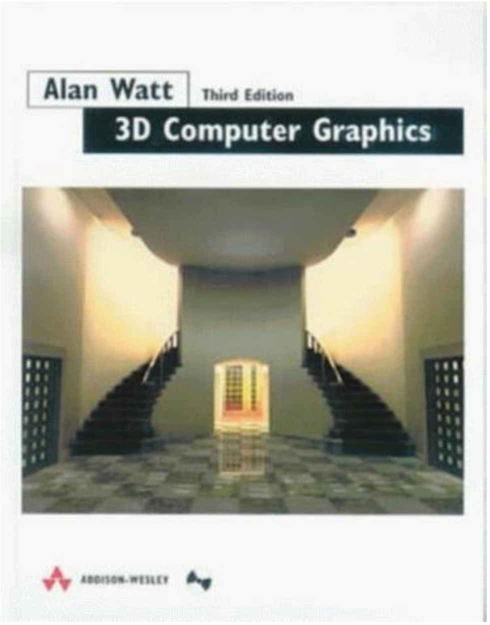
COM3503/4503/6503: 3D Computer Graphics

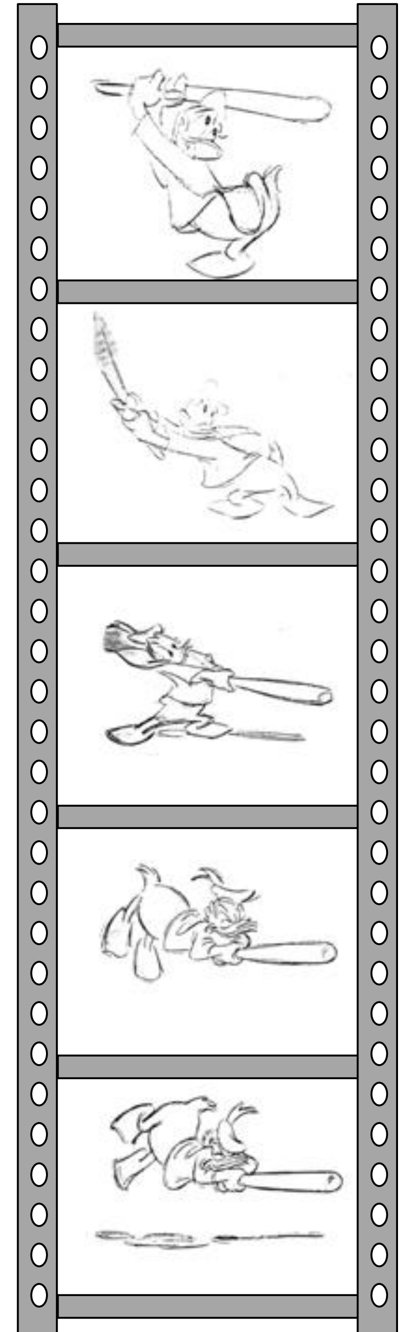
Lectures 15 and 16: Animation



Dr. Steve Maddock
Room G011, Regent Court
s.maddock@sheffield.ac.uk

1. Introduction: The illusion of movement

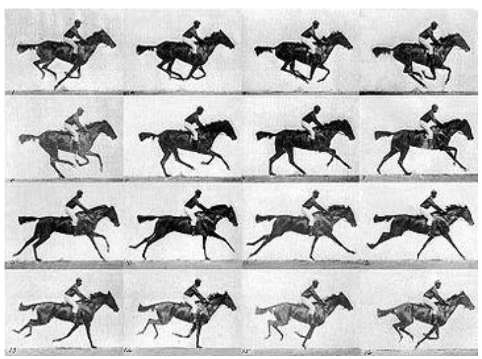
- Traditional animation, e.g. Disney: series of individual drawings on successive frames of film.
- Play at fast enough speed to convince eye it is seeing continuous motion

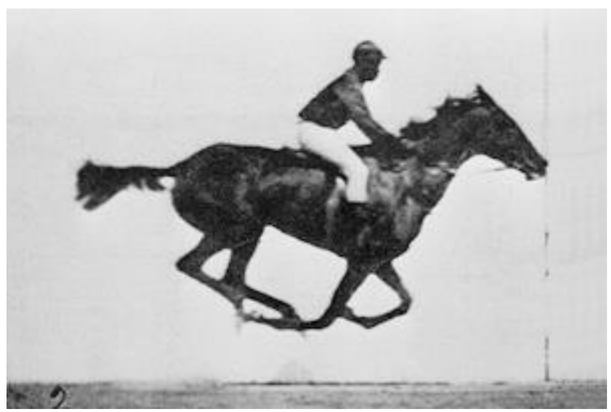


Thomas. and Johnston, “Disney Animation: The Illusion of Life”

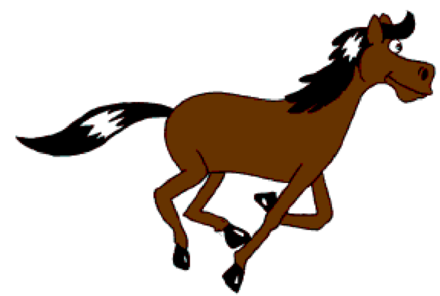
1.1 Using photographs

- From http://en.wikipedia.org/wiki/File:Muybridge_race_horse_animated.gif
- "Animated sequence of a race horse galloping. Photos taken by Eadweard Muybridge (died 1904), first published in 1887 at Philadelphia (*Animal Locomotion*).
- The sequence is set to motion using these frames (Human and Animal Locomotion, plate 626, thoroughbred bay mare "Annie G." galloping)"





http://en.wikipedia.org/wiki/Eadweard_Muybridge

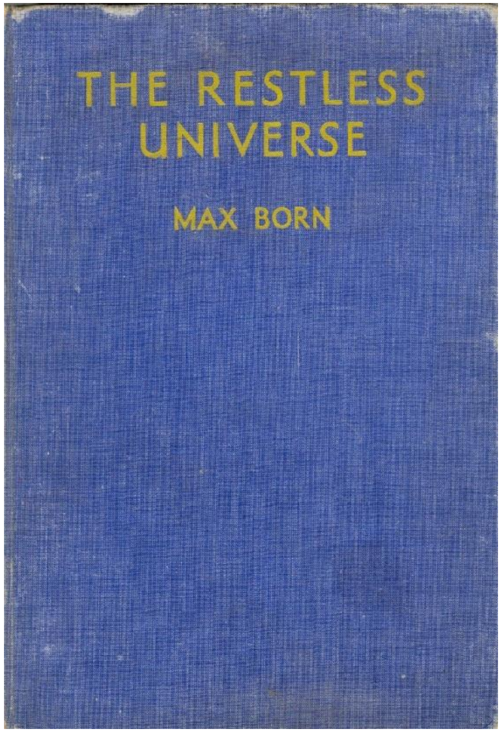


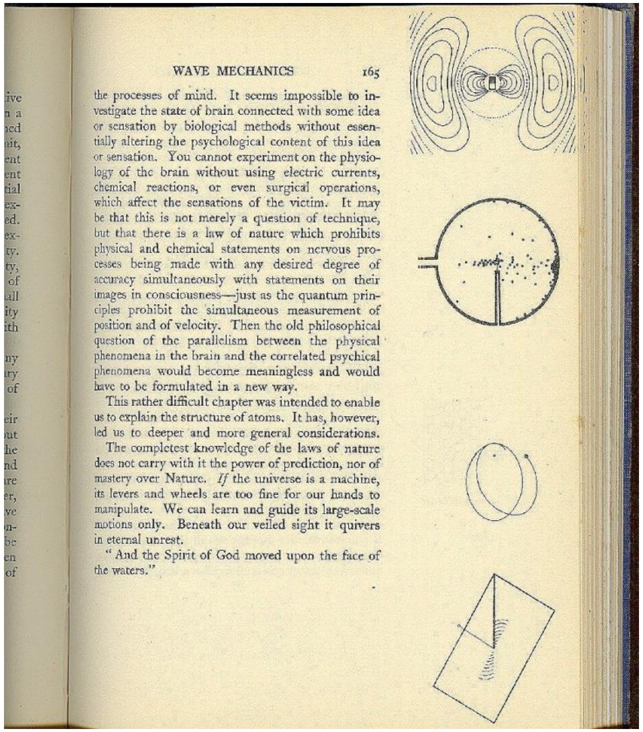
Animated horse, made by rotoscoping 19th century photos by Eadweard Muybridge. Artistic license has been used to achieve the cartoony look. Animated by J-E Nyström

http://en.wikipedia.org/wiki/File:Animhorse.gif

1.2 Animation in science

Max Born 1935





WAVE MECHANICS 165

the processes of mind. It seems impossible to investigate the state of brain connected with some idea or sensation by biological methods without essentially altering the psychological content of this idea or sensation. You cannot experiment on the physiology of the brain without using electric currents, chemical reactions, or even surgical operations, which affect the sensations of the victim. It may be that this is not merely a question of technique, but that there is a law of nature which prohibits physical and chemical statements on nervous processes being made with any desired degree of accuracy simultaneously with statements on their images in consciousness—just as the quantum principles prohibit the simultaneous measurement of position and of velocity. Then the old philosophical question of the parallelism between the physical phenomena in the brain and the correlated psychical phenomena would become meaningless and would have to be formulated in a new way.

This rather difficult chapter was intended to enable us to explain the structure of atoms. It has, however, led us to deeper and more general considerations.

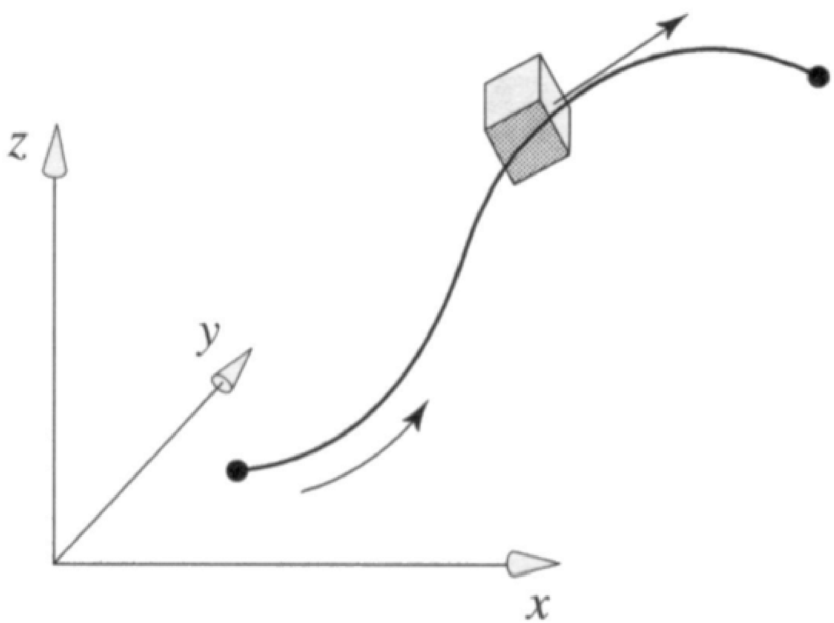
The completest knowledge of the laws of nature does not carry with it the power of prediction, nor of mastery over Nature. *If* the universe is a machine, its levers and wheels are too fine for our hands to manipulate. We can learn and guide its large-scale motions only. Beneath our veiled sight it quivers in eternal unrest.

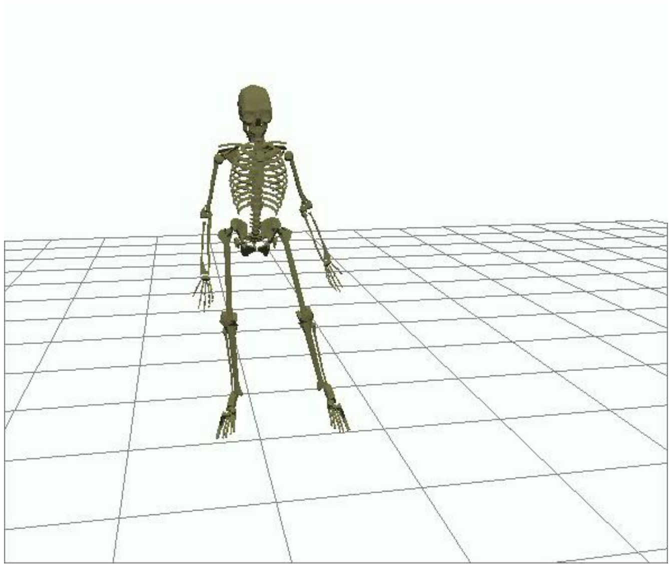
"And the Spirit of God moved upon the face of the waters."

