

Lecture 18: Kinematics and Animation

Animation of 3D Models

In the early days of cinema 3D animation was achieved using a physical 3D model which was adjusted by hand to create each individual frame of the animation. The classic example of the use of this technique was in the movie King Kong (1933), in which the model of Kong was only about one foot high. Animation created by this technique is still popular, a recent example being Wallace and Gromit.

Computer support systems for animation began to appear in the late 1970s, and the first computer generated 3D animated full length film was Toy Story (1995). Although completely created using the computer, Toy Story, and other modern 3D animation films are still enormously costly in



Figure 1: King Kong (from Visions in B & W)

human time. Fully automatic animation is still a long way off, but computer tools to support tools have developed rapidly. These remove much of the tedious work of the animator, and allow the creation of spectacular special effects. Basic approaches are: (i) Physical Models; (ii) Procedural Methods and (iii) Keyframing.

Physical models

This approach is suitable for inanimate objects or effects where there is a simple tractable physical model. Two examples are: bouncing balls and cars driving on rough roads. In both these cases the movement can be calculated simply using Newtonian mechanics. In other cases composite models can be applied, such as spring mass damper arrays for animating fluttering flags or particle systems for creating fire smoke or water.

Procedural models

The behaviour of an object is not modelled at all, but described as a procedure. The description often takes the form of a script. This kind of approach is appropriate for easily defined behaviours in inanimate objects. One example is crack propagation in glass or concrete, which can be used in a variety of effects. Crack propagation is difficult to model physically, since it is influenced by non-uniform material properties, but easy to describe procedurally since cracks tend to follow common patterns.

Keyframes

Keyframing was a traditional method in hand drawn animation where the senior animator drew the key frames at time intervals of around one second, and some stooge would then laboriously draw all the in-between frames. Systems have evolved in which the computer calculates a number of in between positions to effect a smooth movement between the two keyframes. These can be specified using, for example, spline curves to define a path of motion for some particular points on the moving object. However, even this process is difficult to automate while maintaining a plausible movement. In particular it is necessary to (broadly) observe physical laws.

One method to constrain an animation of human or animal figures is to use a jointed skeletons. The skeleton is made up of a set of links. Each link in the chain is rigid, and the movement is constrained by the degree of freedom at each joint. The skeleton can be fleshed out in any way to create the animated figure. One early example was Luxo Jr. (1987) created by Pixar (http://www.pixar.com/shorts/ljr/theater/short_320.html). This was hailed as the first computer generated 3D animation to have the same natural appearance as the older hand crafted animation. However, it still needed considerable human input to create appealing plausible movements. The reason for this is that movements of humans and animals follow rather complex patterns which are difficult to describe.

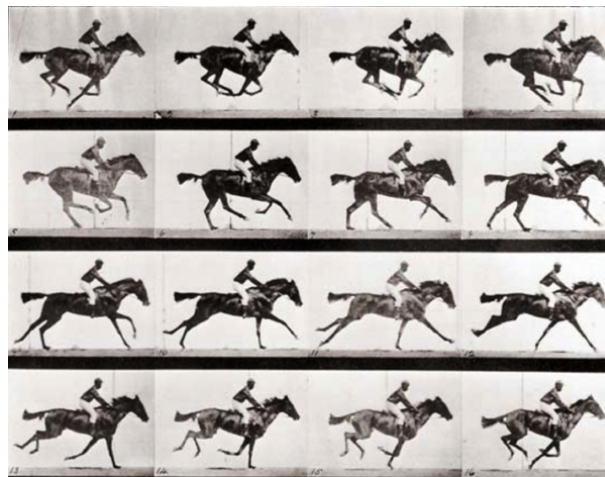


Figure 2: Galloping Horse by Eadweard Muybridge (1878)

The complexity of movement was first observed by Eadweard Muybridge using a set of synchronised cameras to capture frames separated by very short time intervals. His work was intended to create animated photographs, and was a forerunner of the modern cinema. His most famous set of photographs shows the various stages of a horse galloping (showing for the first time that all four legs do leave the ground at one point) and illustrates just how complex the movement actually is.

