Computer Animation

Creating animation sequences
- object definition
- path specification
- key frames
- in-betweening

Displaying animation sequences
- raster animation

Parametric equations in animation

Steps of a simple computer animation

1. Creating animation sequences
   - object definition
   - path specification (for an object or a camera)
   - key frames
   - in-betweening
2. Displaying the sequences
   - raster animation
   - colour-table animation

Displaying animation sequences

- Movies work by fooling our eyes
- A sequence of static images presented in a quick succession appears as continuous flow

Why animation works

- The eye cannot register images faster than approximately 50 frames per second
  (30 is just about adequate)
- If a gap in the projection occurs, the eye seems to perform spatial interpolation over the gap

Displaying animation sequences

- Animations frames can be
  - pre-computed in advance and pre-loaded in memory
  - computed in real time (e.g. movement of the cursor)

CREATING ANIMATION SEQUENCES

Object definition

- In simple manual systems, the objects can be simply the artist drawings
- In computer-generated animations, models are used

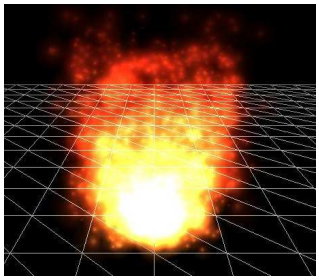
Models can be

- Rigid (i.e. they have no moving parts)
- Articulated (subparts are rigid, but movement is allowed between the sub-parts)
- Dynamic (using physical laws to simulate the motion)
- Particle based (animating individual particles using the statistics of behaviour)
- Behaviour based (e.g. based on behaviour of real animals)

Articulated model

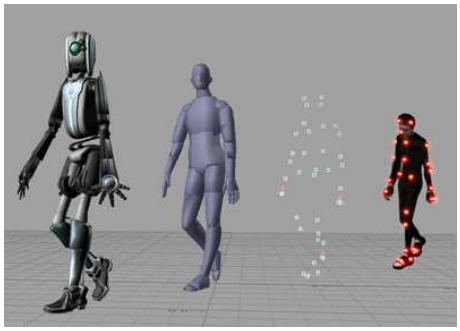
Source: http://www.frontiernet.net/~Imaging/animHierarchy.html

Particle system



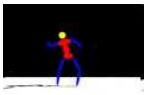
Source: Wikipedia

Behaviour-based (motion capture)



Source: Wikipedia

Examples



Articulated / behaviour based animation

http://www.youtube.com/watch?v=MR0BU7g43p0

Particle system animation

http://vimeo.com/6305219

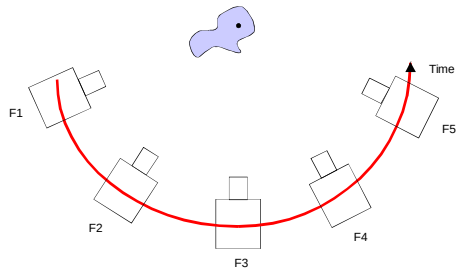
Rigid objects

- Simple rigid objects can be defined in terms of
  - polygon tables (3D)
  - basic shapes such as line segments, circles, splines etc. (2D)
- Rigid body animation is an extension of the three-dimensional viewing

Path specification

- Impression of movement can be created for two basic situations, or for their combination:
  - static object, moving camera
  - static camera, moving object
- The path defines the sequence of locations (for either the camera or the object) for the consecutive time frames

Static object, moving camera



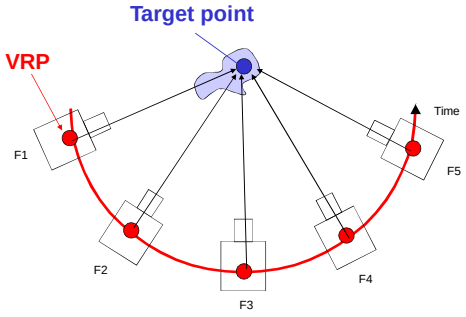
Static object, moving camera

- The path specifies the spatial coordinates along which the camera moves
- The path is usually specified for a single point, e.g. the VRP

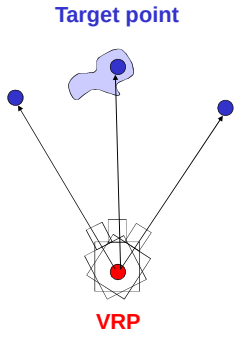
Static object, moving camera

- During movement, the target point in the World coordinate system can
  - remain the same (e.g. when walking or flying around the object to see it from all directions);
  - change (e.g. standing in one location and looking round, or moving along a given path and showing the view seen by the observer while moving).

