAR84]) offer an interesting alternative because they have an implicit as well as a parametric definition. It is because of their robust definition that a superquadric's tessellation can be easily enhanced and rendered.

These properties could be used to decrease the computational cost of muscle and surface animation using techniques for level of detail (LOD) [CH97] adjustment described in section 3. A number of useful formulae are provided by Barr [BAR84], including the formula for finding the normal to the surface of a superquadric which could be used for rendering calculations.

Clearly implicit surfaces are well adapted to physically based animation. In the past the implicit layer has been used to compute response forces that occur as a result of compression and friction. Preservation of the volume of deformed objects is also possible, even when the objects undergo significant changes such as separation and fusion. This level of versatility can be used to animate not only human musculature but also more deformable skeletal structures, as in Industrial Light & Magic's 1997 production of *FLUBBER*.

### 2.2.3 Non-linear Global Deformation

In order to alter the dimensions of an object, a transformation matrix is usually applied to each vertex in the object's definition. Throughout this operation the transformation matrix does not change. Barr [BAR84] suggested changing the transformation matrix while it is being applied to the object. Thus the way in which the matrix is altered becomes a function of the position at which it is applied. In other words the transformation matrix is itself transformed. Barr used this idea to model bending, tapering and twisting. For example, consider the vertex $(x,y,z)$. Let $F$ represent the tapering operation already mentioned. Then:

$$(X,Y,Z) = F(x,y,z).$$

where $(X,Y,Z)$ represent the new tapered vertex.

In order to taper along the x axis we simply scale the y and z axes and preserve the length of x. More precisely:

$$(X,Y,Z) = (x,ry,rz)$$

where $r$ represents the scalar value.

The bending and twisting deformations are achieved in a similar fashion. While this technique is both intuitive and powerful it is wholly geometric. It is conceivable that a physical interpretation could be developed but at this point further investigation is required to determine its feasibility.

### 2.2.4 Free Form Deformations

The free form deformation (FFD) was first introduced by Sederberg and Parry [SED86] as a technique for achieving representational independent deformation. The unique quality of this approach is that instead of applying deformations to the object directly, the object is embedded in a space that is then deformed using Bezier theory. To describe this, Sederberg uses the example of placing a number of objects in a clear and flexible plastic parallelepiped container *(Fig 3)*. When the container is deformed the objects within change shape in a manner that is "intuitively consistent".

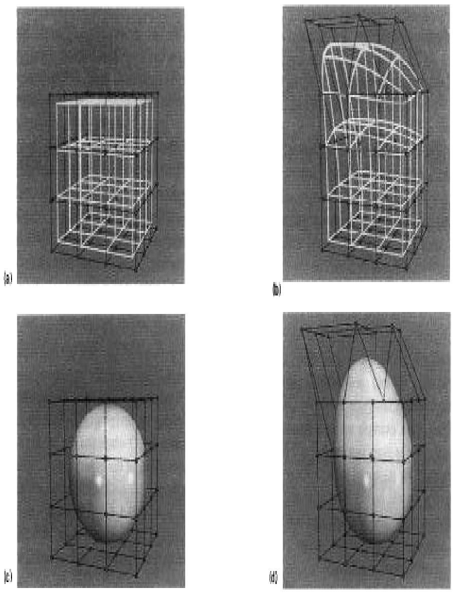
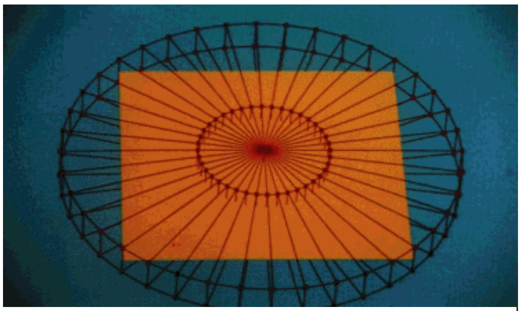


***Figure 3 FFD performed on a sphere***

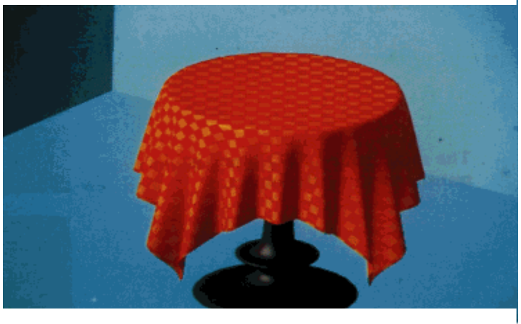
Since all alterations are applied directly to the co-ordinate system, no assumptions need be made about the nature of the object that is to be deformed. Clearly the strength of this approach lies in its generality. Arbitrary muscle shapes can be modelled and with careful construction of FFD blocks around linked skeletal primitives, unwanted blending can be controlled. However, using a single FFD block to control the animation of an object results in transformations being applied to each vertex in the co-ordinate system. This is undesirable in many situations. Changes can be applied more accurately to a particular part of the object by increasing the number of control points of the FFD. Obviously the changes still affect every point in the object but to an increasingly small degree. This also greatly increases the computational cost.

