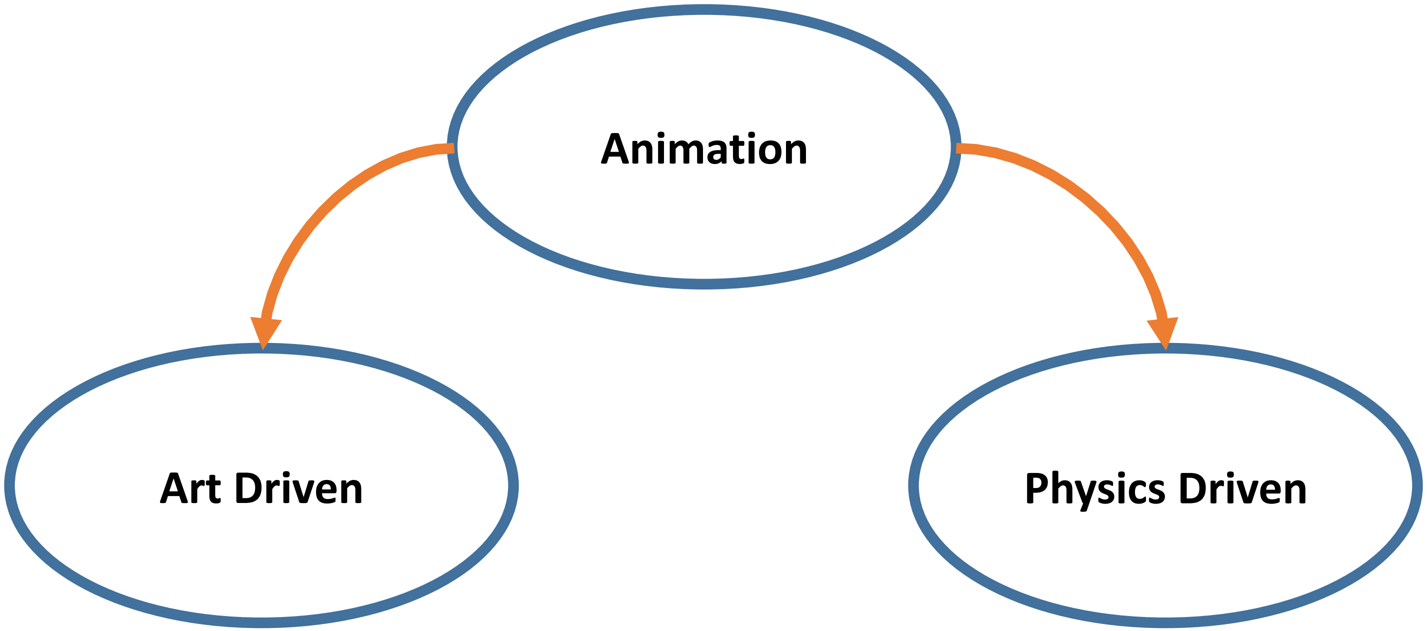
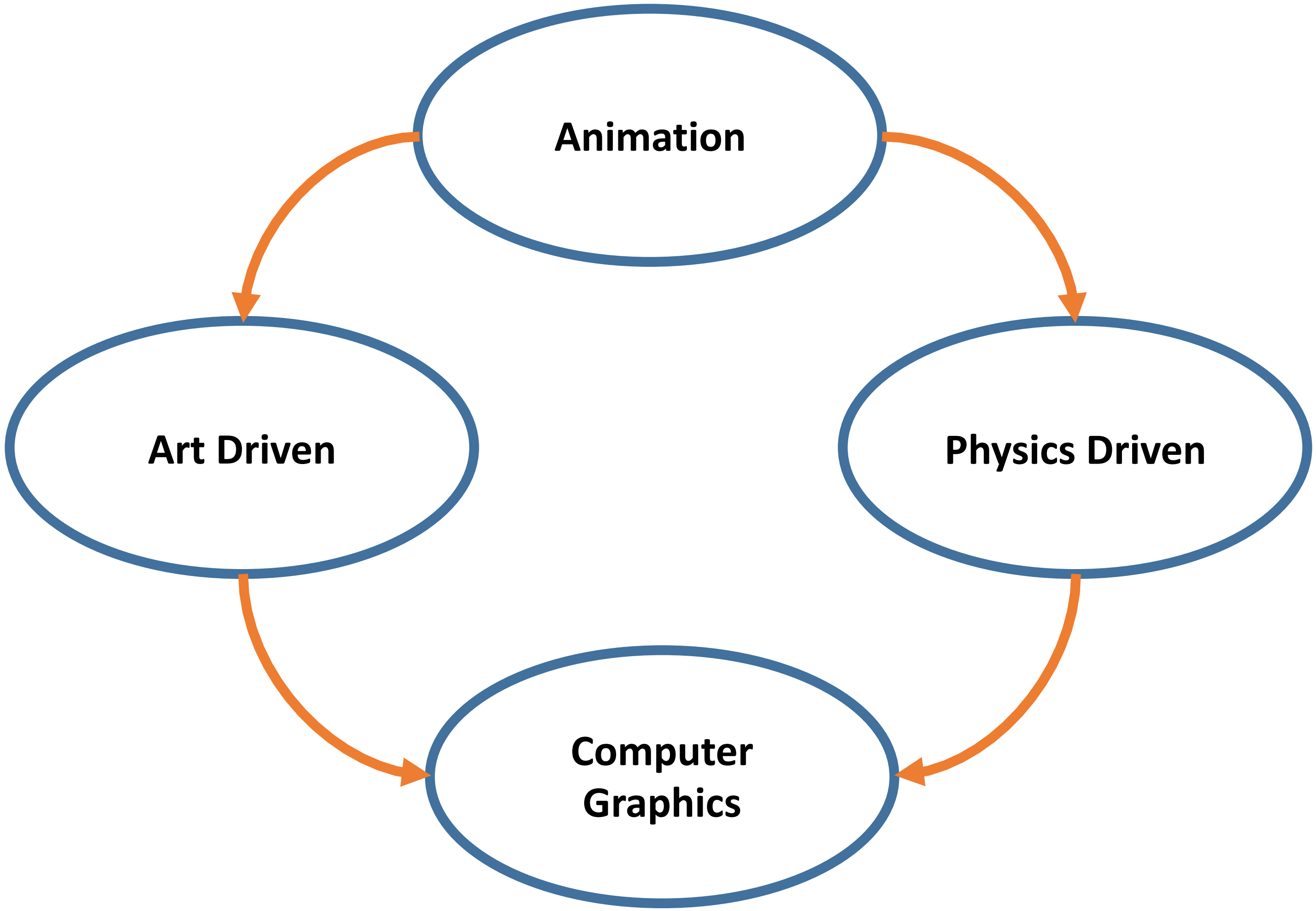
# CS 148: Summer 2016
# Introduction of Graphics and Imaging

Zahid Hossain

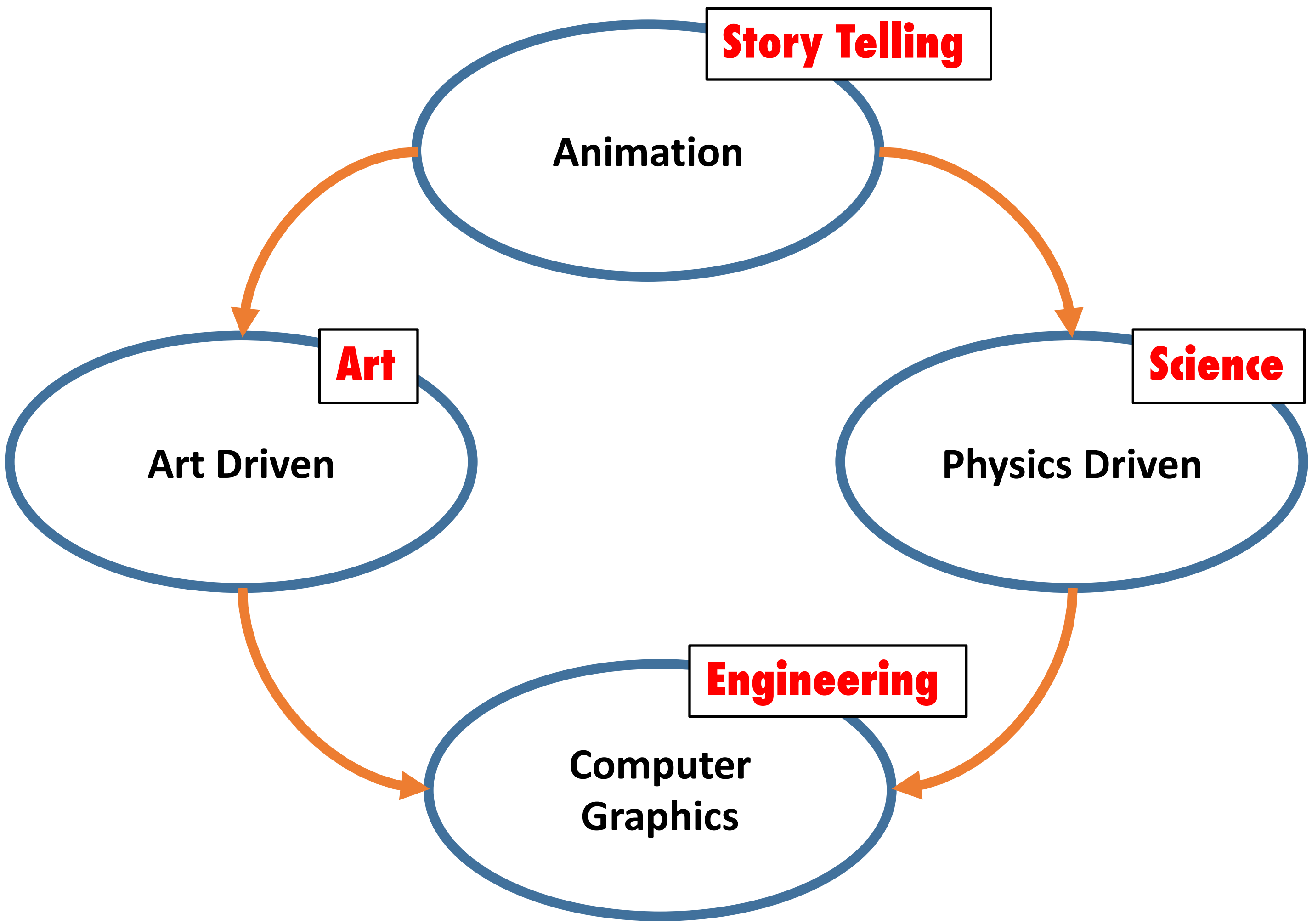
# Animation



# Animation

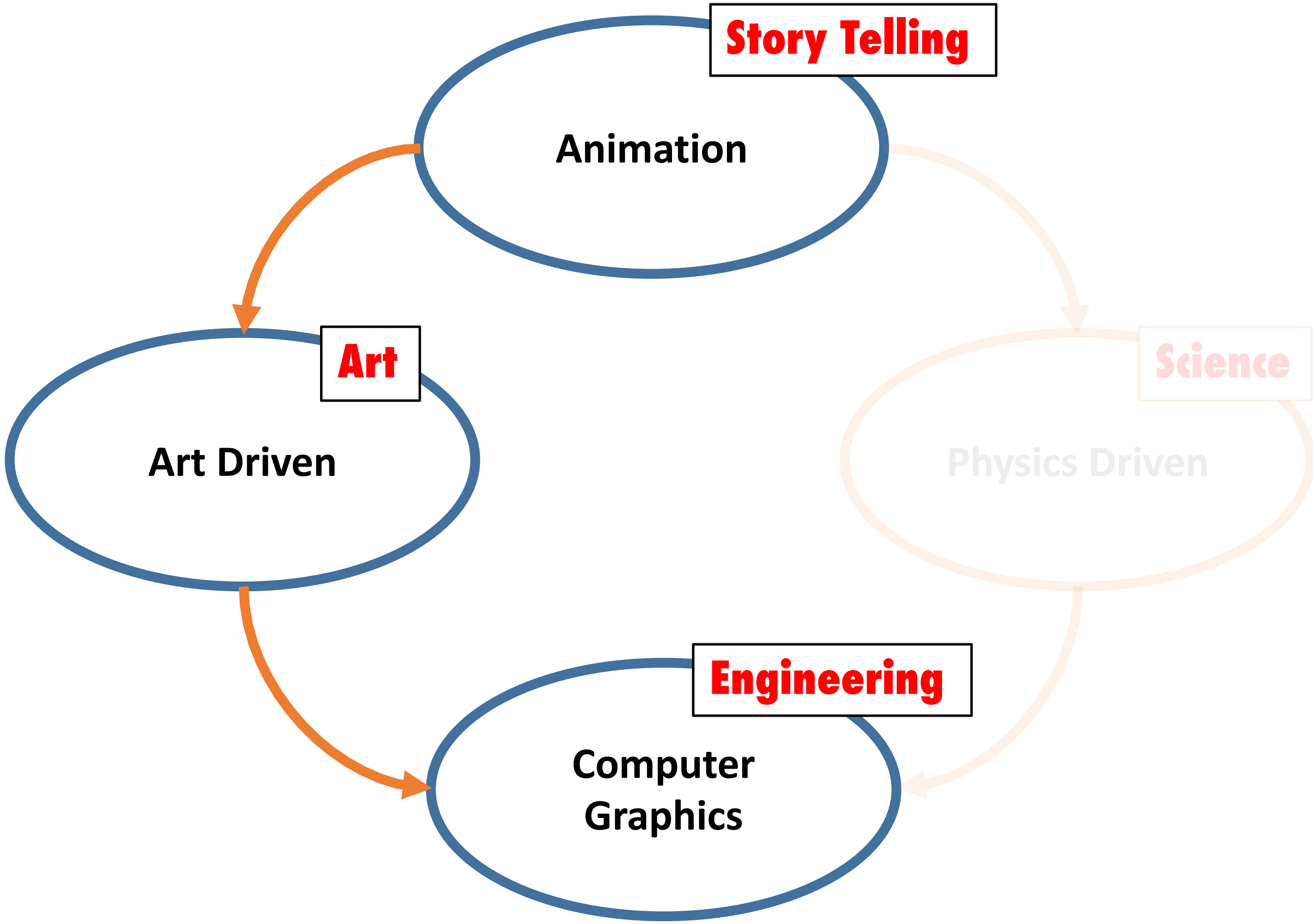


# Animation

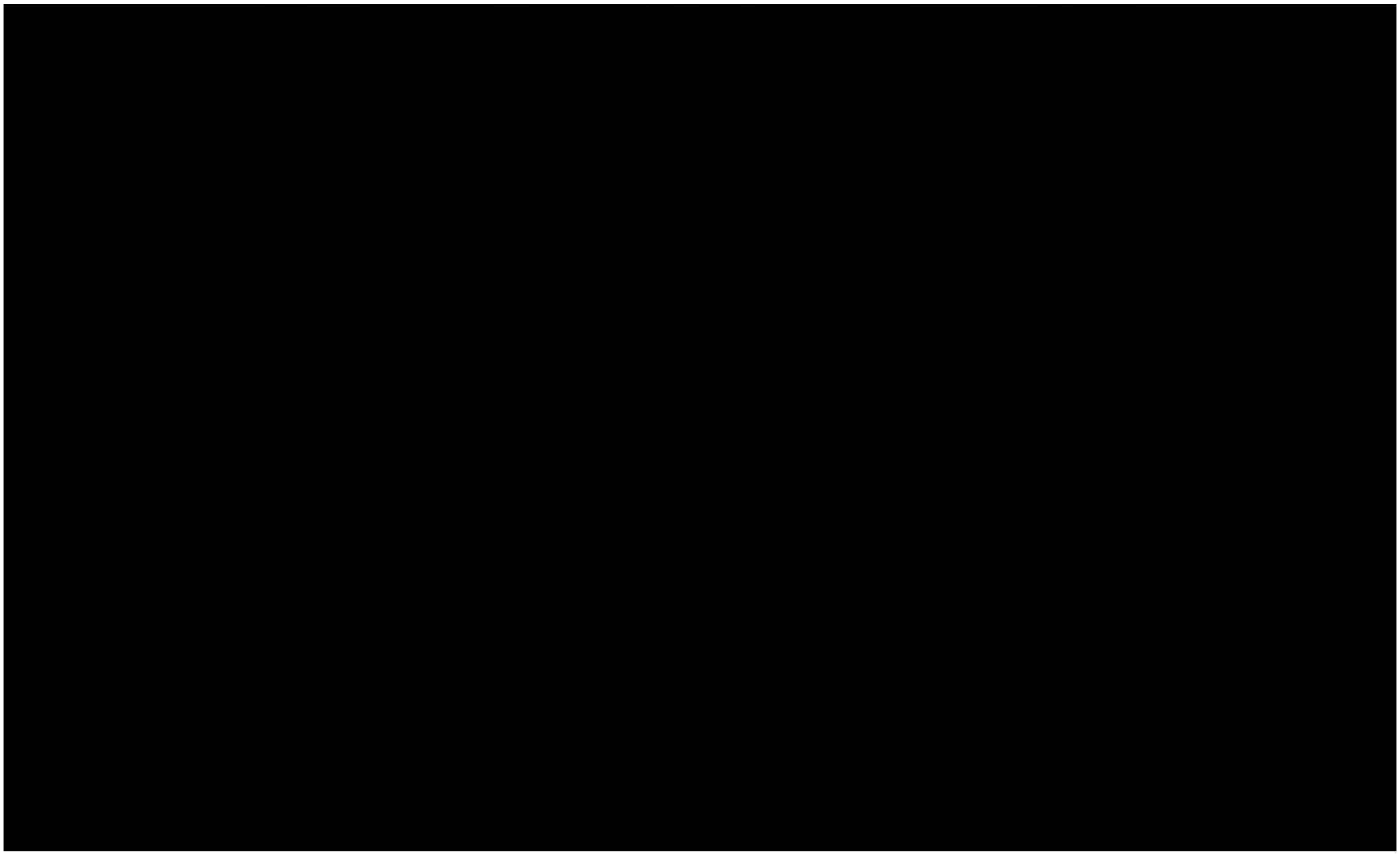


Story Telling
Animation
Art
Art Driven
Science
Physics Driven
Engineering
Computer
Graphics

