Abstract

This paper presents research in enhancing character animation using motion capture data and retargeting motion from one character model to another. Our particular interest lies in animating non-human characters and inanimate objects. The main goal of the project is to use lattice deformations to animate an object and smooth out the kinks in the animation at the bending of the joints.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Animation

Keywords: character animation, motion capture, motion retargeting, deformation

1 Introduction

The greatest advantage of animation over traditional live action cinema is its flexibility in allowing an animator to bring to life characters and scenes that are otherwise impossible to capture or construct. How does one film a dancing teapot? Or a chair performing a cartwheel? One of the greatest aids in the process of modern computer animation is the ability to record human motion and apply it to 3D characters. The actors chosen for such motion capture sections can skillfully deliver emotions through their body movement. This work aims to leverage actor’s talent with animator’s skills by preserving the subtle expressive motions recorded in the motion capture data. The focus of the project is on animating inanimate objects with this data and on enhancing character animation using various deformation techniques.

2 Background

The project is implemented using the professional tools that allow producing animations using motion capture data.

2.1 Motion Capture System

The motion capture data used in this project was obtained using the equipment from the Carnegie Mellon Motion Capture Laboratory [3].

Figure 1: Flower animated using lattices.

The motion is captured using a Vicon motion capture system that utilized 12 infrared MX-40 cameras [9]. Each camera can record data in a space of approximately 3m by 8m at 120Hz.

For the motions recorded in this project the actors wore a suit equipped with 52 strategically placed markers. The markers were attached to
the joint axes and bone areas for the cameras to easily identify their 3D position. As a calibration step, the subject was asked to perform various motions (a T-pose, a "motorcycle" pose and a range of motion) to move each joint through its full range of motion. The Vicon IQ software uses the collected data to automatically calculate appropriate limb lengths for the virtual skeleton resembling the actor.

Using the created skeleton, the motion is further processed and stored in the V file format, which contains information about the absolute root position and orientation, and the relative joint angles of 18 joints. This motion data is then cleaned up to include the occluded markers and is imported into Autodesk® Maya® software [1].

2.2 Autodesk® Maya® software

Maya® is an incredibly powerful 3D modeling and animation software. Its toolkit can be expanded by writing custom scripts in the Maya Embedded Language (MEL) or Python®. For this project the author used Maya’s latest release 2008 (9.0) on a Windows XP platform. To animate a 3D model using motion capture data, the author used MEL scripts that imported .v (motion data) and .vsk (skeleton information) files into Maya®, and constrained the model to the skeleton. The majority of the work explored Maya’s retargeting capabilities and the various approaches and options an animator may choose from.

3 Related Work

There has been relevant work in computer graphics aimed towards retargeting human motion to character models. Gleicher [6] developed a solver, which preserved desirable qualities of the motion and used space-time constraints to allow interactive control of the animation via a user interface. Hecker et al. [7] extended this approach spearheading a unique method of user-generated characters with drastically varied morphologies while preserving the style of motion. The works by Li et al. [8] and Noble et al. [10] introduce methods of including expressive deformations into 3D motion capture animations.

The lattice deformation was presented by Coquillart et al. [4] as part of the Free-Form Deformation (FFD) technique. The paper shows the basics of the lattice deformations and how it can be integrated with traditional animation tools. Faloutsos et al. [5] introduced FFDs as a way of dynamic character animation as opposed to traditional key-frame animation. Capell et al. [2] utilized control lattices as a way of turning interactive simulations into skeleton-driven animations. Their paper also outlines a notable body of work related to FFDs.

4 Character Retargeting in Maya

One of the challenges the animator faces in using motion capture data in the production of an animation is the necessity to audition the actors whose body type and size matches the proportions of the intended character. One of the goals of this project is to investigate Maya’s retargeting capabilities in transferring the motion of any actor to a 3D character. In writing of this section the author used the Autodesk® Maya® help manual for reference.

Before retargeting the motion it is necessary to label or rename the target skeleton’s joints to match the names of the source skeleton’s joints. Any joints in your skeletons that are not labeled or renamed will not be included in the retargeting process, which will render them immobile.

