Computer Animation

Contents

Introduction

- Pixar, 1978, Graphics Group, Ed Catmul – cofounder
- 1986 – acquired by Apple, 2006 – acquired de Walt Disney Company.

Toy Story (1995)
A Bug's Life (1998)
Toy Story 2 (1999)
Monsters, Inc. (2001)
Finding Nemo (2003)
The Incredibles (2004)
Cars (2006)

Ratatouille (2007)
WALL-E (2008)
Up (2009)
Toy Story (2010)
Cars 2 (2011)
Brave (2012),
Monsters University (2013)

Introduction

- DreamWorks, 1994
- Movies:

  Antz (1998),

  Shrek (2001), Shrek 2 (2004), Shrek 3 (2007), Shrek Forever After (2010),

  Spirit: Stallion of the Cimarron (2002),

  Madagascar (2005), Madagascar: Escape 2 Africa (2008), Over the Hedge (2006), Flushed Away (2006), Bee Movie (2007),

  Kung Fu Panda (2008), Kung Fu Panda 2 (2011),

  Monsters Vs. Aliens (2009), How to Train Your Dragon (2010), Megamind (2010), Puss in Boots (2011),

  Madagascar 3 (2012), Rise of the Guardians (2012), The Croods (2013), Turbo (2013), How to Train Your Dragon 2 (2014),

  Penguins of Madagascar (2014), Home (2015)

  Bridge of Spies (2015)

Introduction

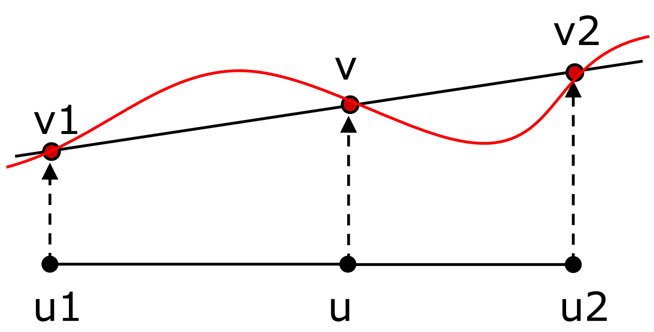
- Walt Disney Company
- Cartoons:
  - The Little Mermaid (1989)
  - Beauty and the Beast (1991)
  - Aladdin (1992)
  - The Lion King (1994)
  - Hercules (1997)
  - Mulan (1998)
  - Tarzan (1999)
  - Monsters, Inc. (2001)
  - The Lost Empire (2001)
  - WALL-E (2007)
  - The Princess and the Frog (2009)
  - Winnie the Pooh (2011)
  - Frozen (2013)
  - Big Hero 6 (2014)
  - Cinderella (2015)
  - Zootopia (2016)
  - Jungle Book (2016)

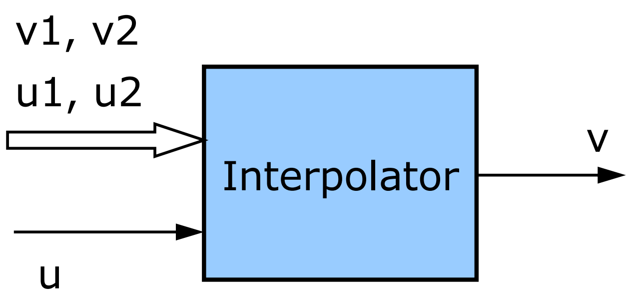
Introduction

- Animation – movement illusion by rapid display of different image sequence
- Classic approach:
  1. Define a sequence of keyframes
  2. Render the intermediate frames computed by the first and the last keyframes

Introduction

- Computation of the intermediate frames creates the sensation of animation, modifying:
  - Object attributes (position, shape, color, etc.)
  - Attributes of the camera (position, orientation, etc.)
  - Attributes of light sources (light intensity, position, color, etc.).
- Computation of intermediate values between two limit values (eg position, color, orientation, intensity, etc.) is performed by interpolation: linear, curve, etc.





Computing key positions

- Direct kinematics
- Reverse kinematics
- Modeling physical systems
- Constraint-based modeling
- Procedural modeling
- Behavior-based modeling
- Capturing key positions through sensors mounted on real objects

Computing intermediate positions

Modifying some features of the visualization system.

1. For objects in the scene:
   - Position
   - Orientation
   - Dimension
   - Shape
   - Color
   - Transparency
2. For the camera:
   - Point of view
   - Point of interest
   - The angle of view
3. For the light source:
   - Position
   - Light intensity

Computing intermediate positions

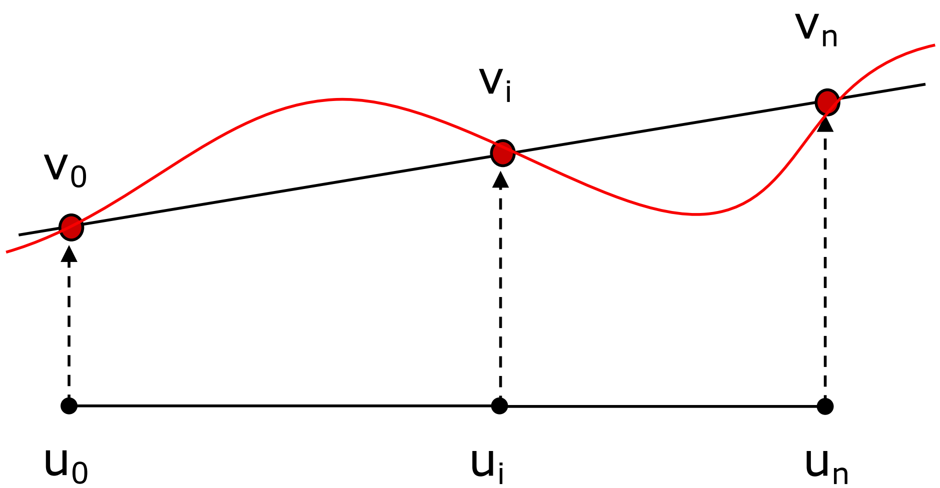
$u_i = u_0 + i\,\Delta u$

$v_i = v_0 + i\,\Delta v$

where

$\Delta u = (u_n - u_0)/n$

$\Delta v = (v_n - v_0)/n$



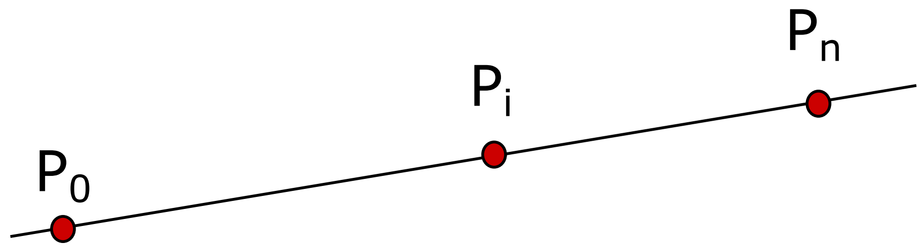
Iterative expression:

$u_{i+1} = u_i + \Delta u$

$v_{i+1} = v_i + \Delta v$

Computing intermediate positions

Value space generalization



Let be P_0, P_n the values of P parameter or attribute on frame 0 and n.