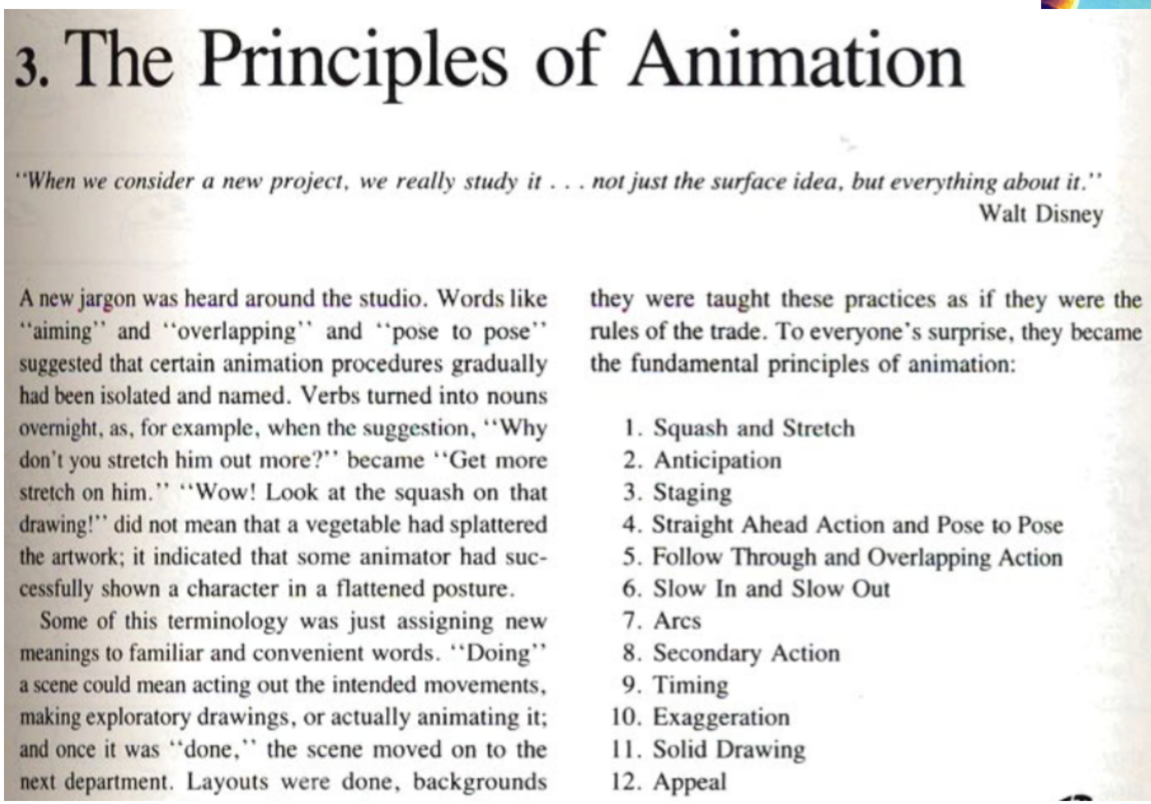
# Animation


Story Telling
Animation
Art
Art Driven
Science
Physics Driven
Engineering
Computer
Graphics

# Luxo Jr. (Pixar Animation 1986)



# Principles of Animation



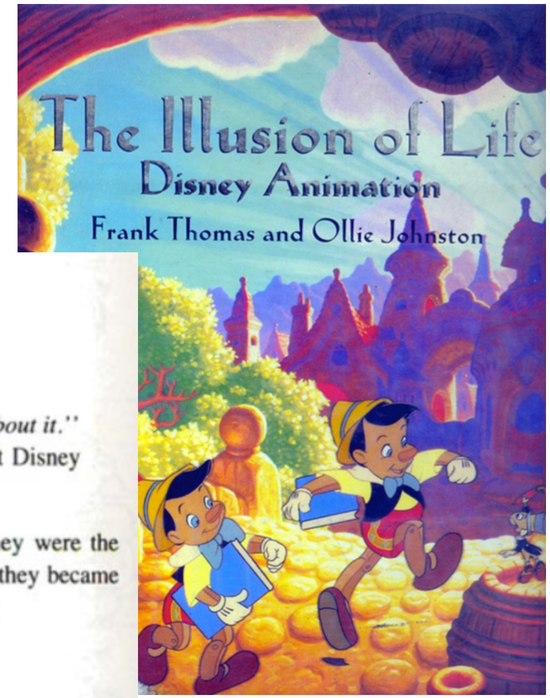
# 3. The Principles of Animation

*"When we consider a new project, we really study it . . . not just the surface idea, but everything about it."*
Walt Disney

A new jargon was heard around the studio. Words like "aiming" and "overlapping" and "pose to pose" suggested that certain animation procedures gradually had been isolated and named. Verbs turned into nouns overnight, as, for example, when the suggestion, "Why don't you stretch him out more?" became "Get more stretch on him." "Wow! Look at the squash on that drawing!" did not mean that a vegetable had splattered the artwork; it indicated that some animator had successfully shown a character in a flattened posture.

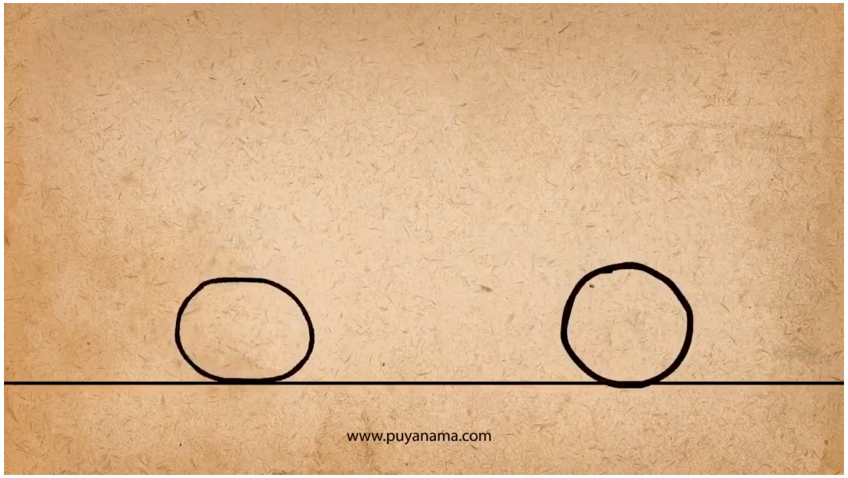
Some of this terminology was just assigning new meanings to familiar and convenient words. "Doing" a scene could mean acting out the intended movements, making exploratory drawings, or actually animating it; and once it was "done," the scene moved on to the next department. Layouts were done, backgrounds they were taught these practices as if they were the rules of the trade. To everyone's surprise, they became the fundamental principles of animation:

1. Squash and Stretch
2. Anticipation
3. Staging
4. Straight Ahead Action and Pose to Pose
5. Follow Through and Overlapping Action
6. Slow In and Slow Out
7. Arcs
8. Secondary Action
9. Timing
10. Exaggeration
11. Solid Drawing
12. Appeal

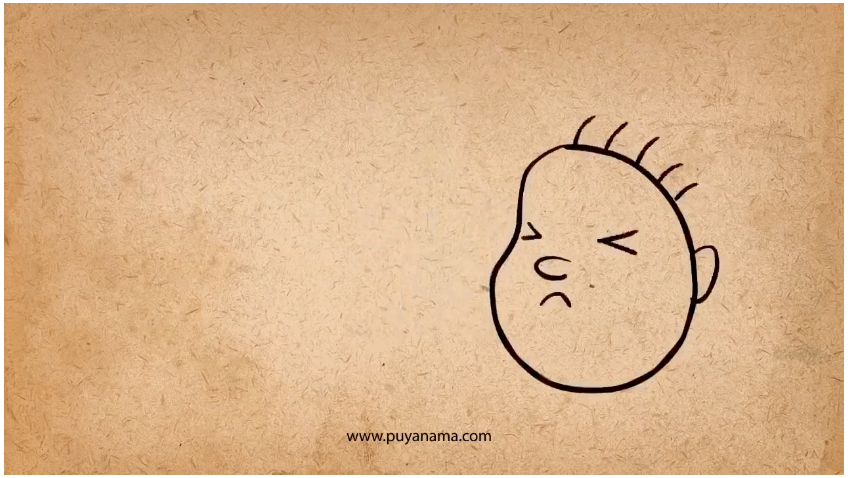



1981

# Squash and Stretch



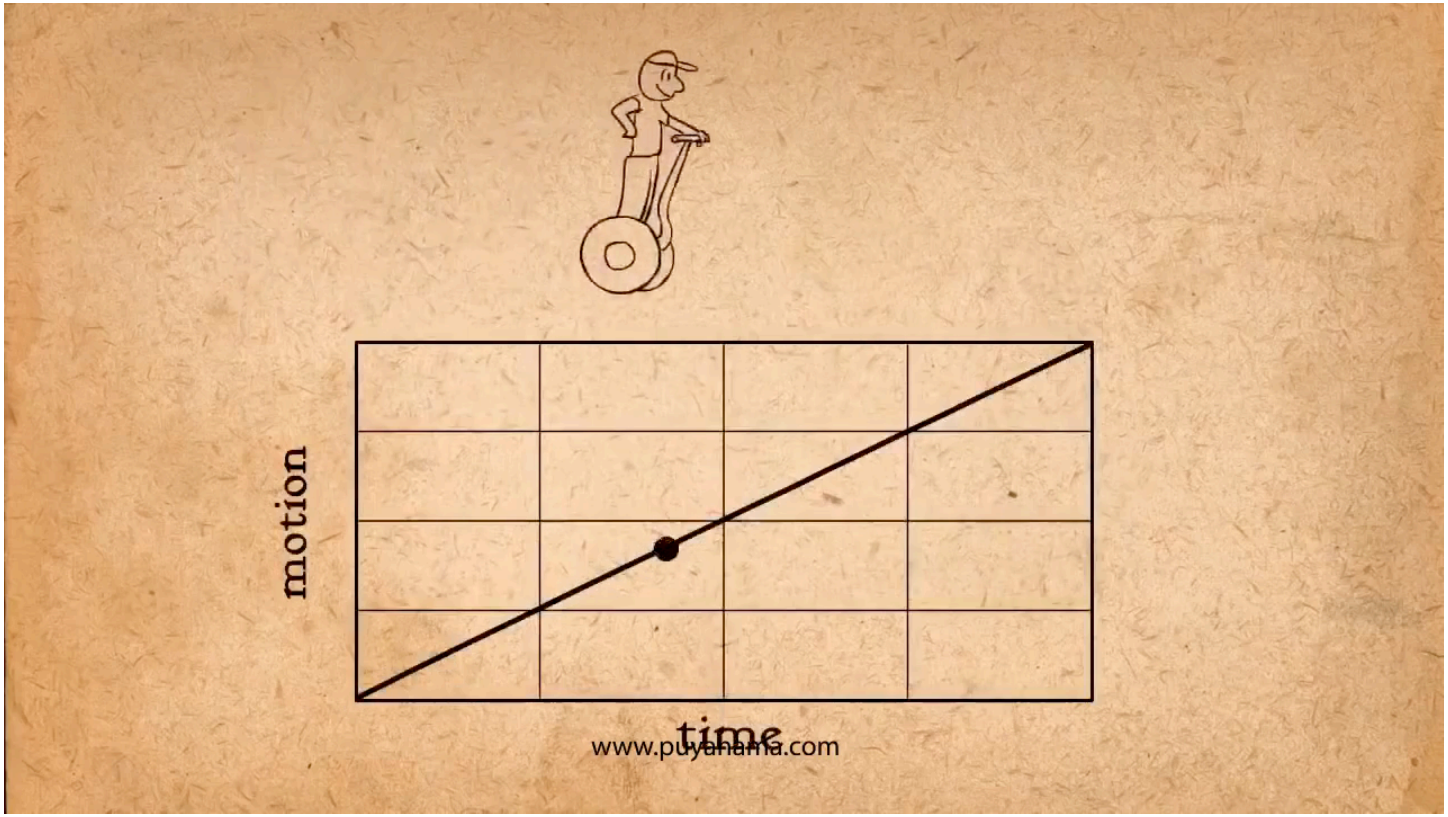

**Rigidity/Flexibility**



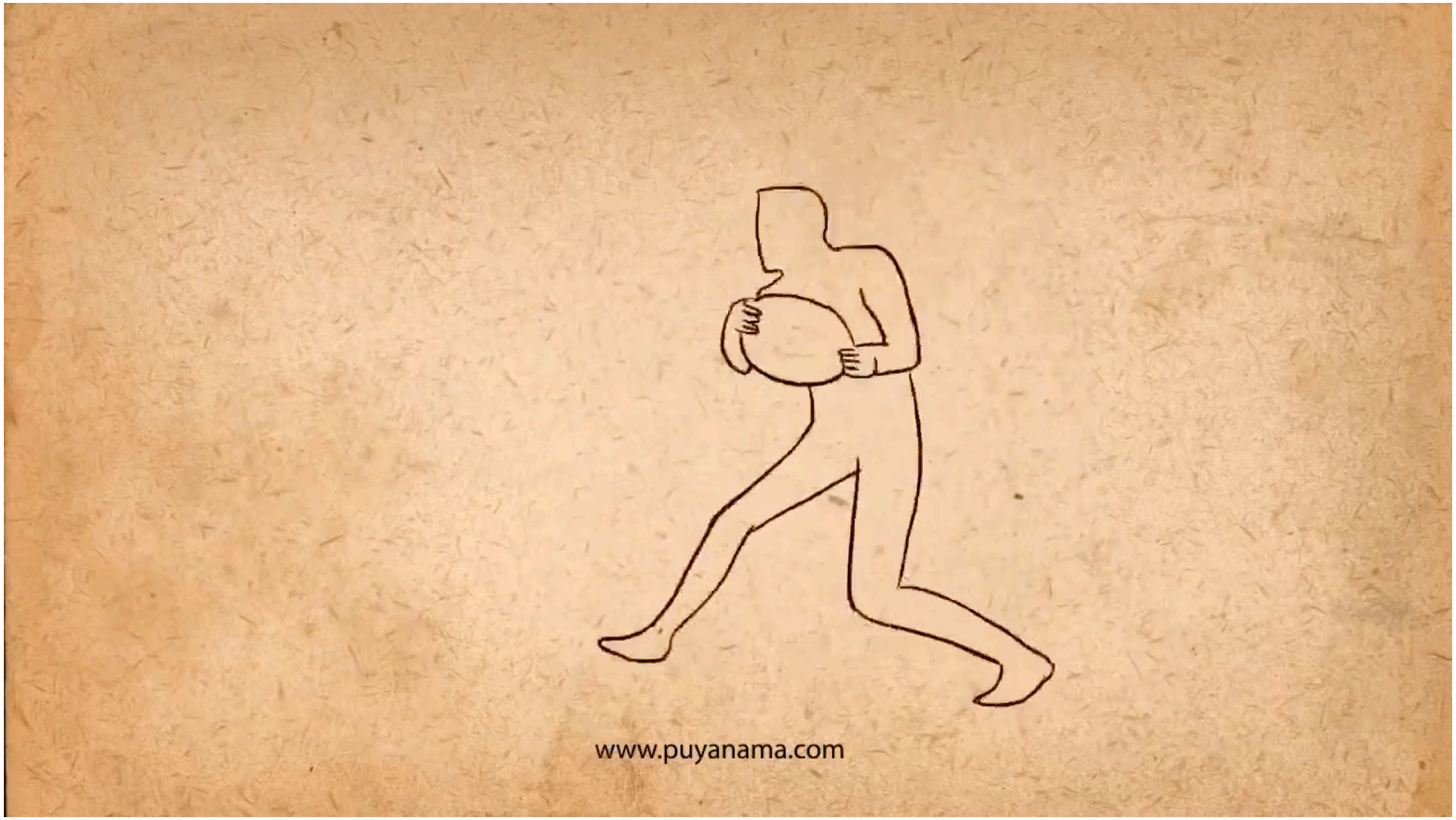

**Lively Expressions**

# Anticipation

# Slow In And Slow Out



# Arcs

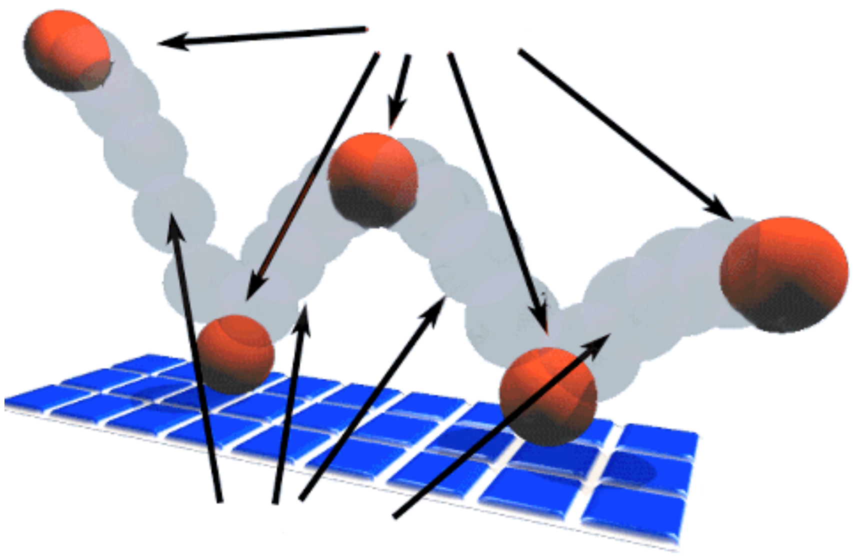


# Principles of Animation


**https://www.youtube.com/watch?v=5l2Aem7Ll3A**

# Keyframe Animation

# Keyframes



- Specify significant poses
- Automatically fill in motion between these points in time.

