Interactive Computer Graphics

Lecture 18

Kinematics and Animation

Animation of 3D models

In the early days physical models were altered frame by frame to create animation - eg King Kong 1933.

Computer support systems for animation began to appear in the late 1970, and the first computer generated 3D animated full length film was Toy Story (1995).

Computer Aids to Animation

Fully automatic animation has not proved successful. However computer tools to support animation have developed rapidly.

These remove much of the tedious work of the animator, and allow the creation of spectacular special effects. Basic Approaches are:

- Physical Models
- Procedural Methods
- Keyframing

Physical Modelling

This approach is suitable for inanimate objects or effects where there is a simple physical model:

- Bouncing balls
- Cars on rough roads

Effects can be created by Newtonian mechanics

In some cases composite models can be applied

- Spring mass damper arrays for fluttering flags
- Particle systems for fire smoke etc.

Procedural Approach

The behaviour of an object is described by a procedure, sometimes specified by a script.

This kind of approach is appropriate for well known behaviours in inanimate objects:

eg crack propagation in glass or concrete which is difficult to model physically but easy to describe procedurally.

Keyframe Approach

Animators specify two positions of the skeleton (keyframes) a short time apart. The computer then calculates a number of “in between” positions to effect a smooth movement between the two keyframes.

This was a traditional method in hand drawn animation where the senior animator drew the key frames and some stooge would then laboriously draw all the in betweens.

Creating In-betweens

The in-betweens are created by interpolation.

For certain objects they can be created by using paths of motion defined by, for example, spline curves.

However there are difficulties:

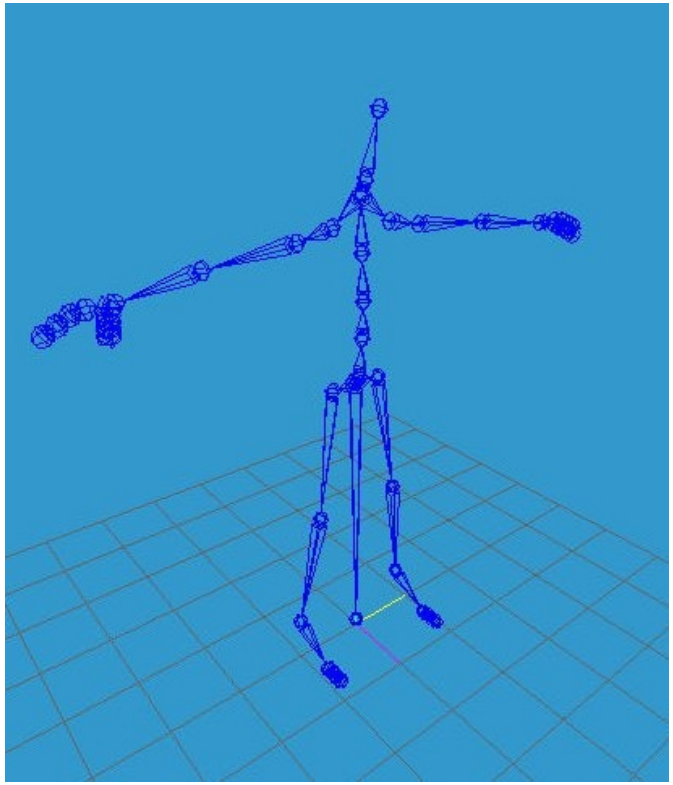
- Observing the laws of physics
- Making plausible movements

Movement Control

When animating vital characters any model animation must be kept plausible. Hence computer systems are often based on jointed skeletons.

Each link in the articulated chain is rigid. The movement is constrained by the degree of freedom at each joint.

The skeleton can be fleshed out in any way.



Luxo Jr. (1987)

This award winning short film was the hailed as the first example of computer generated 3D animation that was as natural as hand crafted animation.

