Motions and Morphing

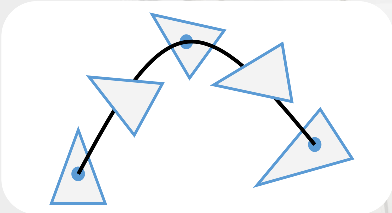
Lesson objectives

By the end of the module, you should be able to:

- Understand two aspects of animation
- Define functions to simulate the effects of acceleration, deceleration, and uniform speed
- Construct time-dependent functions for motions or animation
- Use linear interpolation to create morphing

1. Introduction

- Animation
  - Any time sequence of visual changes in a scene
  - Motion: Location or shape changes with time (objects, light sources, camera, etc.)
  - Updating: Static state or other parameters (pressure, temperature, zoom, focus, color, texture, transparency, etc.)
  - Morphing: Changing one shape into another



Animation

- Introduce time into the definitions of shape, position, orientation, etc.
- Two fundamental aspects:
  - Change of shape, position, orientation, etc.
  - Speed control
- Strategy for specifying **time-dependent** changes:
  - Define the shape, position, or orientation, etc., by functions of some parameter (say, τ)
  - Define function $\tau = f(t)$ to relate the change to time

Problems to be addressed

- How to define function τ = f(t) for three typical speed control?
- How to describe animation by time parameter or frame-index?
- How to make functions time-dependent?
- How to use linear interpolation to generate morphing?

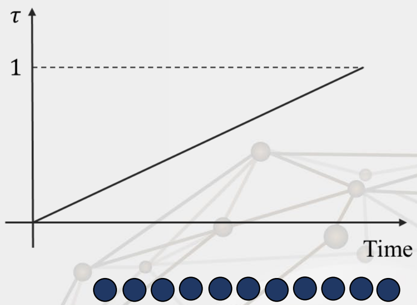
2. Speed specification

- How to simulate:
  - Uniform
  - Acceleration
  - Deceleration

  by defining appropriate function: τ = f(t)
- Two ways to describe animation:
  - Use time parameter
  - Use frame index

Simulating uniform speed

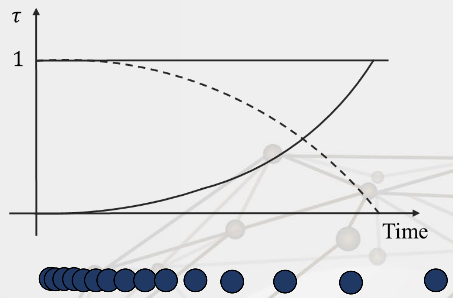
- Example:
  $P(\tau) = P_1 + (P_2 - P_1)\,\tau$,
  $0 \le \tau \le 1$



- Time parameter:
  $$\tau = f(t) = \frac{t - t_1}{t_2 - t_1},\quad t \in [t_1, t_2]$$
- Frame index:
  $$\tau = f(k) = \frac{k-1}{m-1}$$ where *k* is the frame index, 1 ≤ k ≤ m, *m* is the total number of frames

Simulating acceleration

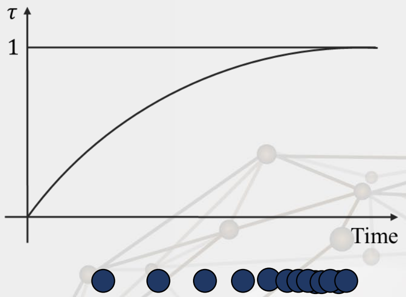
- Example:
  $P(\tau) = P_1 + (P_2 - P_1)\,\tau$,
  $0 \le \tau \le 1$



- Time parameter:
  $$\tau = f(t) = 1 - \cos\left(\frac{\pi}{2}\,\frac{t - t_1}{t_2 - t_1}\right),\quad t \in [t_1, t_2]$$
- Frame index:
  $$\tau = f(k) = 1 - \cos\left(\frac{\pi}{2}\,\frac{k-1}{m-1}\right)$$ where *k* is the frame index, 1 ≤ k ≤ m, *m* is the total number of frames

Simulating deceleration

- Example:
  $P(\tau) = P_1 + (P_2 - P_1)\,\tau,$
  $0 \le \tau \le 1$



- Time parameter:
  $$\tau = f(t) = \sin\left(\frac{\pi}{2}\frac{t-t_1}{t_2-t_1}\right),\quad t \in [t_1, t_2]$$
- Frame index:
  $$\tau = f(k) = \sin\left(\frac{\pi}{2}\frac{k-1}{m-1}\right)$$ where k is the frame index, $1 \le k \le m$, m is the total number of frames

Question for thinking

Q: How to simulate "back and forth" for $P(\tau) = P_1 + (P_2 - P_1)\,\tau,\ 0 \le \tau \le 1$?