http://graphics.stanford.edu/courses/cs148-10-summer/docs/10_anim_interact.pdf

# Not a New Idea

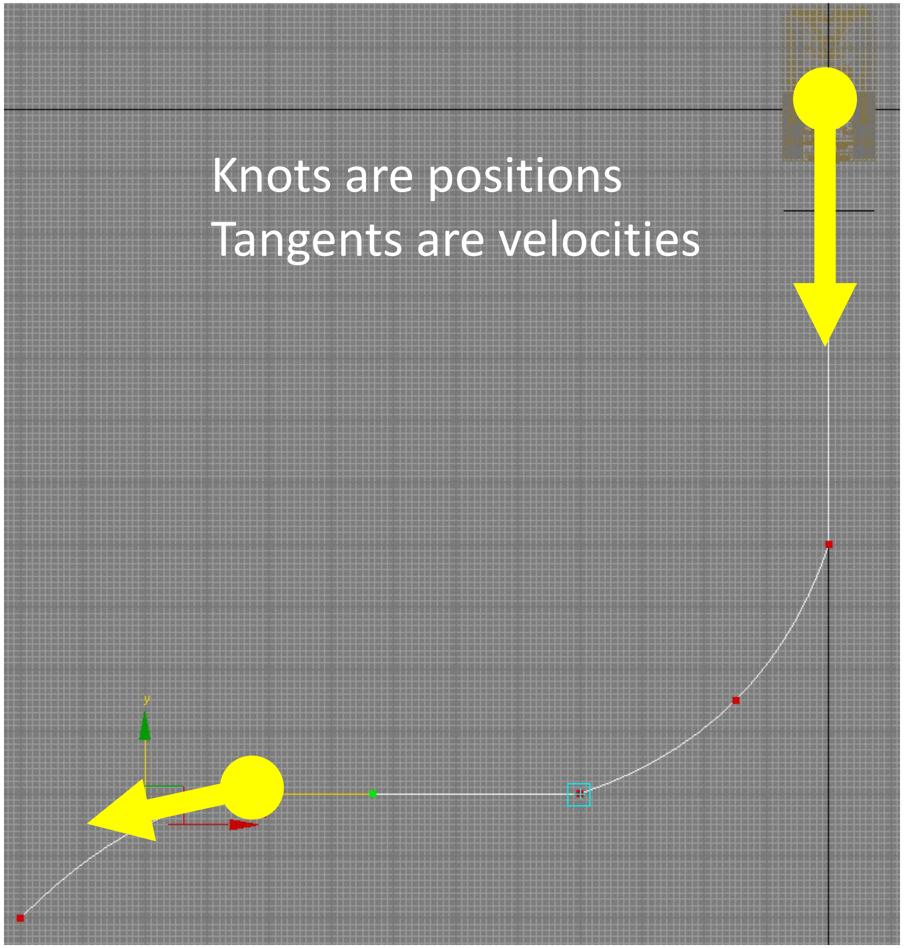
**Tweening: Not a fun job!**

# CAPS

**Computer-Aided Production System: 1980**

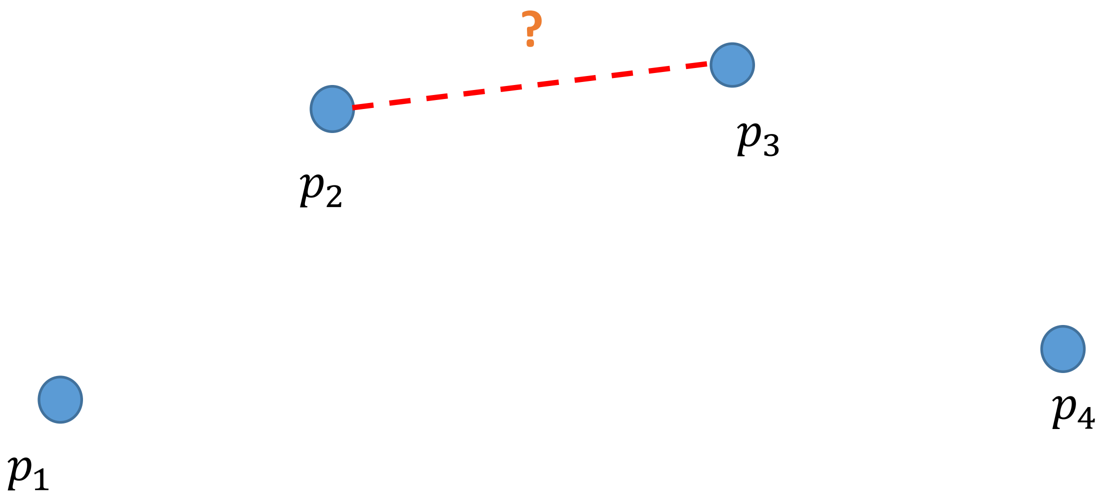
**~~Tweening :~~ Computer Does Interpolation**

# Animation Curves

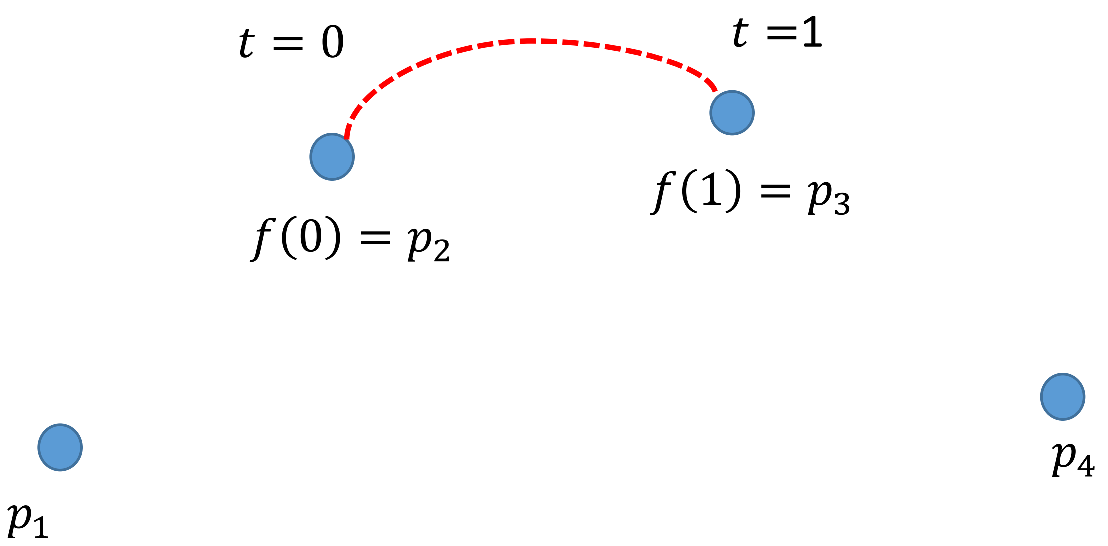



**Curves specify paths that objects take over time**

# Catmull-Rom Spline

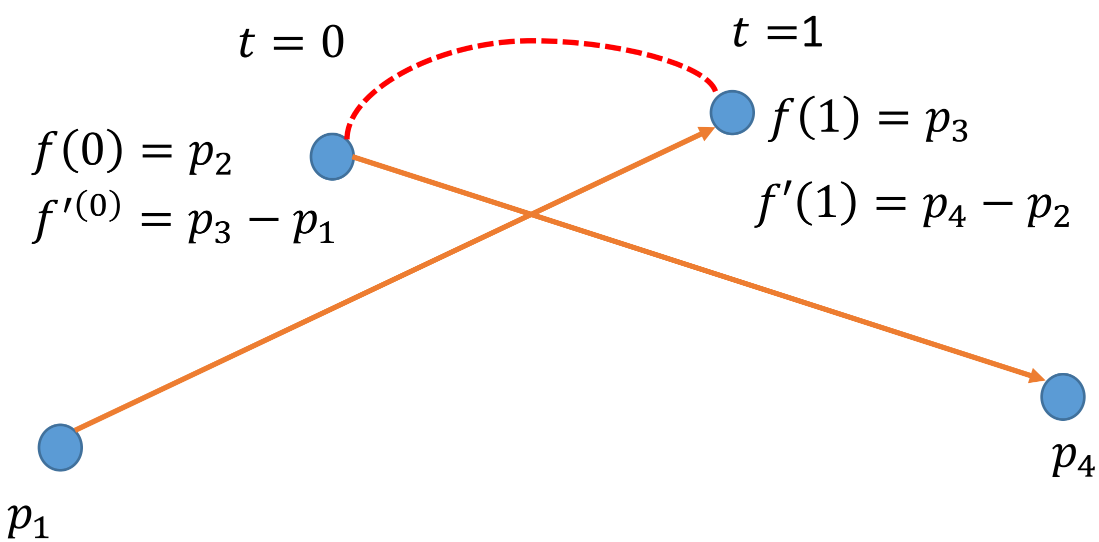


# Catmull-Rom Spline



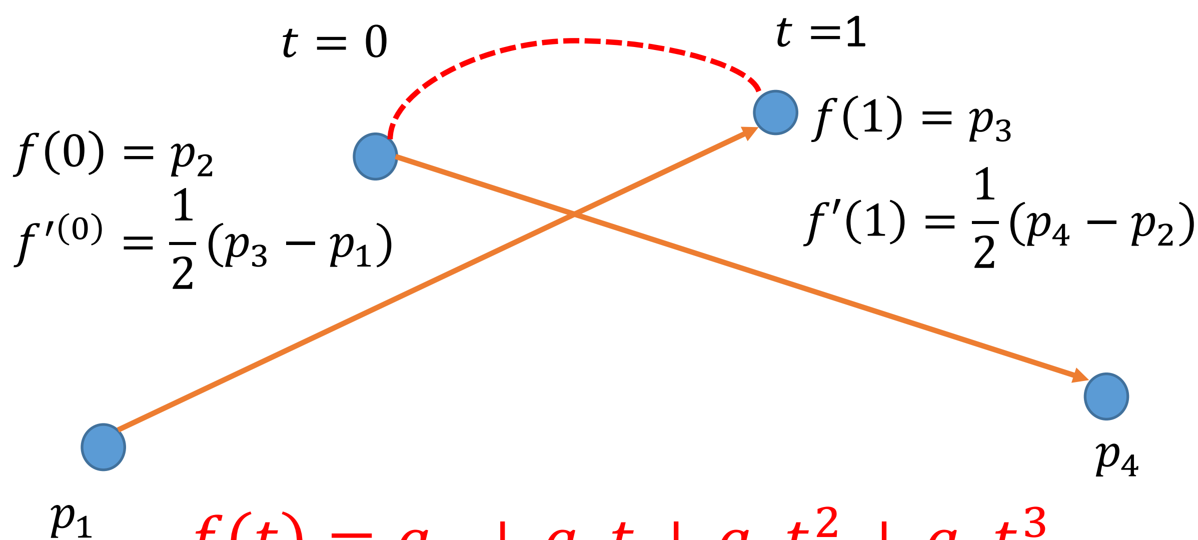

$$f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

# Catmull-Rom Spline




$$f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

# Catmull-Rom Spline




$$f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$f(0) = a_0 = p_2$$

$$f'(0) = a_1 = \frac{1}{2}(p_3 - p_1)$$

$$f(1) = a_0 + a_1 + a_2 + a_3 = p_3$$

$$f'(1) = a_1 + 2a_2 + 3a_3 = \frac{1}{2}(p_4 - p_2)$$

# Catmull-Rom Spline

Writing the control points $p_i$ in terms of the parameters $a_j$

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.5 & 0 & 0.5 & 0 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 1.5 & -1.5 & 0.5 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

# Catmull-Rom Spline

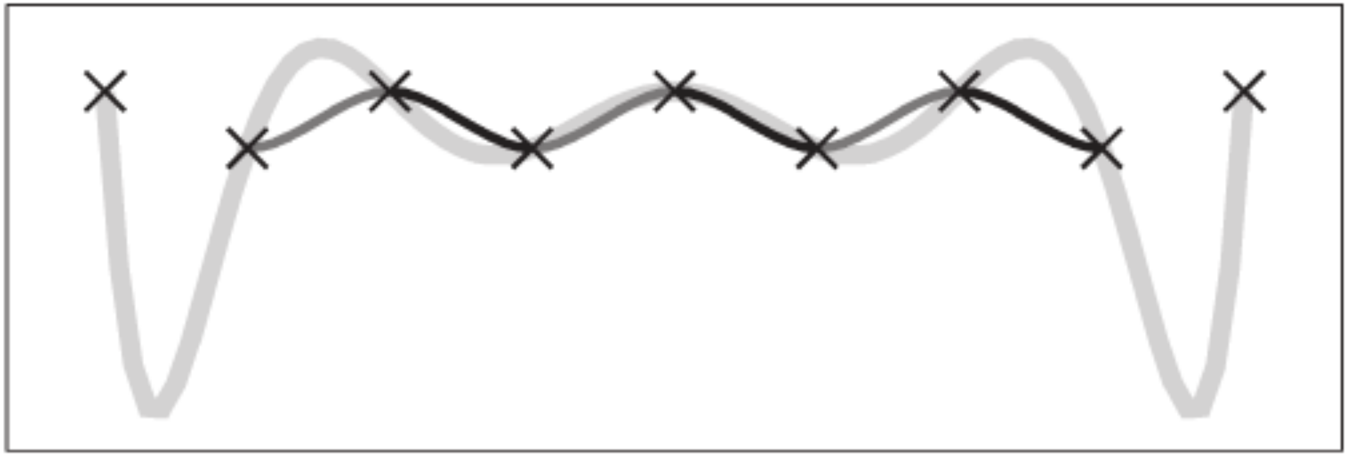
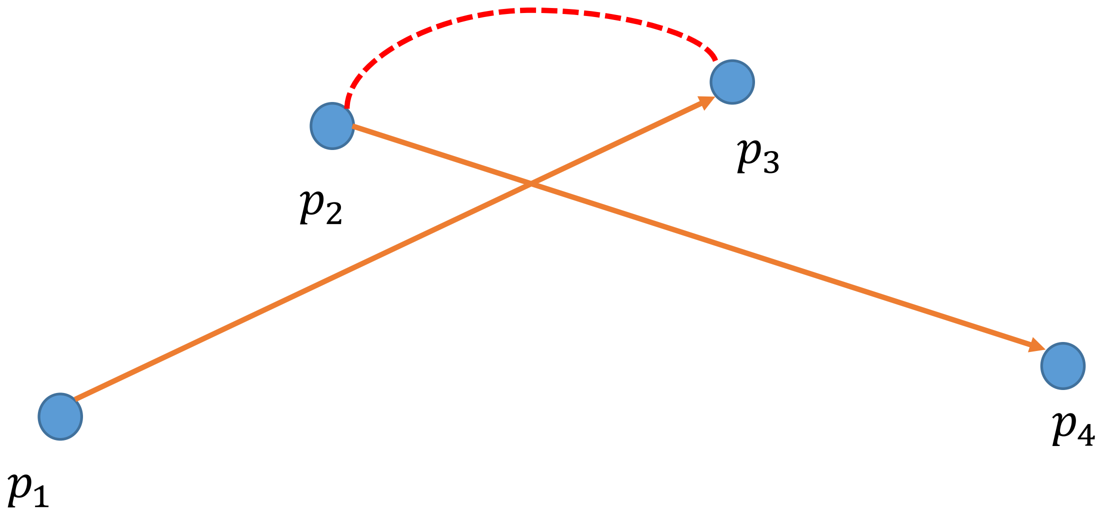


**Figure 15.9.** Splines interpolating nine control points (marked with small crosses). The thick gray line shows an interpolating polynomial. The thin, dark line shows a Catmull-Rom spline. The latter is made of seven cubic segments, which are each shown in alternating gray tones.