$P_i = P_0 + i\ \Delta P$, where $\Delta P = (P_n - P_0)/n$

$P_i = P_0 + i\ \Delta P = P_0 + i/n\ (P_n - P_0)$

If consider $u = i/n$, $u \in [0, 1]$

$P_i = P_0 + u\ (P_n - P_0)$, the parametric equation of linear variation

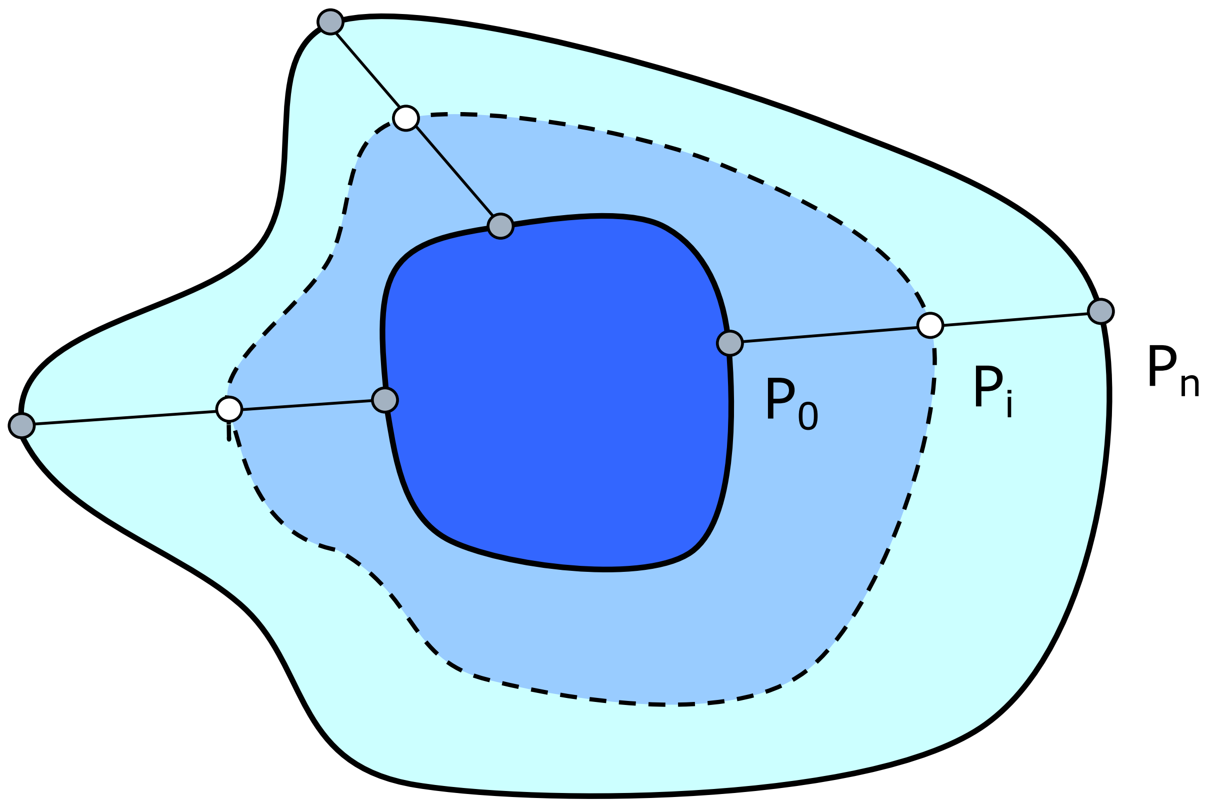
iteratively:

$P_{i+1} = P_i + \Delta P$

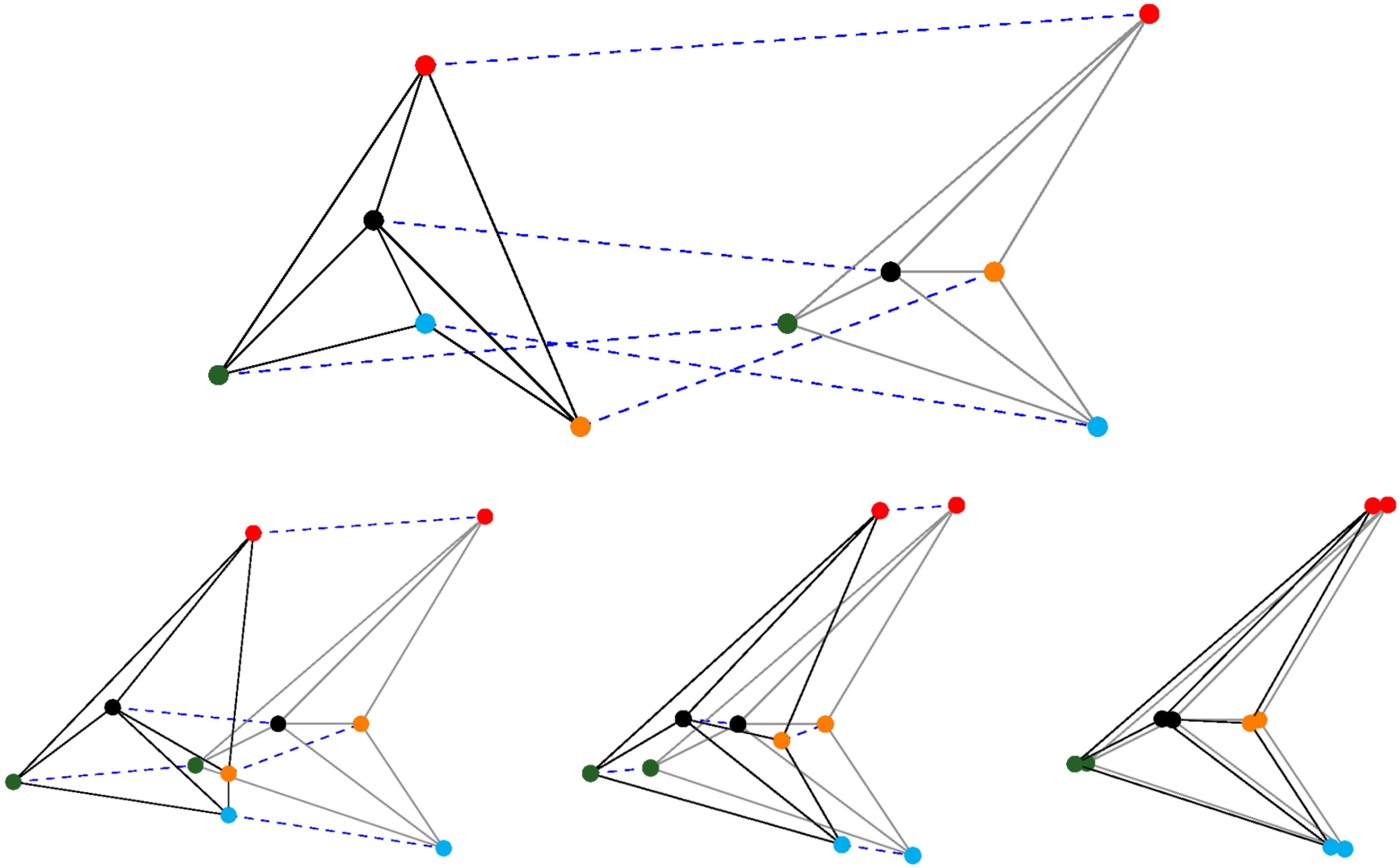
Example: P may be orientation angle, speed, acceleration, position, size, light intensity, transparency level, etc.

Morphing

- change smoothly from one shape or image to another by small gradual steps using computer animation techniques.



Morphing



Classic animation

The traditional animation technique is very laborious:

1. Defining a scenario
2. Drawing the key frames that define the action (main designer)
3. Drawing the intermediate frames to complete the animation (drawing team)

Actual animation

1. Launching an animation story idea.
2. Elaboration of a document that describes the main idea of the animated story.
3. Cartoons that outline the main sequences of actions and dialogues. The main emotional feelings of the characters are highlighted.
4. The initial voices (e.g. of the members of the team of cartoonists and screenwriters) are recorded. These will be later replaced by the voices of some actors.
5. Making the first videotape that already contains the materials from which the animated film will be edited.
6. The art department creates the graphic aspect (e.g. look and feel) of the environment and the characters.
7. 3D modeling of the characters.

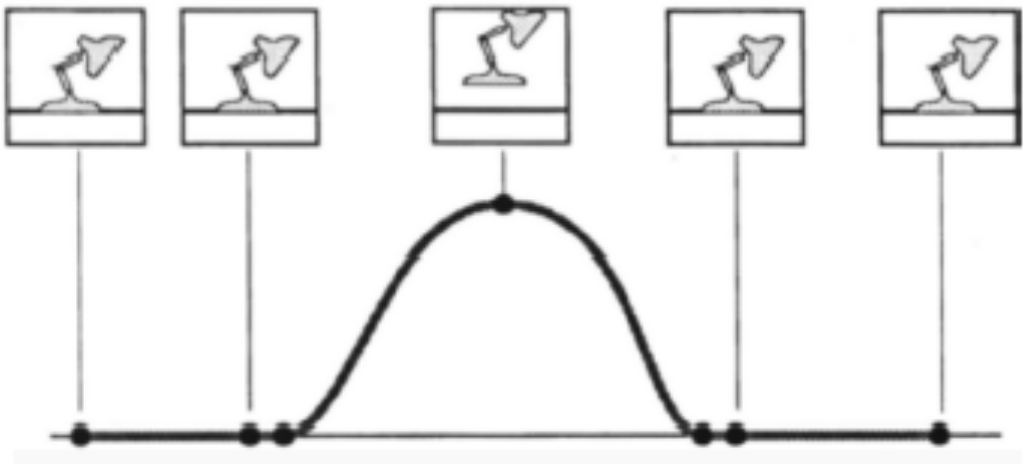
Reference: Pixar methodology, http://www.pixar.com/howwedoit/index.html

Actual animation (2)

8. 3D instances of significant scenes of the characters' positions and emotional states are created. Make pictures for these instances.
9. The instances are animated. Animators do not draw or color the characters. They are like actors or puppets. Using specialized software tools, the animators create the choreography of the characters' facial movements and expressions in each scene. These are key frameworks. Intermediate frames are achieved by the computer.
10. Create shadows and textures on the surfaces of objects. The shape of the objects was given by the model (step 7).
11. Light sources are added and special lighting effects are created.
12. Compute the animation by the computer system, using all the information specified so far. At least 24 frames/sec are created for the film. Computing a frame could take several hours (e.g. from a few to tens of hours).
13. Special sound and animation effects are added.

Automatic animation - fragmentation

- Computation of intermediate frames by interpolation algorithms
- Key frames correspond to the parameter values at certain times
- The quality of fragmentation is less than that obtained manually
- Solutions: (a) more intermediate frames, (b) "smoothing" the transitions from one frame to another

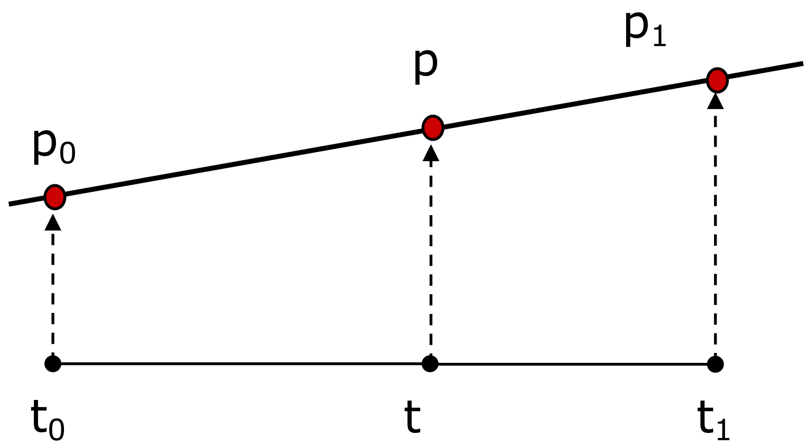


ACM © 1987 "Principles of traditional animation applied to 3D computer animation"

