# Animation

COM3404

Richard Everson

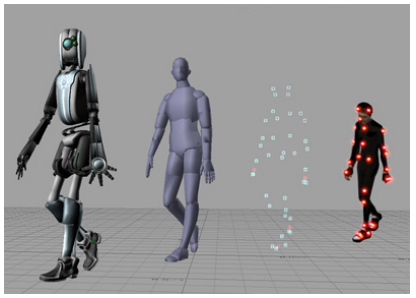
School of Engineering, Computer Science and Mathematics
University of Exeter

R.M.Everson@exeter.ac.uk
http://www.secamlocal.ex.ac.uk/studyres/COM304



## Outline





**References**

- Fundamentals of 3D Computer Graphics. Watt.
- Computer Graphics: Principles and Practice. Foley et al (1995).
- Principles of Three-Dimensional Computer Animation. M. O'Rourke.



## Animation

**Animation**
Bring to life

Moving graphics of any kind: sequences of single images

- flipbooks, kineographs
- phenakistoscope, zoetrope
- cartoon films
- computer animation

**Speed** 6 frames per second for impression of movement

- Films: 24 fps
- TV: 30 fps
- Computers: upto 100 fps


**Double buffering** Render to off-screen buffer; swap with displayed buffer



## Victorian animation

Phenakistoscope

Zoetrope

## Computer Animation

### What can be animated

| | |
|---|---|
| Objects | Size, position, rotation, colour, texture, shape |
| Camera | Position direction, field of view, focal length |
| Lights | Position, type, colour |

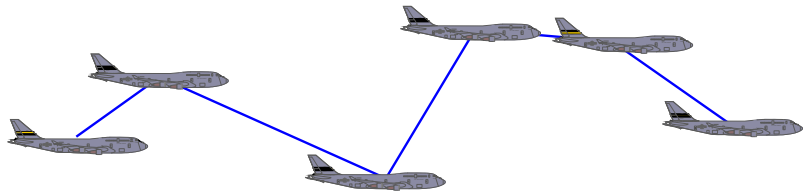
### Methods

- Key-framing/interpolation
- Motion paths
- Hierarchical systems
- Forward and inverse kinematics
- Motion dynamics
- Motion capture

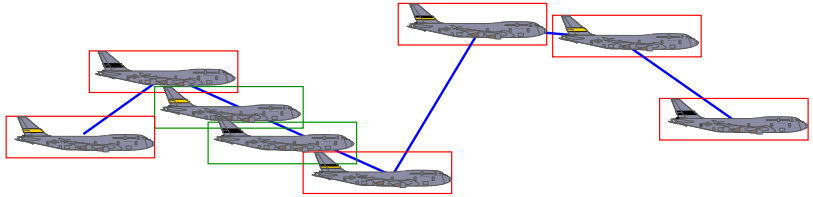


## Key-framing

### Specify positions in a few key frames



### Interpolate intermediate frames: 'in-betweening'





## Key frame interpolation

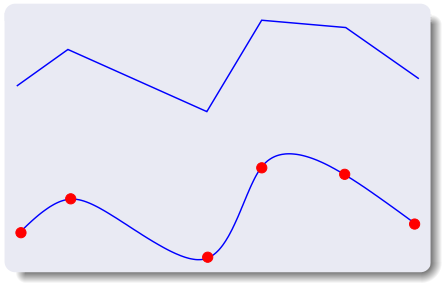
### Linear interpolation

$$\mathbf{v}(t) = \mathbf{v}_0 + \frac{t - t_0}{t_1 - t_0}(\mathbf{v}_1 - \mathbf{v}_0)$$

$$x = x_0 + \frac{t - t_0}{t_1 - t_0}(x_1 - x_0)$$

$$y = y_0 + \frac{t - t_0}{t_1 - t_0}(y_1 - y_0)$$

$$z = z_0 + \frac{t - t_0}{t_1 - t_0}(z_1 - z_0)$$



- Simple
- Motion appears jerky at the key-frames

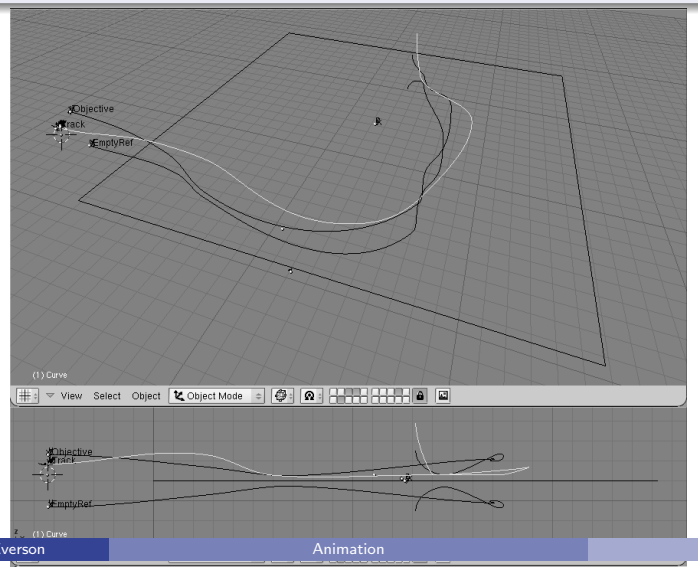
### Spline interpolation

- Hermite splines or NURBS give fine control over location
- Hermite splines guaranteed to pass through control points
- Most modelling packages allow interactive editing of motion path
- Linking of paths to permit camera to track a moving object etc.