VI. MOTION OF THE ELECTRON IN THE HYDROGEN ATOM

1.3 Computer animation

- Computer animation: series of individual states of a dynamic scene.
- Scene state: set of numbers
- Animation: Change numbers over time
  - Example numbers: colour, coordinates, size, velocity, focal length,...
- Display each frame, after rendering, at fast enough speed to convince eye it is seeing continuous motion





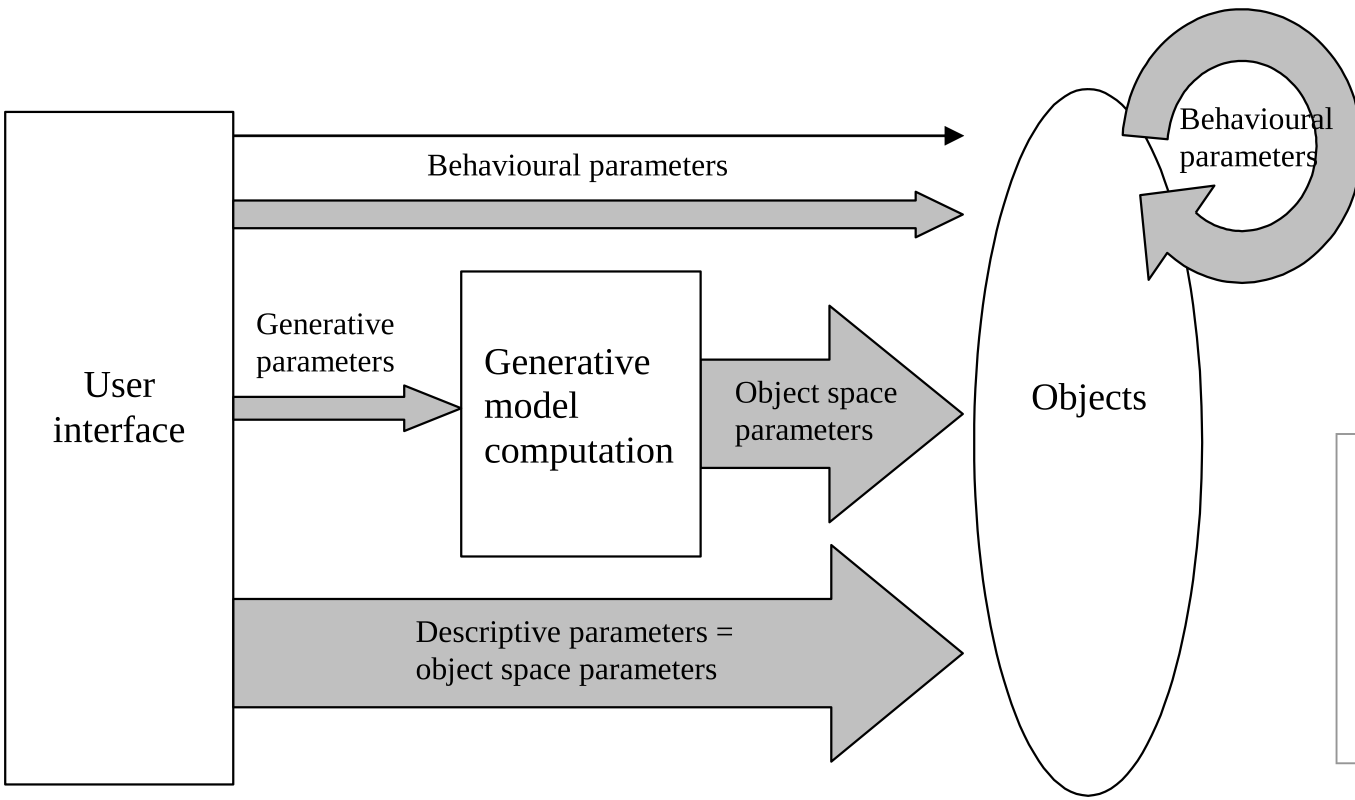
2. Animation control

- Animation techniques differ in:
  - Parameters offered
  - Control given by the parameters

Watt and Watt, 92	**Zeltzer, 85**	**Hegron and Arnaldi, 92**	**Thalmann and Thalmann, 94**	**Examples**
low-level	guiding	descriptive or phenomenological models	geometric	Keyframing, Motion capture
medium-level	animator-level	generative or physically-based models	physical	Particle systems, Procedural
high-level	task-level	behavioural models	behavioural	Flocking, Sensory perception

2.1 Cause and effect (Hegron and Arnaldi, 92)

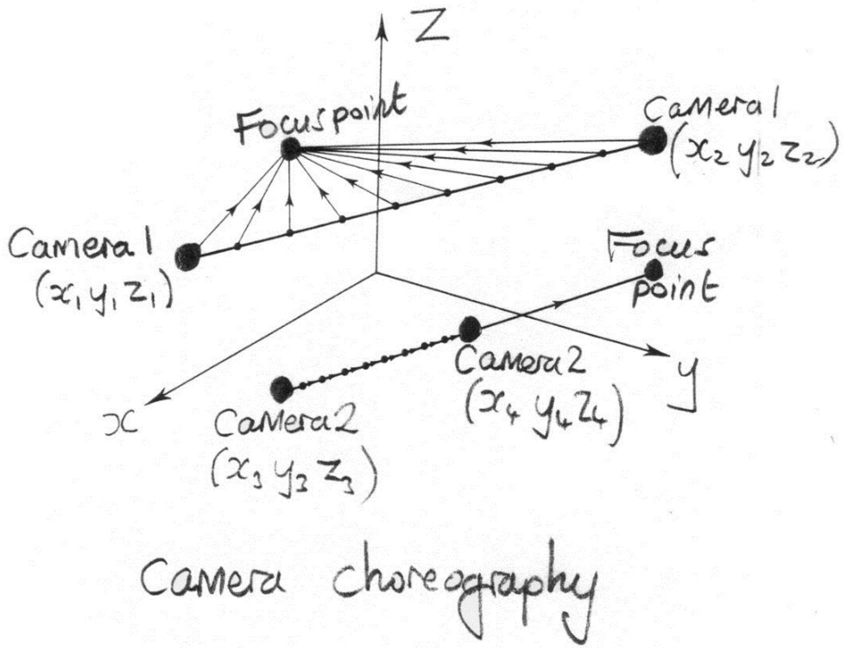
- **Descriptive models**: effect without knowledge of cause
- **Generative models**: describe cause that produces effects
- **Behavioural models**: simulate the actions of living things that respond to internal and external stimulation

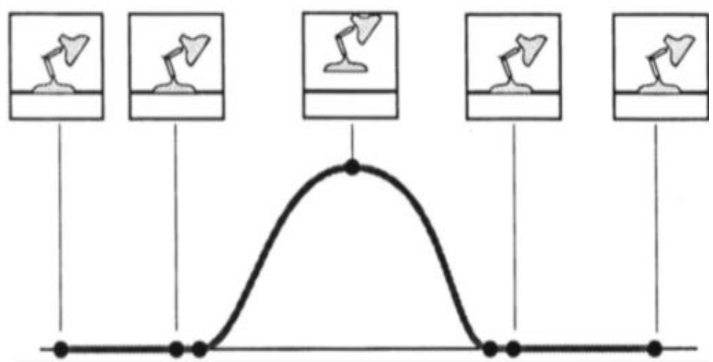


Different sized arrows indicate the quantity of parameters that need to be controlled

2. Low-level control

- Object parameters are explicitly changed over time
  - Examples: camera positions and controls, eye blinking rate for a face, joint angles for a hierarchical object
- *Keyframes*: specify attributes at certain instances.
- *Inbetweening*: calculate intermediate frames.

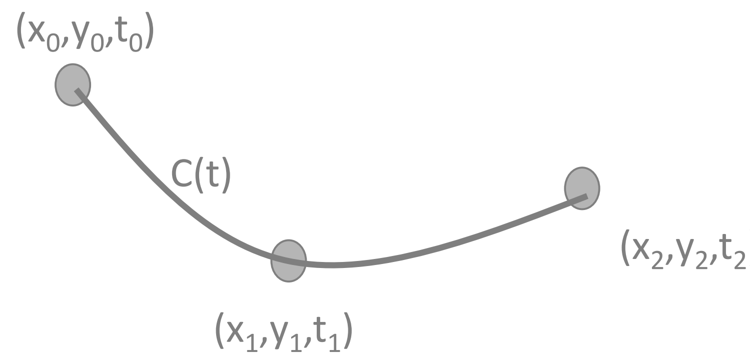




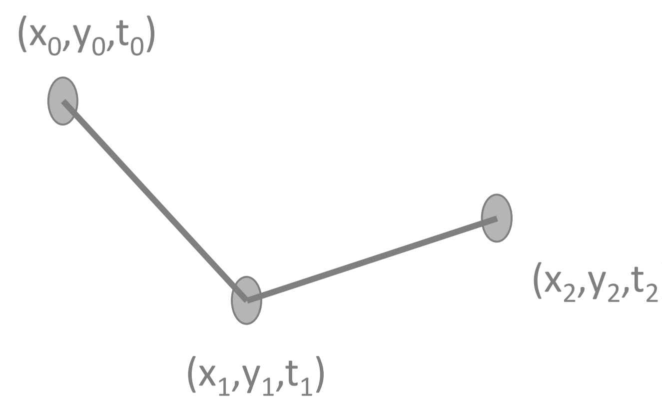
John Lasseter, "Principles of Traditional Animation Applied to 3D Computer Animation", Computer Graphics, pp. 35-44, 21:4, July 1987 (SIGGRAPH 87)."

2.1 Interpolating 2D position

- Given positions: (x_i, y_i, t_i), i=0,…,n
- Find curve $C(t) = (x(t), y(t))$ such that $C(t_i) = (x_i, y_i)$

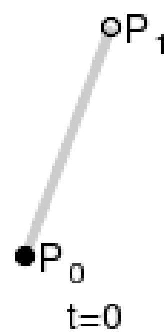


- Linear interpolation
  - For first two points, assuming $t_0=0$ and $t_1= 1$
  - X coordinate: $x(t) = x_0(1-t) + x_1 t$



$\mathbf{Q}(t) = \mathbf{P}_0(1-t) + \mathbf{P}_1 t \quad 0 \le t \le 1$

http://en.wikipedia.org

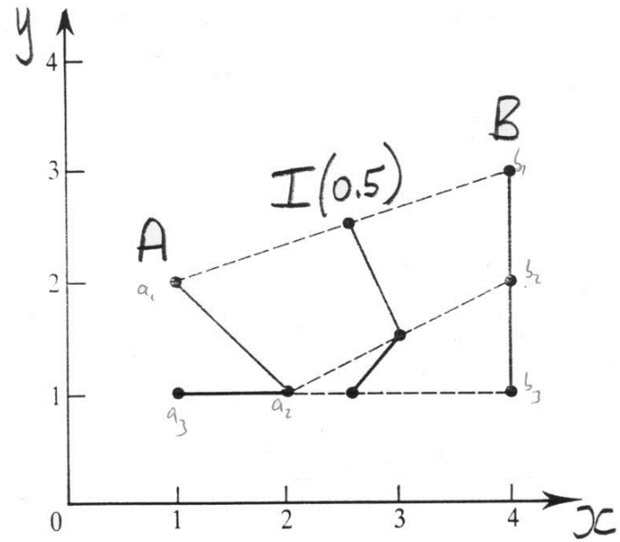


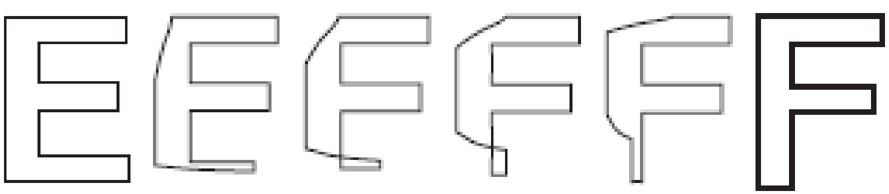
2.2 Shape interpolation

- Given two sets of vertices A_i and B_i
- Inbetween sets can be calculated using, say, linear interpolation:

$$P = A_i\,(1-t) + B_i\,t \qquad i = 1..n$$

- Issues:
  - Corresponding vertices
  - Different numbers of vertices
  - Avoiding scrambling





Example of commercial software, from (Sederberg and Greenwood, 92)

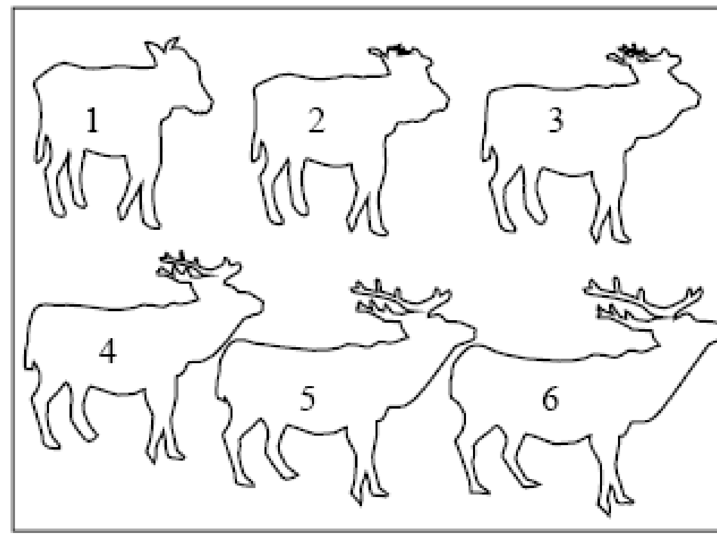
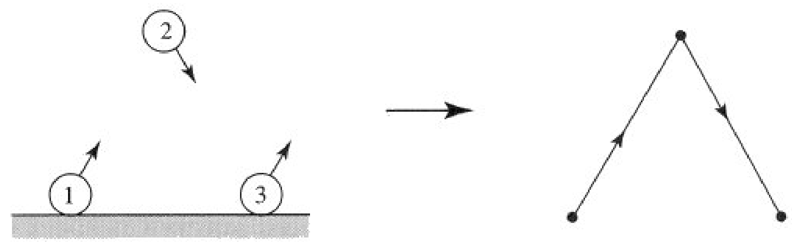


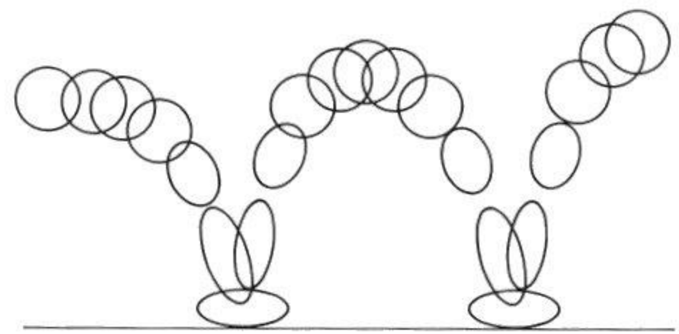
Figure 27: Cow to deer

Sederberg and Greenwood, 92

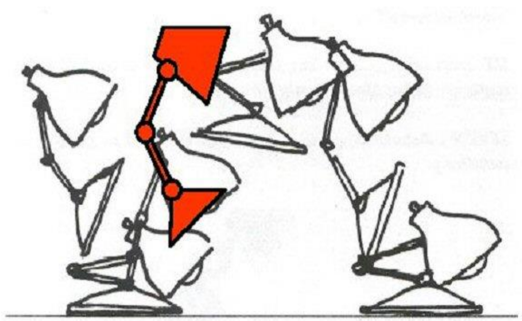
2.3 Interpolating parameters using a curve

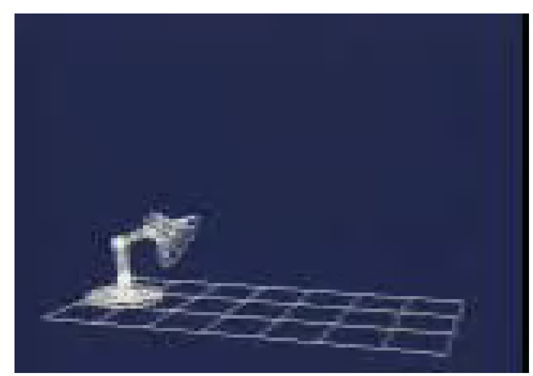
- Linear interpolation can lead to non-smooth movement.
- Solution: Fit a (parametric) curve through the key parameters and use this to derive inbetween points
- The parameter u of the curve q(u) becomes the current animation time