The complexity of natural movement makes it difficult to specify for computer based animation systems. For this reason it is cheaper to use motion capture in film special effects. Two recent examples of this were Gollum, in “The Lord of the Rings” trilogy, and King Kong in the 2005 (not quite as good) remake of the classic 1933 film. In both cases an actor is filmed making the required movements. Then key points are tracked throughout the sequence, and used to control the computer model’s movements. In the case of King Kong the actor spent time at a zoo studying the movements made by gorillas, so that he could then imitate them. This was a cheaper process than trying to specify them in a computer readable form.

Kinematics

Although motion capture is an effective way of collecting complex information about movement, it is still very human intensive and therefore costly. Thus there is still considerable interest in studying the movement of jointed chains. The work also applies to the mechanics of robot movement and is called kinematics. One of the most interesting current problems is the study of inverse Kinematics. Here we specify the movement path of say one point on a chain (for example the foot in figure 3) and then try to reconstruct the movement of all other points that create that movement given certain constraints or possibly the need to minimise the motion as a whole. Normally the links of the articulated chain are taken as rigid, and the joints are constrained by the degrees of freedom they support. Three examples of joints found in the body are illustrated in Figure 4.

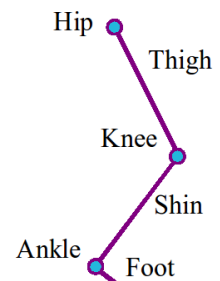
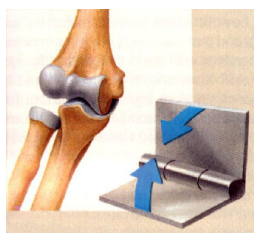
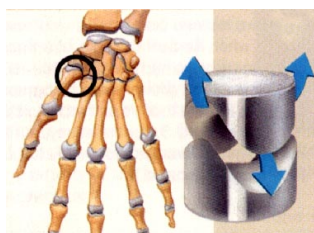


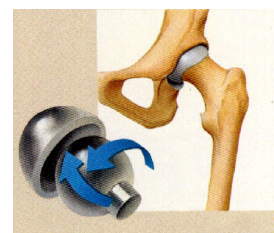
Figure 3: Articulated Chain for a Leg



(a) Hinge: 1 degree of freedom



(b) Saddle: 2 degrees of freedom



(c) Socket: 3 degrees of freedom

Figure 4: Different types of Joints (images from www.shockfamily.net)

Euler Angles

At any joint we can specify the orientation of one link relative to the other by the Euler angles. These encode pitch, roll and yaw. The links are fixed in length (rigid), so the position of end of the chain is completely defined by a set of Euler angles. For the leg shown in Figure 3 we have 3(hip) + 1(knee) + 1 (ankle) = 5 variables (Euler Angles) to determine the end point.

Euler was the first mathematician to prove that any rotation of 3D space could be achieved by three independent rotations. The rotations could be performed in many ways, but the usual convention is to: (i) Rotate about Z; (ii) Rotate about X and (iii) Rotate about Z (again) The complete Euler rotation matrix is therefore:

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) & 0 \\ 0 & -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ -\sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This formulation is not very intuitive, so to see what is happening consider transforming a vector w lying along the z axis.

The first rotation matrix will not change the vector w at all. In this case it turns the u and v directions about the w direction. This would be equivalent to a roll, or the rotation of an image about its centre if w were the viewing direction. The second matrix can rotate w to point to any position in the $y-z$ plane as shown in Figure 5(a), where the axis system is viewed from along the positive z axis.

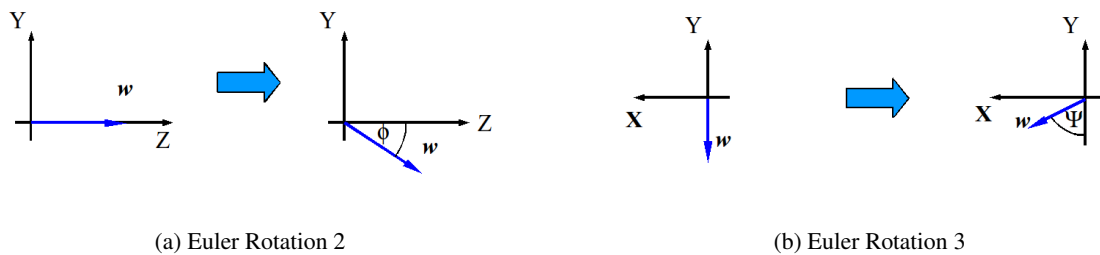


Figure 5: Euler Angles

The last rotation about the z axis can make the projection of w point in any direction in the $x-y$ plane as shown in Figure 5(b). Thus the combined transformation can orient w in any direction in the 3D space with the u and v directions aligned in any direction.