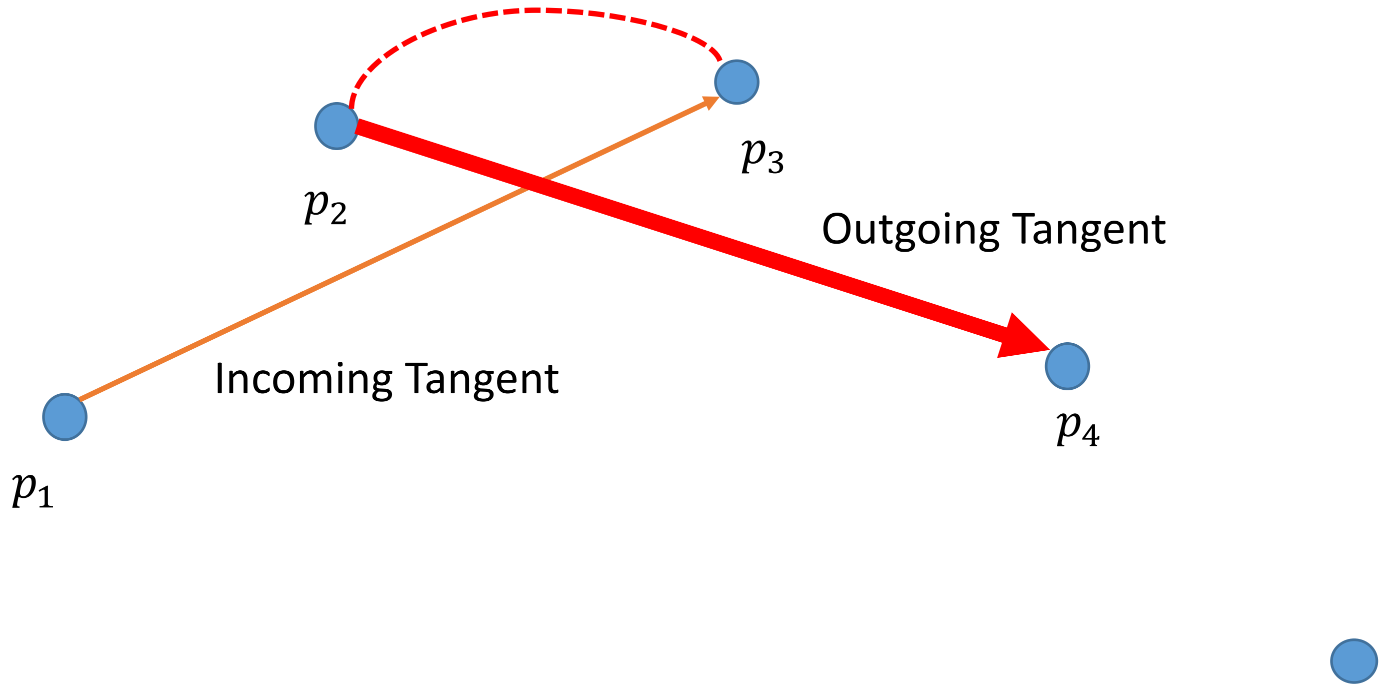
**$C^1$ Continuity**

Fundamentals of Computer Graphics - Shirley Page: 365

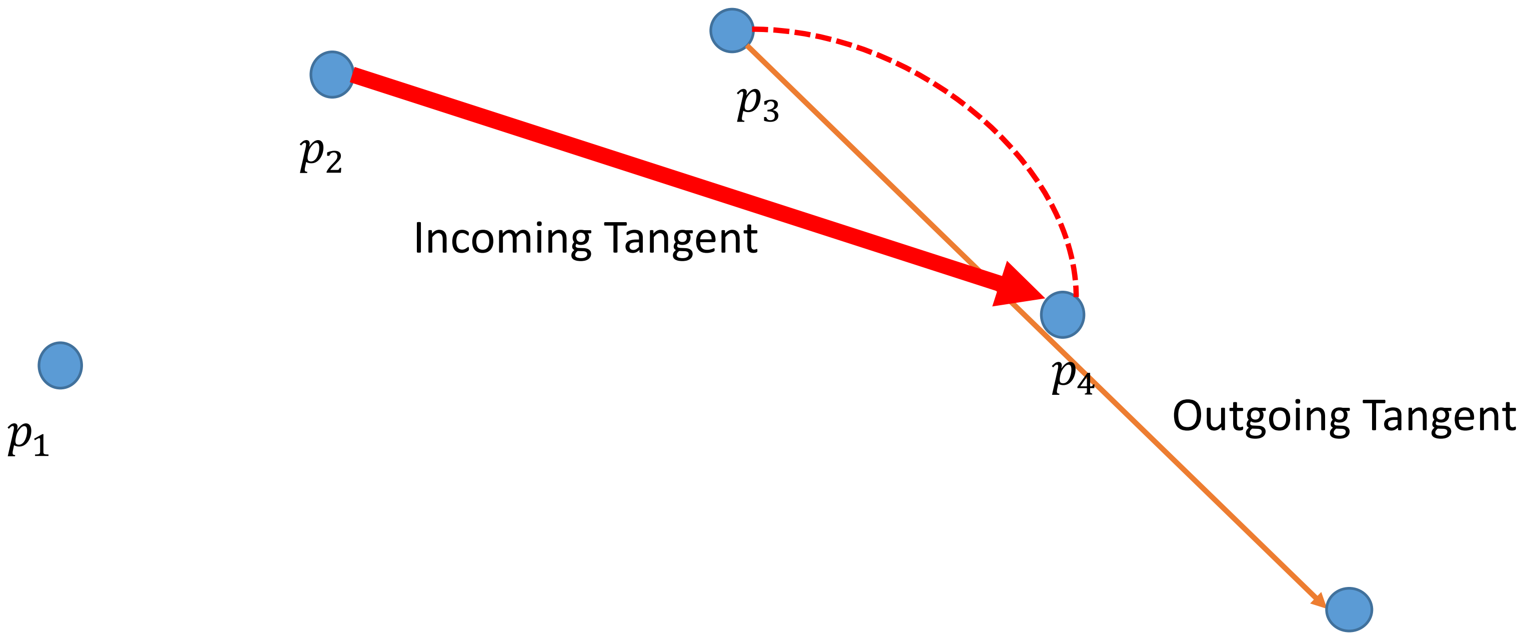
# Catmull-Rom Spline: Continuity



# Catmull-Rom Spline



# Catmull-Rom Spline

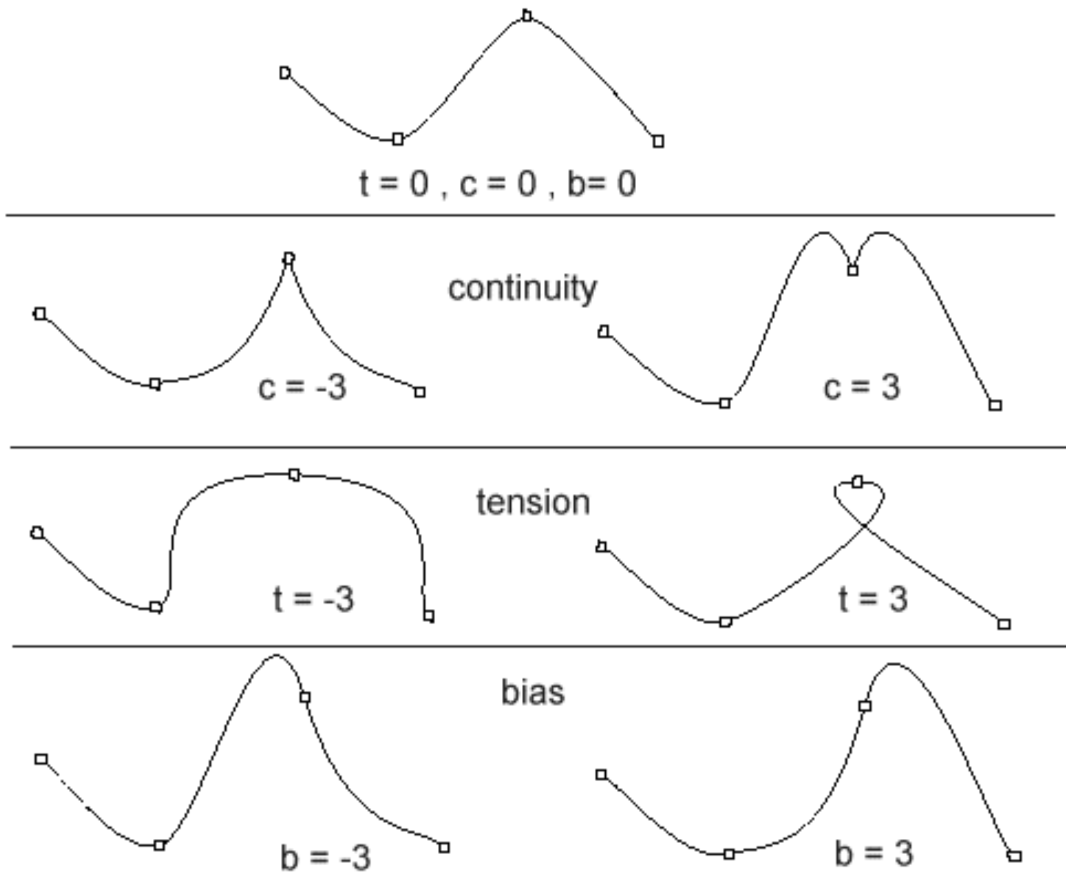



**"Outgoing Tangent" becomes "Incoming Tagent" for the next Segment**

# Framework for Animation Curves

- **Tension**
  - Sharpness near keyframes
- **Continuity**
  - Different in/out tangents
- **Bias**
  - Overshoot or undershoot

# Framework for Animation Curves




Fundamentals of Computer Graphics - Shirley Page:420

# Bezier Curve

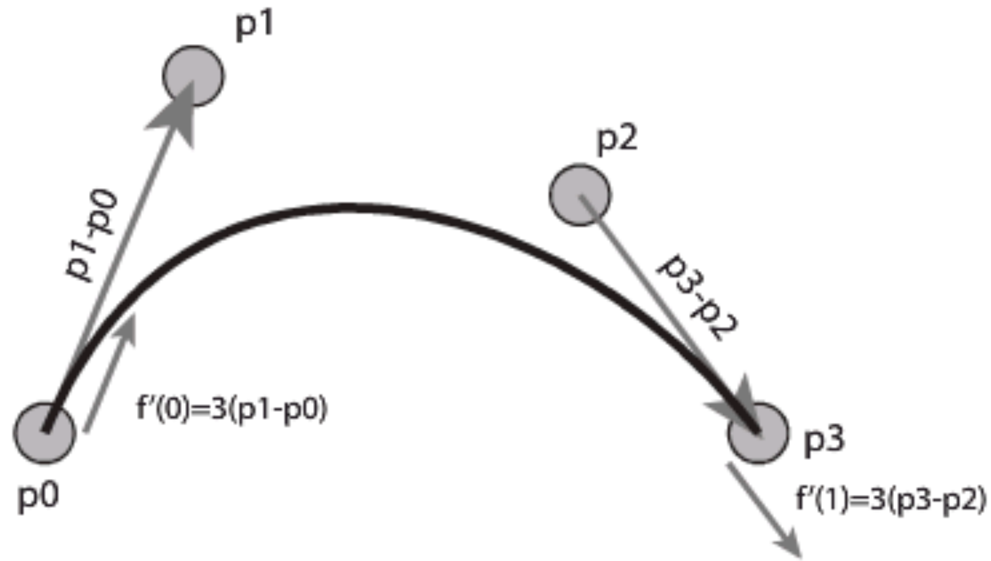
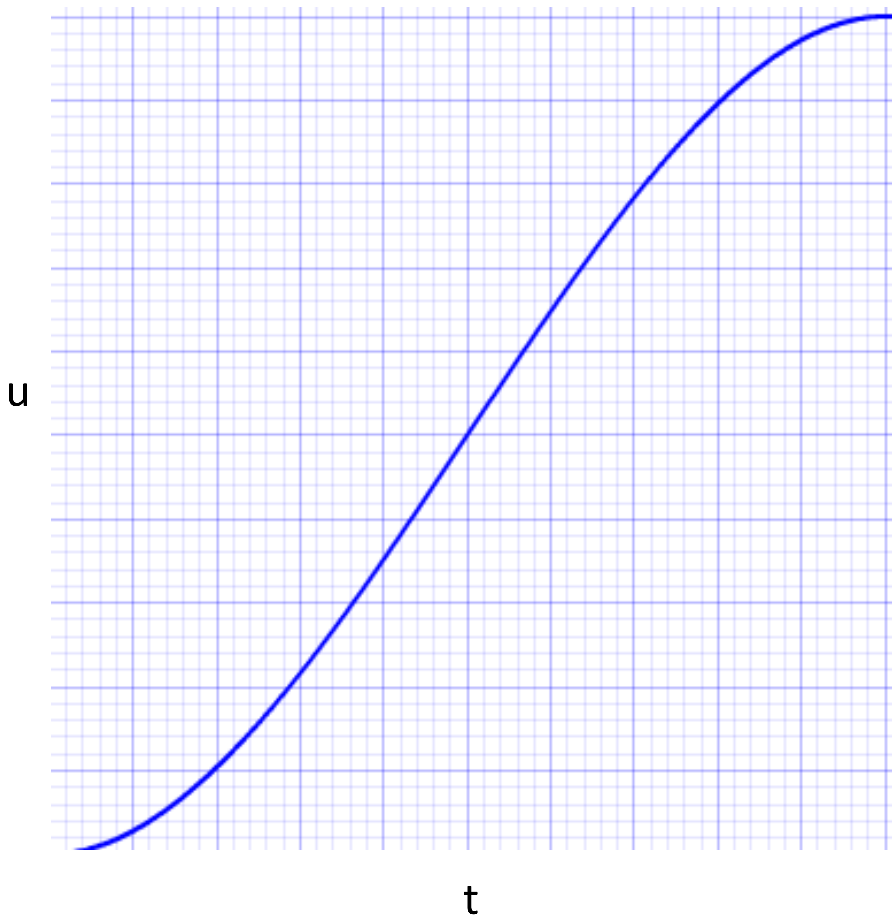



**Figure 15.10.** A cubic Bézier curve is controlled by four points. It interpolates the first and last, and the beginning and final derivatives are three times the vectors between the first two (or last two) points.