Linear interpolation

- The most used technique for computing intermediate values for a given interval
- Let be two values p0 and p1 of the parameter p at times t0 and t1, an intermediate value is:

$$p(t) = \frac{t_1 - t}{t_1 - t_0} p_0 + \frac{t - t_0}{t_1 - t_0} p_1 \qquad (1)$$



or other forms of the same expression:

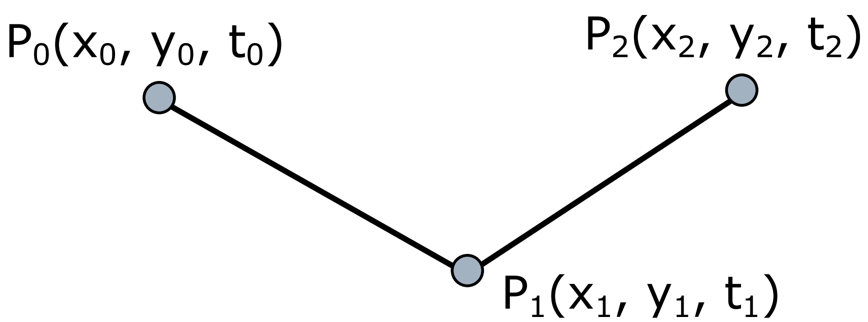
$$p(t) = p_0 + \frac{t - t_0}{t_1 - t_0}(p_1 - p_0) \qquad (2)$$

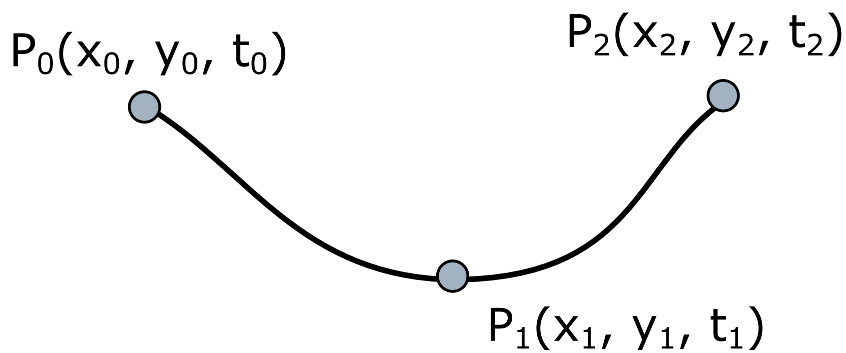
$$p(t) = \frac{p_0 t_1 - p_1 t_0}{t_1 - t_0} + t\frac{p_1 - p_0}{t_1 - t_0} \qquad (3)$$

$$p(t + \Delta t) = p(t) + \Delta t\frac{p_1 - p_0}{t_1 - t_0} \qquad (4)$$

Linear interpolation

- Benefits:
  - Changing the value of a parameter along a segment is linear (3).
  - The rate of changing along a segment is constant and may be controlled (4).
  - Computing the value at time t + Δt is achieved simply by adding a constant to the previous value at time t.
- Disadvantages:
  - There are discontinuities in all segment connection positions (e.g. frames / key positions Po, P1, P2).





Speed control

- Given a function of modifying a parameter (1-4) it is desired to control this modification (animations) depending on the time variable.

$$p(\tau) = \frac{\tau_1 - \tau}{\tau_1 - \tau_0} p_0 + \frac{\tau - \tau_0}{\tau_1 - \tau_0} p_1$$

- Depending on the time, a variable speed of parameter change (animation) can be controlled.

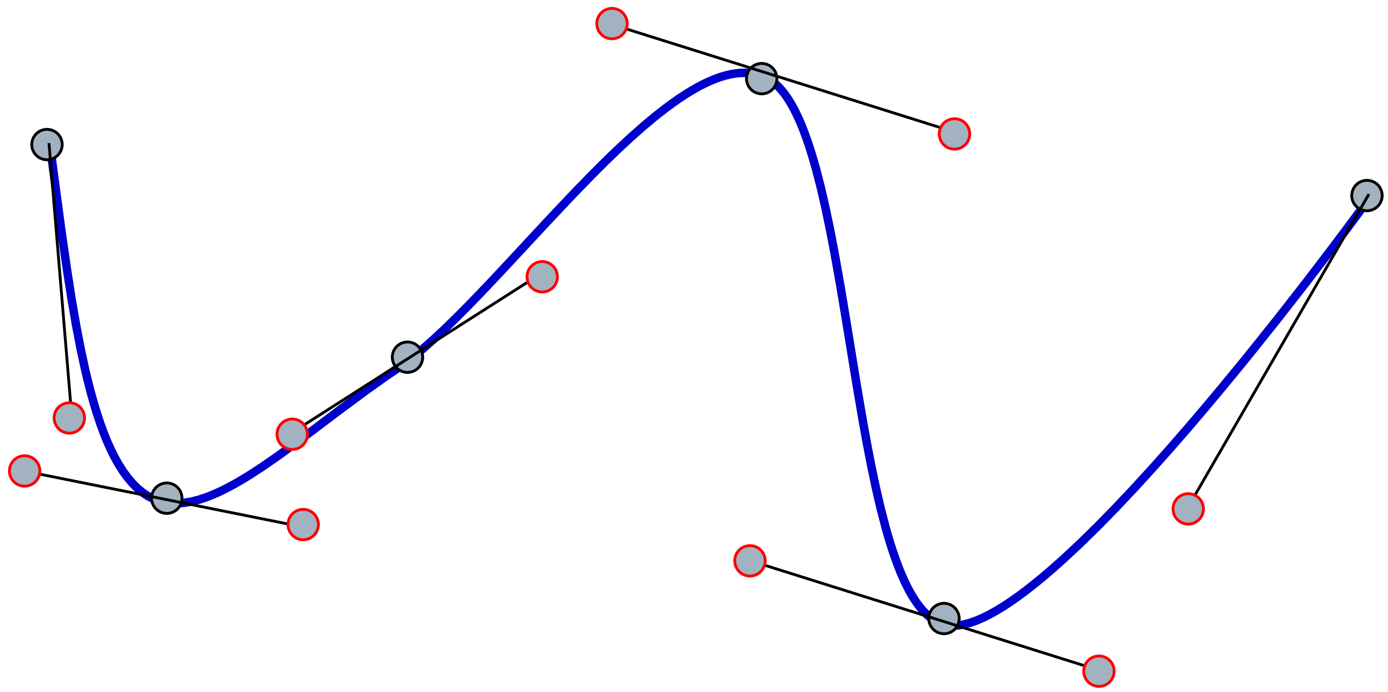
- Let this function be $\tau = f(t)$. We pass to the variable τ in the expression of linear interpolation:

$$p(\tau) = p(f(t)) = \frac{f(t_1) - f(t)}{f(t_1) - f(t_0)} p_0 + \frac{f(t) - f(t_0)}{f(t_1) - f(t_0)} p_1$$

Spline Interpolation

- Along with the linear interpolation, curve interpolation based on Spline curves (parametric curves) is used..
- Advantages:
  - Allows continuous control of the change
  - The number of intermediate frames can be determined depending on the quality required for the animation
- Disadvantages:
  - Computation points of the spline curves is based on interpolation constraints defined for the control points
  - Additional information is required (e.g. tangent vectors at the end points)
  - Computation is more expansive
  - Implementation is difficult

Spline Interpolation



Spline Curves

B-splines: allow automatic control of the continuity, but only approximate the control points, without going through them. Allow local control. Changing a checkpoint affects only the part in the vicinity of that checkpoint.

Non-uniform B-spines: needed to interrupt the continuity.

Non-uniform rational B-splines (NURBS): allow additional local control of the shape of the curve.

Bézier: require additional information for each segment and do not automatically maintain continuity. But it interpolates the ends of the segments. The adjustment of the smooth coupling is done through the control points at the ends.

Hermite: requires the definition of tangent vectors for each keyframe. The adjustment of the smooth coupling is done by adjusting the tangents at the ends.

Catmull-Rom: ensures the continuity of the first derivative (smooth trajectory). Catmull-Rom allow local control.

Beta-splines: allow the automatic continuity, the control over the tangent and approximate the length of the arc of each segment.

Control on the trajectory

- For basic travel, two basic facilities are required:
  1. Independent speed control along the trajectory
  2. Continuity control
- Spline curves allow continuity control, but speed control is more difficult
- The interpolation of the position on the spline curve allows the control of the continuity of the positions, but:
  - An equal increment of the spline parameter does not always correspond to an equal increment in distance along the curve
  - Different segments of the spline curve, corresponding to the same parametric length, may have different physical lengths
  - If the spline curve is parameterized directly according to the time variable, the animation will be with non-uniform speed

Distance parameterization on the curve

Transformation steps:

1. Given a spline step $P(u) = |x(u), y(u), z(u)|$, compute the length of the spline arc as a function of u: $s = A(u)$
2. Compute the inverse function A(u): $u = A^{-1}(s)$
3. Substitute $u = A^{-1}(s)$ from P(u) to find the parameterization of the motion path by the length of the arc, s: $P(u) = P(A^{-1}(s))$

   Note: The parameter u (and thus s) must be a global parameter for all segments of the initial spline curve.

Speed control on the curve

Speed control along the spline trajectory:

- Let $s = f(t)$ be the definition of the distance on the spline curve, as a function of time t
- The function $f(t)$, as a scalar value, can be processed by a functional animation technique (e.g. another spline curve)
- The function $f(t)$ can be expressed as an integral of the velocity function, $v(t) = df(t)/dt$.

  Note: the integral of a spline function can be calculated analytically, through control points

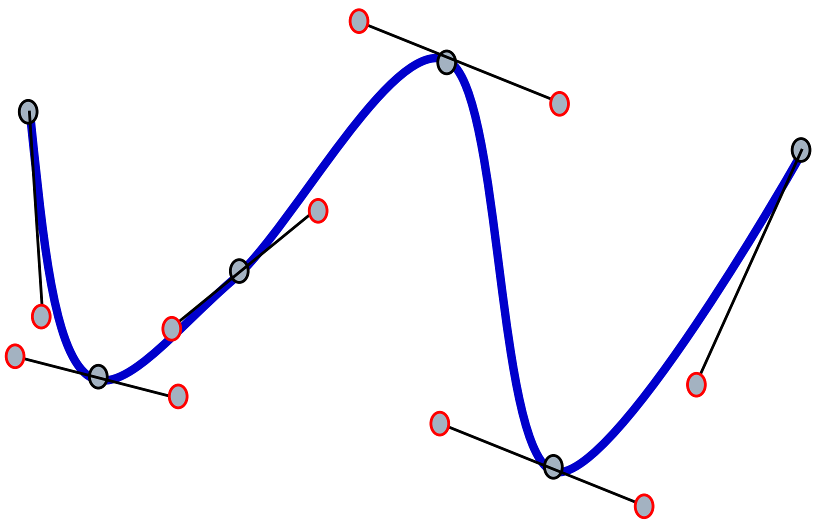
- The step of movement as a function of time is given by:

  $s = A(u)$, $u = A^{-1}(s)$

  $P(u) = P(A^{-1}(s)) = P(A^{-1}(f(t)))$

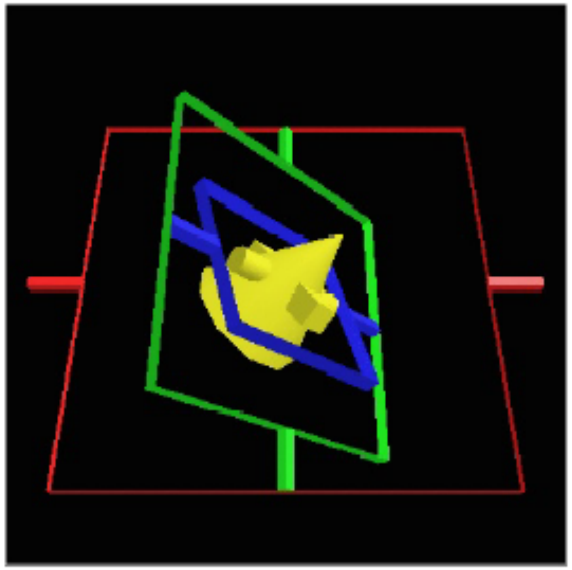
Interpolation of the orientation

- Interpolation of the orientation is computed indirectly by interpolating angles and not by transformation matrices:
  - In 2D the orientation is defined by a single angle, and the animation is relatively simple
  - In 3D, three angles are used, and the topology is of a sphere in four dimensions. Interpolation is more difficult to visualize.
- Methods for interpolating object orientation
  - Euler angles
  - Quaterni
  - Spherical and linear interpolation



Euler angles

- Three angles are used: rotation after x, then rotation after y and then rotation after z
- Disadvantages: difficult to solve the inverse problem - given the orientation, what are the angles?
- Advantages: Widely used in practice, easy to implement, inexpensive method.