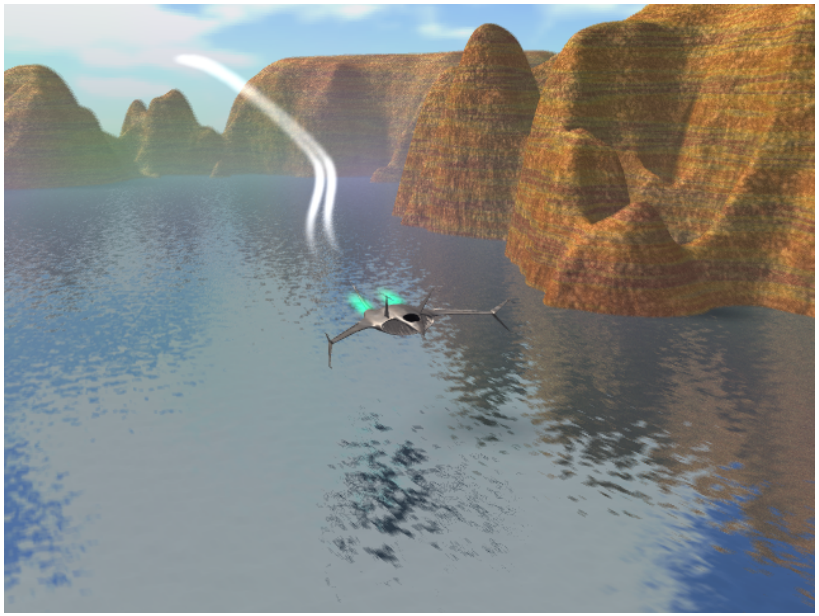


## Example: Blender

A fighter dives into a canyon, flies next to the water and rises again. The camera follows the motion and there is a reflection in the water.