The extended FFD, proposed by Coquillart [COQ90], is based on the observation that altering the control point mesh from a parallelepiped arrangement to a different shape (cylindrical), results in a more closely fitting lattice *(Fig 4)*. This lattice avoids unusual deformations which occur when deforming objects of arbitrary topology.



*a) Cylindrical EFFD block surrounding cloth*

*b) Applying deformations to the EFFD block*



*c) The deformed surface*

***Figure 4  EFFD used to deform a square table cloth***

Coquillart considered EFFD's applied locally to the surface of an object . In this way the EFFD and the object become disassociated from one another. This overcomes the problem of unwanted global deformations which occur as a result of changes to the co-ordinate system.

The AFFD [COQ91] introduced a new way of looking at Free-Form Deformation. Instead of interpolating the metamorphosis of the 3D lattice which lies around the deformable object, the deformation tool (lattice) is differentiated from the object. The animated FFD allows the control point mesh to be moved through the space which it is deforming, causing the object to deform in an animated way as it passes through the spatial deformation .

### 2.2.5 Mass Spring Systems

Particle systems were introduced by William T. Reeves [REE83] in 1983 for the Genesis Scene in Star Trek II, The Wrath of Kahn. In this scene a volcanic eruption engulfs a planet gradually turning it into a molten mass. A particle in this sense was essentially represented by a point with physical attributes such as velocity, acceleration and force. As a particle progressed through space its colour changed (varying shades of orange and yellow for lava) eventually turning black and being removed from the current frame to give the impression that it was being extinguished. In this instance, the particle system (collection of particles) was a straightforward extension of particle dynamics. However, particle systems have been used to model more structured objects such as cloth [BRE 91], trees and grass [REE85], hair and fur [DIS99] etc.

Mass-Spring systems are essentially particle systems whereby each particle is connected to any number of its neighbours with a spring so that a deformable structure is preserved. Nedel and Thalmann [PNT98] used this idea during the ESPRIT sponsored CHARM project to model real-time muscle deformation. To simulate these deformations, a new type of angular spring was developed. These allow the muscle's volume to be controlled during simulation.  Although the results obtained from this research were visually realistic they only succeeded in modelling the human upper-arm, and that relied on specific medical data.  A technique which relied on more general user input, and more intuitive mathematics was developed by Cooper and

Maddock [COO98] but was still only used to model the human-limb and suffered from some unusual effects when a joint bent more than 90 degrees from the rest position. Mass-Spring systems have been used with some success to model skin by binding a surface layer to the skeleton using stiff springs *(Fig 5)*.

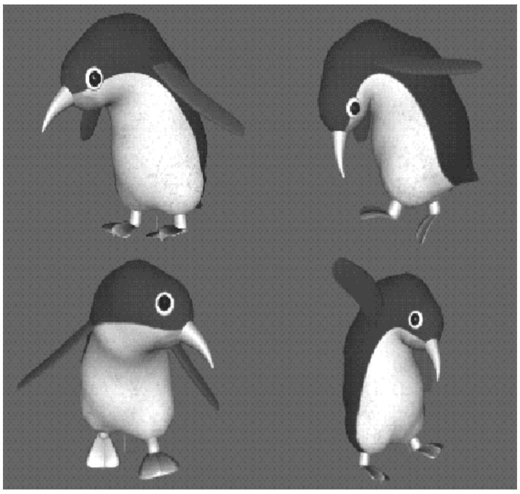


***Figure 5 A Mass-Spring skin model [CHA89]***

## 3. Animation

Traditionally, the art of animation has been a time-consuming pen and pencil affair, highly dependent on the skill of the animators involved. The art of computer animation has taken tremendous strides from these traditional techniques. Perhaps the first piece of work to demonstrate the power of using computers for animation was that of Pixar's 1986 mini masterpiece, '*Luxo Jr'*.

Luxo Jr. was a computer animation that featured a lamp playing with a ball. At that time, the animation was done using a trivial keyframing method.

Since then, computer animation has evolved with leaps and bounds. Animated figures are no longer totally dependent on the skill of the animator for their realism, and new methods provide ways of animating figures with very little intervention from an animator. This section will consider a number of these techniques.