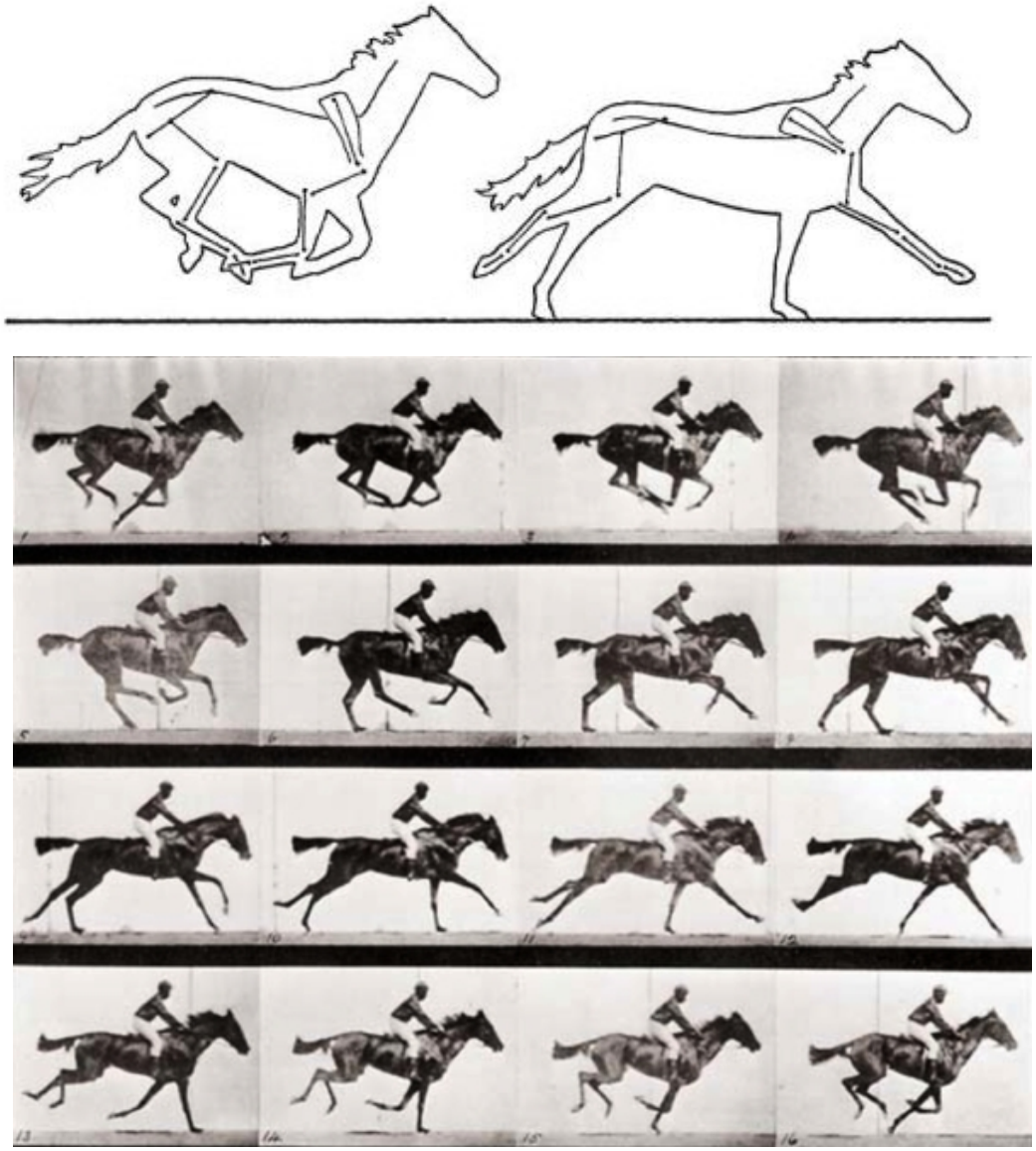
It was based on the use of an articulated skeleton.

http://www.pixar.com/shorts/ljr/theater/short_320.html

Complexities of Natural Motion

Natural motions are complex, as shown by Eadweard Muybridge's horse photographs (1878).

Some systematic approach is needed for "scripting" the motion of the skeleton.