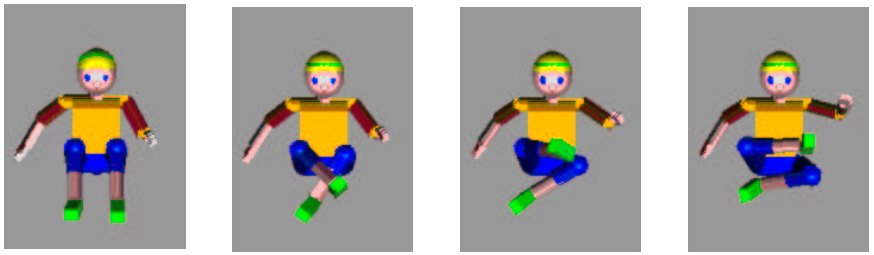


## Example: Blender

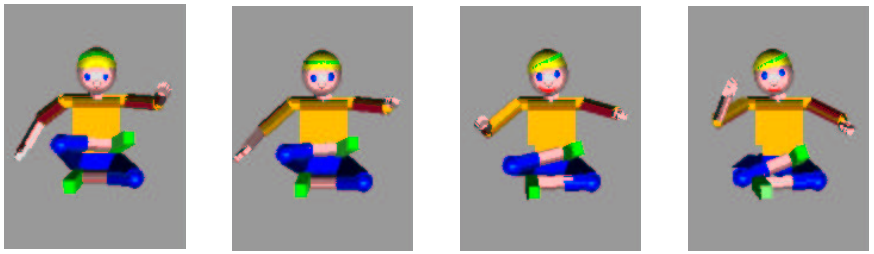




## Keyframing: collision avoidance



Key 1 2 3 4

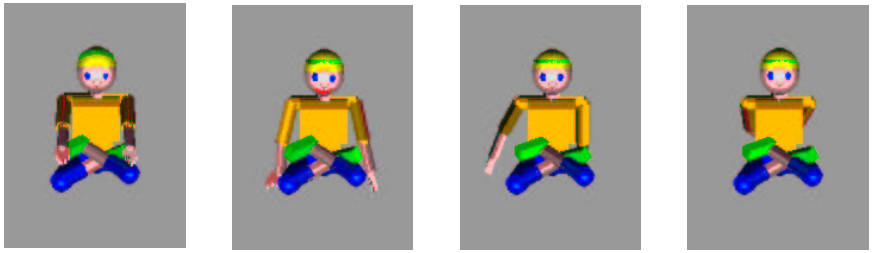


Key 5 6 7 8



## Keyframing: collision avoidance

Key 9 10 11 12



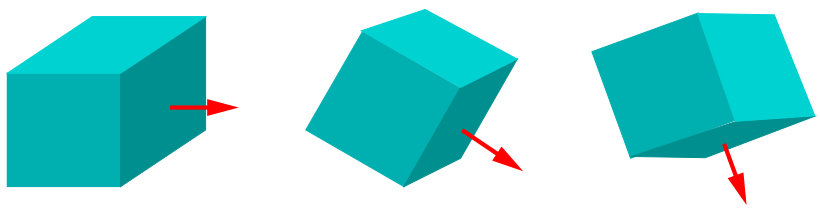
Key 13 14 15 16



## Keyframing

- Keyframing of limbs requires a significant additional effort to avoid collions and unrealistic positions during interpolation.
- Inverse kinematics determines joint motions necessary to achieve a goal: more in following lectures.
- Additional constraints to limit the joint only to realistic motions.
- Lotus position keyframing by Jean-Christophe Nebel, *Keyframe interpolation with self-collision avoidance* in Computer Animation and Simulation '99, pages 77-86. Eurographics, 1999.

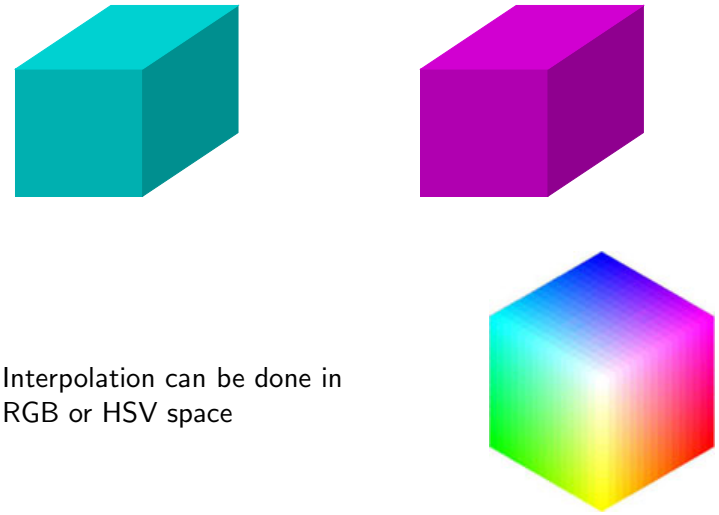
## Key-frame orientation



- Choose rotation axis
- Interpolate angle about axis
- Interpolate takes the shortest path on a unit sphere (geodesic path)



## Key-frame colour



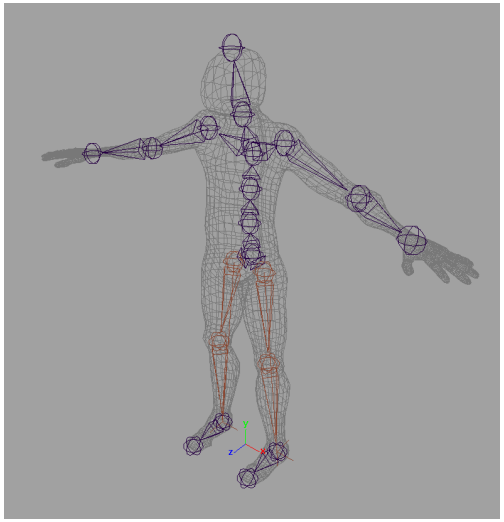
- Interpolation can be done in RGB or HSV space



## Skeletal animation: rigging

Animating complex polygonal meshes

- Construct a *skeleton* of *bones* capturing the components and joints of the body
- Rig the skeleton by attaching vertices to skeleton
- Animate skeleton
- Each vertex location is a weighted combination of bone locations:
  eg: 1 part shin and 2 parts thigh for a vertex in the knee

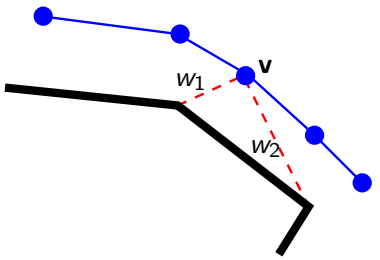




## Matrix palette skinning

**Bones** of skeleton each has a transformation matrix $M_i$ describing its animation from the untransformed location in world coordinates