Forward Kinematics

Forward kinematics is used in robot control. Given a specification of the Euler angles of each joint or an articulated robot arm we calculate the position and orientation of its end point. This is a well posed problem and can be solved by matrix transformations of space similar to the ones that we have already seen in use - for example in first person shoot 'em ups.

We define a coordinate system at the start of a chain and at each joint as shown in Figure 6. Each new coordinate system is defined in the co-ordinate system of the previous joint. Thus the position C^i and the direction vectors $[u^i, v^i, w^i]$ of frame i are defined using the frame $i - 1$ coordinate system. C^i and $[u^i, v^i, w^i]$ can be calculated simply using the Euler rotation matrix \mathcal{R}_E :

$$u^i = [1, 0, 0]\mathcal{R}_E \quad v^i = [0, 1, 0]\mathcal{R}_E \quad w^i = [0, 0, L_{i-1}]\mathcal{R}_E$$

where L_{i-1} is the length of link $i - 1$.

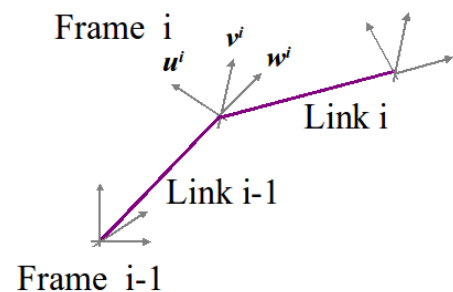


Figure 6: Forward Kinematics

We can express positions and directions in frame i in the coordinate system of frame $i - 1$ by a matrix transformation, which is the inverse of the viewing transformation discussed earlier in the course. This can be found by inspection and verified by multiplying out.

$$\begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ C_x & C_y & C_z & 1 \end{bmatrix} \begin{bmatrix} u_x & v_x & w_x & 0 \\ u_y & v_y & w_y & 0 \\ u_z & v_z & w_z & 0 \\ -C \cdot u & -C \cdot v & -C \cdot w & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To extend this to a whole chain we denote the transformation from frame i to frame $i - 1$ as:

$$T_i^{i-1} = \begin{bmatrix} u_x^i & u_y^i & u_z^i & 0 \\ v_x^i & v_y^i & v_z^i & 0 \\ w_x^i & w_y^i & w_z^i & 0 \\ C_x^i & C_y^i & C_z^i & 1 \end{bmatrix}$$

Then the complete forward transformation (using pre-multiplication) can be seen to be:

$$T_n^0 = \prod_{i=n}^0 T_i^{i-1}$$

Inverse Kinematics

For graphics applications we want the opposite process. We wish to specify some path for the end point of the chain, and calculate the relative positions of all the other links. This is an ill posed problem (there are infinitely many solutions for some chains). Hence we need to find a constrained solution minimising for example, the joint movements.

We would like to use inverse kinematics to calculate in-between frames. For example, to generate ten frames of a leg movement, we define the ten steps that make up the foot movement, and estimate the changes in the Euler angles of the rest of chain that implement those changes. In the simple articulated leg chain there are five Euler angles. The constrained angles are initialised to zero. One way to solve the problem is to use gradient descent.

Let E be the distance between the end point and its target. For each Euler angle θ we find $dE/d\theta$ using forward kinematics, replacing θ with $\theta + \Delta\theta$ and calculating ΔE . We then update the angles using:

$$\theta^t = \theta^{t-1} - \mu \frac{dE}{d\theta}$$

Applying gradient descent with small steps should find a solution that involves only small joint movements. However, over time it is still possible that a skeleton will reach an implausible pose. One approach is to constrain the optimisation to avoid poor poses. Poor pose is difficult to define, and needs analysis of videos of actual motion. This is a current topic for research.

Summary

Currently human intervention is a necessary part of creating realistic living model animation, either using motion capture or expert animator input. Physical modelling and procedural methods are successful in creating movements in inanimate objects. Inverse kinematics can take some of the burden out of model animation simulating living creatures, but is limited and is an interesting current research area.