### 3.1 Basic Techniques

A number of basic techniques have become prominent in recent years and are widely used in many different areas of character animation. The following could be considered to be parts of a 'basic toolkit' that can be used to build powerful animation systems.

The more traditional method of animating an articulated figure in computer graphics involves specifying each particular part at certain key locations in space, and then using some interpolation technique to animate the in-between frames for the motion in question, see Hodgins and O' Brien [HOD98]. This method of creating motion is known as *keyframing* and the manual input of poses is, perhaps, the simplest method of defining motion . The animator must produce the real world motion by hand. Not only is this a tedious and time-consuming task [LAS87], but it also relies on the skill of the animator for realistic, convincing effects.

Witkin and Zoran [WIT95] propose a technique for editing captured or keyframed animation based on warping of the motion parameter curves. The animator defines a set of keyframes, which act like constraints, to derive smooth deformation that preserves the structure of the original motion. Many different realistic motions can be derived from a single captured motion sequence using only a few keyframes to specify the motion warp. This method fits in well with the existing keyframe paradigm. Unfortunately, it also shares some of the limitations of keyframing (enforcing geometric constraints across keys is difficult). The method is not based on any especially deep understanding of the motions structure. Because of this, extreme warps can look distorted and unnatural.

*Kinematic* techniques specify motion independently of the underlying forces that produced the motion. For character animation, it is useful to think of *forward-kinematics* as the process of explicitly specifying all joint motions in order to determine the position of the free end of a chain (referred to as the *end-effector*). In contrast, *inverse-kinematics* determines the position and orientation of all joints in the hierarchy given an end-effector state. Using inverse kinematics, the geometric data need only be specified for one or a few specific branches within the articulated figure, and others within the hierarchy will follow suite, Zhao and Badler [ZHA94]. While such a method provides the animator with a useful tool, generating realistic motion using inverse kinematics is problematic, since the motion of links in any articulated figure is identified using specific motion attributes rather than using forces and accelerations. It does allow for a more dynamic approach to character animation, however, since specific motions do not have to be predefined.

Another very popular technique is that of *motion capture*. Delaney [DEL98] discusses how realistic motion is generated by capturing the motion of a real world actor, either by optically tracking special sensors attached to key points on the actor's body or by tracking them magnetically. The geometric data needed to generate realistic motion is obtained from the data collected from these sensors. This technique has been very popular within the movie and computer games industry over the years. While it has the potential to produce highly realistic motions, these motions are also highly specific and, alone, are not influenced by dynamic environment factors such as gravity and inertia. Some interesting work by Hodgins and Zordan [HOD99] has focused on combining motion capture with dynamic factors.

While kinematics work independent of underlying forces, *dynamics* simulates the laws of Newtonian physics in order to obtain physically plausible animation.

*Forward-dynamics* controls objects by manipulating forces and torques in order to obtain trajectories and velocities. *Inverse-dynamics*, on the other hand, determines the forces and torques that are required to produce motions in a system. Dynamics based systems are often referred to as *physically based*, in order to distinguish them from purely geometric approaches such as kinematics. Two sub-approaches generally require consideration when using physically based methods; forward (step-based) simulation and constraint based methods such as the *space-time constraints*, which will be discussed shortly.

## 3.2 Motion Generation

The animation of aesthetically pleasing, realistic motions for human-like characters has been a challenging problem in the field of computer animation. The fact that humans are very acute at recognising plausible motion makes the task quite a difficult one.

In order to make a character move realistically, a suitable motion generation technique must be used. The technique chosen largely depends on the task at hand and how dynamic the environment will be. In many cases, it may be useful to employ a number of different techniques. For example, some systems may choose to animate the walking motion of a character using motion capture techniques, while giving the character an unscripted grasping ability using inverse-kinematics. The basic techniques employed determine the overall realism, processing speed and dynamic ability of the character. In turn, the dynamic ability determines the amount of animator intervention required to animate the character in different situations.

Multon et al. [MUL98] provide a good overview of the area in a survey of the various techniques involved in animating a walking human. Kinematic and dynamic based approaches for articulated figure animation methods are mentioned, in addition to animation based on motion data. Space-time constraints are touched on for the dynamics based approach and motion warping is also considered.

### 3.2.1 Parametric Generation

A number of parametric motion generation techniques have been developed for special case motion situations.

The *Jack* system described by [BAD93] uses dynamic and kinematic models of humans based on biomechanical data with built-in behaviours, such as walking and grasping, to simulate human motion. These built-in behaviours are generalised from motion capture data to provide realistic motion.