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

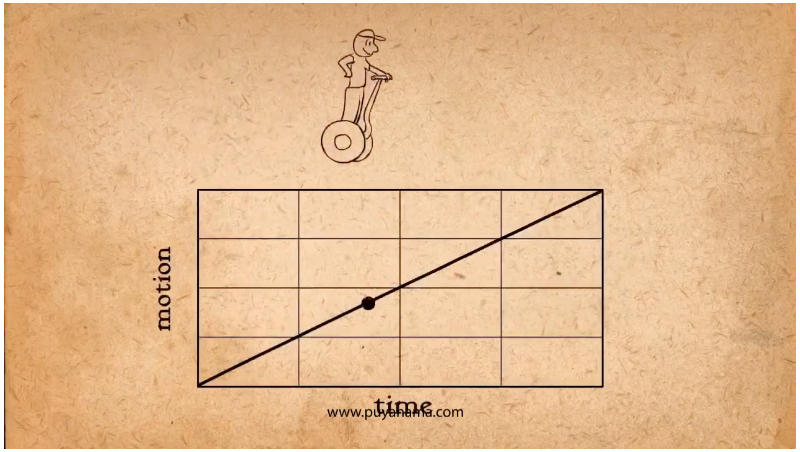
Fundamentals of Computer Graphics - Shirley Page:365

# Slow-In-Slow-Out

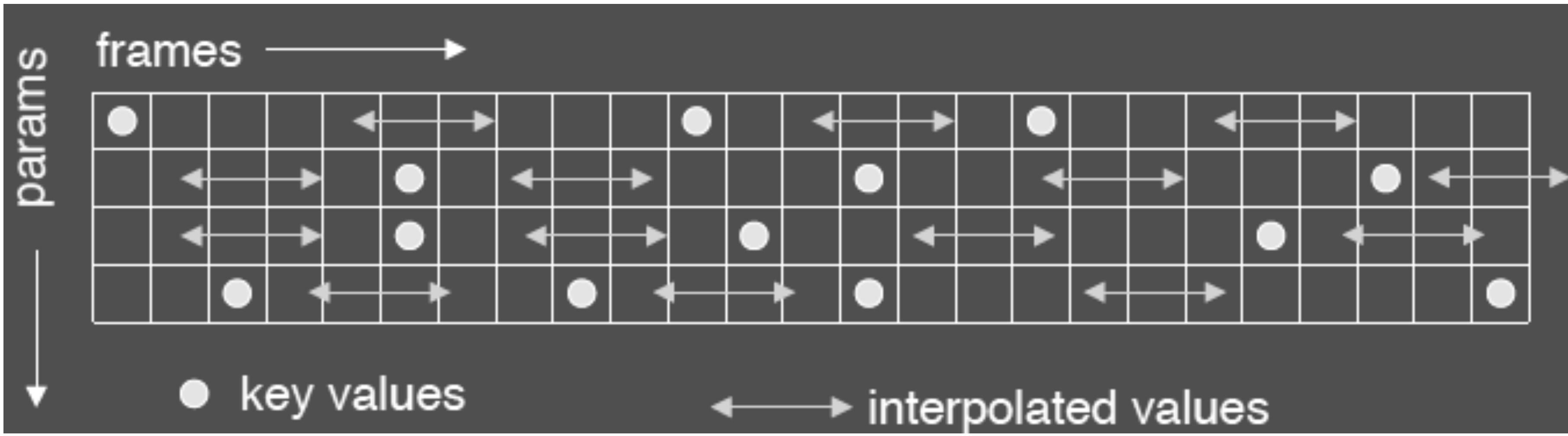



Replace an animation parameter $t$ with $u(t)$

Bezier Curve is tunable way to implement this

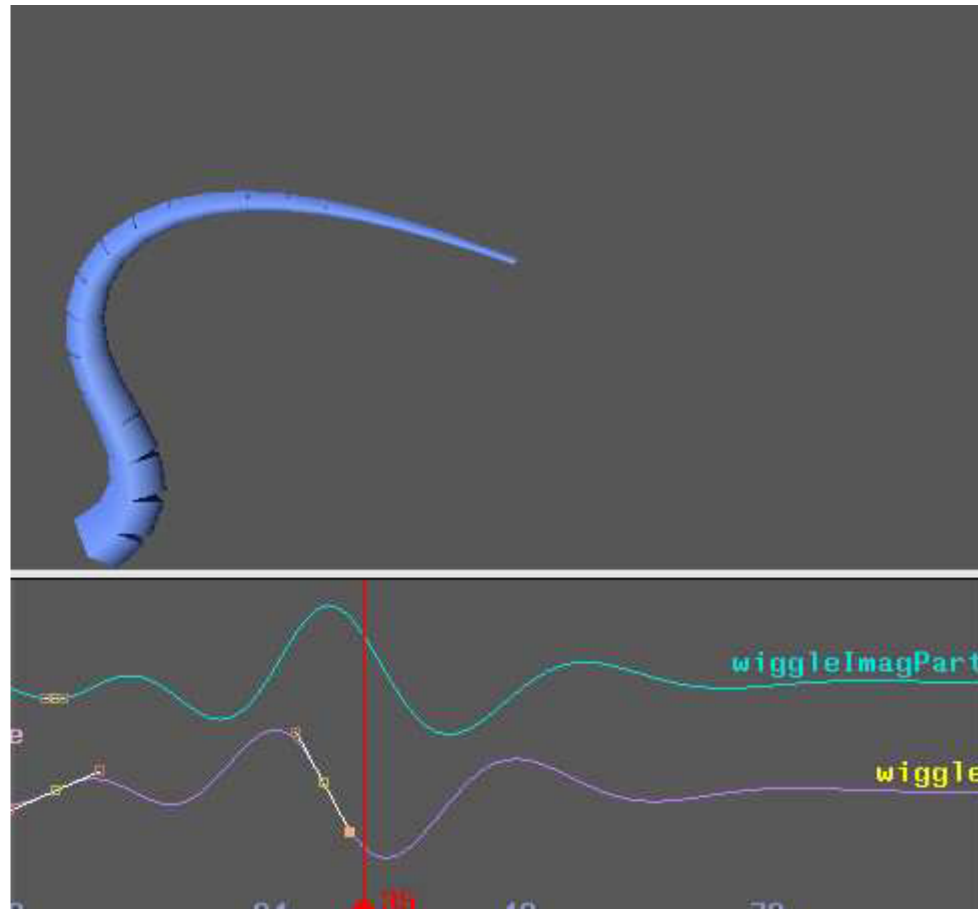


# Key Values vs Key Frames




- Animation composed of several parameters
- Parameters may not agree on which “frame” is important, i.e. there is no “key-frame” per-se.
- Create path for each parameter, which has key-values specified in appropriate frames.

# Specialized Curves for Animation




**"Wiggly Splines"**

**Kass and Anderson, SIGGRAPH 2008**

# Interpolating Orientations

# Interpolating Matrices ?

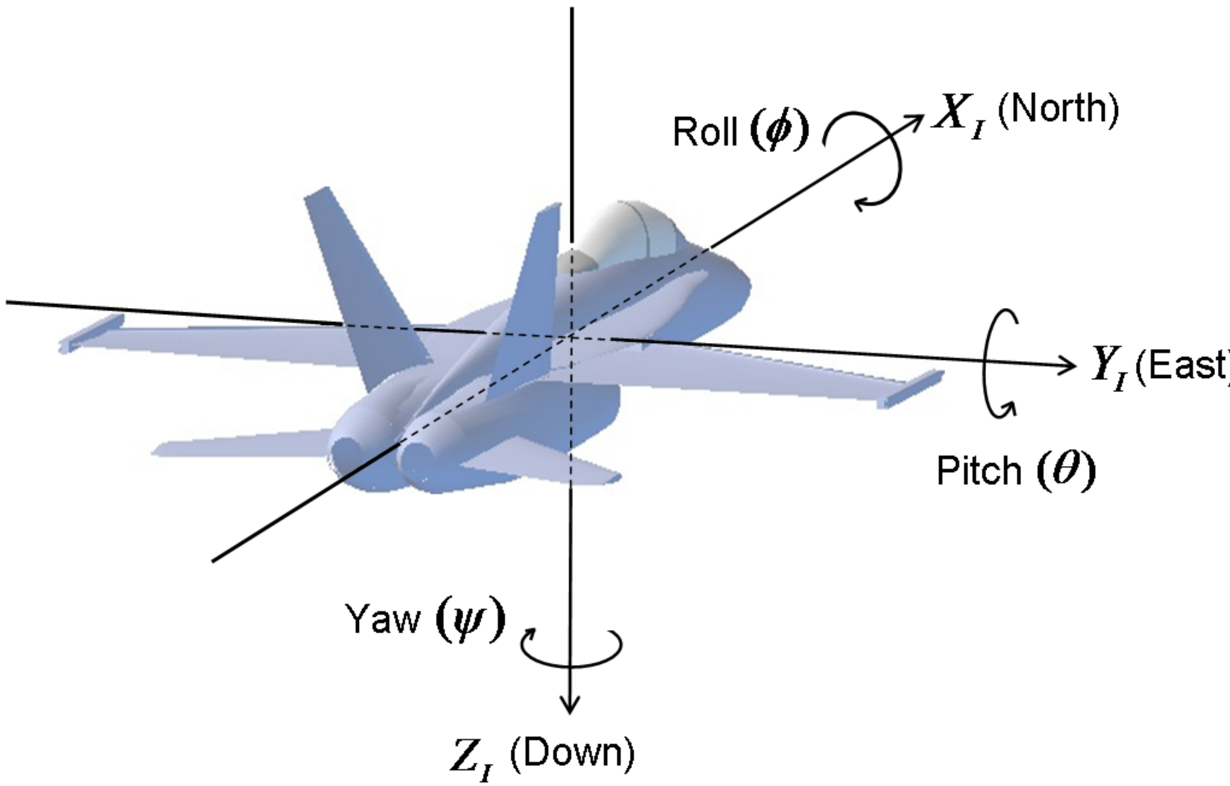
$$\frac{1}{2}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

90° CW

90° CCW

Not a rotation !