[Luxo movie]

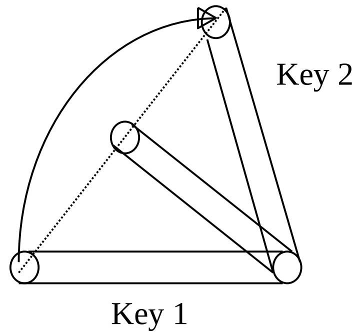


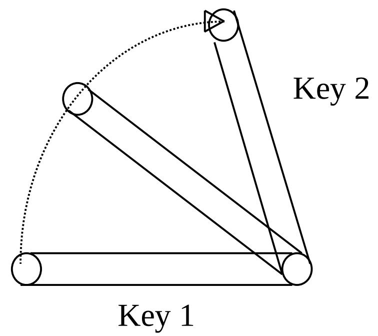


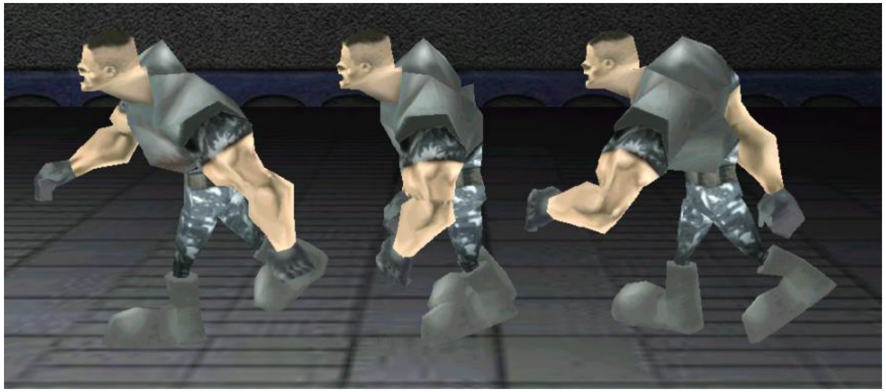
http://www.cs.cmu.edu/~aw/mpg/luxo.mpg

2.4 The ‘right’ parameters?

- Example: a human skeleton
- (Linear) interpolation of skin vertex positions leads to bones shortening in length
- Instead, interpolate the skeleton using joint rotation and then add skin







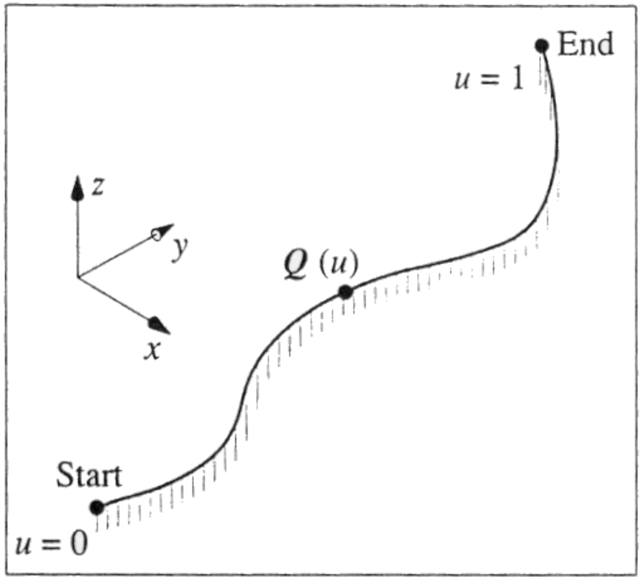
2.5 Timing and inbetweens (Thomas and Johnston, 1981)

- Two drawings of a head:
  1. Leaning toward the right shoulder.
  2. Over on the left with chin slightly raised.
- Different meanings:
  - NO inbetweens – the character has been hit by a tremendous force. His/her head is nearly snapped off
  - ONE inbetween –...has been hit by a brick, rolling pin, frying pan.
  - TWO inbetweens –...has a nervous tic, a muscle spasm, an uncontrollable twitch
  - THREE inbetweens –...is dodging the brick, rolling pin, frying pan
  - FOUR inbetweens –...is giving a crisp order, "Get going!" "Move it!"
  - FIVE inbetweens –...is more friendly, "Over here," "Come on -hurry!"
  - SIX inbetweens –...sees the sports car s/he has always wanted
  - SEVEN inbetweens –...tries to get a better look at something
  - EIGHT inbetweens – searches for the peanut butter on the kitchen shelf
  - NINE inbetweens – appraises, considering thoughtfully
  - TEN inbetweens – stretches a sore muscle

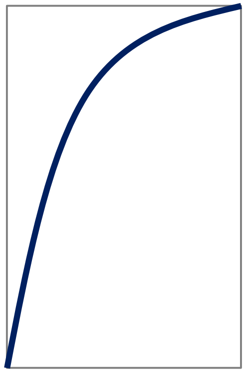
2.6 Specifying velocity characteristics

- Common approach: double interpolant method
  - (Steketee and Badler, 85)
- One (or more) curve(s) for path control:
  - Q(u), 0<=u<=1
  - x(u), y(u), z(u)
- One curve for control of velocity along path:
  - V(u), 0<=u<=1

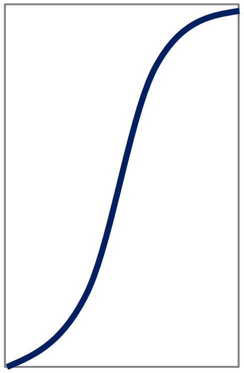
Specify path as a curve in 3D space



Path curve



Sinθ



(1-cosθ)/2

Velocity control

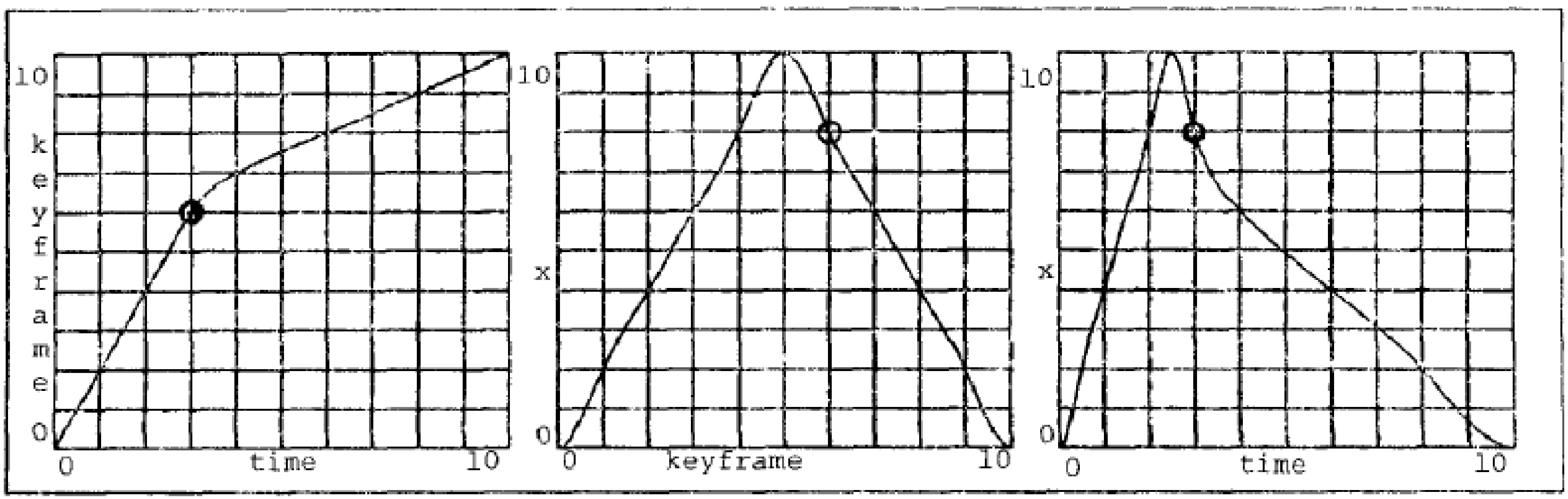
2.6.1 (Steketee and Badler, 85) Double interpolant method

Figure 1: The Double-Interpolant Method.

The left-hand curve is the kinetic interpolant, expressing the keyframe number as a function of time. The middle curve is the position interpolant, expressing the value of the motion variable **x** *as a function of the keyframe number. The right-hand curve is formed by composing these two interpolants, and expresses* **x** *as a function of time. For instance, at time* **t** = 3*, the left-hand graph shows that the keyframe number is* 6*. The middle graph shows that keyframe* 6 *corresponds to an* **x** *value of* 8*. Thus at* **t** = 3 *on the right-hand graph, the* **x** *value is* 8*.*

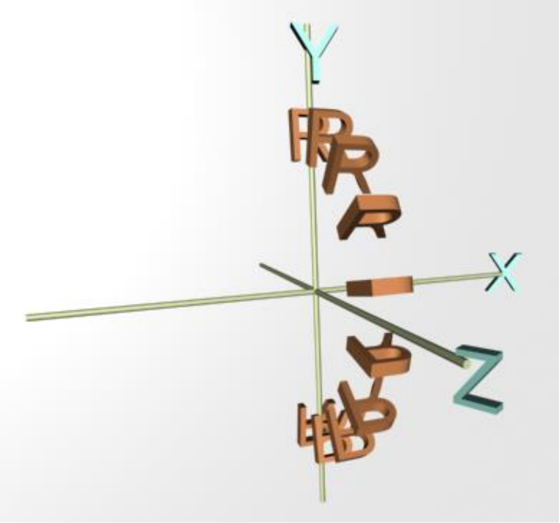
Example (read graphs left to right):

- At time t=3, keyframe number is 6;
- Keyframe 6 corresponds to x value of 8;
- Thus at t=3, x value is 8.



2.7 Interpolation of rotation

- Euler angles: rotation represented by using angles with respect to three mutually perpendicular axes: $R(\theta_1,\theta_2,\theta_3)$
- Rotation matrices are not commutative
  - Following two sequences produce same results:
    $R(0,0,0),\ldots,R(\pi t,0,0),\ldots,R(\pi,0,0)$
    $R(0,0,0),\ldots,R(0,\pi t,\pi t),\ldots,R(0,\pi,\pi)$
    $t \in [0,1]$
- There exist an infinity of ways of getting from one key to another in the parameter space of Euler angles
- Instead, use quaternions



2.8 Low-level control of complex modelling techniques

- Use low-level control with expressive modeling techniques.
- *Example 1*: Controlling Barr's deformation parameters:
  Twist an object about its z axis:

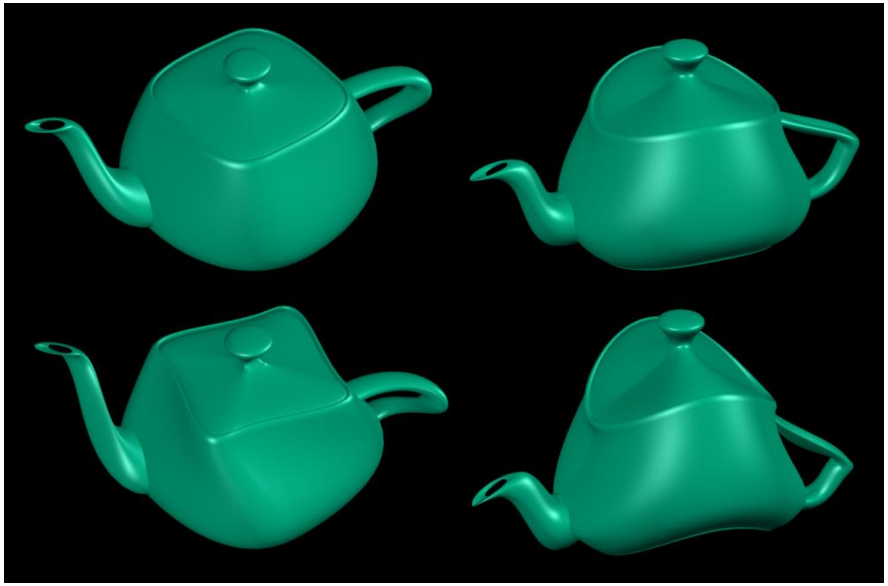
  $$X = x \cos\theta - y \sin\theta$$

  $$Y = x \sin\theta + y \cos\theta$$

  $$Z = z$$

  $$\text{where } \theta = f(z)$$

- *Example 2*: Controlling the free-form deformation (FFD) technique



2.9 FFDs (Sederberg and Parry, 1986)

- An object is surrounded by a 'container'
  - The container is a parametric space - a tricubic Bezier hyperpatch
- The 'container' is distorted and the embedded object distorts with it
- The major advantage of the FFD technique is that it is independent of object representation

