Octonions

(Just kidding)
Assignment 1 discussion
Recap: slerp & quaternions

Rotations are non-Euclidean
- Matrix interpolation does not result in rotations
- Euler angles are full of problems

Unit 4D quaternions
\[ q = (\cos(\theta/2); \mathbf{v} \sin(\theta/2)) \]
represents rotation of \( \theta \) around axis \( \mathbf{v} \)
slerp just works directly!

De Casteljeau for splines

Conversion with matrices
- Polar decomposition if not orthonormal

Cool 4D fractals: quaternion Julia sets
Buzzword

Quaternions are a Lie group:
  manifold+group
Practical issues

slerp is expensive

- lots of sine evaluations

storage

hard to keep the vertical up (for camera interpolation)

\[
slerp(q_0, q_1, t) = q(t) = \frac{q_0 \sin((1-t)\omega) + q_1 \sin(t\omega)}{\sin(\omega)},
\]
Alternative: Normalized lerp (nlerp)

Just do the linear interpolation of quaternions
And then normalize
Not uniform,
   but not so bad if rotations are close enough
Advocated by Casey Muratori (RAD Game Tools)
“Worst case of lerp speed variation. Green: ideal result produced by slerp; Red: distorted result produced by lerp. The error between these two functions should be measured vertically, so they are more different than they may appear at first.”
Normalized lerp (nlerp): analysis

Can be seen as a linearization of the sine
Can be compensated for by an additional cubic term

\[
\mathrm{slerp}(q_0, q_1, t) = q(t) = \frac{q_0 \sin((1-t)\omega) + q_1 \sin(t\omega)}{\sin(\omega)},
\]
Questions?
Options for rotations

Matrix - not great, not rigid
Euler angles - mostly bad
Quaternion
  - slerp
  - nlerp
  - corrected nlerp (with extra cubic)
  - variational splines

Axis-angle, aka exponential map
  - store rotation of $\theta$ around $\mathbf{v}$ using vector $\theta\mathbf{v}$
  - singularities around $2\pi$
  - OK if interpolating small angles
Questions?
Skinning /
Enveloping
Back to hierarchical modeling

Remember forward kinematics?

- Rotation at each joint

Great to model robots

What about smooth surfaces such as human outer skin?
Enveloped (skinned) characters are pervasive.

Skeletons are often used to control meshes.

Slide from Robert Wang
Skinning/Enveloping

We know how to deform bones, now we need to infer how skin deforms.

Most popular technique:
Skeletal Subspace Deformation (SSD)

- Each bone yield a deformation of the space around it (rotation)
- In the middle of a limb, the skin points follow the bone rotation
- At a joint, skin is deformed according to a weighted combination of the bones

aka linear blend skinning, vertex blending

From wikipedia
Skeletal Subspace Deformation (SSD)

Bone rotations $R'$ & $R''$
Vertex p has weights 0.5 0.5
Transform according to $R'_t$ and $R''_t$ yields $p'_t$ and $p''_t$
the new position is the weighted average

Rest pose

After rotations

$R'_t$

$p'_t$

$p_t$

$p''_t$
Not perfect

After rotations

$R'_t$, $R''_t$, $p'_t$, $p_t$, $p''_t$
Questions?
SSD: problem statement

Given:
- Mesh in 1 bind pose
- Skeleton (bones or joints)
- Each vertex has weights for each relevant bone
  - tells how much it’s influenced by those bones
- For animation frame: (rigid) transformation for each bone

Goal:
- new vertex position to “envelope” the new skeleton configuration
Figure 4: Color coded influences maps from two bones on the animal creature. The yellow area for the upper leg and the red area for the lower leg. The darker the area, the smaller the influence.
Pseudocode

Do the usual forward kinematics
- maybe quaternion interpolation
- get a matrix $M_i(t)$ per bone

For each skin vertex $v_j$
- $v_j(t) = \sum w_{ij} M_i(t) v_j$
- normal: $n_j(t) = \sum w_{ij} M_i^{-T}(t) n_j$

- where $w_{ij}$ is the weight map.
- Weights for a vertex, usually sum to one.

Note that the weights are constant over time
- Only a small number of matrices change
- → enable implementation in graphics hardware (little information to update for each frame)
SSD

Image: Nvidia
SSD eye candy

from cgcharacter

cgSkin:
novel skinning technology
Questions?
The secret of SSD

Choose the appropriate weights
- Users can paint weight maps
- Weights can be optimized to match a set of example poses

Choose the bind pose wisely

“Rigging”: Associating a bind pose and a skeleton, figuring out the weights
Least square fit of weights

Given example animation, optimize SSD weights


http://portal.acm.org/citation.cfm?id=545261.545283&coll=GUIDE&dl=GUIDE&CFID=4867415&CFTOKEN=79896358

Figure 1: The multi-weight enveloping skinning process solves for a set of weights that approximate the movement of skin in a training exercise. The result is that the skin movement in the training exercise is generalized so that it applies well to other sequences of animation. 1-a shows some example poses from a training exercise. 1-b shows some frames from a new animation sequence.
SSD limitations

Figure 2: The ‘collapsing elbow’ in action, c.f. Figure 1.