Bruderlin and Calvert [BRU89] use a simplified dynamic model and control algorithm to generate the motions of a walking human in the K.L.A.W. (Keyframe-Less Animation of Walking) system. A telescopic leg model is used with 2 degrees of freedom for a 'stance' phase, while a compound pendulum model is used for a 'swing' phase. A hybrid goal-directed, dynamic-motion system is used for controlling the walking motion. The upper body and arms are added kinematically and moved in an oscillatory pattern that resembles that of a human.

Pai [PAI90] describes a high-level fashion of specifying a walking motion. By using time-varying constraints, such as ‘keep torso vertical’, ‘maintain ground clearance during leg swing’ and ‘lift and put down a book’, a walking motion for a human model is programmed.

Lee et al. [LEE90] focus on the simulation of arm motions for lifting based on human muscle models. Factors such as comfort level, perceived exertion and strength are considered during the simulation. Koga et al. [KOG94] apply manipulation planning to computer animation by considering a path planner that computes collision-free trajectories for several cooperating arms to manipulate a movable object between two configurations. An inverse-kinematics algorithm is utilised by the planner for the generation of realistic human arm motions as they manipulate objects. The inverse kinematics algorithm is specific to human arms and is based on neuro-physiological studies. Dynamics and muscle models are not considered.

Rijpkema and Girard [RIJ91] present a grasping system that allows both automatic and animator chosen grasps. The grasping task is split into three distinct phases: the task initialisation phase, where the target object is classified as a single primitive to determine the grasping strategy; a target approach phase in which the final grasp is decided; a grasp execution phase where the hand closes around the object. Objects are approximated with simple primitives.

Hodgins et al. [HOD95] uses hand designed control algorithms to perform desired manoeuvres such as running and bicycling. Physical intuition about behaviours, observations of humans performing the specified tasks, and biomechanical data are used in establishing these control algorithms. Secondary motion is added to the animations through the use of mass-spring simulations of cloth driven by the rigid-body motion of the simulated humans. Dynamic simulation is used to compute the motions of the behaviours. Constraints are also used to model points of contact between the feet and the ground, and the feet and the bicycle wheels.

Hodgins et al. remarks that hand designed controllers are used since automated techniques are not yet available for finding solutions to systems with the number of controlled degrees of freedom that would be necessary to simulate a plausible human body. Furthermore, such techniques do not generally produce results that appear natural in the sense of resembling the motion of a biological system.

Despite the high-quality and realistic results often produced by these methods, they provide no general motion solutions. New behaviour implementation generally requires additional work on the part of the animator. A number of other techniques were developed in an attempt to generate motion for new behaviours automatically.

Thalmann and Thalmann [THA95] outline the three main phases of task planning as world modelling, task specification and code generation. Grasping problems are discussed, with a focus on ‘workspaces’. These are volumes of space in which the end-effector of the actor can reach. Mas and Thalmann [MAS94] discuss a completely automatic procedure for arm motion, including the use to two hands when necessary. Both direct and inverse kinematics are used to control all the articulations of the hand.

### 3.2.2 Constraint Based Methods

Witkin and Kass [WIT88] and Brotman and Netravali [BRO88] first introduced space-time constraints. The animator specifies the characters’ motion, how the motion should be performed, the characters’ structure, and the physical resources available to complete the motion. Constraints over the entire motion are considered simultaneously in order to find the optimal motions. This essentially treats the problem of automatic motion generation as a trajectory optimisation problem. Although space-time constraints have demonstrated some interesting results, a major problem is that they are currently limited to simple systems, due to the size and complexity of the required computations. Also, optimal physically correct motions are not always desired for realistic character motion, [GLE96].

Liu et al. [LIU94] go some way toward improving the usefulness of the space-time constraints mentioned by Witkins and Kass [WIT88]. They describe the use of the hierarchical space-time constraints technique for creating physically based and goal directed motion of articulated figures. The hierarchical space-time constraint method is the reformulation of a function through time of the generalized degrees of freedom in a hierarchical wavelet representation. The problems encountered in the use of space-time constraint control due to over-complexity in the articulated figure are tackled by the idea of creating space-time windows, as shown in figure 6.