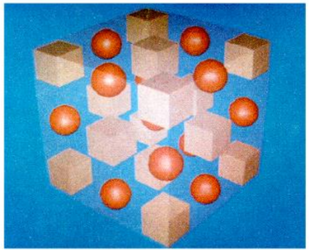


Fig 1. Undeformed Plastic

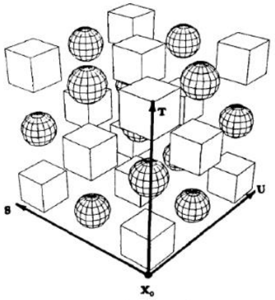


Fig. 3 s,t,u Coordinate System

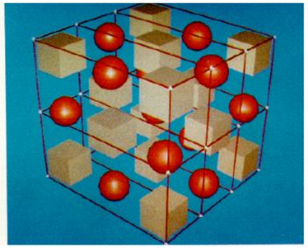
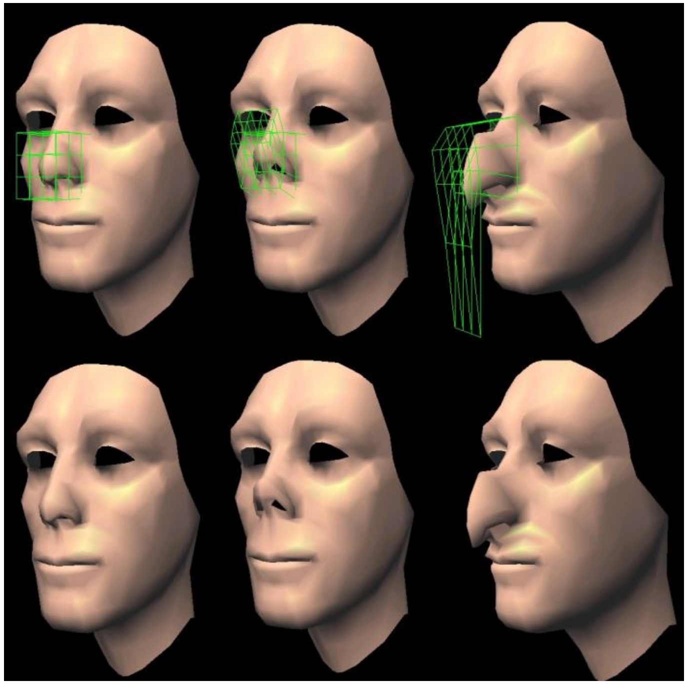
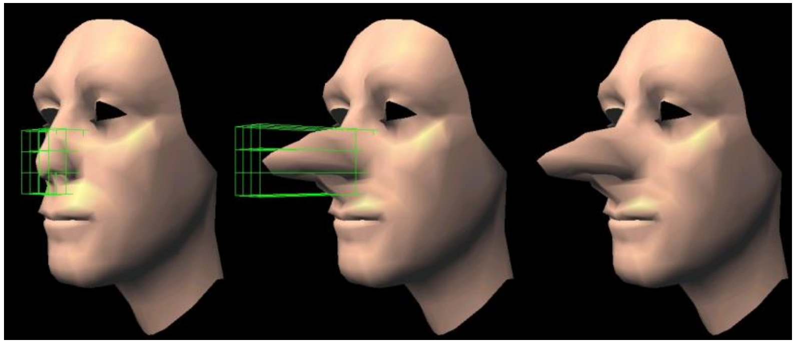


Fig. 4 Undisplaced Control Points

Fig. 5 Control Points in Deformed Position

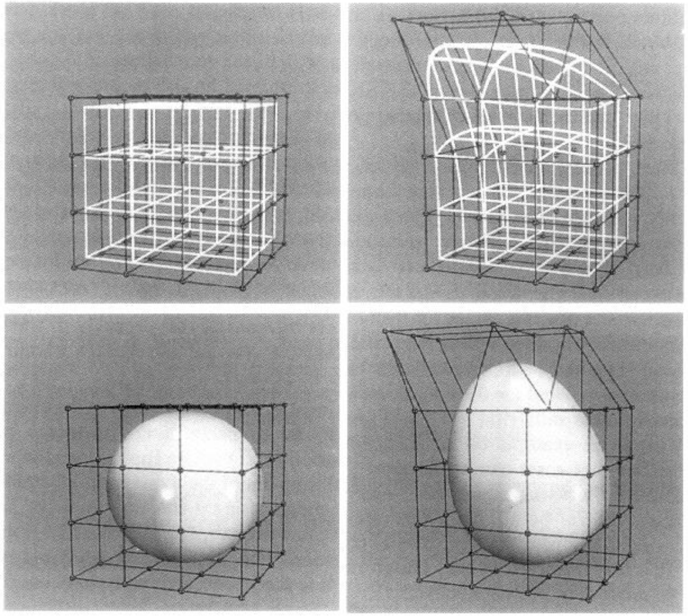




2.9.1 FFDs: details

- Tricubic Bezier hyperpatch:
  - $B_i(u)$, $B_j(v)$ and $B_k(w)$ are Bernstein polynomials of degree 3
  - p_{ijk} are a 3D grid of 64 control points with $u,v,w \in [0,1]$.
- For each object vertex
  - the world coordinates $(x,y,z)_{vert}$ are defined relative to the enclosing parametric space to give $(u,v,w)_{vert}$
  - the control points of the parametric space p_{ijk} are moved to p'_{ijk}
  - the $(u,v,w)_{vert}$ are substituted into $Q'(u,v,w)$ to give the deformed $(x,y,z)'_{vert}$

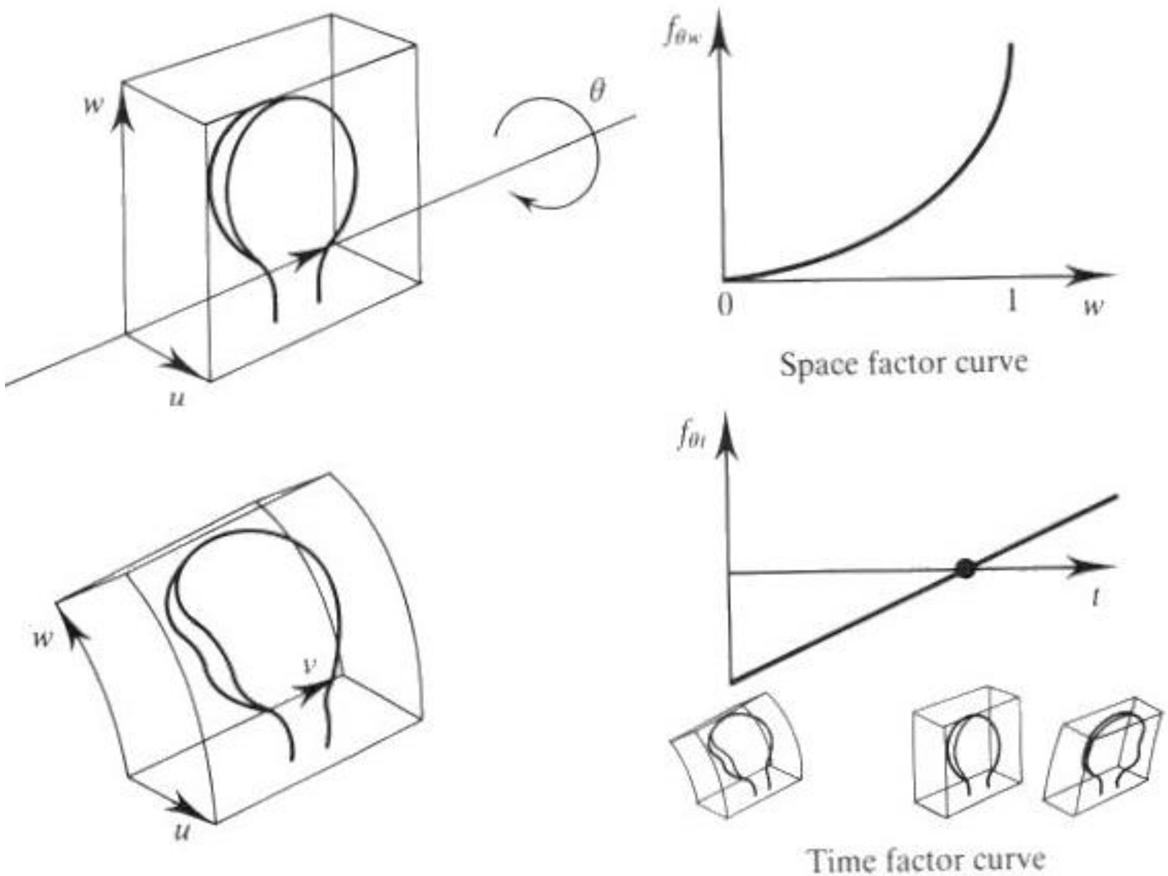
$$Q(u,v,w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} p_{ijk} B_i(u) B_j(v) B_k(w)$$

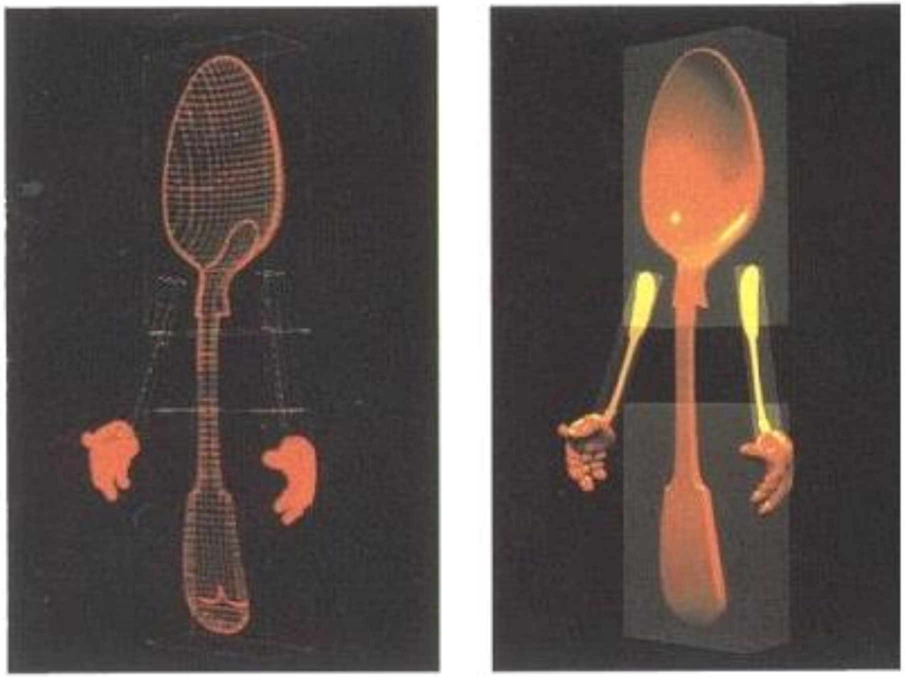


$$Q'(u,v,w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} p'_{ijk} B_i(u) B_j(v) B_k(w)$$

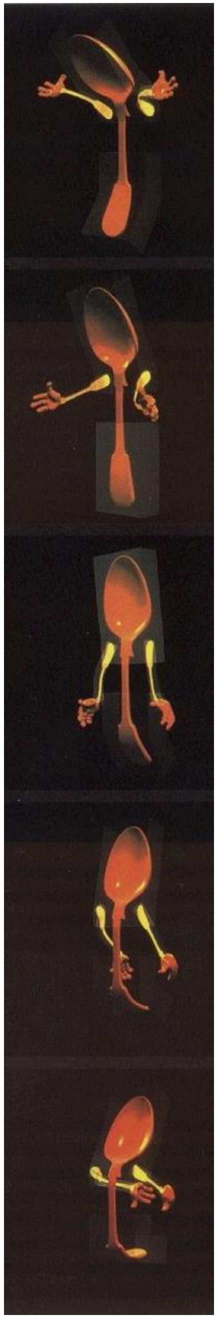
2.9.2 Controlling FFDs

- Spline curves can be used to choreograph the FFD control points in deforming the embedded object (which also has more data points)
- Factor curves [Watt92] can be used to modify the change of deformation over time.





57 FFD animation and characterization: the undeformed spoon model shown in wirefram and rendered form, together with the FFD blocks. Note the polygonal resolution of the model compared with the resolution of the FFD block.



58 FFD animation and characterization: part of an animated sequence for the spoon. Basic characterization becomes possible as even the static frames demonstrate. The sequence shows the spoons jumping. Two FFD blocks are used to control the upper and lower halves of the spoon, and two FFD blocks control the shape of the arms. FFD blocks (not shown) also control the fingers.

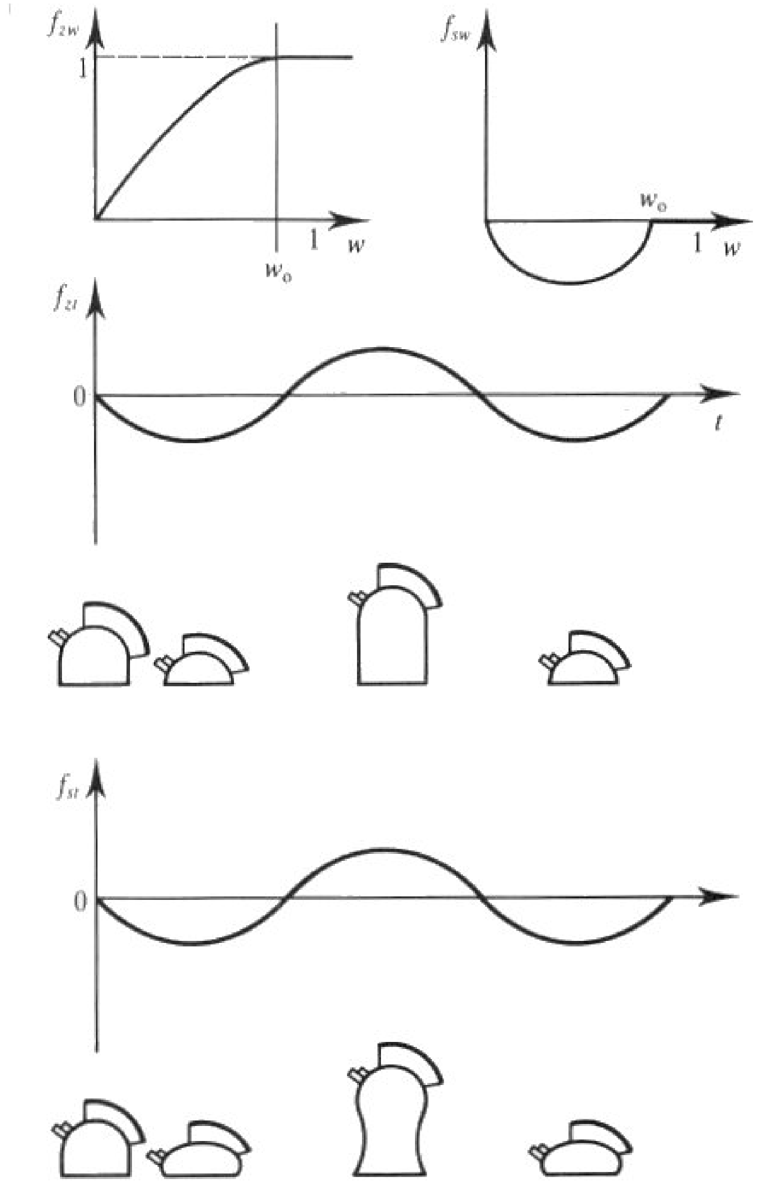
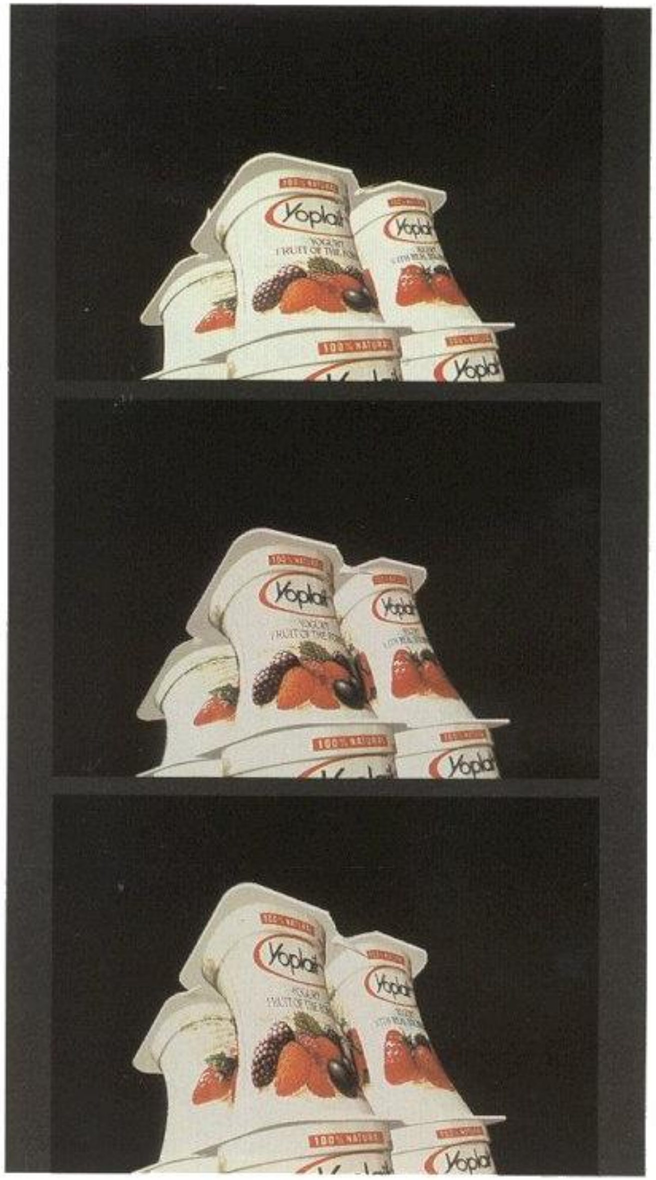