Figure 3: The forearm in the ‘twist’ pose, as in turning a door handle, computed by SSD. As the twist approaches 180° the arm collapses.

From: Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation
J. P. Lewis, Matt Cordner, Nickson Fong
Limitation of SSD

It is a **linear** combination of transformation
Rotations beg to be combined differently (quaternions!!!!!)
Recap: what you should know

Given
- vertex position $v_j$ in bind pose
- per-vertex weights $w_{ij}$
- bone transformations $M_i(t)$

Compute new vertex position as
- $v_j(t) = \sum w_{ij} M_i(t) v_j$

Don’t forget: $M^{-T}$ for normal

Quality depends on rigging
- choice of skeleton and weights

Limitations due to linear interpolation (problem with rotations)
Real-time enveloping with rotational regression
Wang, Pulli, Popovic
We learn a fast model from exported examples.

Black Box Simulation  →  Exported Examples (skeleton-mesh pairs)  →  Fast Model

Our method

\{(q^i, y^i)\}

\(y(q)\)

Real-Time Enveloping with Rotational Regression

Robert Y. Wang\(^1\)  Kari Pulli\(^{1,2}\)  Jovan Popović\(^1\)

\(^1\)Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
\(^2\)Nokia Research Center
The two steps of our work are deformation gradients prediction and mesh reconstruction.
Rotational regression is good at capturing muscle bulges.
Real-time enveloping with rotational regression

Figure 1: Anatomy-based techniques produce good results but are too slow for many real-time applications. Our technique accurately captures muscle deformations from a set of examples generated by the black-box anatomical model and efficiently evaluates them on graphics hardware. Our technique is more accurate than linear blend skinning and almost as fast. (Credit: Joel Anderson)
Questions?
Back to quaternions!

Dual quaternions!
ToG 2008

Geometric Skinning with Approximate Dual Quaternion Blending

Ladislav Kavan\textsuperscript{1}  Steven Collins\textsuperscript{1}  Jiří Žára\textsuperscript{2}  Carol O’Sullivan\textsuperscript{1}

\textsuperscript{1}Trinity College Dublin, \textsuperscript{2}Czech Technical University in Prague

84.9 FPS  
Log-matrix Blending

197.4 FPS  
Dual Quaternions

55.1 FPS  
Spherical Blend Skinning

122 FPS  
Dual Quaternions
Dual numbers

kind of like complex numbers, vector space

a dual number \( \hat{a} = a_0 + \varepsilon a_\varepsilon \)

where \( a_0 \) and \( a_\varepsilon \) are scalar

and \( \varepsilon^2 = 0 \)

- then you can deduce multiplication rules by linearity
Dual quaternions

\[ \hat{q} = q_0 + \varepsilon q_\varepsilon \]

8 dimensional!
Can represent rigid transformations naturally
- rotation + translation

Does the right interpolation when the axis of rotation translates

Figure 21: Conversion of a rotation about axis \( s_0, \|s_0\| = 1 \), followed by translation \( t \) (top) to the corresponding screw (bottom).
Skinning with Dual Quaternions

L. Kavan, S. Collins, J. Zara, C. O'Sullivan

Trinity College Dublin
Czech Technical University in Prague
Questions?
Cool demo

When you have no skeleton, but a source animation for the full mesh

Skinning Mesh Animations

Doug L. James        Christopher D. Twigg

Carnegie Mellon University

Figure 1: Stampede! Ten thousand skinned mesh animations (SMAs) synthesized in graphics hardware at interactive rates. All SMAs are eformed using only traditional matrix palette skinning with well-chosen nonrigid bone transforms. Distant SMAs are simplified.
Skinning Mesh Animations

Doug L. James
Christopher D. Twigg

Carnegie Mellon University
Questions?
Automatic rigging

When you just have a mesh, and some reference skeleton animation

Automatic Rigging and Animation of 3D Characters
Ilya Baran & Jovan Popović
http://www.mit.edu/~ibaran/autorig/

Fredo Durand 6.837  MIT EECS
First fully automatic rigging algorithm

Input

3D Character

Generic Skeleton

Processing

Pinocchio

Output

Rigged Character

Slide from Ilya Baran
Pinocchio

Automatic Rigging and Animation of 3D Characters

SIGGRAPH 2007

papers_0030