Motion Capture

One approach to defining the movement is to use motion capture - for example Gollum in the Lord of the Rings films.

These methods require laborious data capture and are surprisingly hard to automate.

Motion Capture

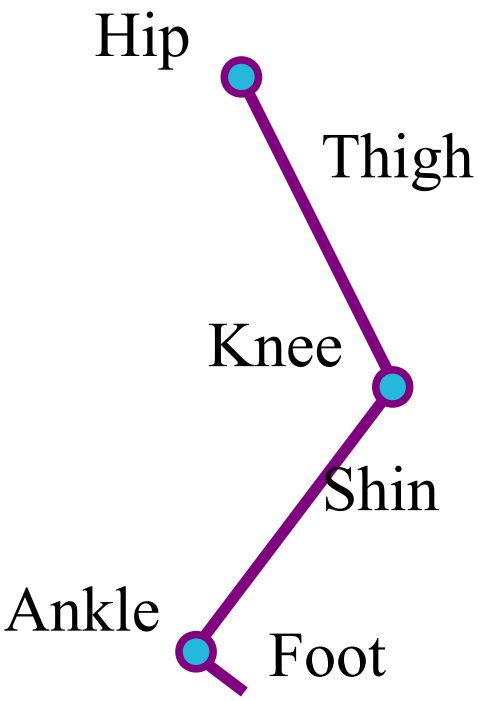
King Kong (2005) also used motion capture.

The actor wore a blue suit and was marked with around 80 identifiable spots that could be tracked by cameras.

The spots were mapped (not linearly) to corresponding points on the computer Kong model.

The actor had to study gorillas and then imitate their movements.

Kinematics



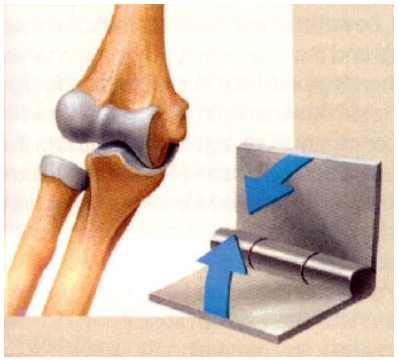
Although natural skeletons are not strictly rigid, it is a good approximation to treat them as such and thus constrain the movement.

The study of the movement of articulated chains began in mechanical engineering of robots and is called kinematics.

Degrees of freedom

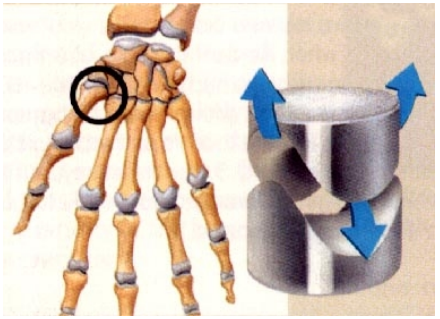
Hinge Joint - One degree of freedom

Knee or elbow joint



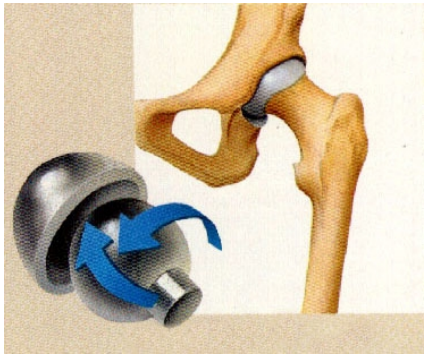
Saddle - Two degrees of freedom

Wrist/Hand joints



Socket Joint - Three degrees of freedom

Hip, shoulder neck



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 we have 3(hip) + 1(knee) + 1 (ankle) = 5 variables (Euler Angles) to determine the end point.

Euler Angles

Euler was the first mathematician to prove that any rotation of 3D space could be achieved by three independent rotations.