Figure 17.12 Effect of vertical deformation followed by combined effect of vertical and horizontal deformation

63 FFD animation: an environment-mapped kettle jumping exhibiting the classic squash and stretch technique.



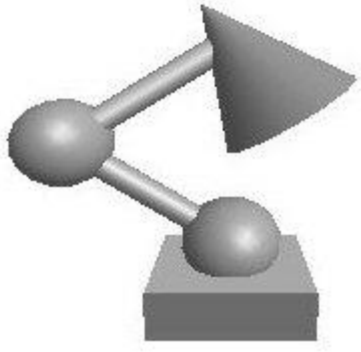
64 A frame from an animated sequence that contains the jumping kettle.

66 A frame from an animated sequence that contains the yoghurt cartons.

65 FFD animation: yoghurt cartons taking off and flying out of a fridge.

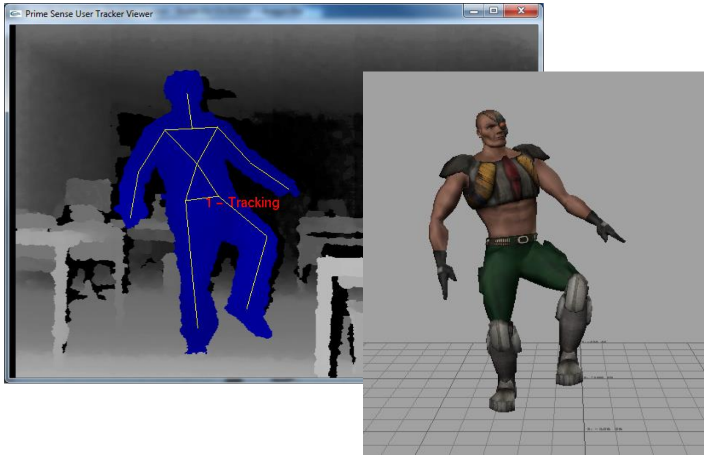
2.10 Amount of data: How to specify?

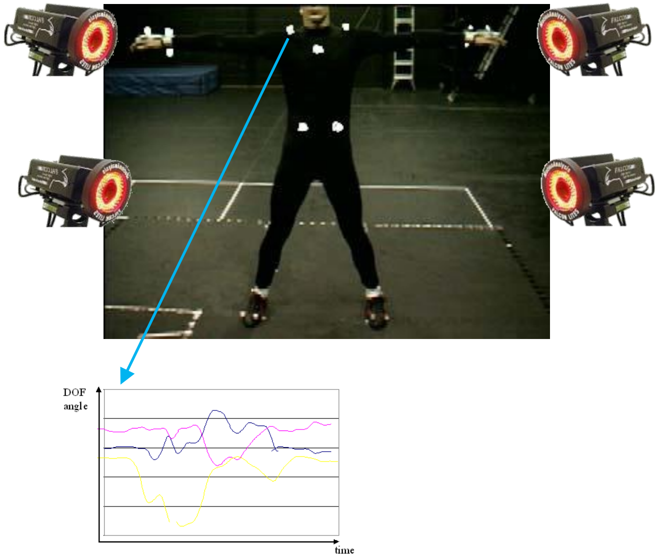
- Instead of using keyframes and fitting a curve through these we can use techniques to produce every frame of an animation sequence
- Examples:
  - Use a formula to create the data for every frame (e.g. equation for a circle)
  - Motion capture
  - Stop-frame animation

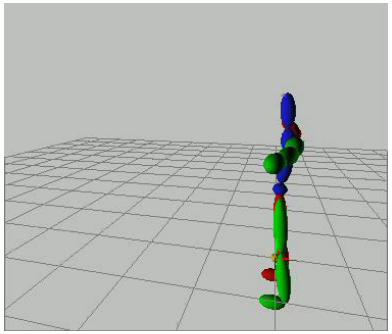


2.10.1 Motion capture for human figure animation

- Measure real-world data
- Skeletal animation of virtual characters
- Add a skin to the skeleton
- Dynamic units are concatenated to make longer sequences

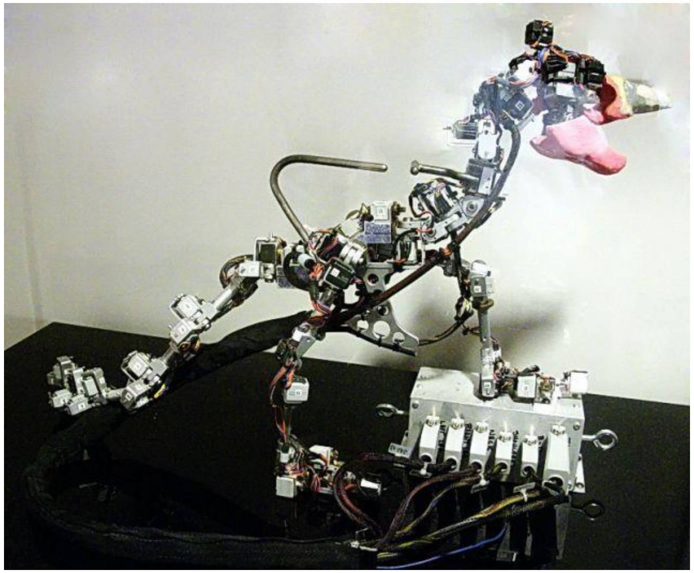


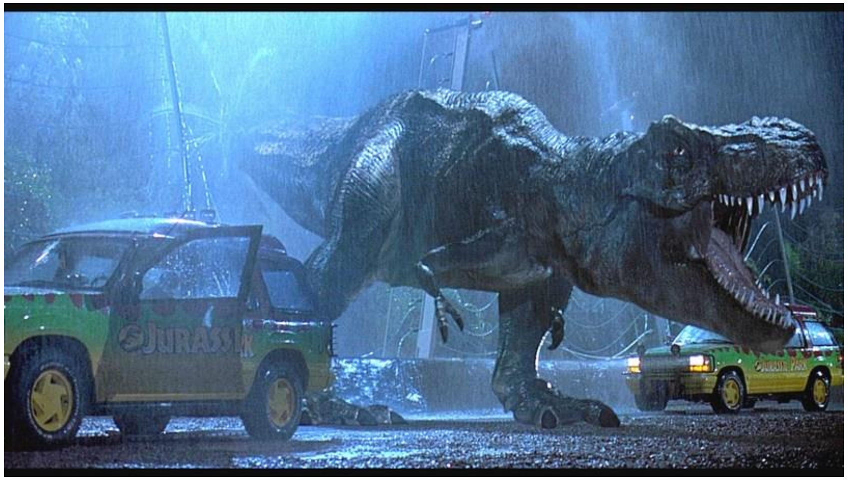




2.10.2 Posing using an inanimate object

- Stop motion animation, with a 'Dinosaur Input Device', was used in the film Jurassic Park to script the computer models



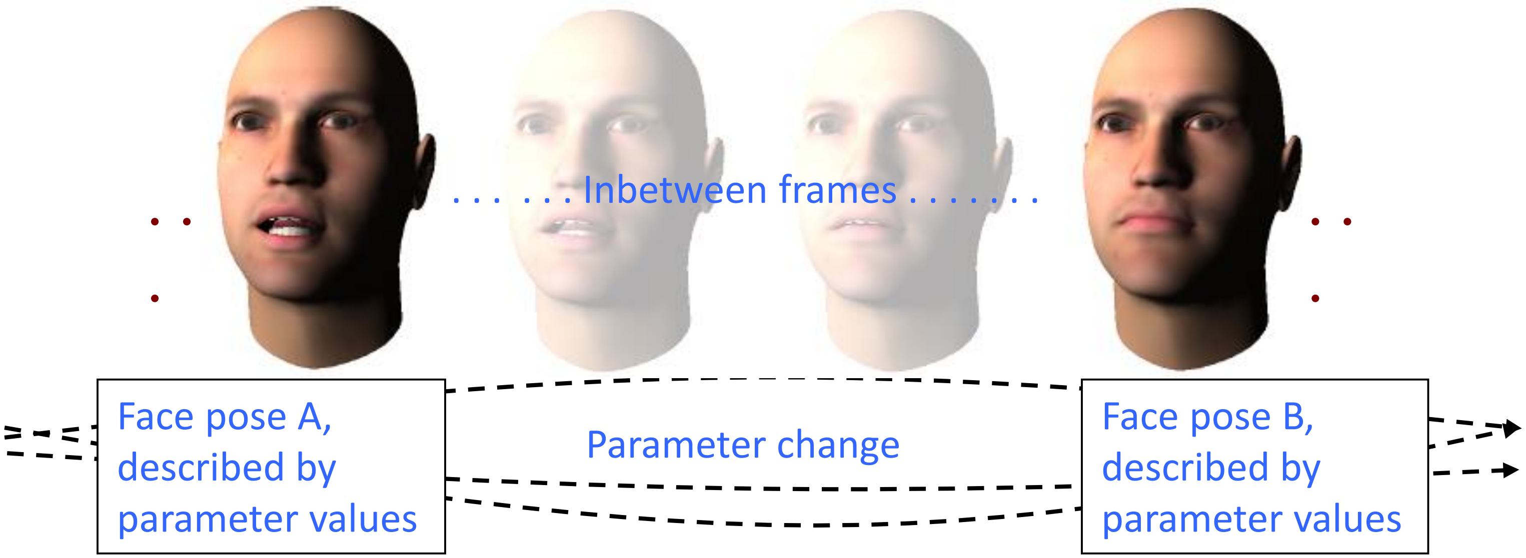


http://www.blep.com/rd/special-effects/dinosaur-input-device/

Knep et al, 95. Dinosaur Input Device

2.11 Facial animation

- Create pose targets at discrete time locations and interpolate
- (we looked at this last week)



3. Medium-level techniques

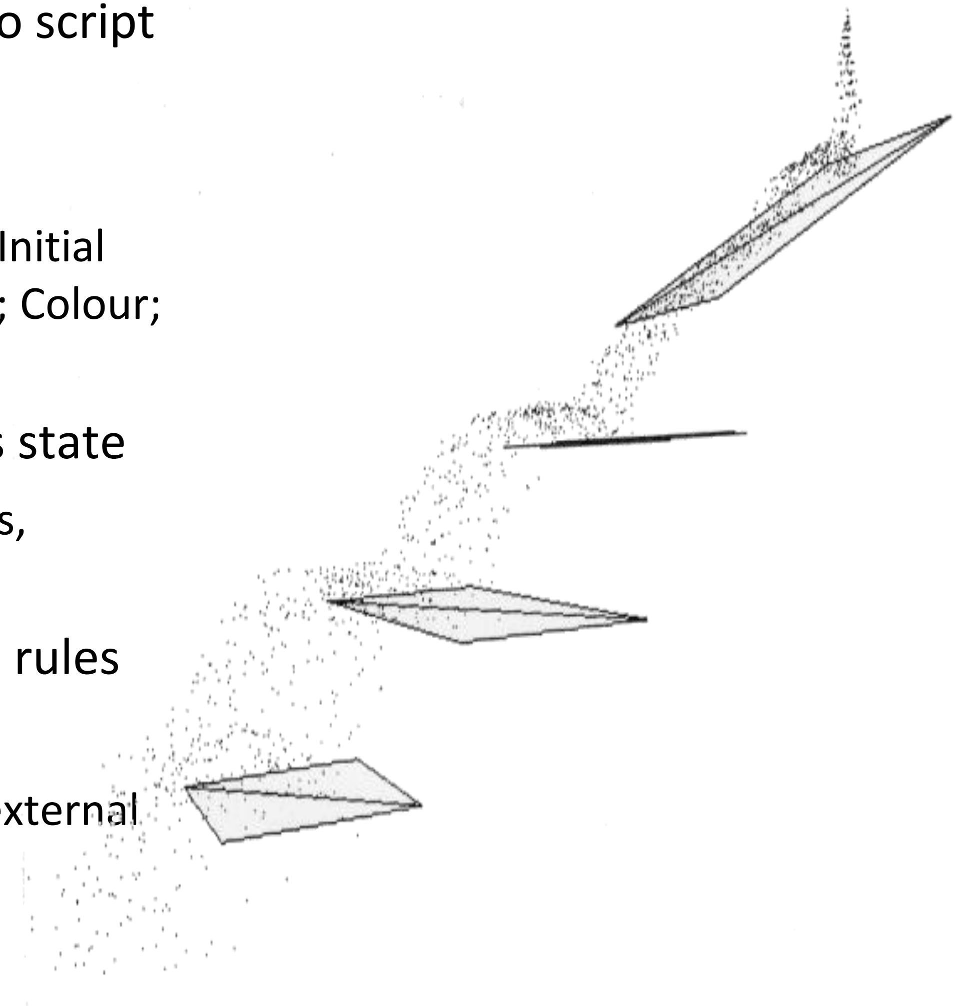
- A generative process converts a few parameters into many parameters in a process of database amplification.