In Maya retargeting of a character is performed by selecting all of the source skeleton’s joints and the corresponding target skeleton’s joints. The retargeting solver transfers the motion of the source skeleton to the corresponding, labeled joints of the target skeleton. The Retarget Options window (Skeleton > Retargeting > Retarget Skeleton >) allows fine control over the solver’s options.
4.1 Neutral Pose Options

Neutral pose options specify whether the animator set a neutral pose for both characters (Skeleton > Retargeting > Set Neutral Pose) or a pose will be specified at a particular frame. Predefined pose is the default and it was used in this project.

4.2 Maintain Offset Option

This option maintains the source and target skeleton's global positions so that the target skeleton does not move to the position of the source skeleton during retargeting. Having both characters side-by-side is helpful during the comparison when playing back the original and the retargeted motions.

4.3 Lower Body Options

The following options allow the animator to select how the lower body animation of the source skeleton is transferred to the lower part of the target character.

"Joint Rotation Only" option preserves the joint rotations. It was not used for this project since this project utilized the 3D models whose bone lengths were different. Instead "Scaled Foot Placement" option was used as it retargets all animation data. It is recommended to use this option when retargeting from one skeleton to a completely different skeleton (such as we did in this project by retargeting human motion to a flower model). This is the default option and it considers the position and orientation of the source skeleton's feet. By orienting and translating the target skeleton's feet, the resulting motion looks more natural as the model's feet move in relation to its proportions.

When this option is on the animator has a choice of re-scaling lower body so that the motion is best retargeted to the corresponding leg joints of the target skeleton. The options include Hip to Foot, Hip to Toe, Overall Height and a Lower Scale Factor, which allows one to input a custom scale factor for the target's lower-body animation.

"Absolute Foot Placement" option ensures precise placement of the target skeleton's feet. This option can be used when it is necessary for the target skeleton's feet to land at very specific locations, for example, if the target character is interacting with another object (soccer ball or a hand-shake sequence).

4.4 Upper Body Options

The options for upper body are similar to the lower body options in that they allow the animator to control the end effector locations and scale. "Joint Rotation Only" and "Scaled Hand Placement" options allow the animator to preserve the rotations of the arm joints and retarget the arm animation based on the character proportions respectively. In addition to scaling the motion from Shoulder to Hand, Shoulder to Wrist and by specifying a custom scaling factor, the animator can choose "Scale Hand From" option that identifies the pivot used for retargeting the target skeleton's hands with respect to the source skeleton's hands.

4.5 Time Range

It is possible to retarget the motion within the selected time range. The range can either be set to the interval selected on the timeline, the current range of the timeline or to the specified Start Time and End Time.

5 The Deformations

The main idea of this project is to use various deformations to enhance motion capture assisted character animation. Two deformation techniques were chosen to be compared and contrasted: lattice deformations and spline IK deformations. The focus of this paper is on lattice deformations and lattice-assisted animation. To give the reader an idea about what spline deformations are, I will just say that spline IK allows an animator to use a curve to control several joints, especially if they are connected together. This technique eliminates the necessity to separately move and rotate each joint.
5.1 Lattice Deformations

A lattice deformation is a free-form deformation, which is applied to any deformable object (i.e. you cannot apply it to joints). When first created, a lattice resembles a rectangular box of points that encompasses an area selected by an animator. If the points of the lattice are modified, then consequently they will affect the underlying mesh.

A lattice is composed of two parts: a base and an influence lattice. When an animator manipulates the “lattice”, the influence lattice is the one that is being altered, since the base lattice is hidden. The deformation effect is achieved by computing the difference between the points of a base and an influence lattice.

The lattices in turn are deformable objects, which means that one can apply deformer objects to a lattice and thus modify the mesh. This paper explores this property and shows how a character can be animated by binding a lattice to the skeleton driven by the motion capture data.