There is no standard way of achieving this, but the usual convention is to:

1. Rotate about Z
2. Rotate about X
3. Rotate about Z

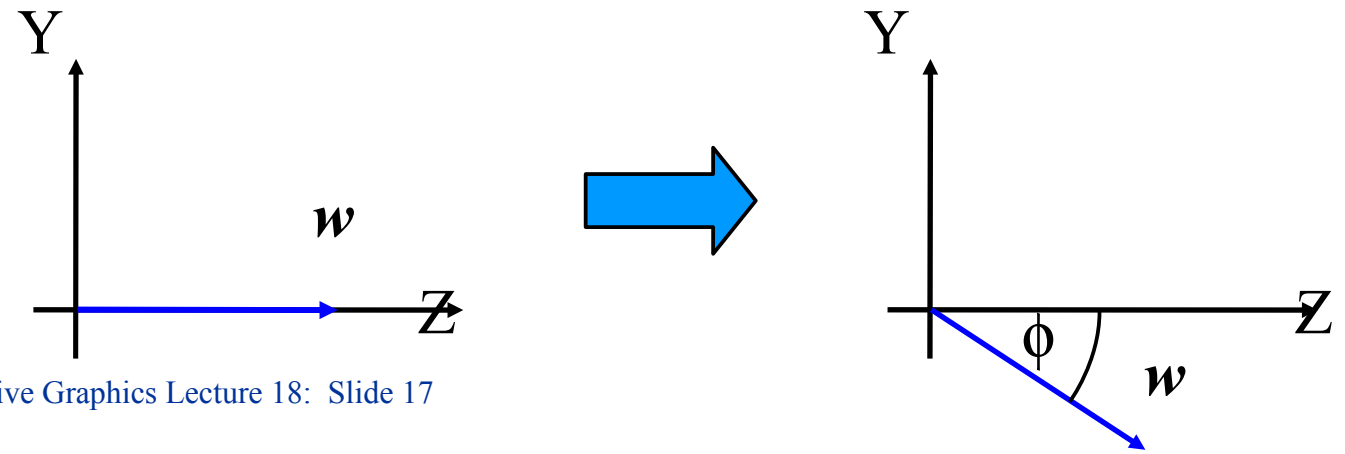
The complete Euler rotation matrix R_E is therefore:

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

Euler Angles

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

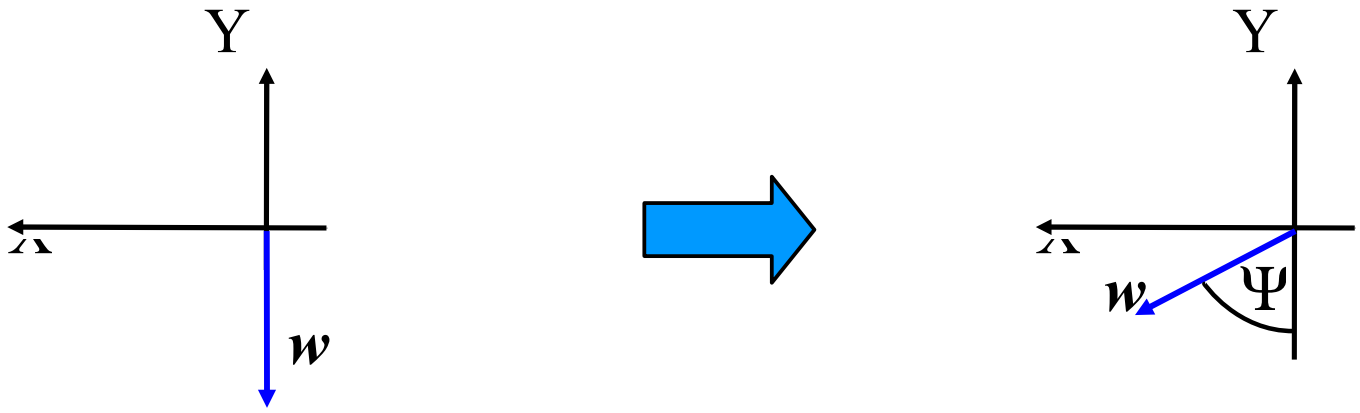
The first rotation will not change ***w*** but turns the ***u*** and ***v*** directions about the ***w***. The second matrix can rotate ***w*** to point to any position in the *y*-*z* plane.



