CS 775: Advanced Computer Graphics
Lecture 4: Skinning
Character Animation

- Skinning

http://www.okino.com/conv/skinning.htm
Character Animation

- Skinning
  - Binding
Character Animation

- Skinning
  - Binding
  - Always done in a standard rest or *bind* pose.
Character Animation

- Skinning
  - Binding
  - Always done in a standard rest or bind pose.
  - Associate all skin mesh vertices to some skeleton joint(s).
Character Animation

- **Skinning**
  - Moving the skin vertices when the skeleton is moved.
  - Blending the various parts of the mesh.

http://www.okino.com/conv/skinning.htm
Character Animation

• Skinning
  - The skin is a (polygonal) mesh.
  - A mesh is a collection of connected (polygonal) primitives.

http://udn.epicgames.com/Three/UT3CustomCharacters.html
Character Animation

- Skinning
  - Binding
    - The skin mesh is defined in some local frame.
Character Animation

- Skinning
  - Binding
    - The skin mesh is defined in some local frame.
    - The skeleton joints are defined in their own local frames.
    - Let the transformation between any local frame \{ j\} of the skeleton and the local frame of the skin be given by:

\[ B_j \]
Character Animation

- Skinning
  - Binding
    - The skin mesh is defined in some local frame.
    - The skeleton joints are defined in their own local frames.
    - Let the transformation between any local frame \{ j \} of the skeleton and the local frame of the skin be given by:
      \[ B_j \]

- Binding Matrix
  - or Bind Pose Matrix
Character Animation

- Skinning
  - Binding
    - Associate a group of vertices to a single skeleton link
    - Every vertex of the mesh that is associated to the link \{ j\} is initially given, in the *bind* pose, in local skin space as \( v_k \)
Character Animation

- Skinning
  - Binding
    - Every vertex of the mesh that is associated to the link \{ j\} is initially given, in the *bind* pose, in local skin space as \( v_k \).
    - Binding expresses each skin vertex in the global frame as:
      \[
      0v_k = T_j \cdot B_j^{-1} \cdot v_k
      \]
Character Animation

- Skinning
  - Deforming the mesh
  - When the skeleton links move, $T_j$ changes.
Character Animation

- Skinning
  - Deforming the mesh
    - But the relative position of the vertex in the local joint frame does not change.

\[
{0}v_k = T_j \cdot B_j^{-1} \cdot v_k
\]
Character Animation

- Skinning
  - Deforming the mesh
    - This is known as **Rigid** or Simple skinning.

\[
\mathbf{v}_k = \mathbf{T}_j \mathbf{v}_0
\]
Character Animation

- Skinning
  - Rigid Skinning
    - Simple but low quality skinning.
    - Large distortions happen at bends.
Character Animation

● Skinning
  - Linear Blend Skinning
  • Vertex Blend Skinning, Skeletal Subspace deformation
  • Associate multiple joints with vertices and blend the effect of each joint on the vertex using weights.

\[ v_k^0 = \sum_i w_{i,k} T_i B_i^{-1} v_k \]
Character Animation

- Skinning
  - Linear Blend Skinning
  - Associate multiple joints with vertices and blend the effect of each joint on the vertex using weights.

\[
0v_k = \sum_i w_{i,k} B_i^{-1} v_k
\]

For every \( k \), \( \sum_i w_{i,k} = 1 \) and \( 0 < w_{i,k} \leq 1 \)

Here \( i \) is the index over all joints associated with the vertex \( v_k \)
Character Animation

• Skinning
  – Linear Blend Skinning
• Deforming the mesh

\[ 0v_k = \sum_i w_{i,k} T_i B_i^{-1} v_k = \sum_i w_{i,k} M_i v_k \]
Character Animation

- Skinning
  - Linear Blend Skinning
- Mesh normals

\[
0 n_k = \frac{\sum_i w_{i,k} N_i n_k}{\|\sum_i w_{i,k} N_i n_k\|} \\
\]

where \( N_i \) is the first 3x3 submatrix of \( M_i \)

Here we have a \( N_i \) that is rigid. What if it was an Affine transformation?
Character Animation

• Skinning
  - Algorithm
    • Skin::Update()
      - Compute $M_i = T_i B_i^{-1}$ for each joint. $B_i^{-1}$ can be precomputed and stored.
      - Loop through the vertices and blend positions and normals.
    • Skin::Draw()
      - Set matrix state to Identity
      - Loop through skin polygons and draw using global vertex positions and normals

Why separate the two?
Character Animation

- Skinning
  - Algorithm
    - Skin::Update() (*view independent processing*)
      - Compute $M_i = T_i B_i^{-1}$ for each joint. $B_i^{-1}$ can be precomputed and stored.
      - Loop through the vertices and blend positions and normals.
    - Skin::Draw() (*view dependent processing*)
      - Set matrix state to Identity
      - Loop through skin polygons and draw using global vertex positions and normals
Character Animation

- Skinning
  - Limitations of Vertex Blend Skinning
- Skin collapse - Bending

Figure 1: The skeleton subspace deformation algorithm. The deformed position of a point $p$ lies on the line $p'p''$ defined by the images of that point rigidly transformed by the neighboring skeletal coordinate frames, resulting in the characteristic ‘collapsing elbow’ problem (solid line).
Character Animation

- Skinning
  - Limitations of Vertex Blend Skinning
  - Skin collapse – Twisting (Candy Wrapper effect)
Character Animation

- Skinning
  - Limitations of Vertex Blend Skinning
    - Skin collapse
      - A quick solution used to prevent collapse during bending is to dynamically add more bones.

Was done in many games but does not solve the problem completely.
Character Animation

- Skinning
  - Limitations of Vertex Blend Skinning
  - Skin collapse
    - A better solution is to use dual quaternions

Geometric Skinning with Approximate Dual Quaternion Blending, Kavan Collins, Zara and O'Sullivan, ACM TOG 2008
Character Animation

- Skinning
  - Limitations of Vertex Blend Skinning
- Skin Binding
  - Containment Binding
Character Animation

- Skinning
  - Limitations of Vertex Blend Skinning
- Skin Binding
  - Containment Binding
  - Point-to-line Mapping
  - Manual (combined with adding weights)
Character Animation

● Skinning
  - Limitations of Vertex Blend Skinning
    • Indirect control via weights is non-intuitive
      - Weights are added either via simple heuristic rules like
        \[ w_{i,k} \propto \frac{1}{d_{i,k}} \]
        where \( d_{i,k} \) is the distance from the skin vertex \( v_k \) to the skeleton joint \( i \)
      - Added manually
      - Demo/Video
Character Animation

- Skinning
  - Limitations of Vertex Blend Skinning
    - Skin collapse
    - Skin Binding is difficult
    - Indirect control via weights is non-intuitive
    - No anatomical basis
  - Advantages
    - It is simple to do and so is very widely used
    - Good starting point for more complex skinning
    - Implementation on hardware is easy