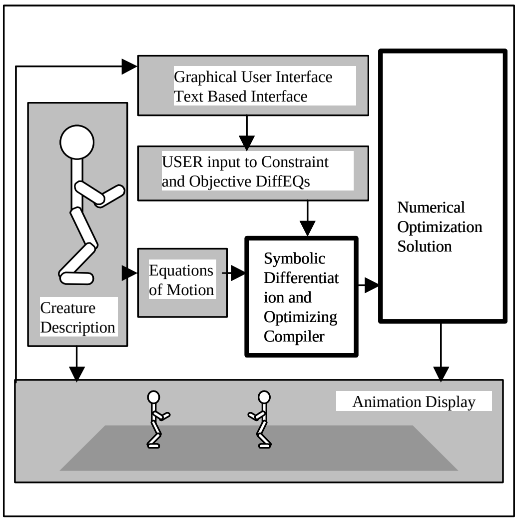


***Figure 6 The Hierarchical Space-time Constraints System proposed by Liu et al. [LIU94]***

Rose et al. [ROS96] use a combination of space-time constraints and inverse kinematics to create seamless and dynamic transitions between segments of human body motion. The method allows the users to break motion data into smaller pieces and reassemble it into new, more complex motions. This works by generating high quality motions (known as 'basis motions') using motion capture techniques and creating transitions between these motions so that they can be concatenated into further animations. The combination of these methods discussed, and [LIU94] could make space-time constraints a viable option for generating motions for limited parts of a complex articulated structure.

Barzel and Barr [BAR88] present a modelling system based on dynamic constraints. Models are built from collections of primitive elements with pre-defined geometric constraints; constraint forces pull the components in the model into the proper configuration and the elements move to satisfy the constraints. The motion of each rigid body is calculated from the effects of inertia, forces and torques acting on the body. Inverse dynamics is used to determine the forces that will produce a given behaviour, while maintaining existing constraints.

## 3.2.3 Motion Control

Some alternate approaches to those previously discussed are based on attempts to find control algorithms for models rather than desired motion trajectories.

Ngo and Marks [NGO93] propose a global search algorithm that can generate a number of trajectories for space-time constraints from scratch for 2D articulated figures. These search strategies rely on a method of encoding trajectories as behaviours, and then a genetic search algorithm for choosing behaviour parameters.

Van de Panne and Fiume [PAN93] detail a method by which the user specifies the configuration of a mechanical system that has been augmented with sensors and actuators. The sensor actuated network (S.A.N.) consists of weighted connections between the sensors and actuators. Suitable S.A.N.s are selected and improved using a stochastic procedure. This process provides an automatic means of discovering many possible modes of locomotion for a given object. The S.A.N.s are particularly adept at controlling creatures that warrant non-linear and non-smooth systems.

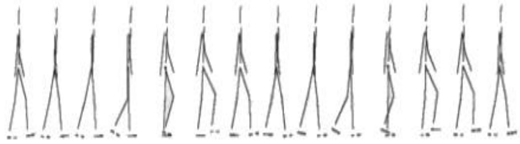


***Figure 7 Illustration of walk using limit cycle control as discussed by Laszlo et al [LAS96]***

Laszlo et al. [LAS96] [LAS97] also discuss an approach that adds closed-loop control to open-loop motions to control simulated walking (illustrated in figure 7). The animator provides the open loop control, treating walking like the repetitive motion of a wind-up toy. Limit cycle control is then used to automate the calculation of the control adjustments required to maintain balance. Walking direction and speed are kept under the animator's control. The paper makes note of two simplifications: a linear predictive model which can be applied to motions such as walking, despite their non-linearity, and the introduction of a reduced state vector which reflects changes in the motion of the majority of the body mass. The walk controllers contain a number of parameters that need to be tuned by hand.

Gritz and Hahn [GRI95] use genetic programs to allow the animator/user to find/evolve the appropriate controller program for the agents to achieve their goals. This work relates to that of Witkin and Kass on space-time constraints and especially to that of Karl Sims [SIM91] who used a similar approach to evolve procedural textures.

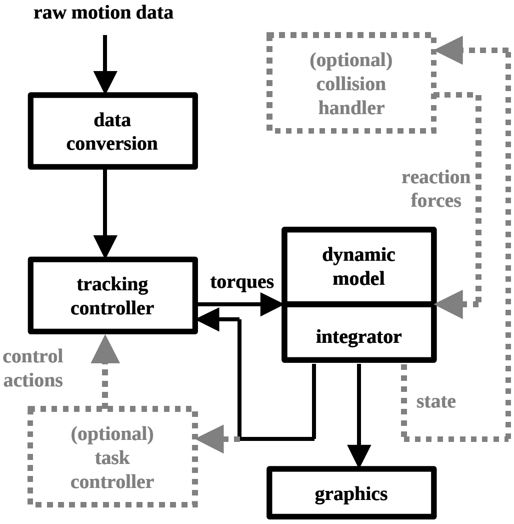


*Figure 8 Tracking system layout for system proposed by [HOD99]*

Hodgins and Zordan [HOD99] combine motion capture with dynamic simulation for the purpose of producing realistic human like motion. The system uses the motion capture data as the input to a tracking controller for a dynamic simulation. This system uses forward dynamics and control algorithms rather than inverse dynamics and torque minimization. Figure 8 provides a system layout of the proposed system. Raw motion data is converted into continuous joint angles and used by the tracking controller. The tracking controller uses this data to calculate torques, which is then integrated to generate motion. An optional task controller and collision handler may be added to this system to achieve more complex behaviours.