Euler Angles

The last rotation about the *z* axis can make the projection of ***w*** point in any direction in the *x*-*y* plane.

Looking from the positive z axis we have:



So the last two rotations can orient ***w*** in any direction in the 3D space.

Euler Angles

The first rotation turns the ***u*** and ***v*** directions about the ***w***. This would be equivalent to a roll, or the rotation of an image about its centre if ***w*** were the viewing direction.

Thus the specifying the three Euler angles allows us to rotate a (***u***,***v***,***w***) axis system to any orientation we require, while preserving its orthogonality.

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.

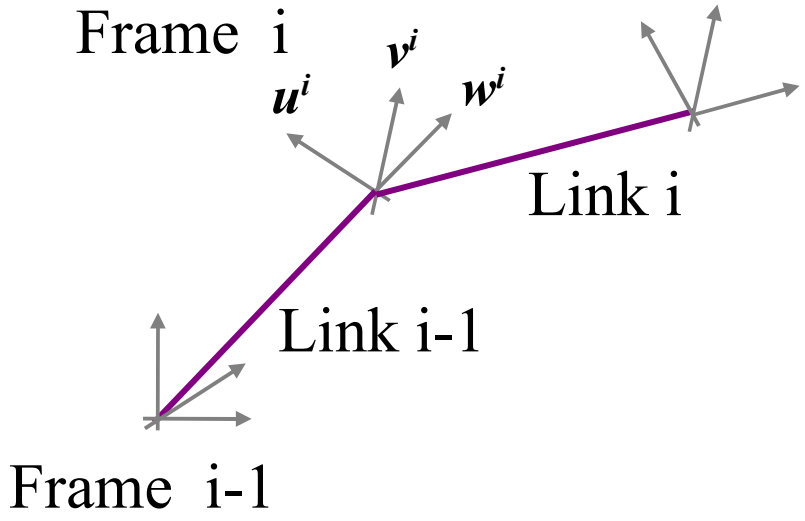
This is a well posed problem and can be solved by matrix transformations of space similar to the ones that we have already seen in use - for example in first person shoot 'em ups.

Forward Kinematics Computation

We define a coordinate system at the start of a chain and at each joint.

Each new coordinate system is defined in the co-ordinate system of the previous joint.

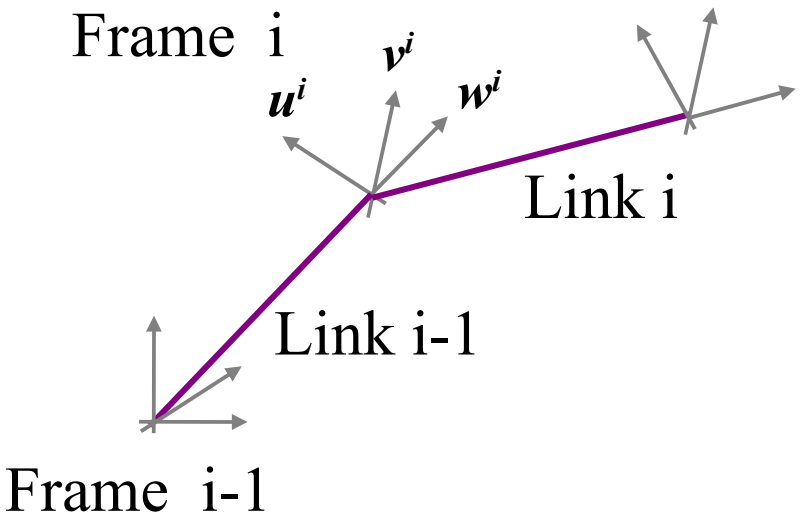
Thus the position $\mathbf{C}^i$ and the direction vectors $\{\boldsymbol{u}^i, \boldsymbol{v}^i, \boldsymbol{w}^i\}$ of frame i are defined using the frame i-1 coordinate system.



