23 - Computer animation

Video

Frames Per Second: 30 ~~FPS~~ · Broadcast TV -
- 30 FPS - NTSC
- 25 FPS - PAL

Movies
- 24 FPS - standard
- 48-120 FPS - High frame rate

Computer monitors
- ~60 Hz or more (Hz = 1/s, 1 Hz = 1 FPS)

Video games
- 60-120 FPS

Video resolution:
- 4:3 aspect ratio
  - VGA 640x480 (1987) SD
  - XGA 1024x768 (1990)
- 16:9 aspect ratio
  - 480p 854x480
  - 720p 1280x720 HD
  - 1080p 1920x1080 HD
  - 4k 3840x2160 UltraHD
  - 8K 7680x4320 UltraHD

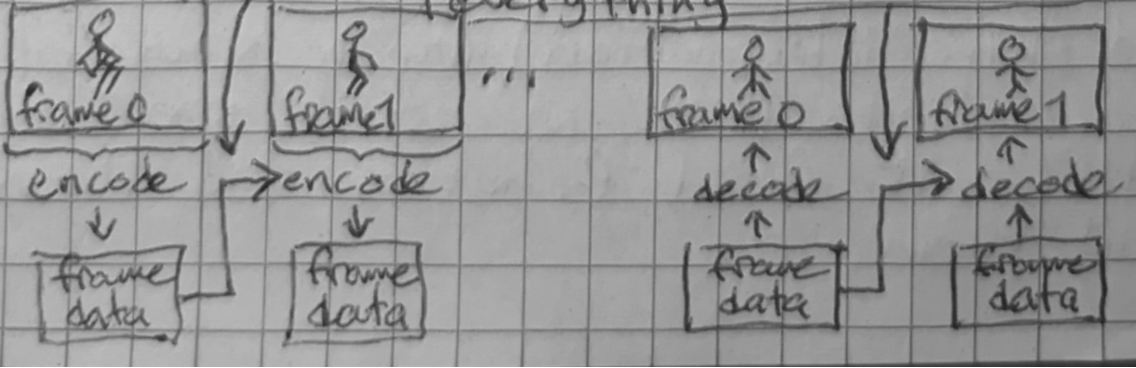
Video data: If RGB = 3 bytes per pixel, then

720p resolution	1080p resolution
1280x720 = 921600 pixels	1920x1080 = ~2,1 M pixels
~2,8 Mb per frame	~6,2 Mb per frame
30 fps	30 fps
~83 Mb per sec	~186 Mb per sec
~664 Mbits per sec	~1,5 Gbit per sec

Massive!
So we need to compress

Video compression: Next frame encoding takes takes previous into account, so we don't have to recalculate everything

There is hardware to encode/decode



Computer animation

What is animation? can be specified as procedural animation, keyframing, motion capture or physics-based animation.

Transformation, i.e. rotating, scaling and translating, or deformation, i.e. more complex, probably what we think about animation.

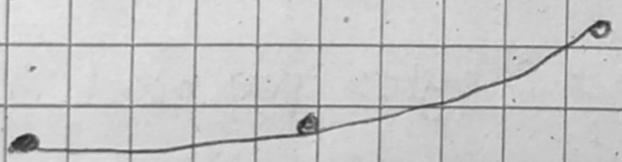
Procedual animation:

Animation based on some function. Can be a transformation function like rotating, or deformation function like rocking. Can also be more complex like flocking, where entities move based on some function.

Keyframing:

Back in the days when handdrawing, the lead artist drew keyframes, then other artist drew tweens (in between frames). This is applied to computer animation as well, where tweens are auto- or semi-auto generated.

Keyframes



X_n = shape at n
t_n = time at n

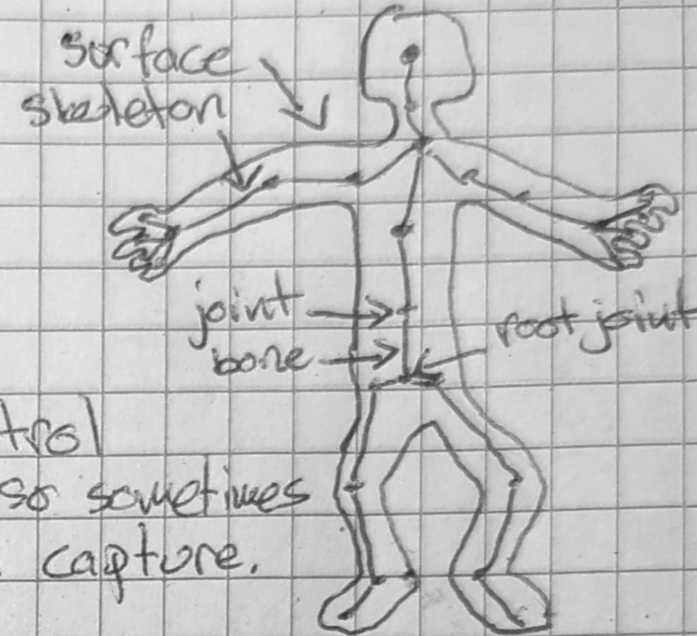
$$\underbrace{\begin{matrix} X_0 \\ t_0 \end{matrix}}_{\text{keyframe0}} \Longrightarrow \begin{matrix} X_1 \\ t_1 \end{matrix} \Longrightarrow \begin{matrix} X_2 \\ t_2 \end{matrix} \Longrightarrow \begin{matrix} X_3 \\ t_3 \end{matrix}$$

we then use keyframe interpolation, like linear or spline:

$$X(t) = \text{catmull-Rom\_spline}(X_0, X_1, X_2, X_3, t_0, t_1, t_2, t_3)$$

We mostly use tools for specifying keyframes, like <u>morphing</u> (blend shape interpolation), which is useful for facial animations. Here an artist specifies sliders for different facial formations, and then we can blend them.

We also have <u>skeletal animation</u>. Here we use a process called rigging, which creates a skeleton for a model. We can then use, for example, <u>inverse kinematics</u>, where mathematics figures out where joint positions are supposed to be when moving a hand for example, or <u>forward kinematics</u>, where we specify the joint rotations.
<u>Skinning</u> is the process of attatching a surface to a skeleton.



For each vertex, we can specify ~~a~~ <u>weights</u>, describing how much a joint impacts that vertex.

These can be a lot of control points on complex models, so sometimes it's easier to use motion capture.

<u>Motion capture:</u>
Using a soit with markers, we can estimate the ~~joints~~ bone motion. Algorithms are used ~~for~~ estimating. ~~Wan~~ can also be used for facial animations.

Lastly we also have physics-based animation/simulation, which is the topic for the next lecture.

Also noted that often times several of the animation techniques are used simultaniousely