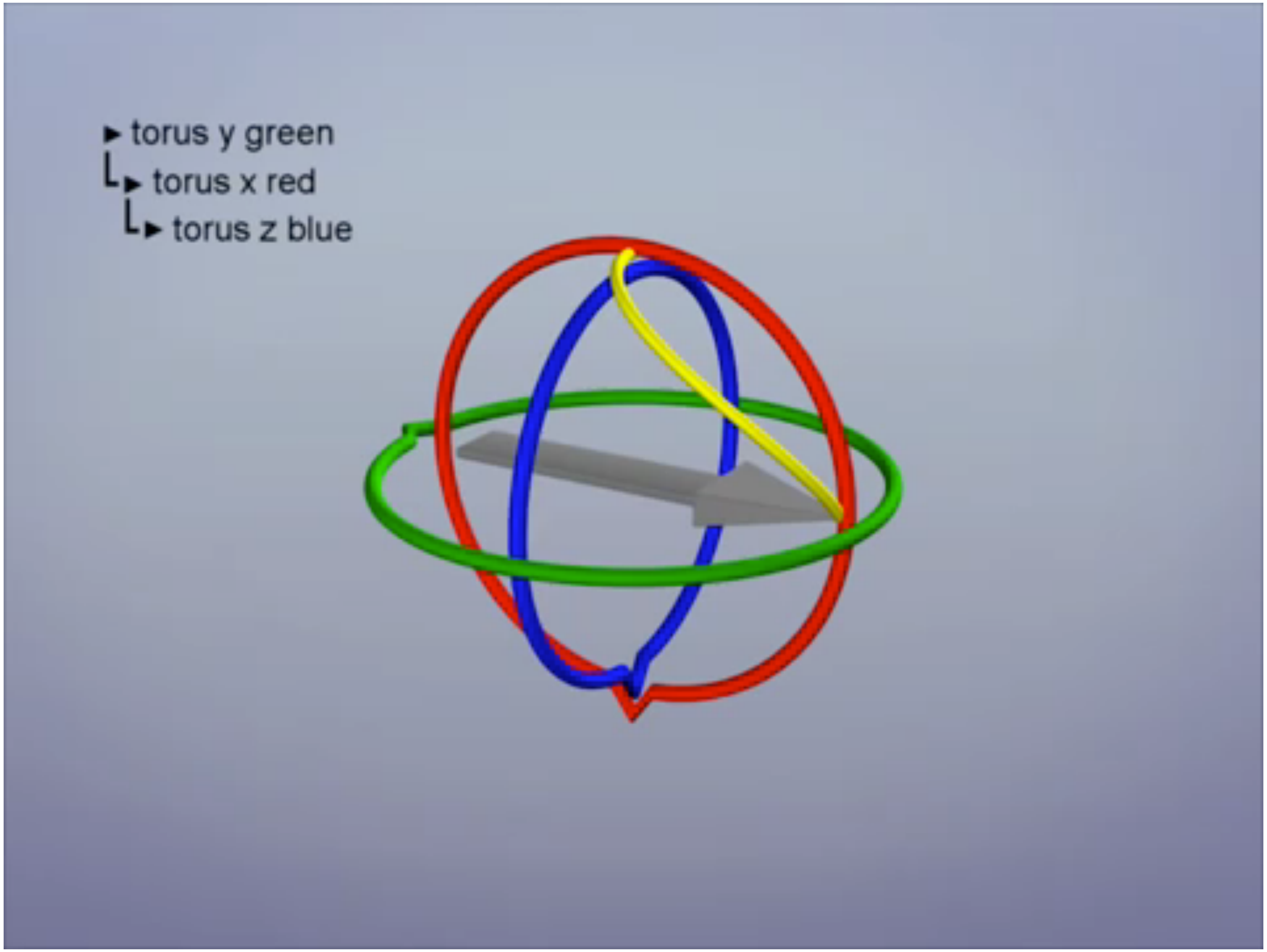
# Euler Angles



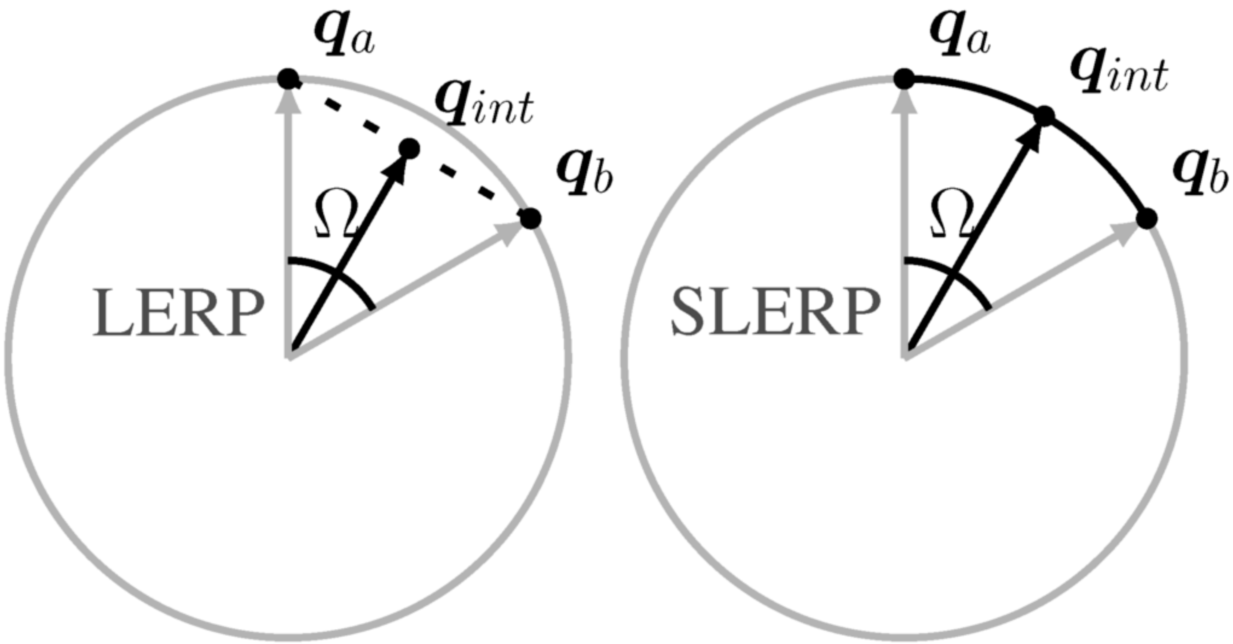

$$R = R_z(\psi) R_y(\theta) R_x(\phi)$$

http://www.chrobotics.com/wp-content/uploads/2012/11/Inertial-Frame.png

# Gimbal Lock

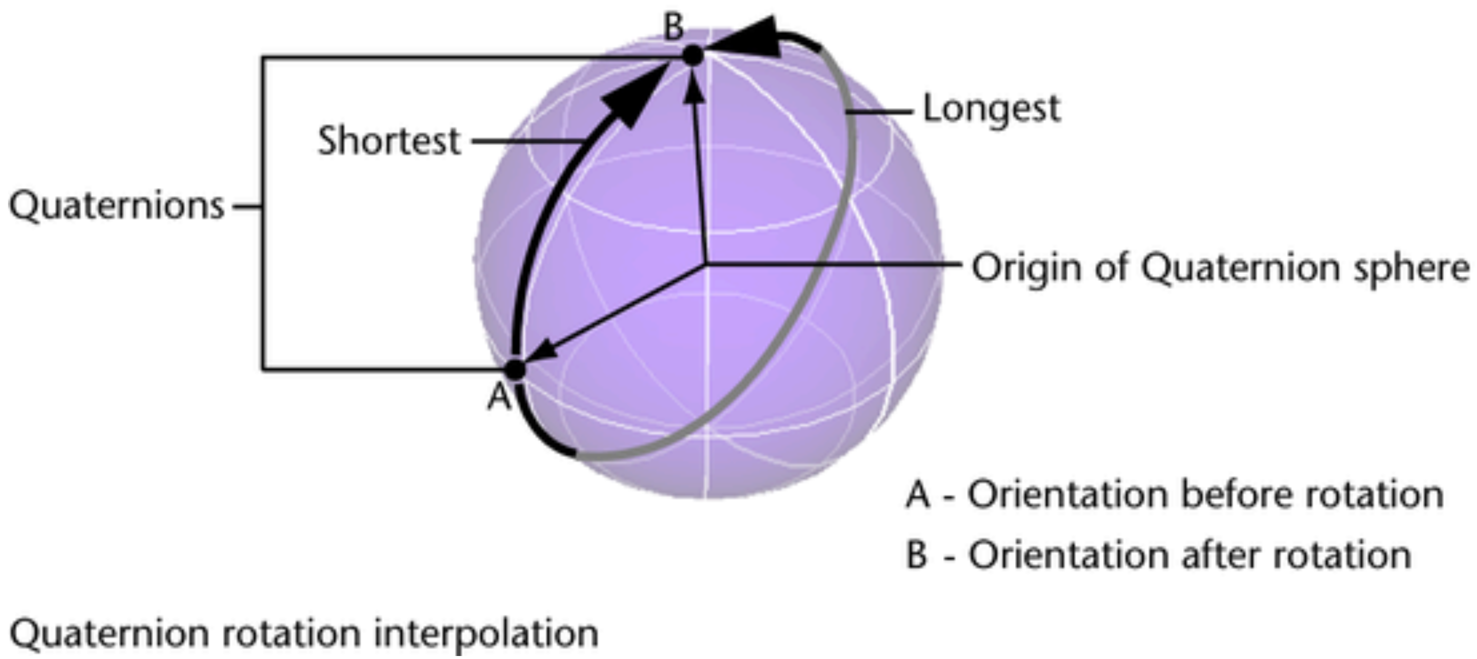


# Spherical Linear Interpolation (SLERP)




**LERP: Linear Interpolation**

# Spherical Linear Interpolation (SLERP)

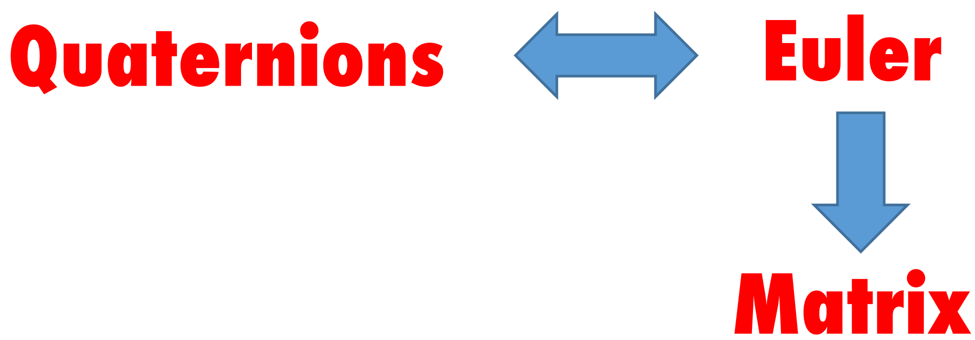



**Quaternions [Hamilton 1843]**

# Quaternions

$$q = [q_0, q_1, q_2, q_3]^T$$

$$|q|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

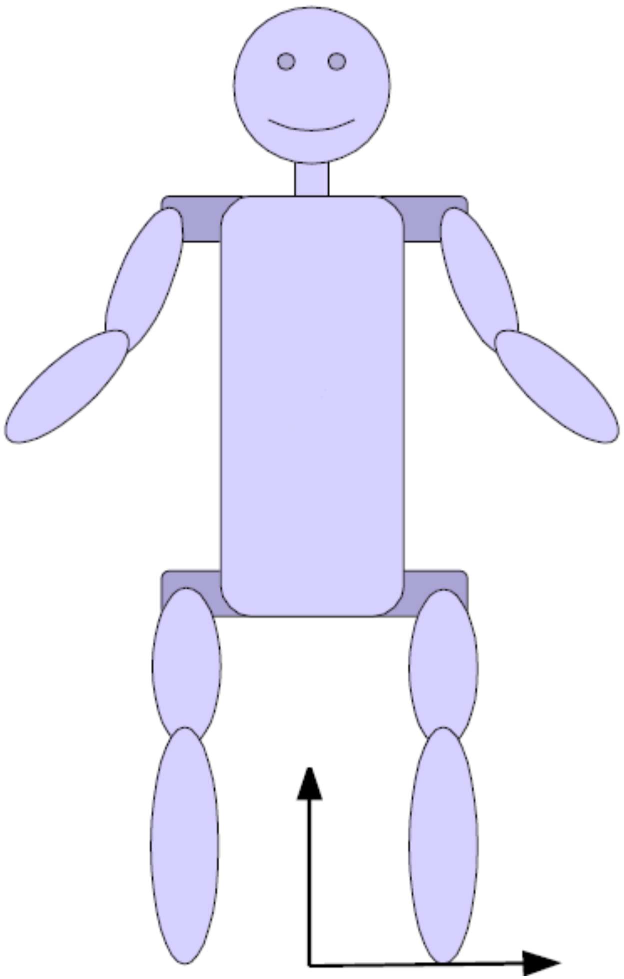



**For conversion formulae**

https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles

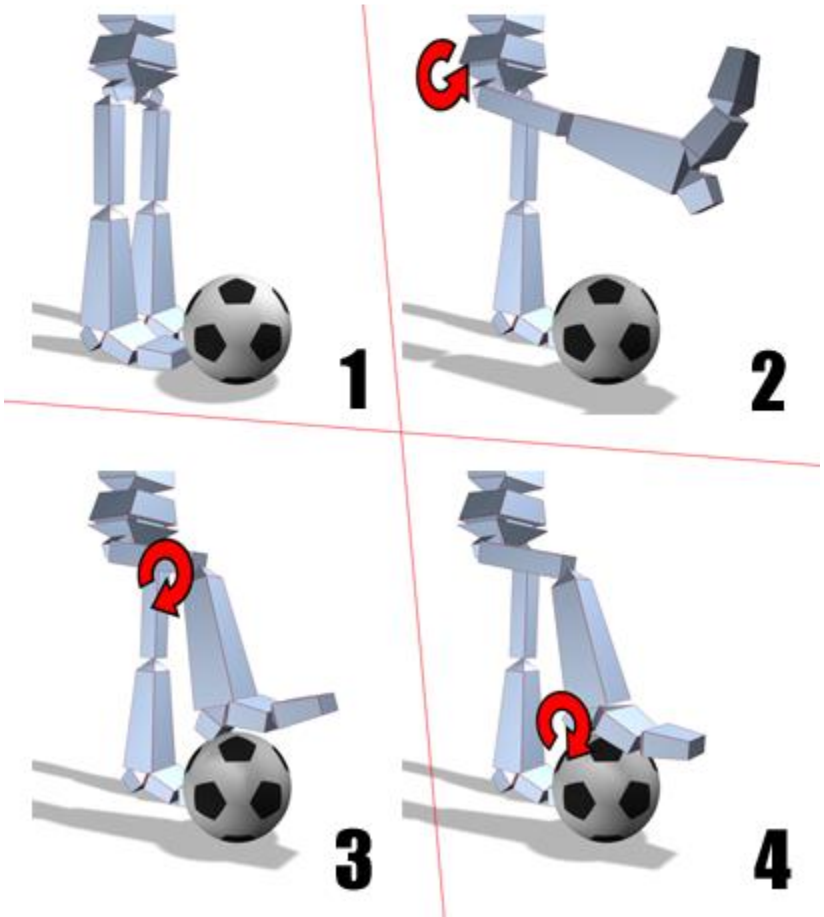
# Character Animation

# Recall: Hierarchical Modeling



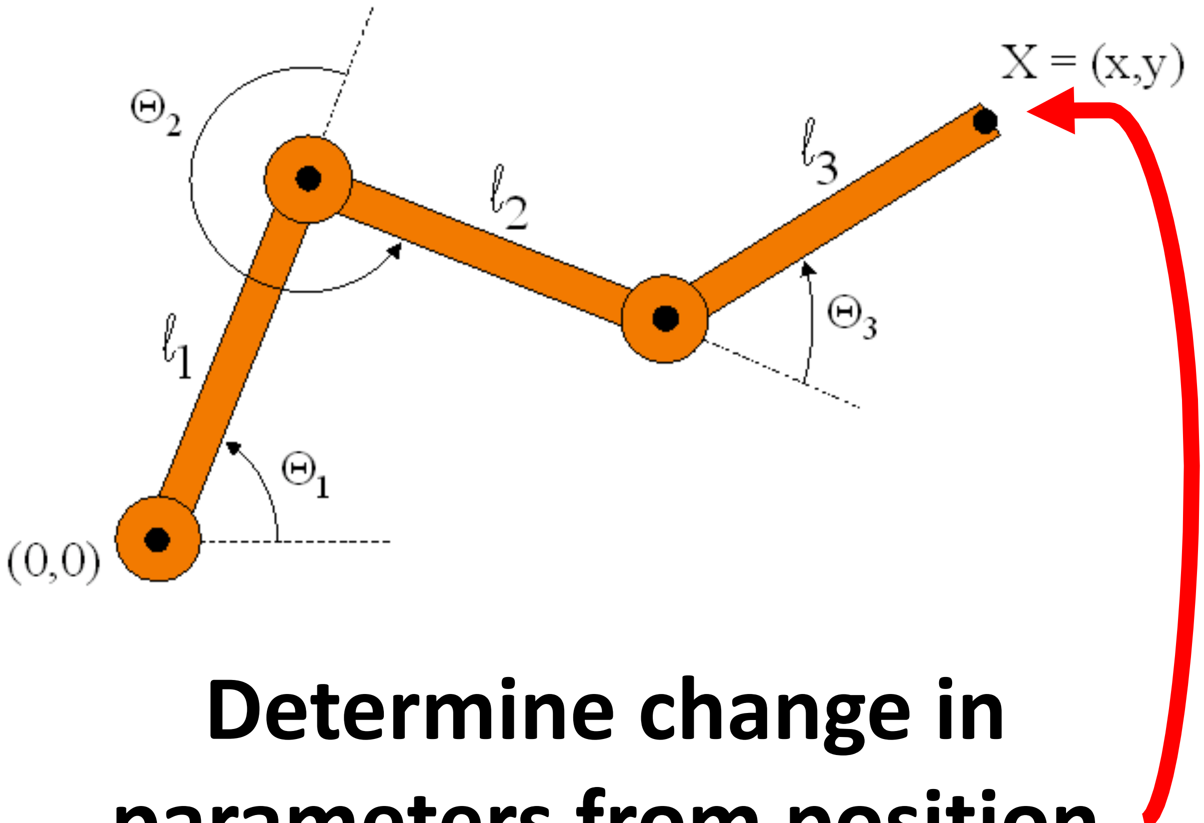
```
Body
 Torso
  Head
  Shoulder          Hips
   LeftArm            LeftLeg
    UpperArm           UpperLeg
    LowerArm           LowerLeg
    Hand               Foot
   RightArm           RightLeg
    UpperArm           UpperLeg
    LowerArm           LowerLeg
    Hand               Foot
```

# Forward Kinematics (FK)




**Manipulate degrees of freedom directly, construct geometry hierarchically.**

# Inverse Kinematics (IK)




**Determine change in parameters from position of end effector.**

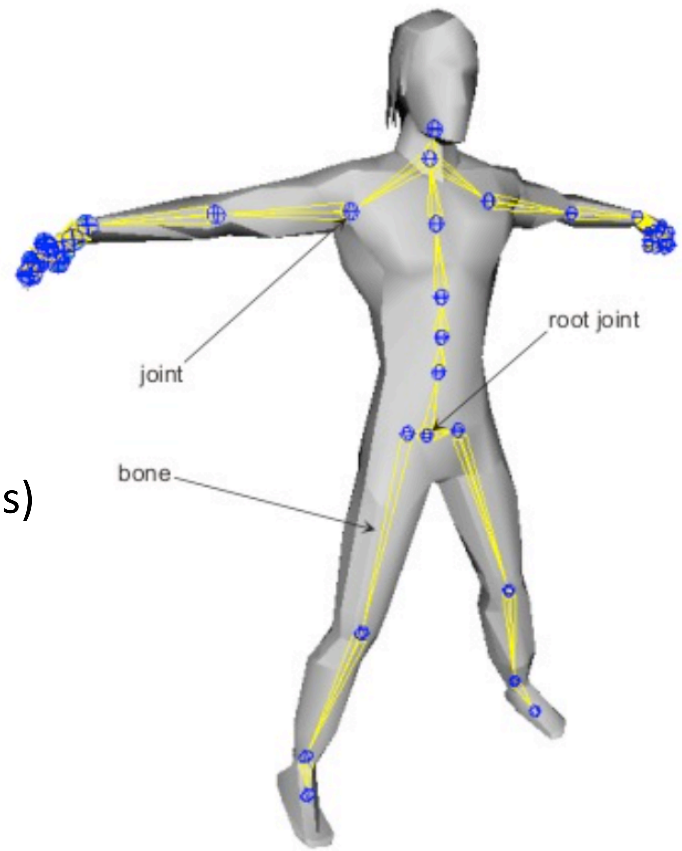
# Inverse Kinematics (IK)


**Non-Linear Optimization**

**Determine change in parameters from position of end effector.**

# Skinning and Bone Animation

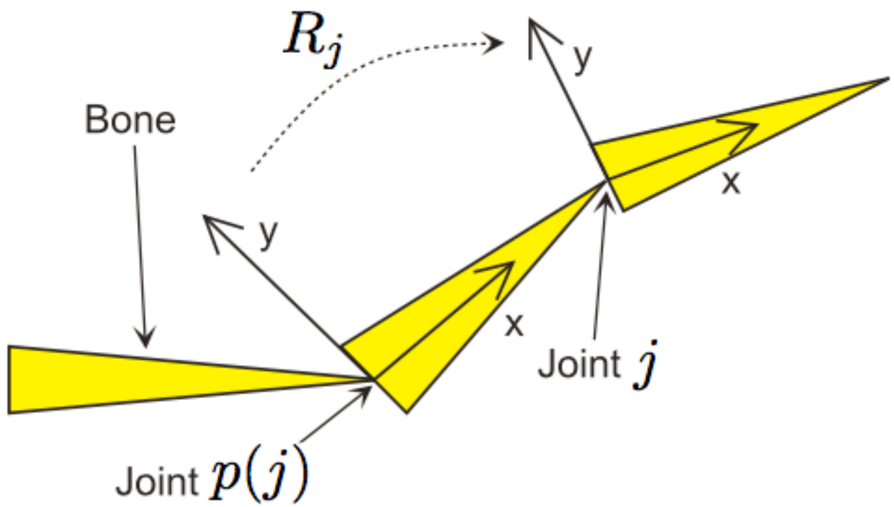
- **Skeleton**
  - Nodes represents joints
  - Joints are local Coordinate Systems (frames)
  - Edges represents bones
- **Skin**
  - 3D model/surface driver by skeleton




**Both are designed in a reference pose (rest pose)**

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Skeleton in Reference Pose

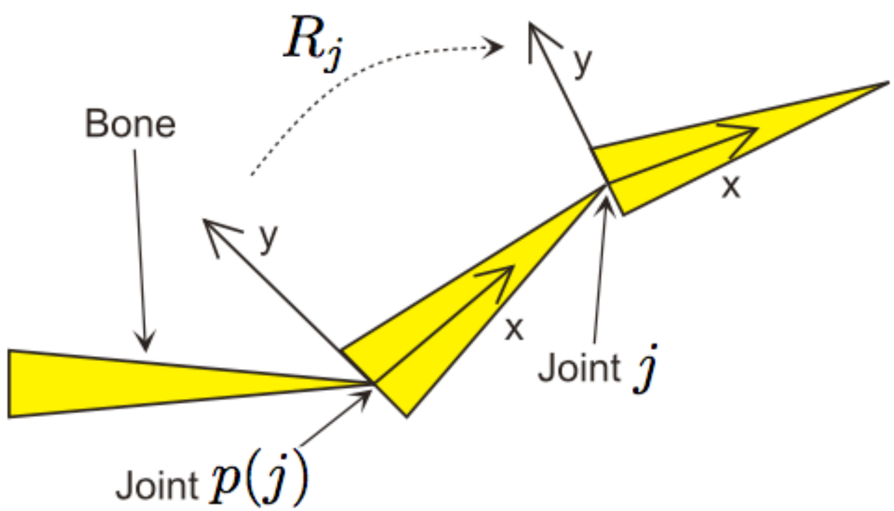



- Root Frame expressed with respect to the world: $R_0$
- Relative Joint Coordinate Frames - $R_1, R_2, R_3 \ldots R_j$

$$R_j = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Skeleton in Reference Pose




- Root Frame expressed with respect to the world: $R_0$
- Relative Joint Coordinate Frames - $R_1, R_2, R_3 \ldots R_j$
- Mapping from local frame to world
  - $A_j = R_0 \ldots R_{p(j)} R_j$, $p(j)$: Parent of joint j