## 3.3 Enhancing Motion

Although there are many ways to generate character motion, most methods share two common problems. Firstly, because of the mathematical nature of the methods used, formulaic motions are often obtained; cyclic motions look robotic in nature, and do not contain the human 'flaws' necessary to convey a realistic movement to any human viewer. For the average human being, no two steps are ever the exact same The human visual system is perhaps the most critical observer of human motion, since it is well versed in observing normal day-to-day human motions.

The second problem is that of speed. For dynamic motion generation, complex and time-consuming calculations must be performed. Speed becomes a particularly critical concern when simulating crowds of virtual characters.

The remainder of this section reviews research into possible solutions to these problems.

### 3.3.1 Noise

Perlin [PER95] uses principles from procedural texture synthesis to create subtle human movements such as shifting of weight and fidgeting while standing. Rhythmic and stochastic noise functions determine time varying parameters that drive characters. Computation of dynamics and constraint solvers is avoided since only the 'texture' of motion needs to be conveyed. Expressions are tuned to contain pseudo random noise. The fundamental motion of each joint is sinusoidal with additive noise used to prevent the motion from becoming repetitive.

Bodenheimer et al. [BOD99] introduce natural looking variability into cyclic animations of human motion. Noise functions that are based on biomechanical considerations are used. These are applied to a base running motion that is produced either by motion capture data or dynamic simulation and introduce natural-looking perturbations. Tests conducted on the male running simulation did not generalise well to a female running simulation. It was concluded that although some noise is preferable in an animation, attaining the most preferable noise level for a specific animation is not a trivial matter.

### 3.3.2 Level Of Detail

Because dynamic simulation methods are inherently expensive, a number of techniques have been investigated for reducing the computational cost involved in real-time animation. As mentioned, these techniques are critical when simulating a large number of characters in the same scene.

Carlson and Hodgins [CAR97] explore techniques for the reducing the computational cost of simulation by using less accurate simulations for characters when they are less important to the viewer or to the action in a virtual world. Rigid-body dynamics was chosen as the highest level of detail, followed by point-mass simulation with kinematic joints and finally the lowest level of detail was a point-mass simulation with no kinematic motion of the legs. Using these methods, better real-time performance for a larger group of creatures can be achieved than would be possible if each creature were dynamically simulated. Carlson and Hodgins outline two criteria for simulation level of details: Firstly, the outcome of the simulation must be essentially unchanged. Secondly, the viewer's perception of the motion must be the same. It was noted that the main problem during tests was meeting the first criterion.

Chenney et al. [CHE99] suggest animation levels of detail according to computation frequency. Each animation level of detail has a specific interval of frame rate. The high frequency model is a "classic" mechanical model, whilst a neural network is used for low frame rates.

Cozot et al. [COZ99] examine a pipeline of sub-models. Each sub-model performs a given task in the animation process: animation of the body or the legs for example. Several methods are encapsulated for a given task with different computation costs. The most suitable animation method is selected according to two criteria: the complexity of the environment and the distance from camera. The process facilitates transitions between one level and another while ensuring coherency between the motions captured by the different levels. Switching from one level to another consists of adding a module to, or retrieving a module from this pipeline. As remarked by Carlson and Hodgins, the difficult part of animation levels of detail is ensuring that the different models produce coherent outputs.

## 4. Conclusion

It is clear from the literature reviewed in this paper that there are many choices available when implementing an animation system. However, when considering a method for implementing realistic, real-time character animation, it becomes apparent that the field may not be as broad as it may seem.

In the areas examined for character deformation it is clear that each has its own advantages and disadvantages. However, FFD's undoubtedly offer the most versatility for muscle deformation, although implicit surfaces will be examined in further detail. Mass-spring systems also offer a wide range of applications and have a particular suitability for surface and outer layer modelling e.g. skin, hair, fur and clothing. Further research will derive how many layers will be appropriate for the intended character model and therefore will dictate whether mass-spring systems will be useful in the proposed system.

Although space-time constraints offer an interesting and powerful way to generate motion, the complexity of generating real-time motions for something as complicated as a virtual human is currently too high to contemplate using. Space-time constraints appear to have more useful properties for evolving motions for general articulations rather than proving realistic motions for well established articulations (the human model for example). Windowed space-time constraints, however, appear to be more manageable and may prove to be useful for specific tasks. This is one area that will warrant further research.