Forward Kinematics Computation

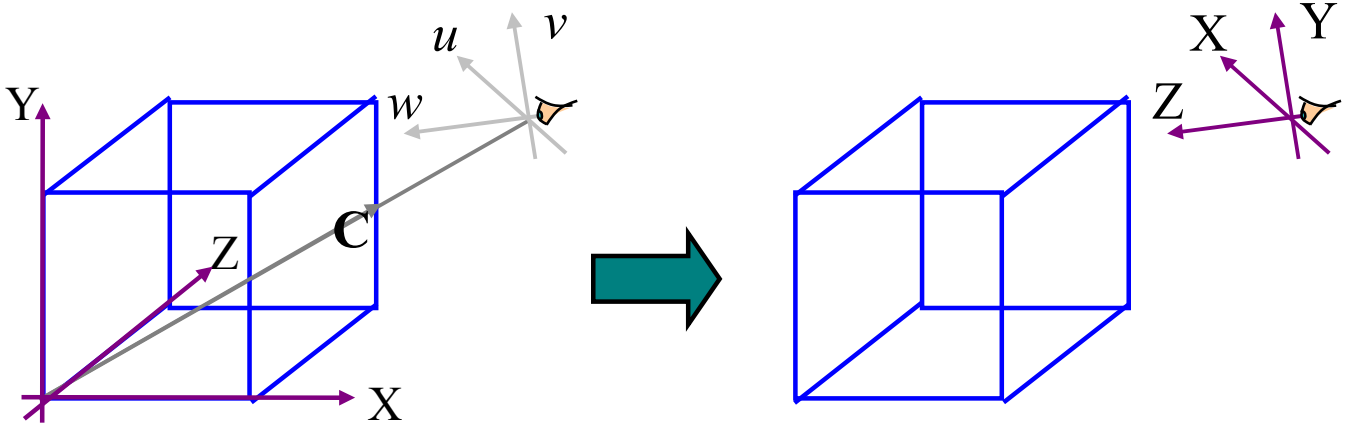
The position $\mathbf{C}^i$ and the direction vectors $\{\boldsymbol{u}^i, \boldsymbol{v}^i, \boldsymbol{w}^i\}$ of each frame can be calculated simply using the Euler rotation matrix R_E: $\boldsymbol{u}^i = [1,0,0]R_E$, $\boldsymbol{v}^i = [0,1,0]R_E$, $\boldsymbol{w}^i = [0,0,1]R_E$ and $C^i = [0,0,L_{i-1}]R_E$ where L_{i-1} is the length of link (i-1).

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.



Revision - the viewing transformation

To transform the scene coordinates to the axis system $\{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}\}$ system (using pre-multiplication) we used:



$$[P_x^t, P_y^t, P_z^t, 1] = [P_x, P_y, P_z, 1] \begin{bmatrix} u_x & v_x & w_x & 0 \\ u_y & v_y & w_y & 0 \\ u_z & v_z & w_z & 0 \\ -C \cdot u & -C \cdot v & -C \cdot w & 1 \end{bmatrix}$$

Revision - Viewing transformation

The inverse of the viewing transformation (which we need for forward kinematics) can be written as:

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

which can be verified by multiplying out:

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

Forward Kinematics Computation

If we denote the transformation from the i^{th} to the $i\text{-}1^{th}$ frame 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.

Inverse Kinematics

In practice, inverse kinematics can be used 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.

Inverse Kinematics by Gradient Descent

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:

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

Further Constraints

Applying gradient descent with small steps should find a solution that involves only small joint movements.

However, over time it is possible that a skeleton will reach an implausible pose.

One solution is to constrain the optimisation to avoid poor poses. Poor pose is difficult to define, and needs analysis of videos of actual motion.

In Summary

Currently human intervention is a necessary part of creating realistic living model animation. Either 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 living model animation, but is limited and is an interesting current research area.

Revision Sessions

MEng 3 Students

Wednesday 24th March 2010 at 10.30

All other students:

In the week beginning 3rd may

See the web page for details nearer the time.