5.2 Creating a Lattice Deformer

Unless specified otherwise the default settings will be applied when a lattice is created using the Deform > Create Lattice menu. It is suggested that the user customizes the lattice creation menu by adjusting the following basic options:

5.2.1 Divisions

The lattice is composed of S, T and U divisions. By changing the number of divisions in these categories, an animator is changing the number of lattice points. The default settings are two divisions for S and for U, and five divisions for T, totaling in 16 lattice points. By increasing the number of lattice points, the animator may gain a better control over the deformation, yet the performance may decrease.

5.2.2 Local Mode

Local mode determines if each lattice point can influence only the deformable object's points that are next to it (local mode is on), or can influence all the mesh's points. It is on by default, and the animator can specify Local Divisions. For this project, turning the local mode off produced chaotic deformations because the character's arms were initially away from the body but as they got closer to the other parts of the body, the lattice influenced their points as well.

5.2.3 Local Divisions

Local divisions determine how far the lattice's influence is propagated. By default, S, T and U have two divisions, which means that each lattice point affects the mesh's points that are at most two divisions away from it in each direction.

The animator has to be mindful about how close the lattice comes in contact with the other points of the mesh that do not need to be deformed. For example, if an animation includes a hand with a pencil that will write something on a sheet of paper, then if the pencil is affected by the lattice whose local divisions are set too high, then the lattice will also affect the points of the paper's mesh, possibly creating undesired deformations.

5.2.4 Positioning

The lattice can be centered around the selected object, or set at the origin of workspace. For this project, the default option with centering the lattice was used. Otherwise, if the lattice is positioned at the workspace origin, the object is deformed only when it moves through the lattice.

5.2.5 Grouping

This option allows grouping the influence and base lattice together. By doing so, it is possible to transform both of them at the same time. It is off by default, since if both, the influence and
base lattice, are transformed simultaneously, then there is no difference between them to create deformation.

5.2.6 Parenting

The parenting option lets the animator parent the lattice to the selected deformable object at the lattice creation time. By parenting them, both will be transformed in unison. By default this option is off, which was the selection used in this project.

5.2.7 Freeze Mode

This option is off by default and was not used in this project. If the components of the object are frozen, then they remain inside of the lattice even if the object itself is transformed.

5.3 Skinning With Lattices

Before adding a lattice, it is necessary to ensure that the mesh’s topology does not need to be altered any further. Changing the number of points on a character’s mesh after the lattice has been applied may lead to undesirable deformation effects.

6 The Process and Results

The retargeting process is essential in animating inanimate objects using motion capture data. This project included multiple stages during which various lattice application options were explored.

For some of the stages, the researcher asked people who are not involved with the project (referred to as “a user”) to identify the motion used in the resulting animation (e.g. walking, jumping, etc.), and to evaluate how effective the animation was in conveying the emotional context of the motion (i.e. sad, happy, etc.).

6.1 Initial Work

Initially, a simple cylindrical model of a pencil (referred to as “character”) was used to illustrate the basic ideas of lattice-assisted animation. The character was first animated using the motion of a person walking with both feet, only using the skeleton’s right foot. A lattice over the entire character, which was fixed to the skeleton’s head and the foot, did not produce the desired smooth bending motion (see Figure 2). The portion of the body affected by the lattice was compressing in response to the skeleton’s motion instead of bending with it, as was desired. Sub-dividing one lattice into three different lattices produced an animation, which followed closely the original movement of the skeleton.

![Figure 2: One lattice fixed at the head and the foot.](image)

When animating a pencil using human motion, the appearance of the rough edge around the knee area, made the animation look too fake. Thus, one of the major results of this project was the discovery that the smoothness of the joint areas is essential in creating a natural looking animation.

In order to get rid of the rough borders in the mesh, a few edges positioned close to each
other were added to the character’s mesh. Since the edges that are influenced by a lattice move in response to the motion, it is possible to position the lattices very close to each other to have a smoother transition between the edges. However, in the resulting animation one could still notice a narrow strip, which was not under the influence of any of the lattices.

During this animation, I discovered that if the lattices overlap, they interfere with one another and deform the character chaotically.

Figure 3 shows one of the possible solutions: it’s possible to use a full-body lattice with a larger number of subdivisions and apply smoothing to the areas around the joints. The other solution is to use separate lattices and smooth out the areas that are not affected by the lattices.