For cyclic motions (such as walking and running), a specific motion planner based on dynamics coupled with some form of noise may offer the most pleasing solution in terms of realism and speed, despite the inherently rigid nature of any such system. Extending the dynamic nature of these specific motions (similar to the work outlined by Laszlo et al.) would be rewarding research. Coupling this with some form of simulation level of detail approach could provide a powerful system.

## References

[BAR84] *Alan H Barr, "Global and Local Deformations of Solid Primitives"*
*Proc. SIGGRAPH '84, Computer Graphics, Vol. 18, No3, pp21-30*

[BAR88] R. Barzel, A.H. Barr, *"A Modelling System Based on Dynamic Constraints"*, Computer Graphics, Vol. 22, No. 4, 1988, pp179 – 188.

[BRE91] D.E. Breen, D.H. House, Phillip H. Getto. " A Particle-Based Computational Model of Cloth Draping Behavior", Scientific Visualization of Physical Phenomena (Proceedings of CG International '91), pp. 113-134 (1991).

[BRU89] A. Bruderlin, T.W. Calvert, *"Goal-Directed, Dynamic Animation of Human Walking"*, Proc. SIGGRAPH'89, pp233-242.

[BRO88] LS. Brotman, A. Netravali, *"Motion Interpolation by Optimal Control"*, Computer Graphics, 22 (4), Proc. SIGGRAPH '88, 1988.

[BAD93] N.I. Badler, C.B. Philips, B.L. Webber, *"Simulating Humans"*, Oxford University Press, 1993.

[BOD99] B. Bodenheimer, A.V. Shleyfman, J.K. Hodgins, *"Effects of Noise on the Perception of Animated Human Running"*, EG CAS 99, 1999.

[CAR97] D.A. Carlson, J.K. Hodgins, *"Simulation Levels of Detail for Real-time Animation"*, Proceedings of Graphics Interface '97, pp 1-8, 1997.

[CH97] D.A. Carlson, J.K. Hodgins, *"Simulation Levels of Detail for Real-time Animation"*, Proceedings of Graphics Interface '97, pp 1-8, 1997.

[CHA89] John Chadwick , David Haumann, Richard Parent, *"Layered Construction for Deformable Animated Characters"*, Computer Graphics, vol.23, no 4,1989, pp.243-252 (Proceedings SIGGRAPH '89)

[CHE99] S. Chenney, J. Ichnowski and D. Forsith, *"Dynamics Modelling and Culling"*, IEEE Computer Graphics and Applications, pp. 79-87, March 1999.

[COO98]
Lee Cooper, *"Physically Based Modelling of Human Limbs"*, Ph.D. thesis 1998, Dept. Computer Science, University of Sheffield, England

[COQ90] *S. Coquillart, "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modelling", pp.187-196 (SIGGRAPH 90)*

[COQ91] *S. Coquillart,P Jacene, "Animated Free Form Deformation: An Interactive Animation Technique", Proc. SIGGRAPH '91 (Las Vegas, July 28-August 2 1991)*

[CRE99] Benoit Crespin, *"Implicit Free Form Deformation"*, Proc. Fourth International Workshop on Implicit Surfaces, 1999, pp 17-23.

[DIS99] J. Dischler, "A General Model of Animated Shape Perturbation", Graphics Interface '99, pp. 140-147 (June 1999).

[GAS91] MP Gascuel, A Verroust, C Puech , *"A modeling System for Complex Deformable Bodies Suited to Animation and Collision Processing"* The Journal of Vizualization an Computer Animation, 1991, Vol.2, No3, pp 82-91.

[COZ99] R. Cozot, F. Multon, B. Valton, B. Arnaldi, *"Animation Levels of Detail Design for Real-time Virtual Humans"*, EG CAS 99, 1999.

[DEL98] B. Delaney, *"The Mystery of Motion Capture"*, IEEE CGA, 14-19, September, 1998.

[GRI95] L. Gritz, J.K. Hahn, *"Genetic Programming for Articulated Figure Motion"*, 1995.

[GLE96] M. Gleicher, P. Litwinowicz, *"Constraint-Based Motion Adaption"*, Advanced Technology Group, Apple TR 96-153, June 1996.

[HOD95] J.K. Hodgins, W.L. Wooten, D.C. Brogan, J.F. O' Brien, *"Animating Human Athletics"*, Proc. SIGGRAPH'95, 1995.

[HOD98] J.K. Hodgins, J.F. O'Brien *"Computer Animation"*, Encyclopaedia of Computer Science, 1998.

[HOD99] V.B. Zordan, J.K. Hodgins, *"Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation"*, 1999.

[KOG94] Y. Koga, K. Kondo, J. Kuffner, J.C. Latombe, *"Planning Motions with Intentions"*, Proc. SIGGRAPH'94, 1994.