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Animating Skeleton

Rotation at a joint j

$$T_j = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Mapping from Local Frame to World in **Reference** Pose

$$A_j = R_0 \cdots R_{p(j)} R_j$$

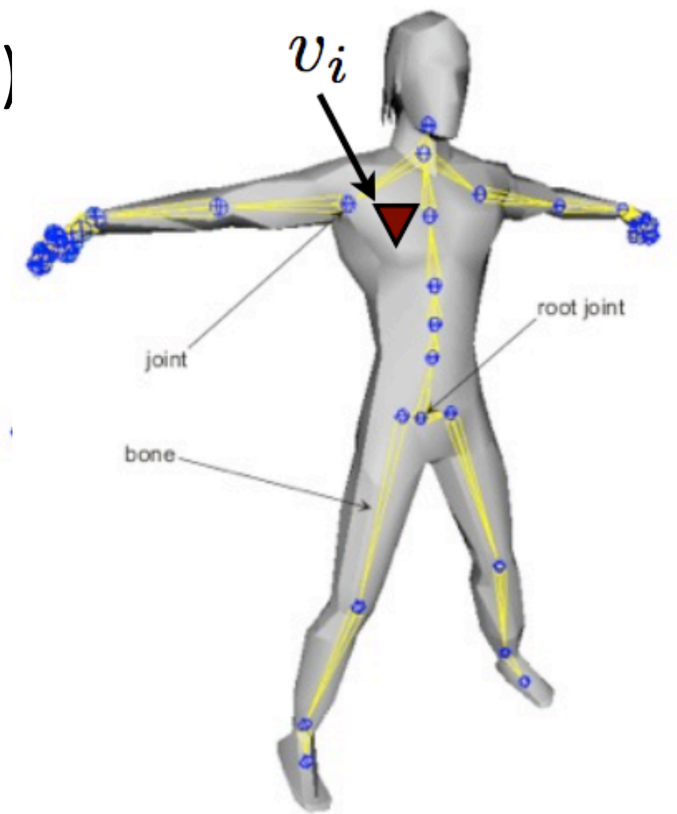
Mapping from Local Frame to World in **Animated** Pose

$$F_j = R_0 T_0 \ldots R_{p(j)} T_{p(j)} R_j T_j$$

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Character Rigging

- Embed Skeleton into a Mesh (Skin)
  - Assign Mesh Vertex to one or more bones to allow




$$\hat{v}_i = F_j (A_j)^{-1} v_i$$

$v_i$ - position of vertex in reference mesh

$A_j$ - joint $j$ in reference mesh

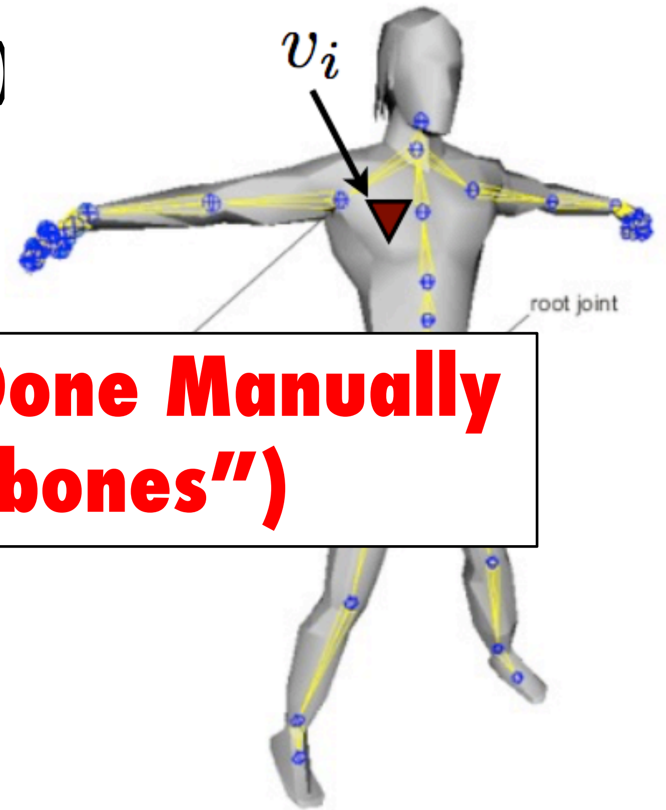
$F_j$ - joint $j$ in animated mesh

$\hat{v}_i$ - position of vertex in animated mesh

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Character Rigging

- Embed Skeleton into a Mesh (Skin)
  - Assign Mesh Vertex to one or more bones to allow




**Vertex Assignments Often Done Manually (Choose the closest "bones")**

$v_i$ - position of vertex in reference mesh

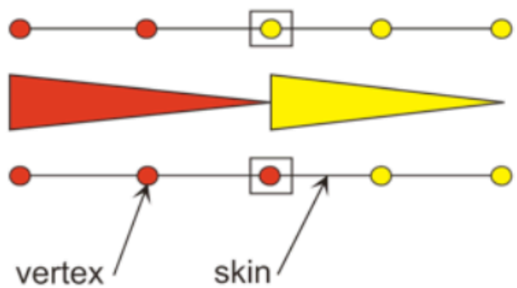
$A_j$ - joint $j$ in reference mesh

$F_j$ - joint $j$ in animated mesh

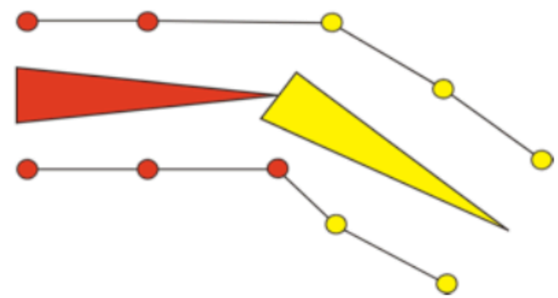
$\hat{v}_i$ - position of vertex in animated mesh

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Rigid Skin Limitations



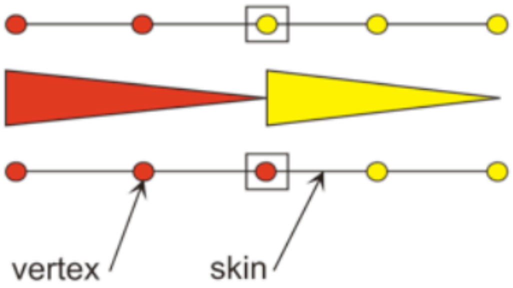

Reference Pose



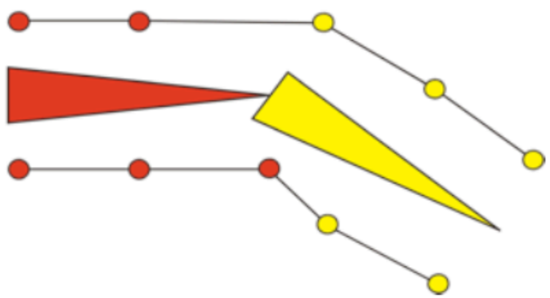
Animated Pose

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Rigid Skin Limitations




Reference Pose



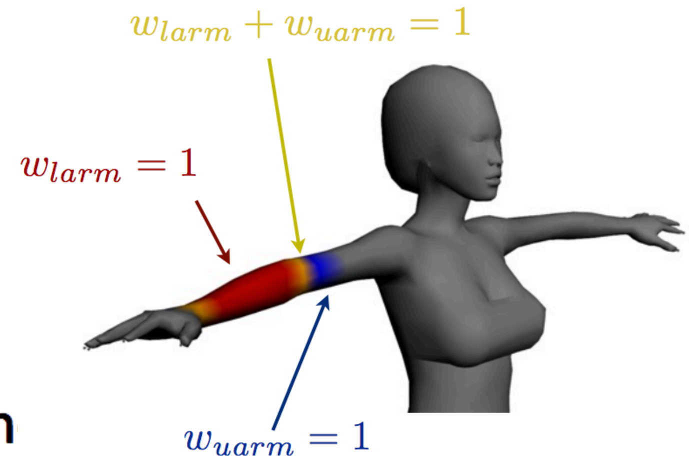
Animated Pose

**Leads to abrupt motion near the joint**

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Linear Blend Skinning

$$\hat{v}_i = \sum_{j=1}^{N} w_{ji} F_j (A_j)^{-1} v_i$$




$v_i$ - position of vertex i in reference m

$A_j$ - joint j in reference mesh

$F_j$ - joint j in animated mesh

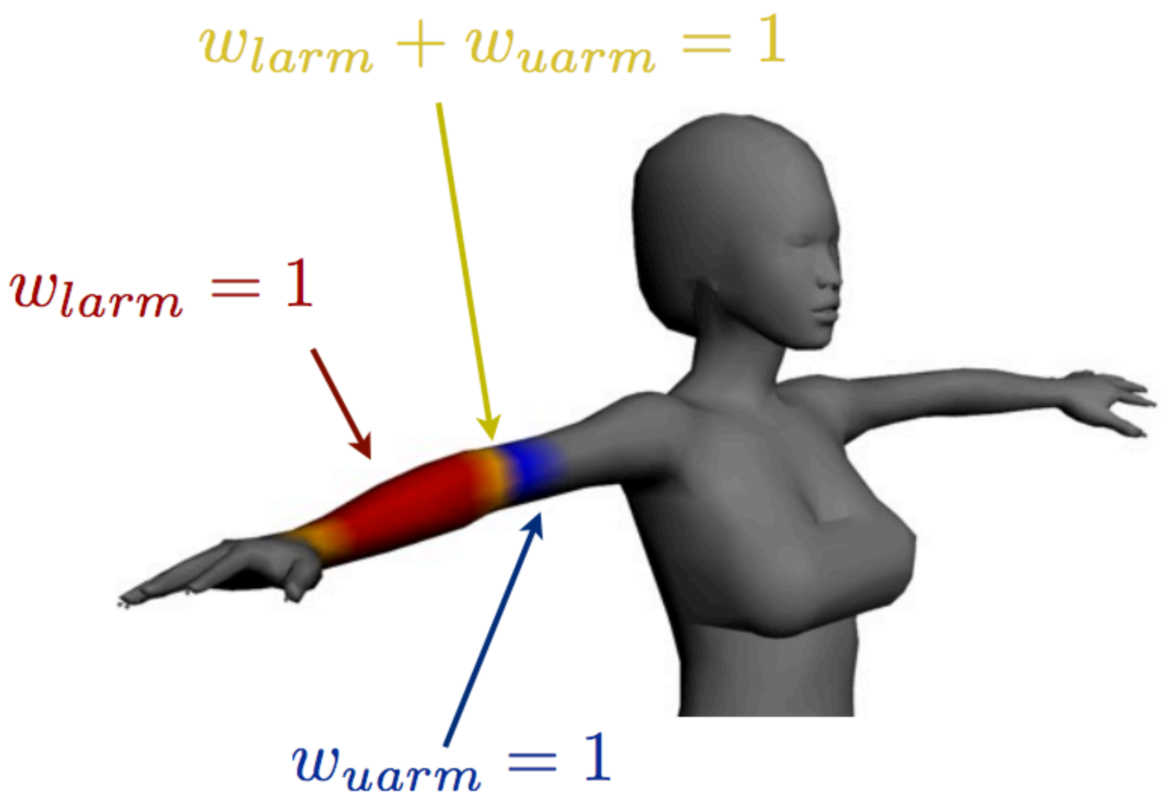
$\hat{v}_i$ - position of vertex $i$ in animated mesh

$w_{ji}$ - influence of joint $j$ on the vertex

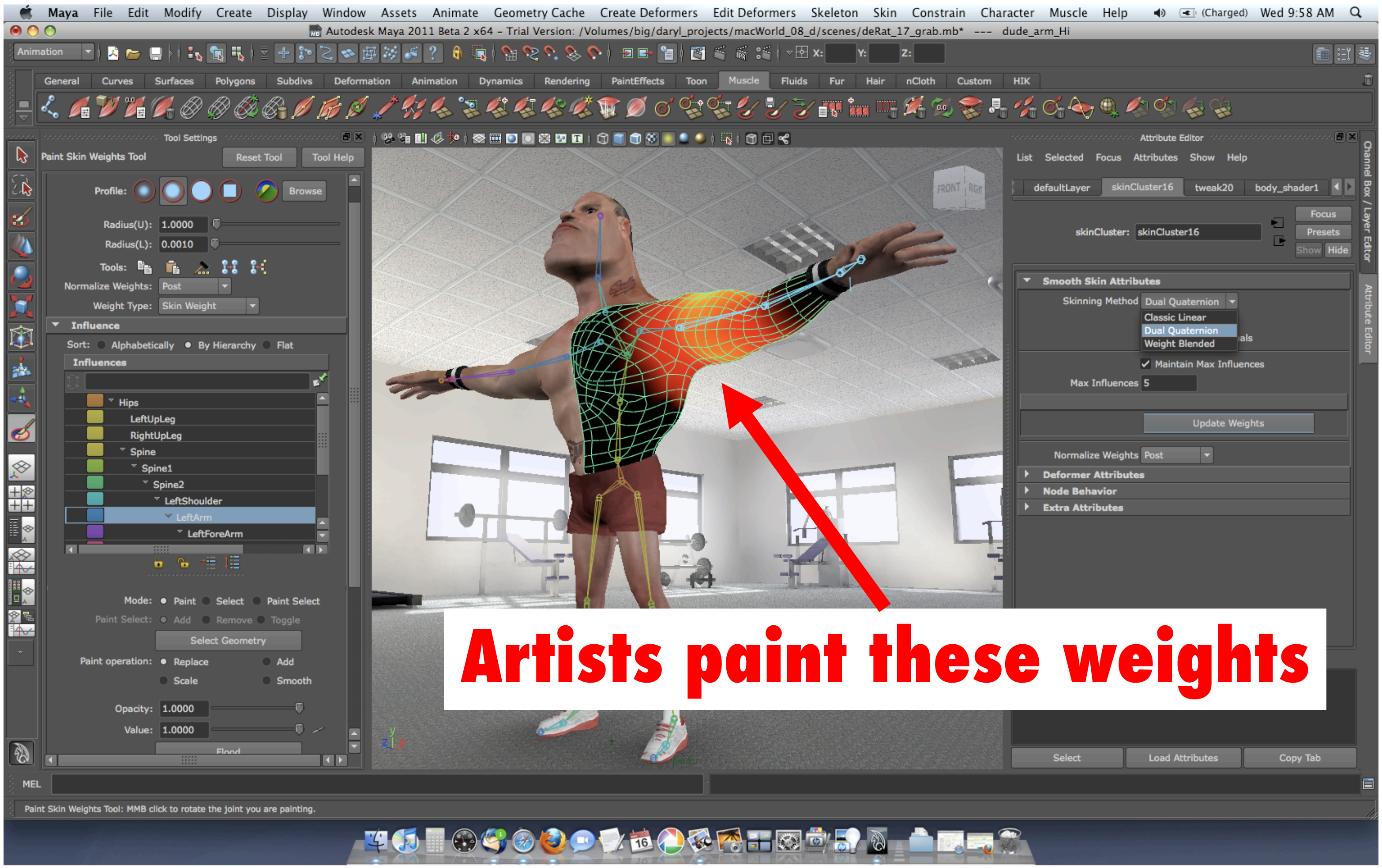
Weight needs to be convex $\sum_{j=1}^{N} w_{ji} = 1, w_{ji} \geq 0$

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Linear Blend Skinning



# Linear Blend Skinning




Maya

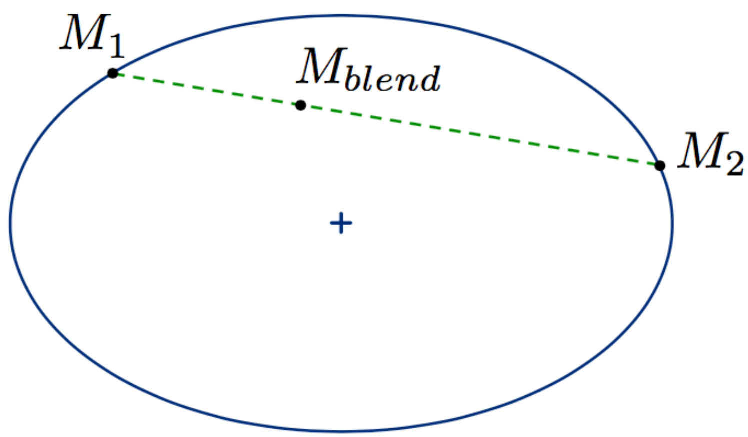
# Linear Blend Skinning: Limitations

**Candy Wrapper Effect**

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

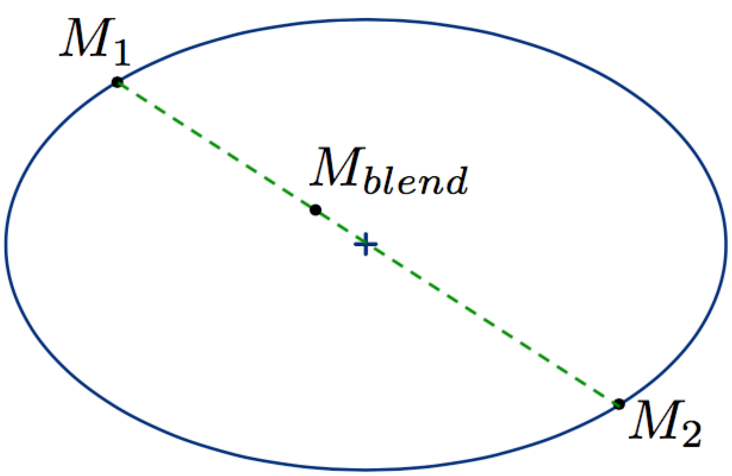
# Why Candy-Wrapper in LBS ?

$$\hat{v}_i = \sum_{j=1}^{N} w_{ji} F_j (A_j)^{-1} v_i \longleftrightarrow \hat{v}_i = \left( \sum_{j=1}^{N} w_{ji} M_j \right) v_i$$

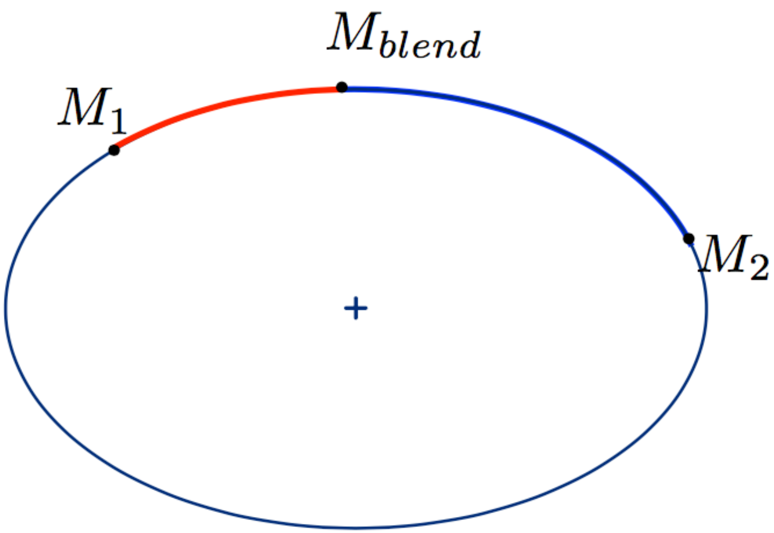



http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Why Candy-Wrapper in LBS ?