Examples:

- Particle systems
  - Specify start values and frame-to-frame update rules
- Cloth animation, wind waves on deep water and waves approaching a beach
  - A few parameters control a generative animation process.
- A hierarchical data structure to enable a cascading of information from a single input value.
- Physically-based simulations

3.1 Particle animation (Reeves, 83)

- Too many individual particles to script using low level techniques
- Specify starting values
  - Examples: Emitting direction; Initial velocity; Bump/bounce factor; Colour; Transparency; Radius
- Each particle is described by its state
  - Position, velocity, colour, mass, lifetime, shape, etc
- Specify frame-to-frame update rules
  - Geometric
  - Physics – laws of mechanics, external forces



3.1 Particle animation

Five steps for a frame:

- (Generator:) New particles are generated and injected into the current system

  $N(t) = M(t) + rand(r)V(t)$

  where $M(t)$ is the mean number of particles perturbed by a random variable of variance V.
- Each new particle is assigned its individual attributes (e.g. initial velocity)
- Any particles that have exceeded their lifetime are extinguished.
- The current particles are moved according to their rules.
- The current particles are rendered.

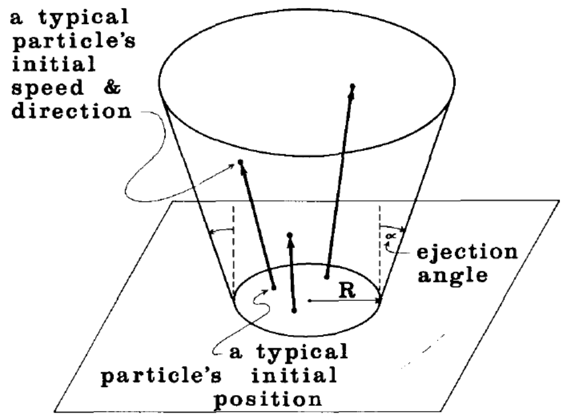


Fig. 3. Form of an explosion-like particle system.

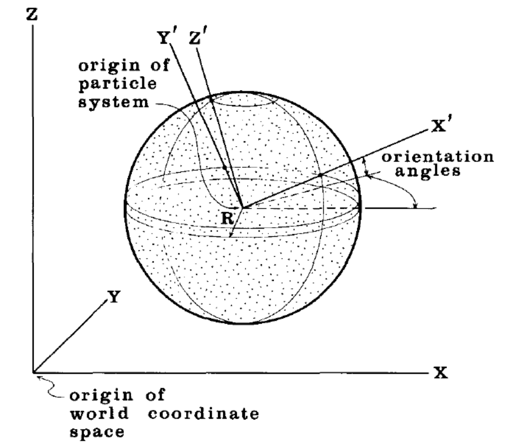
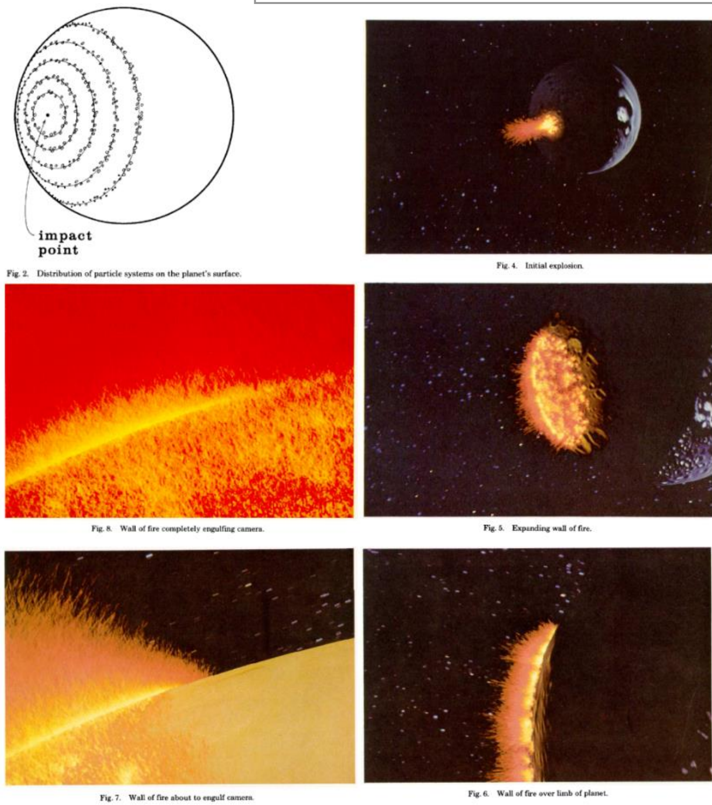


Fig. 1. Typical particle system with spherical generation shape.

W.T.Reeves, "Particle Systems—a Technique for Modeling a Class of Fuzzy Objects" ACM Transactions on Graphics, Vol. 2 , No. 2 (April 1983), pp 91 - 108

"Genesis Effect from Star Trek II: The Wrath of Khan“, 1982



[“Particle Dreams”, K.Sims, 1988]

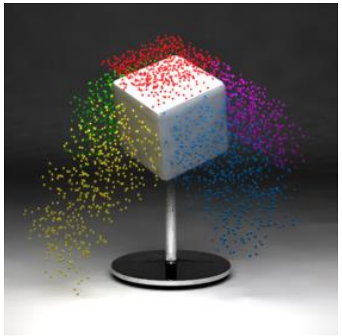
3.1.1 Rendering particle systems

- Points, line trails, billboards, blobs

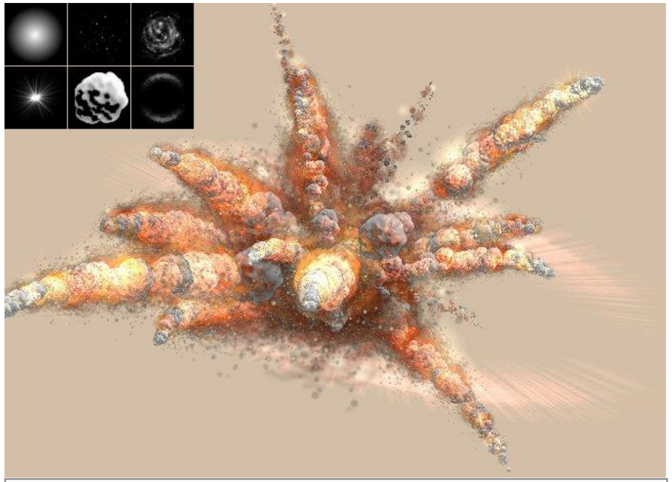
Fig. 12. *white.sand.*

By leaving a trail from one particle position to the next, the effect of grass growing can be created. Reeves, 83

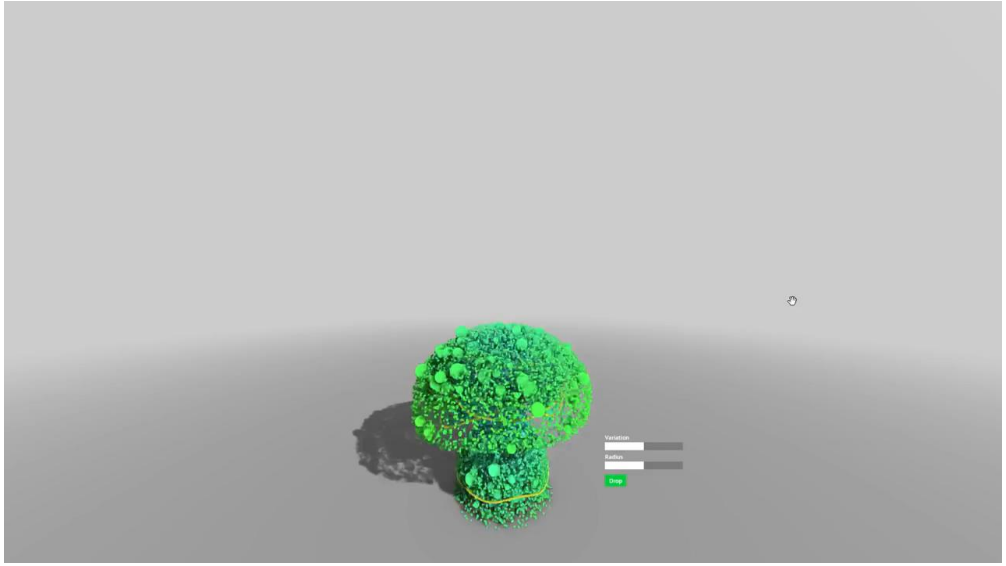
A cube emitting 5000 animated particles, obeying a "gravitational" force in the negative Y direction.
CC BY-SA 3.0, https://en.wikipedia.org/w/index.php?curid=14699785



The same cube emitter rendered using static particles, or strands.
CC BY-SA 3.0, https://en.wikipedia.org/w/index.php?curid=14699814

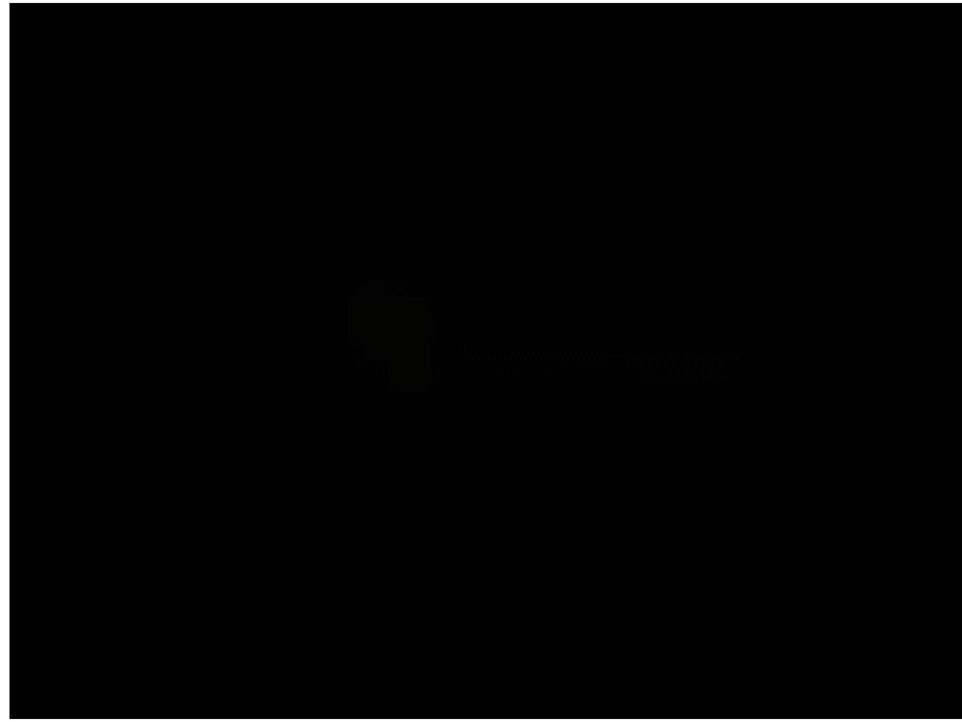


"Pi-explosion" by Sameboat - Own work. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Pi-explosion.jpg#/media/File:Pi-explosion.jpg



http://david.li/vortexspheres/

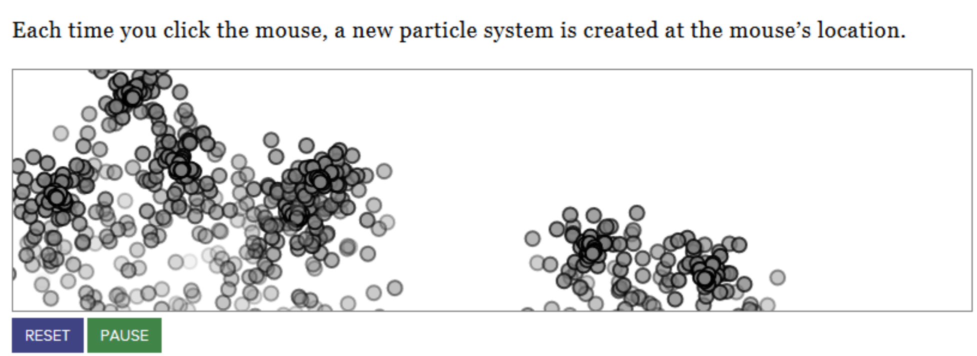
3.1.2 Other



Can mix effects with live video

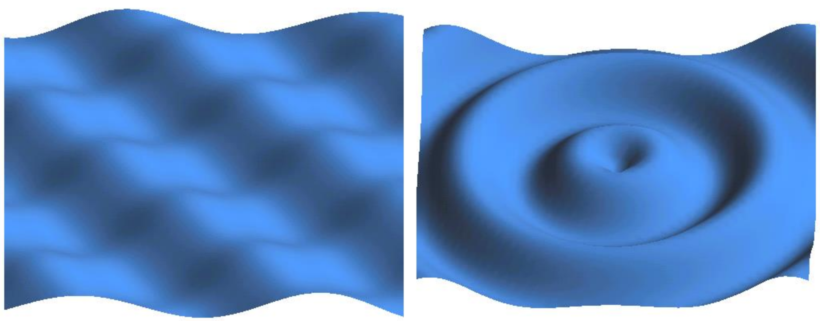
- See Particle Illusion (http://www.wondertouch.com/)

- Daniel Shiffman, "The nature of code" http://natureofcode.com/book/chapter-4-particle-systems/

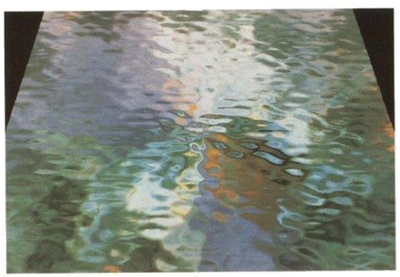


3.2 Deep water waves

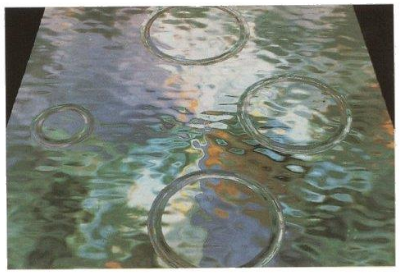
- Sinusoidal waves are added to a surface represented as a mesh of polygons
- The geometric resolution of the surface representation must be high in order to produce a smooth profile.