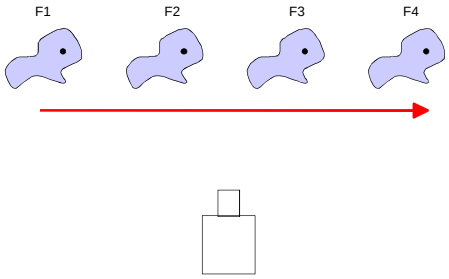
Static object, moving camera



Static object, moving camera



Static camera, moving object



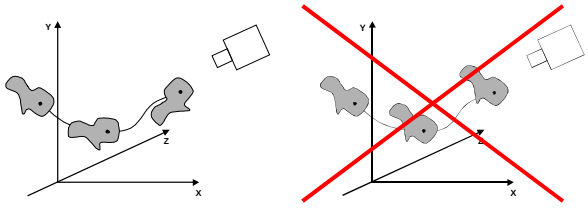
Static camera, moving object

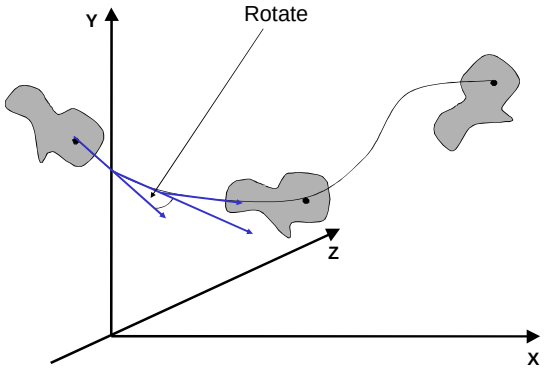
- Path specifying the object movement has to be defined
- The path is defined as the spatial coordinates along which the object moves

Static camera, moving object

- Objects and their parts are defined in a local coordinate system
- Animation path is defined in the World coordinate system
- The path is specified for a single point, e.g. the centre of the object's local coordinate system
- Coordinates of the actual points describing the object are calculated afterwards

It is important to remember that when the object moves along the path, not only its position changes, but also its orientation





KEY FRAMES AND IN-BETWEENING

Rigid body animation

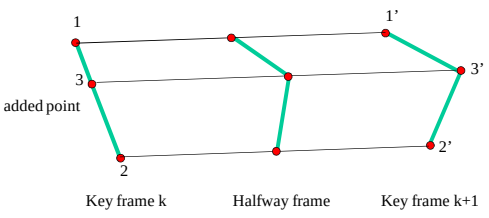
- Rigid body animation uses standard 3D transformations
- At least 30 frames per second to achieve smooth animation
- Computing each frame would take too long

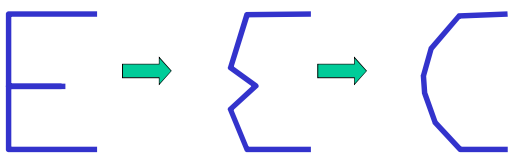
Key frames

- Compute first a small number of key frames
- Interpolate the remaining frames in-between these key frames (in-betweening)
- Key frames can be computed
  - at equal time intervals
  - according to some other rules
  - for example when the direction of the path changes rapidly

In-betweening

- The simplest method of in-betweening is linear interpolation
- Interpolation is normally applied to the projected object points





In-betweening - example

- Given coordinates of a 2D point
  - key frame n: (x_n,y_n)
  - key frame n+1: (x_{n+1},y_{n+1})
  - time interval between the two key frames: 1/10 second
- To get smooth animation, needs at least 30 frames per second
- **Solution**: insert at least further 2 frames between the given two key frames

In-betweening - parametric equations

Parametric equations – a flexible tool for interpolation

Example for line segment between two points, (x_n,y_n) and (x_{n+1},y_{n+1})

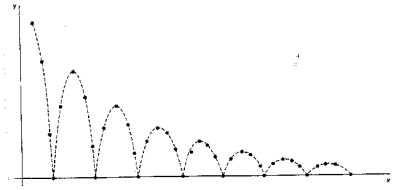
- calculate points in between the two given points
- $x_i = x_n + t\ (x_{n+1} - x_n)$
- $y_i = y_n + t\ (y_{n+1} - y_n)$
- **t** is the parameter which always changes between 0 and 1
- when $t = 0$, we get x_n
- when $t = 1$ we get x_{n+1}
- for $0 < t < 1$ we get the points in between