Quaterni

- Four elements analogous to complex numbers. The computations are based on operations with complex vectors and numbers
- Disadvantages: requires more math, expensive interpolation
- Advantages: simple solution of the inverse problem, simple user interface

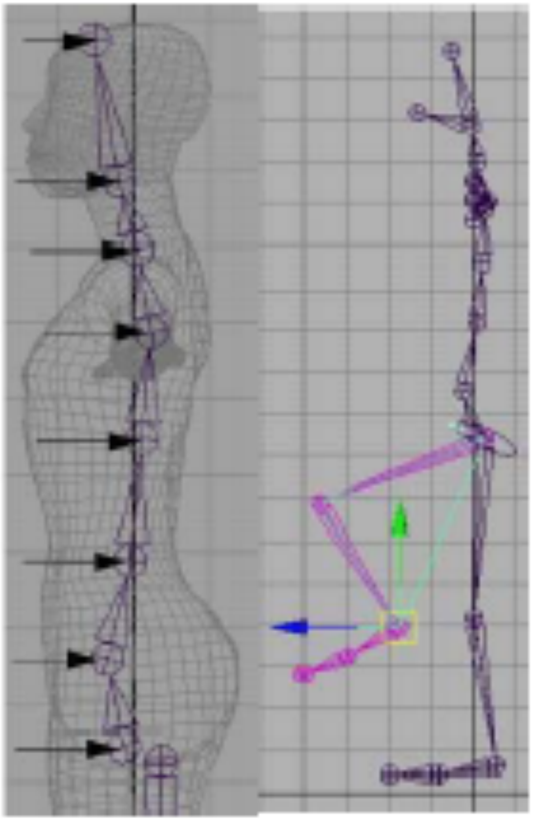
Camera movement

- The animation of the camera can be done by combining the interpolation of positions and orientation. However, there are some specific requirements such as:
  - The camera must always be horizontal, unless otherwise specified
  - The images of the objects of interest must be stable in the plane of the film
  - The specification of the camera movement must comply with certain kinematic conditions

Animation of objects with complex shapes

- Given a model with a very large number of vertices, they can be grouped and processed as a unit. For example, the dots that describe the forearm move more or less like a rigid body
- Connectivity between groups can be represented by linking each group to a "bone" that is part of an articulated "skeleton"
- The associated bones and tips are treated as a single rigid object in transformation operations
- The movement of the bones is constrained by the connections in the skeleton. The skeleton is an articulated structure
- The different forms of connection (rotation, angle, sphere) allow different forms of movement
- The animation can be viewed quickly using only the skeleton drawing. Finally, the model is viewed without bones

Animation of objects with complex shapes



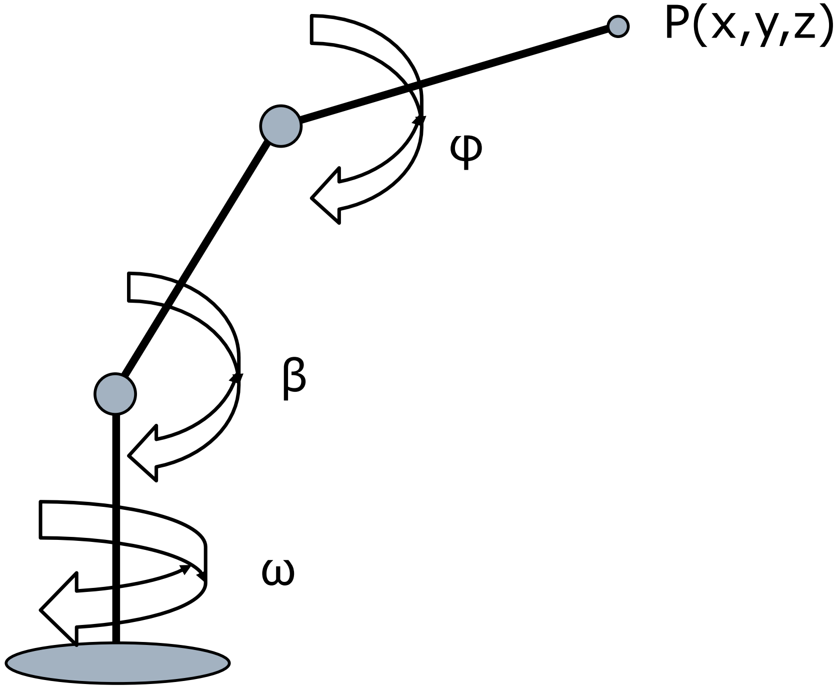
Images from the Maya tutorial

Ref: Automatic Splicing for Hand and Body Animations Majkowska, A., Zordan, V. B., Faloutsos, P. ACM SIGGRAPH/Eugorgraphics Symposium on Computer Animation (SCA) 2006

Kinematics of articulated structures

- **Kinematics:** the study of motion independent of the forces that determine motion. Consider position, speed and acceleration.

  1. Direct kinematics (φ, β, ω -> P(x,y,z))
  2. Inverse kinematics (P(x,y,z) -> φ, β, ω)



Direct kinematics

- Forward Kinematics
- It is calculated for all the links in an articulated model:
  - Positions
  - Speeds
  - Accelerations
- Are given:
  - For the root node:
    - Position
    - Speed
    - Acceleration
  - All the transformations between connections
- Direct kinematics is important in animation based on skeletal frames. It is a relatively easy method to implement.