69 Simulating wind driven water waves: two frames from a chrome-mapped animation sequence showing waves approaching a beach.



70 A surface modelled from long-crested wave sets at two main orientations.

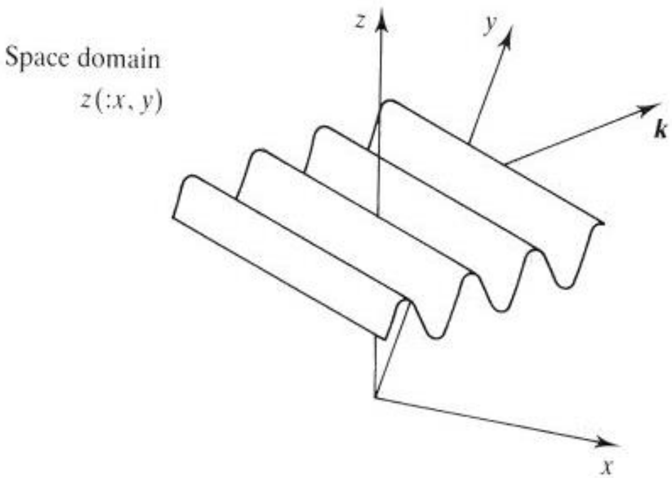


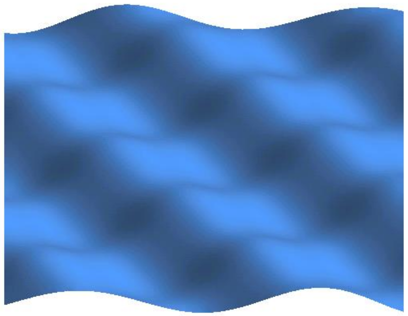
71 Adding radial waves as bump map packets.

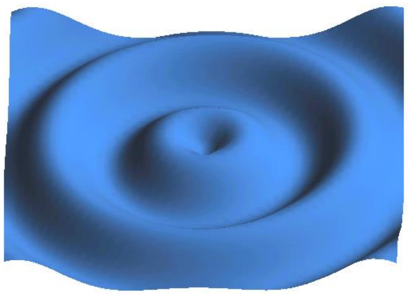
72 The chrome map used in the wind waves illustration.

3.2 Deep water waves

- Parallel waves can be animated using: z(x,y,t) = A cos(**k**.(**x**-**c**t))
- where:
  - A is the amplitude
  - **k** is the wave vector (u,v,0)
  - **c** is a velocity vector that lies in the direction of **k** and is given by **c** = v**k**
  - v = velocity = (g/w)½
  - g is acceleration due to gravity
  - w = wave number = 2π/L
  - L is the wavelength
  - **x** is the vector (x,y,0)







3.2.1 More complex models

Figure 3-13. Water spray generated when two waves of opposite direction meets.

Deep-Water Animation and Rendering, By Lasse Staff Jensen and Robert Golias, www.gamasutra.com, Sep 26, 2001

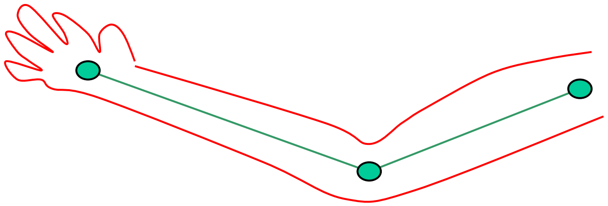
Figure 13
Beach at sunset. Notice the breaking of waves on the shore, and how the crests take the shape of the shore.

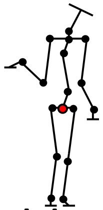
Fournier, A. and W.T.Reeves, "A Simple Model of Ocean Waves", Proc. SIGGRAPH 86, pp.75-84.

Damien Hinsinger, Fabrice Neyret, Marie-Paule Cani, "Interactive Animation of Ocean Waves", ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA) - July 2002

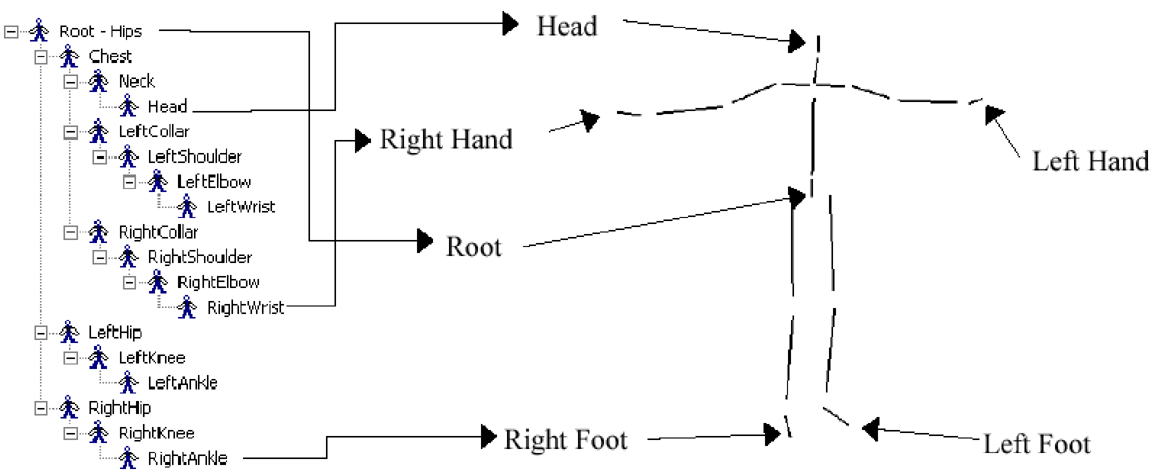
Gamito, M.N., Musgrave, F.K., "Procedural Terrains with Overhangs", *Proc. of the 10th Portuguese Computer Graphics Meeting*, pages 33-42, 2001.

3.3 Human figure animation

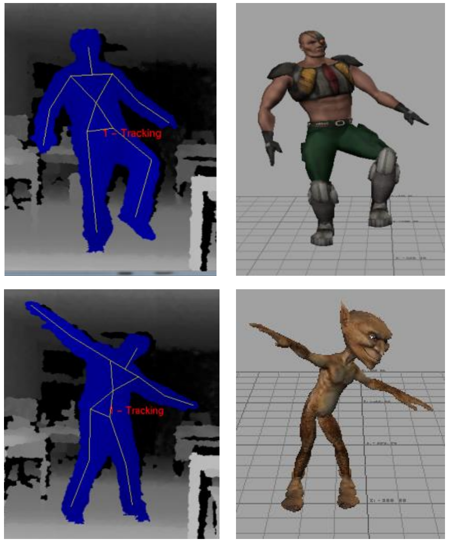




- A hierarchical data structure organises input from animator – enables cascading of information.
- Human skeleton > 250 DOFs;
  - Most joints 1 or 2 DOFs, few (e.g. shoulder) 3 DOFs;
  - Computer Graphics skeleton ~ 50 DOFs.
- A ‘skin’ is attached to the skeleton to give the final look of the figure



Hierarchical Structure for a Human Figure



3.3.1 Skeleton hierarchy

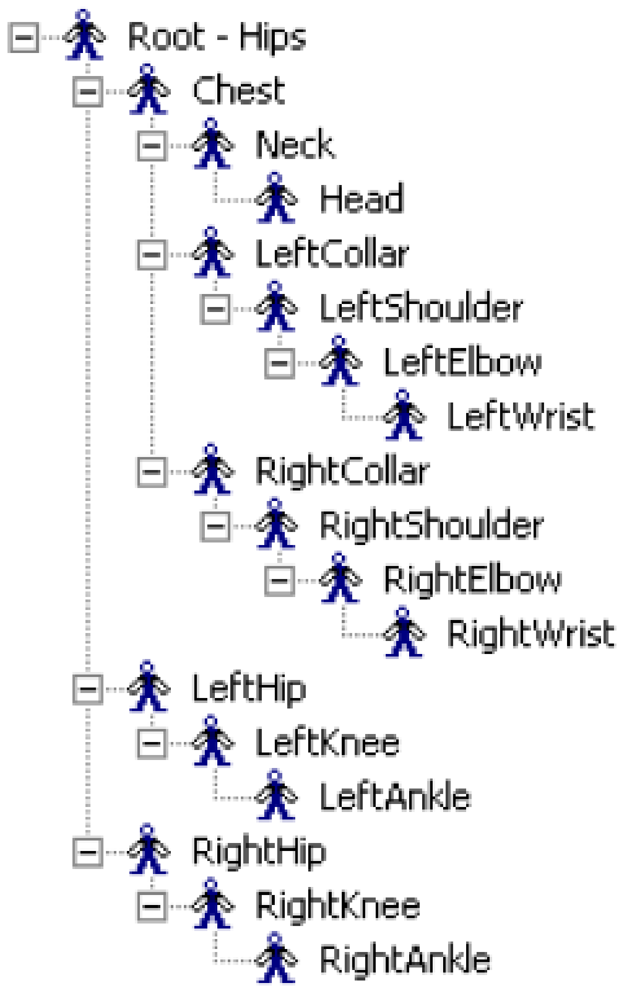
- To map a vertex, v, in the left hand into the root node, we use

  $v_w = Mv$

  $M = T_{Hips} R_{Hips} T_{Chest} R_{Chest} T_{LeftCollar}$
  $R_{LeftCollar} T_{LeftShoulder} R_{LeftShoulder}$
  $T_{LeftElbow} R_{LeftElbow} T_{LeftWrist} R_{LeftWrist}$

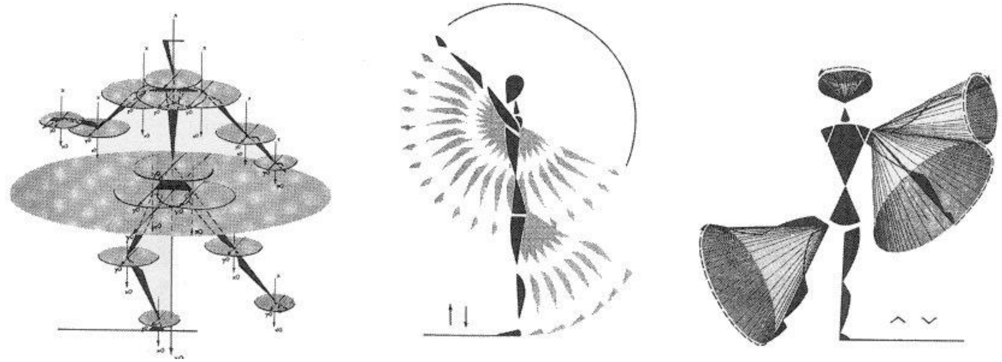
  $$M = \prod_{i=1}^{n} M_i$$

  where $M_i = T_i R_i$, and M_i is the local matrix for a particular node (T = translate, R = Rotate)
- All Ts except T_{Hips} remain constant throughout an animation sequence



3.3.2 Kinematic control

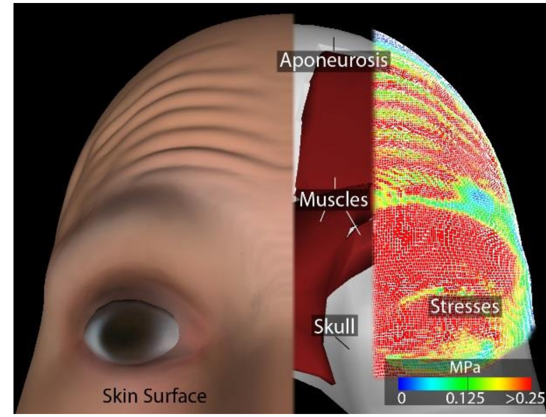
- Kinematics: study of motion independent of underlying physical forces
  - Includes position, velocity and acceleration.
- Forward kinematics
  - $X = f(\theta)$
- Inverse kinematics
  - $\theta = f^{-1}(X)$



Allegory of FK scripts – dance notations

Allegory of an IK script

3.4 Physically-based simulations



- We can also include physically-based simulations in this section. With these it is less clear how to produce 'controlled' motion.

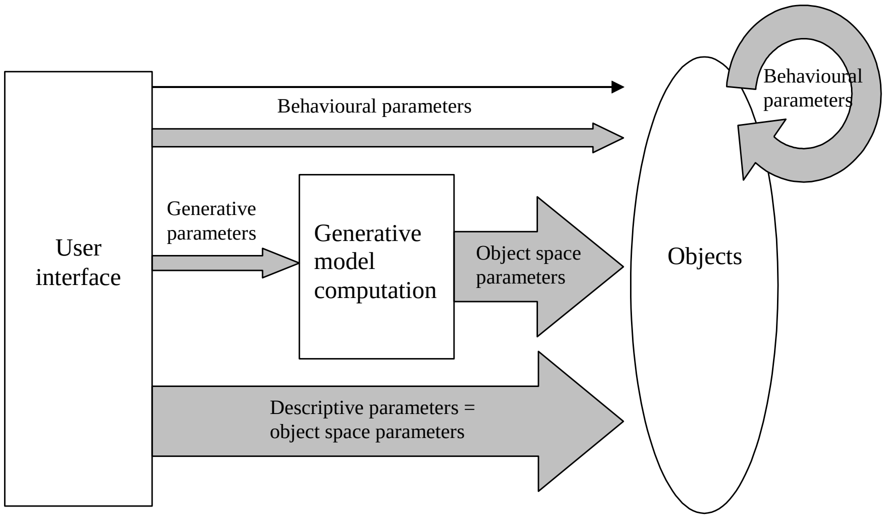
Lee Cooper. Physically Based Modelling Of Human Limbs , PhD thesis, Dept Computer Science, Univ. Sheffield, Sep 1998 (Supervisor: S Maddock)

Mark Warburton, PhD student, 2010-14 (Supervisor: S Maddock)

4. High-level techniques

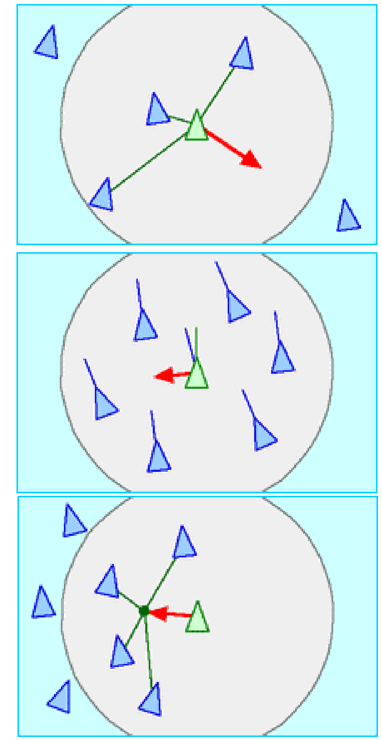
- An object internally transforms itself under the control of a set of rules mimicking a biological process, or acts and reacts to an external stimulus.
- Example internal production rules
  - Plant models expressed by L-systems - based on botanical knowledge.
- Example external stimuli:
  - Biological need
  - High-level task set by an animator such as “walk to the window and look at the garden”

- The control process is characterised by continual feedback in the process of animation control. Direct animation control is limited.