3. Time-dependent functions

Basic idea:

- Make the functions for
  - Shape definition
  - Location
  - Transformation

  parameterized by τ.
- For example, a dynamic point is created from a static point by a time-dependent affine transformation.

$$\begin{bmatrix} x(\tau) \\ y(\tau) \\ z(\tau) \\ 1 \end{bmatrix} = \begin{bmatrix} a(\tau) & b(\tau) & c(\tau) & l(\tau) \\ d(\tau) & e(\tau) & f(\tau) & m(\tau) \\ g(\tau) & h(\tau) & k(\tau) & n(\tau) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},\ \tau \in [0,1]$$

Examples

Translation

$$\begin{bmatrix} x(\tau) \\ y(\tau) \\ z(\tau) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & l\tau \\ 0 & 1 & 0 & m\tau \\ 0 & 0 & 1 & n\tau \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling

$$\begin{bmatrix} x(\tau) \\ y(\tau) \\ z(\tau) \\ 1 \end{bmatrix} = \begin{bmatrix} A\tau & 0 & 0 & 0 \\ 0 & B\tau & 0 & 0 \\ 0 & 0 & C\tau & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation about the x-axis

$$\begin{bmatrix} x(\tau) \\ y(\tau) \\ z(\tau) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha\tau) & -\sin(\alpha\tau) & 0 \\ 0 & \sin(\alpha\tau) & \cos(\alpha\tau) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$ where $0 \le \tau \le 1$

"Motion by path" approach

- Motion by path is an approach to defining a motion or animation by explicitly specifying the motion path.
- Method:
  - Step 1: Represent the path by parametric equations: $(x, y) = (x(\tau), y(\tau))$, $\tau \in [0, 1]$.
  - Step 2: By linking the path to the target animation, derive the representation of the motion object.
  - Step 3: Appropriately define $\tau = f(t)$, $t \in [t_1, t_2]$ or $\tau = f(k)$, $k = 1,\ldots,m$ to control the speed.

4. Linear interpolation

Problem: Given two values *A* and *B*, we want to compute an intermediate value $v(\tau)$ which gradually changes from *A* to *B* at a constant rate.

- Linear interpolation model: $v(\tau) = (1 - \tau)A + \tau B$, $0 \le \tau \le 1$.
  - When $\tau = 0$, $v(\tau) = A$
  - When $\tau = 1$, $v(\tau) = B$
  - For intermediate values of parameter τ, $v(\tau)$ is linear combination of A and B
- Linear interpolation can be applied to many quantities.

Example 1: Constructing time-dependent transformation

Given 2 matrices M_1 and M_2, one way to construct a time-dependent matrix is to use linear interpolation: $M(\tau) = (1 - \tau)M_1 + \tau M_2$, $0 \le \tau \le 1$.

For example, if $M_1 = \begin{bmatrix} a_1 & b_1 & c_1 & l_1 \\ d_1 & e_1 & f_1 & m_1 \\ g_1 & h_1 & k_1 & n_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $M_2 = \begin{bmatrix} a_2 & b_2 & c_2 & l_2 \\ d_2 & e_2 & f_2 & m_2 \\ g_2 & h_2 & k_2 & n_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

then $M(\tau) = \begin{bmatrix} a(\tau) & b(\tau) & c(\tau) & l(\tau) \\ d(\tau) & e(\tau) & f(\tau) & m(\tau) \\ g(\tau) & h(\tau) & k(\tau) & n(\tau) \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

where: $a(\tau) = (1 - \tau)a_1 + \tau a_2$, $b(\tau) = (1 - \tau)b_1 + \tau b_2$, etc., with $\tau \in [0, 1]$.

Example 2: Morphing of implicitly-defined objects

Given:

Object A is a sphere with radius 5:
A: $f_A(x, y, z) = 5^2 - x^2 - y^2 - z^2 \ge 0$.

Object B is a cylinder with radius 5 and height 10:
B: $f_B(x, y, z) = \min(z + 5, 5 - z, 5^2 - x^2 - y^2) \ge 0$.

Propose a morphing from object A at time t = 2 to object B at time t = 10 with uniform speed.

Answer:

$C = A \rightarrow B$
C: $f_C(x, y, z) = f_A(x, y, z)(1 - \tau) + f_B(x, y, z)\tau \ge 0$
with $\tau = \frac{t-2}{10-2}$, $t \epsilon [2, 10]$.

Example 3: Morphing of parametric surfaces

Input:

Surface A: $x = h_1(u, v)$, $y = h_2(u, v)$, $z = h_3(u, v)$

$u \in [u_0, u_1]$, $v \in [v_0, v_1]$

Surface B: $x = g_1(r, s)$, $y = g_2(r, s)$, $z = g_3(r, s)$

$r \in [r_0, r_1]$, $s \in [s_0, s_1]$

Method:

Step 1: Re-parameterization

$(u - u_0)/(u_1 - u_0) = (r - r_0)/(r_1 - r_0) \Rightarrow u = u(r)$

$(v - v_0)/(v_1 - v_0) = (s - s_0)/(s_1 - s_0) \Rightarrow v = v(s)$

$\Rightarrow$ A: $\mathbf{x = h_1(u(r), v(s))}$, $\mathbf{y = h_2(u(r), v(s))}$, $\mathbf{z = h_3(u(r), v(s))}$

5. Summary

- Simulation of three types of speed
  - Uniform speed
  - Acceleration
  - Deceleration
- Time dependent functions for motions
  - "Motion by path"
- Linear interpolation for morphing
- Formulation of animation in terms of
  - Time parameter t
  - Frame index k