Inverse kinematics

- Inverse Kinematics
- Determining the movement of intermediate links in an articulated body. Is given the movement of key links.
- Characteristics of the method based on inverse kinematics:
  - Nonlinear, undetermined
  - Complexity of computing the solution is proportional to the number of free links
  - Free joining between two fixed links can be solved efficiently only by constraints on a degree of freedom
  - Additional constraints are needed to obtain a unique and stable solution. Example, consideration of gravitational force, constraints imposed by some types of joints (rotation vs. spherical, knee vs. shoulder).
  - Additional optimization objectives. The result is the optimization problem solved iteratively as an animation process. Example: minimizing the kinetic energy of the structure.

Alternative animation techniques

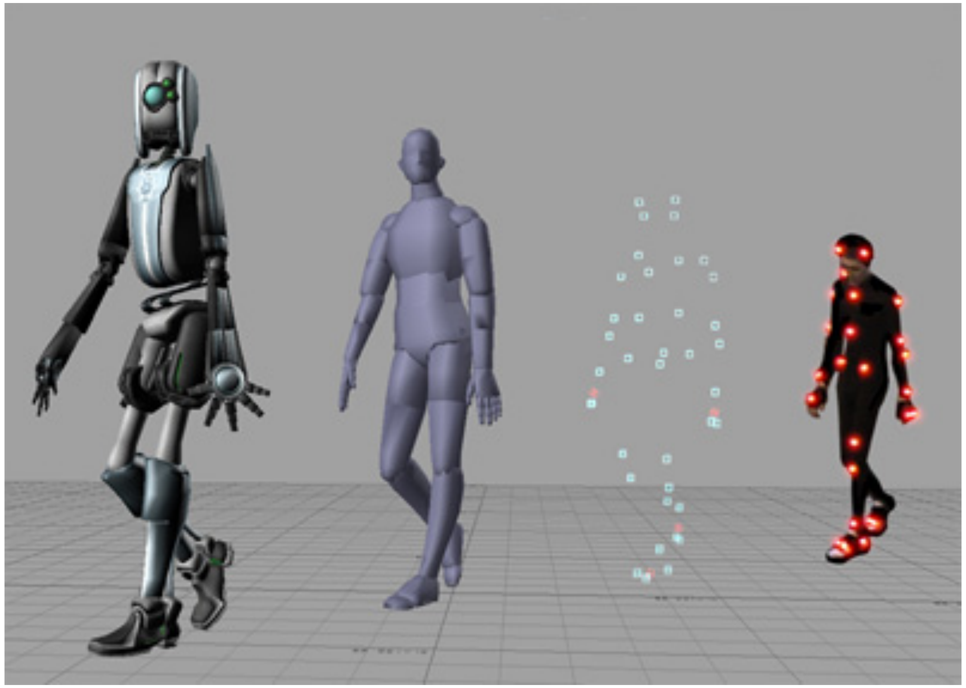
- Morphing transformation
- Movement capturing
- Particle systems
- Group (Flocking)

Morphing transformation

Morphing transformation

- Interpolation transformation between images (the ‘70s).
- Good effects at a low cost
- Main steps:
  1. Partitioning images into appropriate pieces. E.g. eye to eye, etc.
  2. Interpolation of parts (geometry)
  3. Attribute interpolation (pixels, colors)
  4. Filtering

Movement capturing



Movement capturing

Movement capturing

- Complex animations require a lot of key frames (for example, walking a person)
- Capturing key positions from real objects. The positions of some key points on the real object (e.g. an actor) are recorded in the model
- Position tracking technologies:
  - Electromagnetic sensors for position and orientation. Requires cables to each sensor. Reading may be disturbed by magnetic fields. Movement space is limited
  - Ultrasonic triangulation. Simple but less accurate than magnetic sensors
  - Optical triangulation. Transmitters are attached to the tracked object. Two cameras determine the position. It can track multiple points without using cables
  - Body suits. Sensors that determine the angles of the links are inserted into the material of the clothes.

Facerig application

- Allows the tracking of the human face by the web camera (https://facerig.com/).
- Markerless tracking. It recognizes and follows facial features such as nostrils, corners of lips and eyes, or wrinkles..