Figure 3: (Left) A lattice with 25 subdivisions. (Right) Same lattice - bent areas are smoothed.

This example also showed the importance of the application of the appropriate motion: since the motion of only one leg was used, the animation looked perplexing and a user could not identify that the pencil was supposed to be “walking.”

For the next animation, a motion of a jump with both feet together was applied to the pencil model. In addition, a simple checkered pattern was applied to the mesh. The resulting animation was recognizable, and identifiable by a user. As shown on Figure 4, the texture seemed to be unaffected by the lattice deformation.

Figure 4: Top - an area around a joint without a lattice. Bottom - same area affected by a lattice.

6.2 Flower Animation

The major body of the work was done with the flower model. The flower was animated using the lattices and the IK-splines. The subsequent animations were then compared to each other and to the original human motion to find out which one followed the motion the closest and had the most natural looking animation.

The flower was animated using both techniques with the following motions: an actor was asked to act like a sad, motherly, talking and walking flower. Two different actors with different proportions were asked to portray a happy flower.
Figure 5: Left - flower animated using lattices. Middle - original motion’s skeleton. Right - flower animated using IK-splines.

Figure 5 shows screenshots of the animation at the exact same frame. The original motion capture skeleton is shown next to the animated flower models. It is evident, even just by looking at the screen captures, that the IK-splines are more pliant, providing a more elastic smooth animation (especially around the joints).

It was concluded that IK-splines are better for animation of bendable and curvy objects as opposed to the lattices that are more appropriate for bulky and heavy models.

6.3 Issues Discovered

During the work on this project, a few problem areas have been revealed. The retargeting of the human motion to a character with different body structure often introduces self-collisions. For example, the waiving of the arms close to the actor’s head in our case turned into the leaves going through the flower’s petals. It is possible to eliminate the collisions by tweaking
the original motion before the retargeting. However, it is a tedious manual process, which would benefit greatly from automation by detecting the contact between the various parts of the mesh.

Also, for this project the motions started with the T-pose (described in the Introduction). If the motion does not start with the T-pose, then for some models it can be hard to select vertices to which to apply a lattice (e.g. when the leaves are touching the pot).

The local influence of a lattice (see section 5.2.2) is a major attribute an animator has to keep in mind. For instance, a manipulation of lattice’s local divisions allowed subtle deformations in the flower’s petals. By increasing the local influence of the lattice points, an effortless effect of slightly curved petals was achieved. The petals curled slightly every time the flower bowed or swung from side to side. Despite the fact that the original actor’s motion did not have this feature, the lattice-assisted animation resulted in this inadvertent benefit, and allowed the addition of this detail to be automatically synchronized with the motion.

Ultimately, it was discovered that the resulting animation is highly dependent on the animator and his or her vision of the final product. The final system should be rigid enough to conform to the animator’s exact specifications of how the motion is supposed to be transferred and whether or not inadvertent benefits are acceptable. Depending on their choice and the desired effect, the system may choose to encompass the entire mesh into a lattice or just the parts that deform.

7 Future Work

This work can be expanded in many different directions because this project was intended to be the preliminary ground for continuous research in this area. To further support the results of this project it would be beneficial to animate a few large and weighty characters using lattices. Other lattice options can be explored. For instance, one may resize the base lattice (try with it smaller first, then larger than the influence lattice) and note the effect of the deformation. What is the scale for which a certain effect is achieved?

As was mentioned earlier, this work can be extended by automatically detecting and preventing self-collisions.

In this paper, the lattice deformation was the main animation tool. For the final product, however, it would be beneficial to combine lattice-assisted animation with an IK-spline animation to achieve smooth and natural looking motion of bending of the joints.

Acknowledgment

This work was supported by the Computing Research Association’s Distributed Mentor Project (CRA DMP). I am grateful to Dr. Jessica Hodgins and Alla Safonova for their guidance on the project. The author also wishes to thank Laurel Bancroft, Moshe Mahler and Elise Walton for their assistance with the Maya software.

References