Linear




SLERP

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Solution to Candy-Wrapper

- Dual Quternion Skinning [Clifford 1873]
  - Kavan et al.,ACMTOG 2008
- Model rigid transformation (Rotation+Translation)
- Map 6 dimensional manifold into 8 dimensional space

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# LBS vs Dual-Quaternion Skinning

Linear Blend Skinning

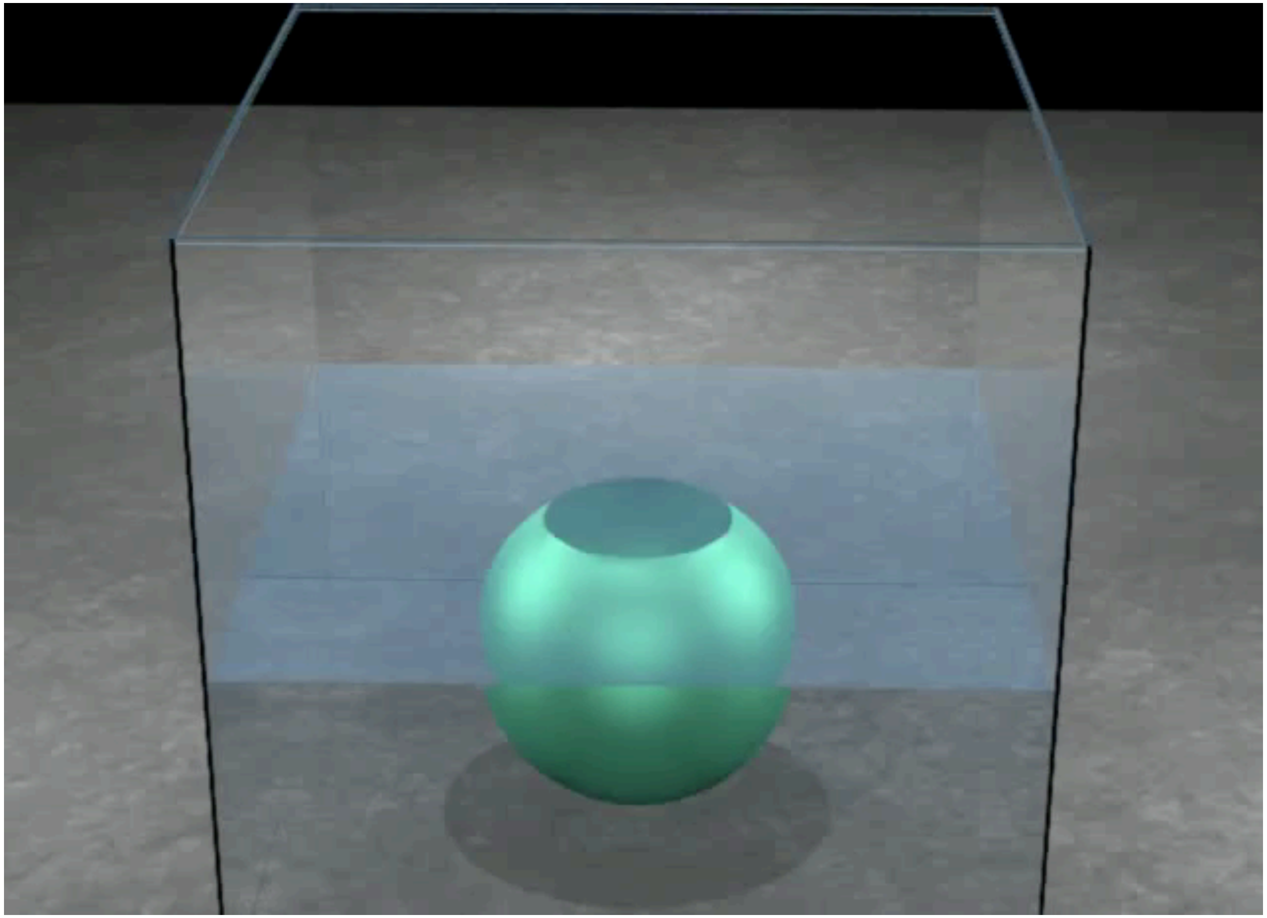
Dual-Quaternion Skinning

http://www.cs.cmu.edu/~yaser/Lecture-9-Skinning%20and%20Body%20Representations.pdf

# Physically Based Animation

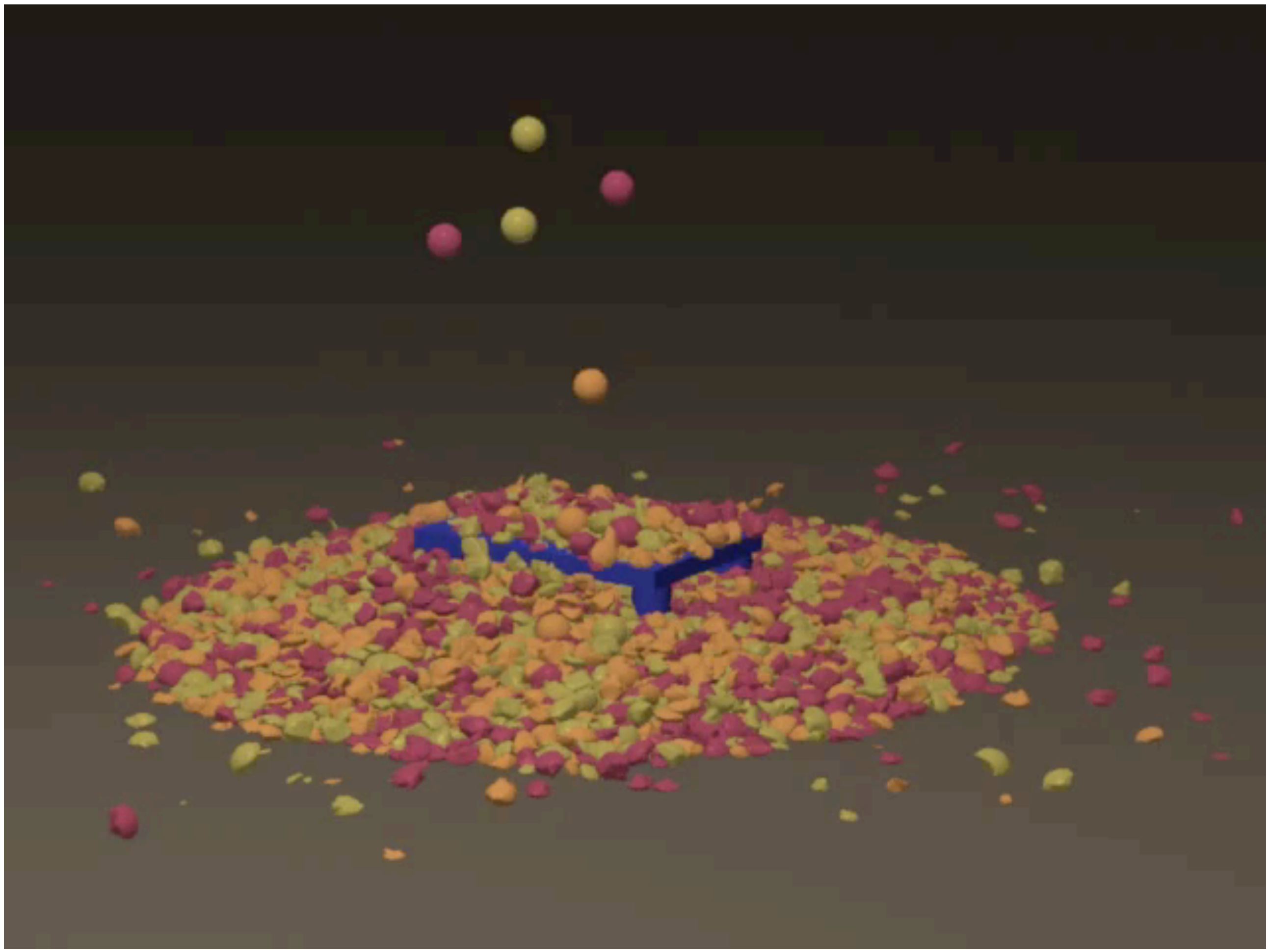
Teaser (David will teach next week)

# Two-way Solid-Fluid Coupling

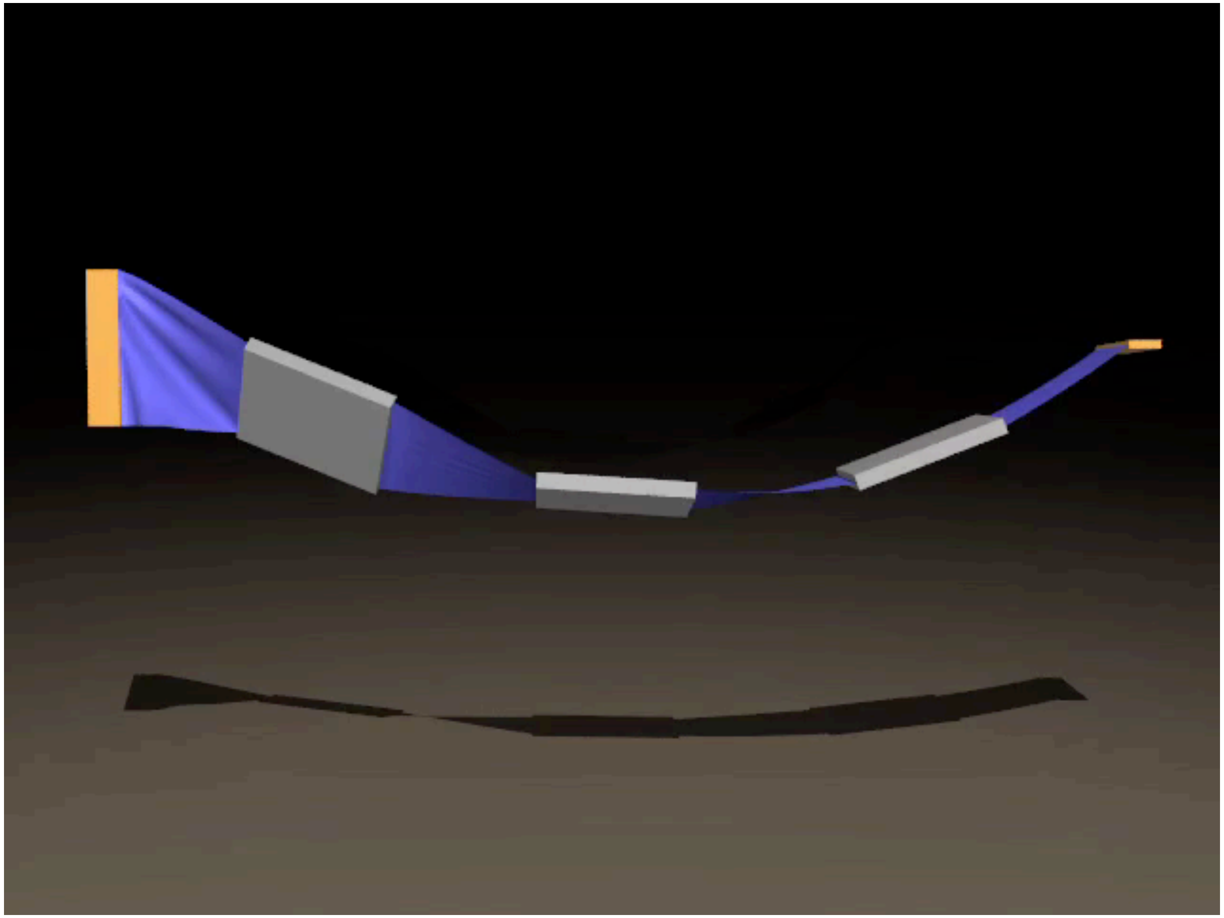


# Detailed Cloth Simulation

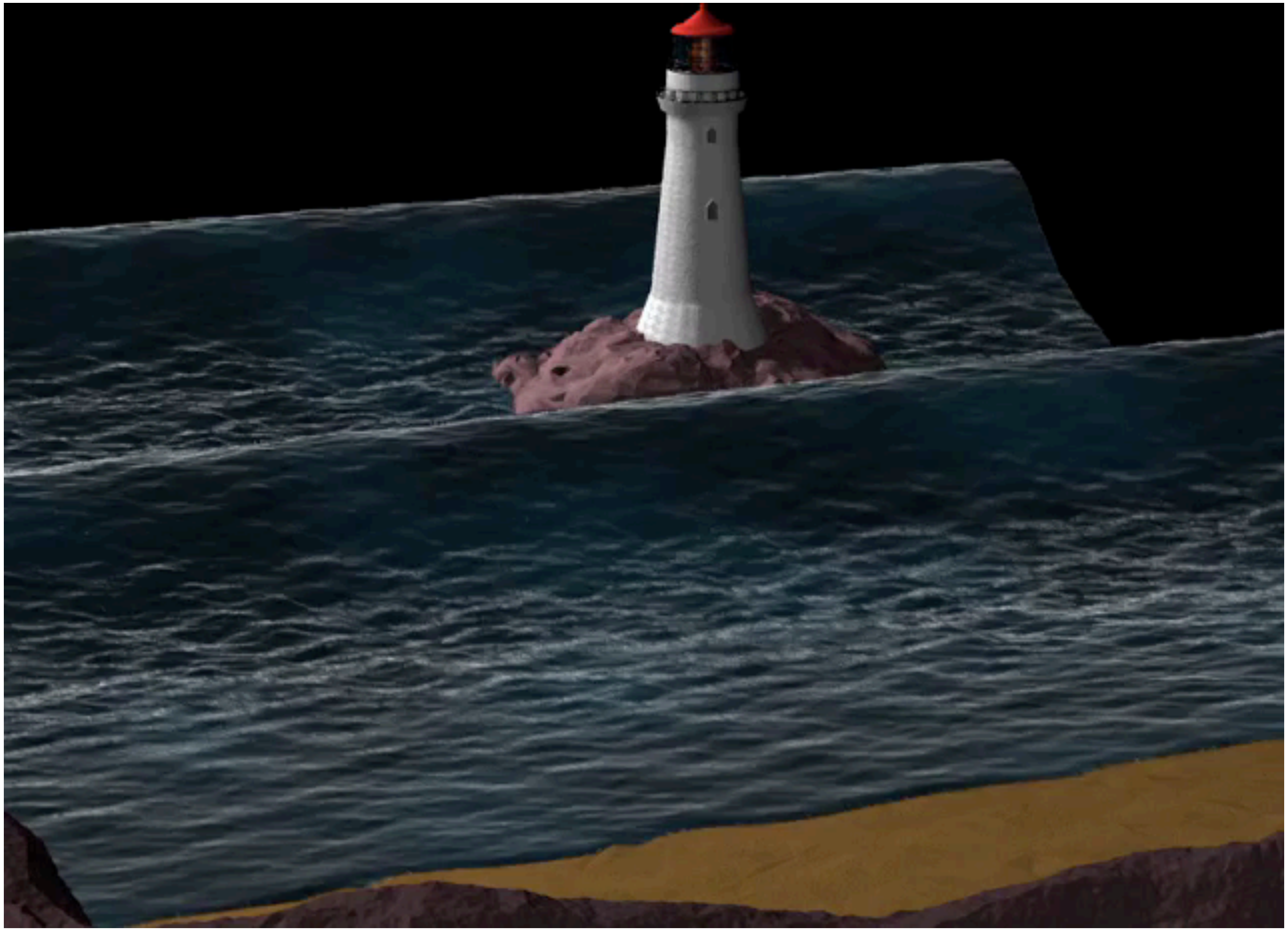
# Fracture and Rigid Body



# Hybrid Daisy Chain



# Fluid Simulation



# CS 148: Summer 2016
# Introduction of Graphics and Imaging
Zahid Hossain