Particle systems

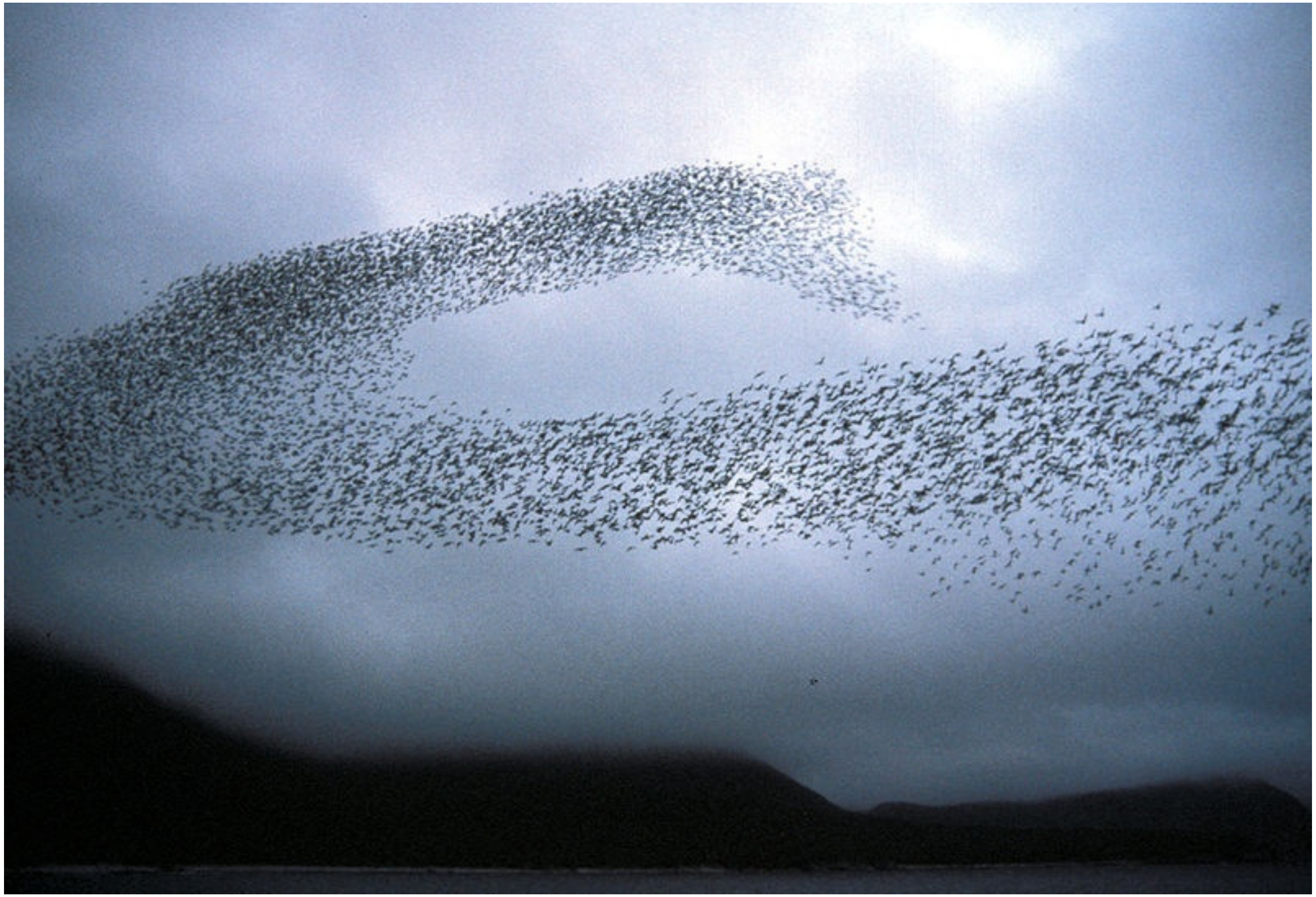

Particle systems

- Model "fuzzy" objects with:
  - blurred edges
  - changing edges
  - chaotic behavior. E.g. clouds, fire, water, grass, fur, etc.
- Principle: it models fuzzy objects as sets of particles.
- Each particle has properties such as:
  - Geometry (point, sphere, right segment)
  - Position
  - Vector speed
  - Dimension
  - Color
  - Transparency
  - Status
  - Lifetime

Particle system animation

- New particles are born, old ones die
- Status information describes the particle type
- At any time:
  - Updates the attributes of all particles
  - Remove old particles
  - Create new particles
  - Displays the current status of all particles
- For antialiasing, draw a right segment from the old to the new position
- To draw grass or fur, draw the entire trajectory of the particle

Flocks animation

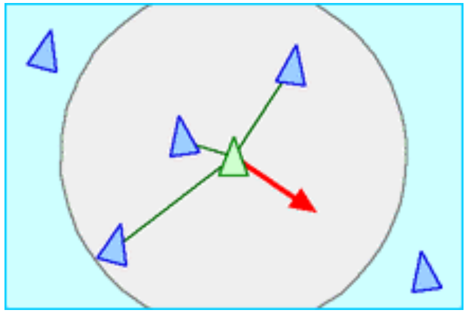


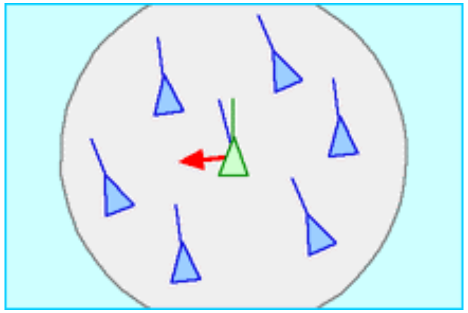
Animation of groups

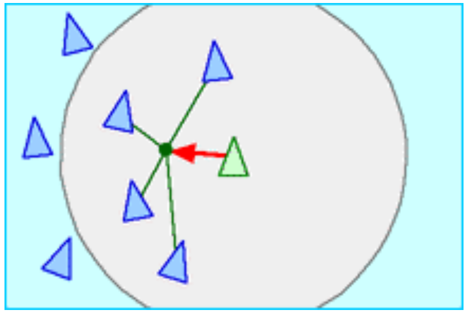
- Animation of animal groups: flocks of birds, fish, herds, etc.
- There are too many individuals to be animated each by keyframes
- They are fewer elements than in particle systems, but with more interactions
- It is necessary to avoid collisions with other objects (e.g. buildings, trees) or between them
- The group has a center: each member tries to remain a member of the group, the center of the group is a global one, it does not allow the division of the group to pass an obstacle

Boids

- Group operations
  1. Separation
     - Avoid congestion
  2. Alignment
     - Alignment with the average orientation of the local group
  3. Cohesion
     - Move to the average position of the local







groupRef.: Craig Reynolds, Boids, http://www.red3d.com/cwr/boids/ (1986).

Questions and proposed problems

1. Exemplify the animation by modifying the object’s attributes.
2. Exemplify the animation by modifying the camera’s attributes.
3. Exemplify the animation by modifying the light source’s attributes.
4. Exemplify the value space for temperature, and an animation by modifying the temperature. What could be the reference variable and the value for a linear interpolation within the value space?
5. What is the main advantage in linear interpolation if the step of the variable is a constant? How are computed the intermediate values?
6. Explain a solution for animated morphism of a triangle into a pentagon. How do you control the transformation to be achieved by the same: (a) time; (b) speed.
7. Explain a solution for animated morphism of a cube into a sphere. How do you control the transformation to be achieved by the same: (a) time; (b) speed.

Questions and proposed problems

8. Describe and exemplify the steps of the actual cartoon video development methodology to animate a talking person.
9. Why initially to register the voices is necessary, even finally they will be replaced?
10. Explain what means to build up the 3D model of the characters in the cartoon video development methodology?
11. Explain and exemplify how are created the key frames for a cartoon movie?
12. Why to animate a faster movement you need to register more than 24 frames per second? Explain why the movie does not replay all the details of the movement?
13. Explain why the 12^{th} step of the cartoon video development methodology requires so high performance computation resources?
14. Explain the linear interpolation of the movement if the reference variable is the time. The animation is performed between the time t1 and t2.

Questions and proposed problems

15. Describe what are the main issues on connecting two intervals of linear interpolation movement? How do you achieve constant speed on the trajectory?
16. Explain why the user cannot control the animation by the abstract parameter u, on the parametric definition of the linear movement? What parameters support the user control of the animation?
17. Exchange the abstract variable u to time, in order to control the animation along a spline trajectory by the time.
18. Why the animation on the spline trajectory is controlled by difficulty? What are the main issues?
19. Give examples in real life for direct and inverse kinematics.
20. Exemplify and explain a morphism for animating the development of a leaf, movement cloud, and jumping ball.