In-betweening - parametric equations

- The only thing to decide is the number steps between point n and point n+1
- This allows us to set the value of Δt , which is 1 divided by the number of steps
- For example, for 10 steps, $\Delta t = 1/10 = 0.1$
- This formula works also for points in 3D

In-betweening

- Linear interpolation will not always produce realistic results.
- Example: an animation of a bouncing ball where the best in-betweening can be achieved by dynamic animation



In-betweening

- In-betweening should use interpolation based on the nature of the path, for example:
  - straight path — linear interpolation
  - circular path — angular interpolation
  - irregular path — linear interpolation, spline
- For in-betweening use parametric representation of lines and curves, e.g.
  - line segment
  - circle
  - Bezier curve

Examples

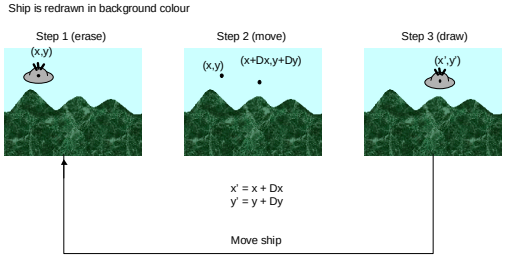
- Bouncing Ball
  - Basic animation
    - http://www.youtube.com/watch?v=ih-sqwHEx7A&NR=1
  - Elastic deformation
    - http://www.youtube.com/watch?v=moGZi1kim-8
  - Story 1
    - http://www.youtube.com/watch?v=-rnbylVH4tM&NR=1
  - Story 2
    - http://www.youtube.com/watch?v=yyRUrDUHLPo&feature=related

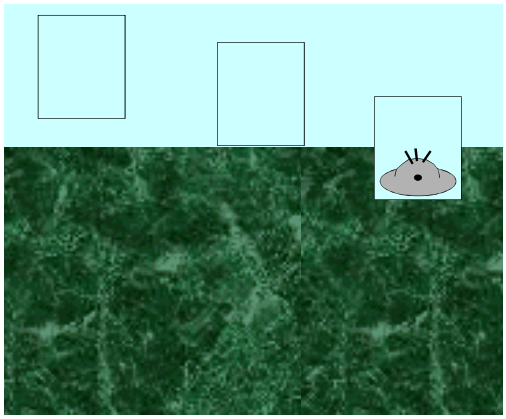
RASTER BASED ANIMATION

Raster animation

- This is the most common animation technique
- Frames are create off-line and copied very fast from the off-screen memory to the frame buffer
- Copying usually done with bitBLT-type operations
- Copying can be applied to
  - complete frames
  - only parts of the frame which contain some movement

Examples





Raster animation - procedures

- A part of the frame in the frame buffer needs to be erased
- The static part of the frame is re-projected as a whole, and the animated part is over-projected.

BitBLT

Bit **BL**ock **T**ransfer

- Copying rectangles of images
  - From one part of the screen to another
  - From off-screen images to on-screen images
  - Implemented on video cards – hence very fast
- Automatic rescaling if rectangles are of different size
- BitBLT often includes logic operations (NOT, XOR etc)

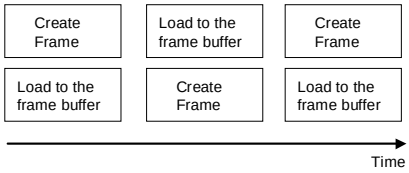
BitBLT

Java methods implementing BitBLT

- Graphics.copyArea( )
  - To move image from one part of the screen to another
- Graphics.drawImage
  - To copy between off-screen images, or off-screen and on-screen images
- java.awt.Robot.createScreenCapture
  - Capture the screen as an Image

Double buffering

- Used to achieve smooth animation
- The next frame of animation is computed to an off-screen buffer at the same time when the current frame is transferred to the frame buffer.



Double buffering

- Java methods implementing double buffering
  - createBufferStrategy( )

Example of raster animation

http://www.youtube.com/watch?v=9mpr6WaTsfU

3D animation overview tutorial

http://www.youtube.com/watch?v=Aks7qowB-U0&feature=related

Next lecture

Efficient algorithms for line and circle