[LAS87] J. Lasseter, *"Principles of Traditional Animation Applied to 3D Computer Animation"*, Computer Graphics, 24(4). Proc. SIGGRAPH 87, 1987.

[LIU94] Z. Liu, S.J. Gortler, M.F. Cohen, *"Hierarchichal Spacetime Control"*, Proc. SIGGRAPH'94,1994.

[LAS96] J. Laszlo, M.V.D. Panne, E. Fiume, *"Limit Cycle Control And Its Application To The Animation Of Balancing And Walking"*, 1996.

[LAS97] J.F. Laszlo, M.V.D. Panne, E. Fiume, *"Control of Physically-based Simulated Walking"*, Proc. IMAGINA '97 Monaco, February 19-21, 1997, Monaco, 231-241.

[LEE90] P. Lee, S. Wei, J. Zhao and N.I. Badler, *"Strength Guided Motion"*, Computer Graphics, Vol. 24, 1990.

[MUL98] F. Multon, L. France, M-Paule Cani-Gascuel, Gilles Debunne, *"Computer Animation of Human Walking: A Survey"*, June 1998.

[MAS94] R. Mas, D. Thalmann, *"A Hand Control and Automatic Grasping System for Synthetic Actors"*, Proc. Eurographics'94, Oslo, Norway, 1994.

[NGO93] J.T. Ngo, J. Marks, *"Spacetime Constraints Revisited"*, Proc. SIGGRAPH'93, 1993.

[PAI90] D. Pai, *"Programming Anthropoid Walking: Control and Simulation"*, Cornell Computer Science Tech Report TR 90-1178.

[PER95] K. Perlin, *"Real-time Responsive Animation with Personality"*, IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 1, March 1995.

[PAN93] M.V.D. Panne, E. Fiume, *"Sensor-Actuator Networks"*, Proc. SIGGRAPH'93, 1993.

[PNT98] Luciana Porcher Nedel, Daniel Thalmann, *"Real-Time Muscle Deformations Using Mass-Spring Systems"*, Computer Graphics International'98, June, 1998

[REE83] William T. Reeves, *"Particle Systems: A Technique for Modeling a Class of Fuzzy Objects"*, Proc SIGGRAPH '83, Computer Graphics, Vol. 17, No3, pp359-376

[REE85] William T. Reeves, Rick Blau, *"Approximate and Probabalistic Algorithms for Shading and Rendering Structured Particle Systems"*, Proc SIGGRAPH '85, Computer Graphics, Vol. 19, No3, pp313-322

[RIJ91] H. Rijpkema, M. Girard, *"Computer Animation of Knowledge-based Grasping"*, Proc. SIGGRAPH'91, Computer Graphics, Vol. 25, No.4, 1991, pp339-348.

[ROS96] C. Rose, B. Guenter, B. Bodenheimer, M.F. Cohen, *"Efficient Generation of Motion Transitions using Space-time Constraints"*, Proc. SIGGRAPH'96, 1996.

[SCHE 97] Ferdi Scheepers, Richard E. Parent, Wayne E. Carlson, Stephen F. May, *"Anatomy-Based Modeling of the Human Musculature"*,

[SEDE 86] T. W, Sederberg, Scott R. Parry , *"Free Form Deformations of Solid Geometric Models*", pp.151-60 ( Proceedings SIGGRAPH '86)

[SHEN] Jihanua Shen, Ph.D. proposal, *"Human Body Modeling and Deformation"* http://ligwww.epfl.ch/~shen/bb.html, Swiss Federal Institute of Technology, Luasanne, Switzerland.

[SIM91] Karl Simms, "Artificial Evolution for Computer Graphics", Computer Graphics, 24(4), 405-13, Proc. SIGGRAPH 1991.

[THAL 88] Nadia Magnenat-Thalmann, Richard Laperriere, Daniel Thalmann, *"Joint-Dependant*

*Local Deformations For Hand Animation and Object Grasping"*, Proc. Graphics Interface '88, Edmonton 1988.

[THA95] D. Thalmann, N.M. Thalmann, "*Finite Elements in Task-Level Animation"*, FEM95-J, Switzerland, 1995.

[TUR93] *Russel Turner, Ph.D. Thesis 1993, "Interactive Construction and Animation of Layered Elastic Characters", Swiss Federal Institute of Technology, Luasanne, Switzerland.*

[WIT88] A. Witkin, M. Kass, "*Spacetime Constraints"*, Proc. SIGGRAPH'88, 159-168, 1988.

[WIT95] A. Witkin, P. Zoran, "*Motion Warping"*, SIGGRAPH'95, pp. 105-108, 1995.

[ZHA94] J. Zhao, N. Badler, "*Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures"*, ACM Transactions on Graphics, 13(4), October 1994.