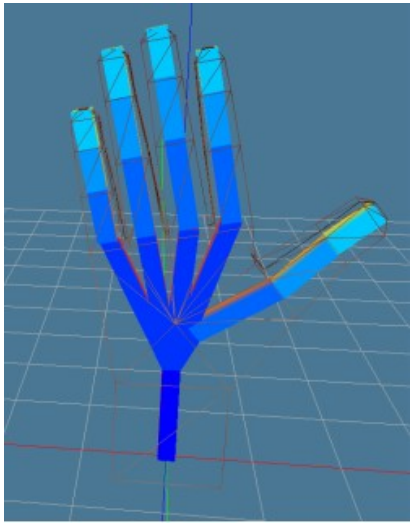
**Vertices** are a linear combination (blend) of a palette of neighbouring bone locations:



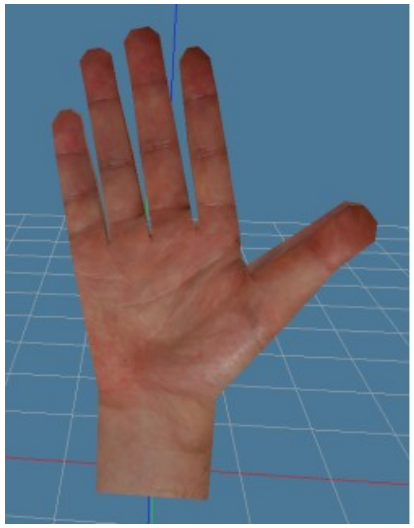

$$\mathbf{v}' = \sum_i w_i M_i \mathbf{v}$$

- $\mathbf{v}$ untransformed vertex location
- $\mathbf{v}'$ transformed vertex location
- $M_i$ transformation matrix describing motion of bone $i$
- $w_i$ weighting for $i$th bone. $\sum_i w_i = 1$

## Matrix palette skinning



Untransformed bones and mesh

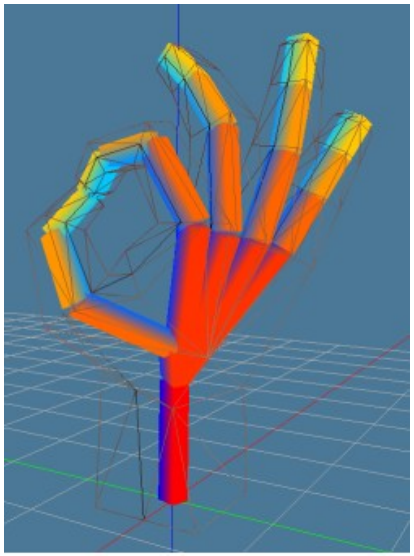


Texture mapped mesh

http://www.gup.uni-linz.ac.at/~gk/Praktika/meshskin_webdata/



## Matrix palette skinning



Transformed bones and mesh

Texture mapped mesh

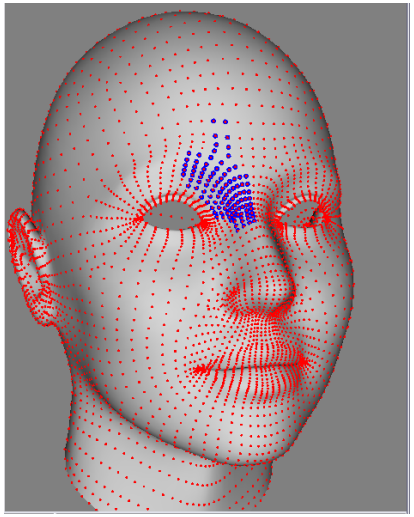
http://www.gup.uni-linz.ac.at/~gk/Praktika/meshskin_webdata/



## Morph target animation

- Vertices of key-frames are edited manually
- Intermediate frames interpolated

- Artist has more fine control over location
- Libraries of facial expressions corresponding to emotions or speech to permit 'procedural' animation
- Expensive to define the vertex positions
- Non-realistic distortions as the vertices take different paths





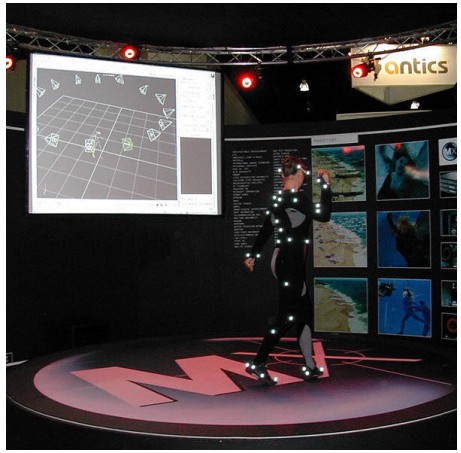
## Morph target animation

http://ivizlab.sfu.ca/research/iface

## Motion capture

- Place markers on the human body
- Surround body with (at least 8) cameras
- Position of each marker inferred if visible to at least 2 cameras
- Rapid acquisition of motion

- Active markers to avoid ‘marker swapping’






## Motion capture