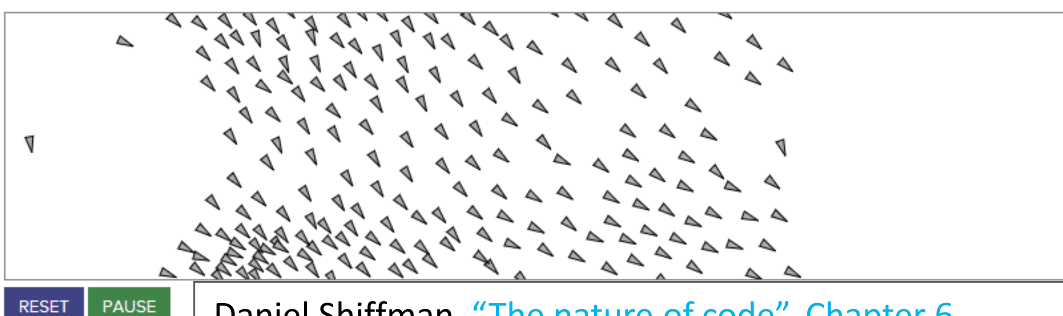


4.1 Flocking (Reynolds, 87)

- Three rules to organise individuals with respect to neighbours, i.e. the ‘internal’ behaviour of the flock:
  - *Collision avoidance*: avoid collisions with nearby flockmates
  - *Velocity matching*: attempt to match velocity with nearby flockmates
  - *Flock centring*: attempt to stay close to nearby flockmates
- A migratory point ahead of the flock is animated (possibly using a low-level technique) and the flock behaviour emerges.



[Original Boids demo]

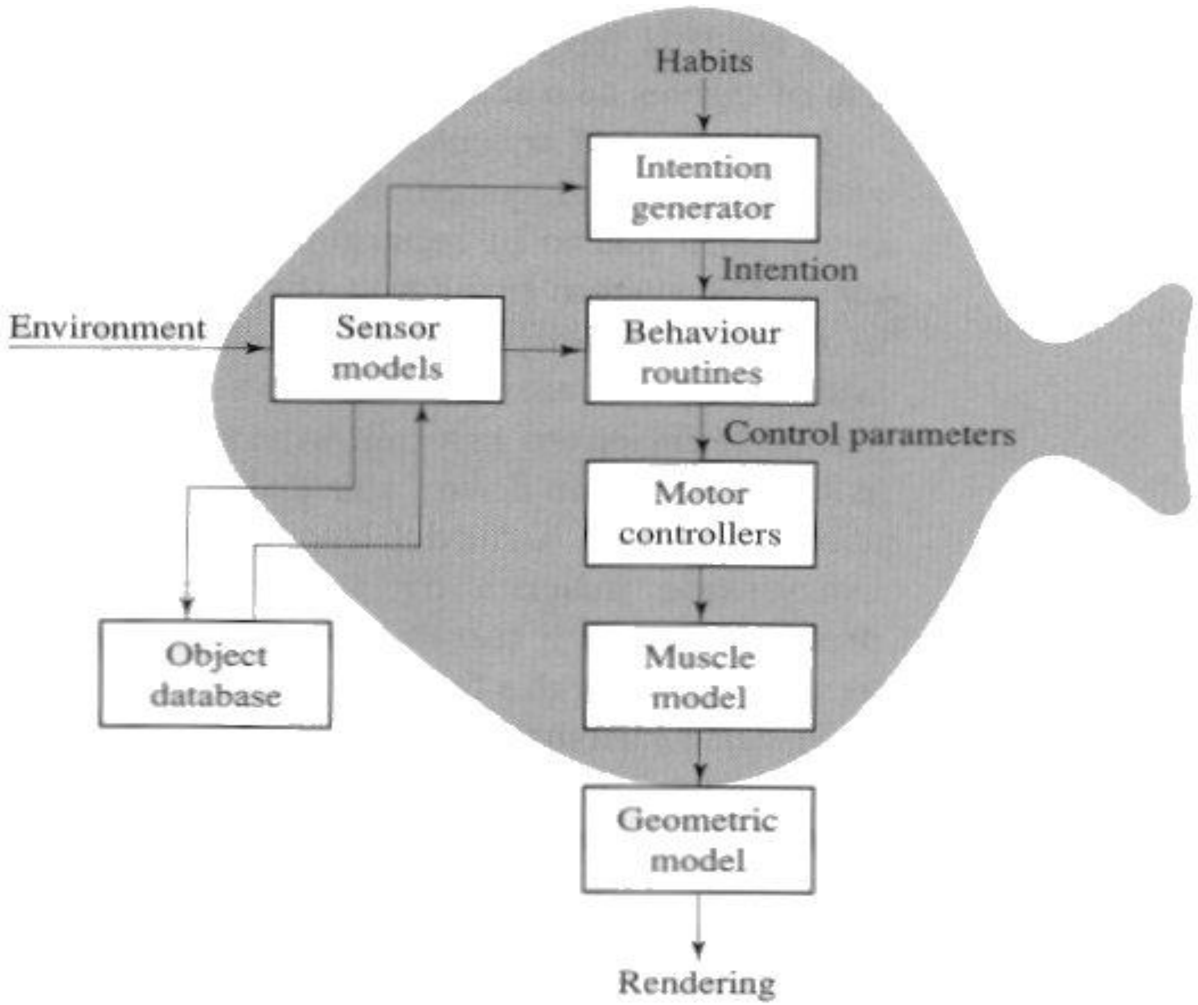


Daniel Shiffman, [“The nature of code”. Chapter 6]
[http://processingjs.org/]

Example: The Lion King, Disney, 1994

4.2 Sensors

- Example: Simulation of the behaviour of a school of fish (Tu and Terzopolous, 94)
- Individual fish are given a vision sensor and temperature sensor to help govern their behaviour, e.g. search for food and avoidance of predators.
- Agent-based systems

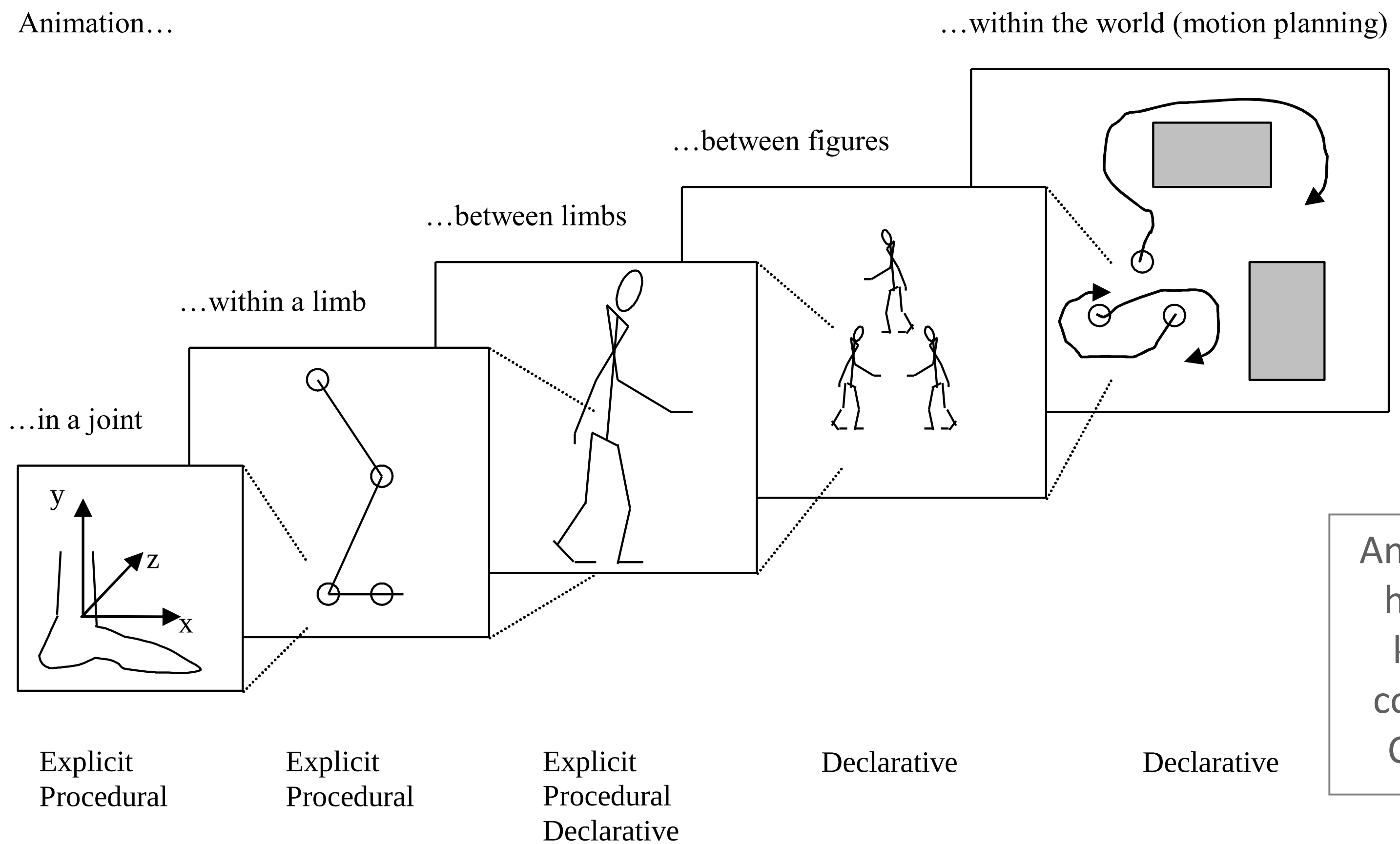


http://www.dgp.toronto.edu/~tu/

http://www.youtube.com/watch?v=aP1_XkCdAaE

5. Summary

- Low-level motion specification can be inordinately tedious (depends on DOFs) but offers full expressive control
- High level specification is easier for a 'general user' to understand and use, but generally results in less artistic freedom



Animation as a hierarchy of knowledge control (after Calvert, 94)

References

- Burtnyk N, Wein M (1971) Computer-generated Key-frame Animation, Journal SMPTE, 80, pp.149-153.
- Calvert, T., R.Ovans, S.Mah, “Towards the Autonomous Animation of Multiple Human Figures”, Proc. Computer Animation 94, (Geneva, Switzerland, May 25-28, 1994), pp.69-75.
- Hegron, G. and B.Arnaldi, "Computer Animation: Motion and Deformation Control", Eurographics '92 tutorial notes (Eurographics technical report series ISSN 1017-4656), 1992.
- Brian Knep, Craig Hayes, Rick Sayre. Tom Williams. Dinosaur input device Proceedings of the SIGCHI conference on Human factors in computing systems, Denver, Colorado, United States 1995, Pages: 304 - 309
- Prusinkiewicz, P. and A.Lindenmayer, “The Algorithmic Beauty of Plants”, Springer-Verlag, New York, 1990.
- Reynolds, C.W., “Flocks, Herds, and Schools: A Distributed Behavioural Model”, Computer Graphics, 21(4), (Proc. SIGGRAPH 87), pp.25-34.
- Sederberg, T.W., S.R.Parry, "Free-Form Deformation of Solid Geometric Models", Computer Graphics, 20(4), (Proc. SIGGRAPH 86, Dallas, Aug 18-22), 1986, pp.151-160.
- Sederberg, T.W. and E. Greenwood, “A Physically-Based Approach to 2-D Shape Blending” Proc. SIGGRAPH’92, pp.25-34.
- Ken Shoemake. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques* (SIGGRAPH '85). ACM, New York, NY, USA, 245-254.
- Steketee, S.N. and N.I.Badler, “Parametric Keyframe Interpolation Incorporating Kinematic Adjustment and Phrasing Control”, Computer Graphics, 19(3), (Proc. SIGGRAPH 85), pp.255-262.
- Thalmann, N.M. and D,Thalmann, “Computer animation: a key issue for time visualisation”, in Rosenblum, L., R.A.Earnshaw, J.Encarnacao, H.Hagen, A.Kaufman, S.Klimenko, G.Nielson, F.Post, D.Thalmann (eds.), “Scientific Visualization: advances and challenges”, Academic Press, 1994, chapter 13, pp. 201-222.
- Thomas, F. and O.Johnston, “Disney Animation: The Illusion of Life”, Abbeyville Publishers, NY, 1981.
- Tu, X. and D.Terzopoulos, “Artificial Fishes: Physics, Locomotion, Perception, Behavior”, Proceedings of SIGGRAPH 94 (Orlando, Florida, July 24-29, 1994). In Computer Graphics Proceedings, Annual Conference Series, 1994, pp.43-50.
- Watt, A. and M.Watt, “Advanced animation and rendering techniques: theory and practice”, Addison-Wesley, 1992.
- Zeltzer, D., “Towards an Integrated View of 3D Computer Animation”, The Visual Computer, 1(4), 1985, pp.249-259.

Particle systems:

- Reeves, W.T., “Particle systems – a technique for modelling a class of fuzzy objects”, Computer Graphics, 17(3), (Proc. SIGGRAPH 83), 1983, pp. 359-376.
- K.Sims, "Particle Animation and Rendering Using Data Parallel Computation", Computer Graphics (Siggraph '90 proceedings), Aug. 1990, pp.405-413.
- http://threejs.org/examples/#webgl_points_sprites
- Tamas Umenhoffer, Laszlo Szirmay-Kalos, Gabor Szijarto, “Spherical Billboards and their Application to Rendering Explosions”, Proc. GI'06, pp. 57-63
- http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/