A few (pre)processing tasks for skeletal animations

- Compression
  - input: ani with N keyframes
  - output: ani with M<N keyframes
- Retargeting
  - input: Rig1 + (Skel animat for Rig1) + Rig2
  - output: (Skel animat for Rig 2)
- Building from a blend-shape animation
  - input: Blend-shape
  - output: Rig + Skinned Mesh + Anim
  - note: the opposite is a trivial («baking»)

110

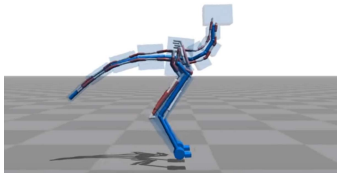
Compression of skeletal animations

- Objective: remove keyframes
  - the “redundant” ones
  - preprocessing task (e.g. as a game tool)
- Basic algorithm concept:
  - for each keyframe P_x
    - tentatively remove P_x
    - compute interpolated version P_i from remaining keyframes
      - (the prev and next ones)
    - if *distance*(P_i , P_x) > MAX_ERR then reinsert keyframe P_x

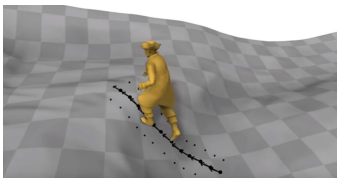
111

Research topic: apply ML to skeletal animations

- A very active area of research…



Flexible Muscle-Based Locomotion for Bipedal Creatures
Thomas Geijtenbeek, Michiel van de Panne, A. Frank van der Stappen
SIGGRAPH 2013

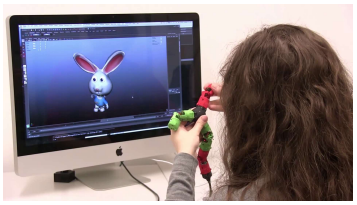


Phase-Functioned Neural Networks for Character Control
Daniel Holden, Taku Komara, Jun Saito
SIGGRAPH 2017

(Among MANY others)

112

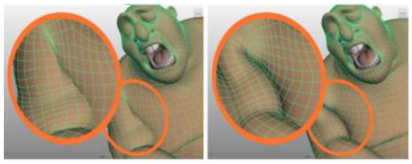
Research topic: better interfaces to author animations



Tangible and Modular Input Device for Character Articulation
Alec Jacobson, Daniele Panozzo, Oliver Glauser, Cedric Pradalier, Otmar Hilliges, Olga Sorkine-Hornung
SIGGRAPH 2014

113

Research topic: Deformation beyond standard skinning



Efficient Elasticity for Character Skinning with Contact and Collisions
Aleka McAdams et al (Disney animation)
SIGGRAPH 11

Note: usually way more complex than direct methods (LBS / DQS).
More offline animation oriented than videogames

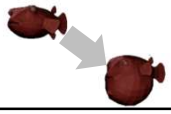
114

Per-vertex animations VS Skeletal-Animations

- Per Vertex animations
  - can interpolate keyframes (but linear trajectories)
  - heavy in RAM
    - replications of normals / positions
  - light to render / compute
- Skeletal animations
  - can interpolate keyframes *better* (curved trajectories)
  - light in RAM
    - animations / models orthogonality
  - minor overheads
    - transform interpolation (x vert!)
    - updates final transoform before (unless can be baked)

115

Animations in games
(of 3D Solid Objects)

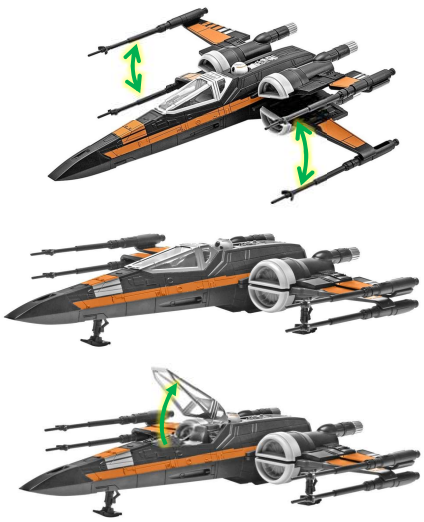
	Non-Procedural (ASSETS)	Procedural (e.g. PHYSIC ENGINE)	
Rigid 	Kinematic animations	Rigid body dynamics	
Articulated 	Skeletal Animations	Ragdolling	Inverse kinematics
Free form 	Blend-Shapes	~~(generic) deformable object simulation~~ *usually too expensive*	Cloth/ garments Ropes

116

Non-procedural Animations: which one to pick?

- Which format to pick?

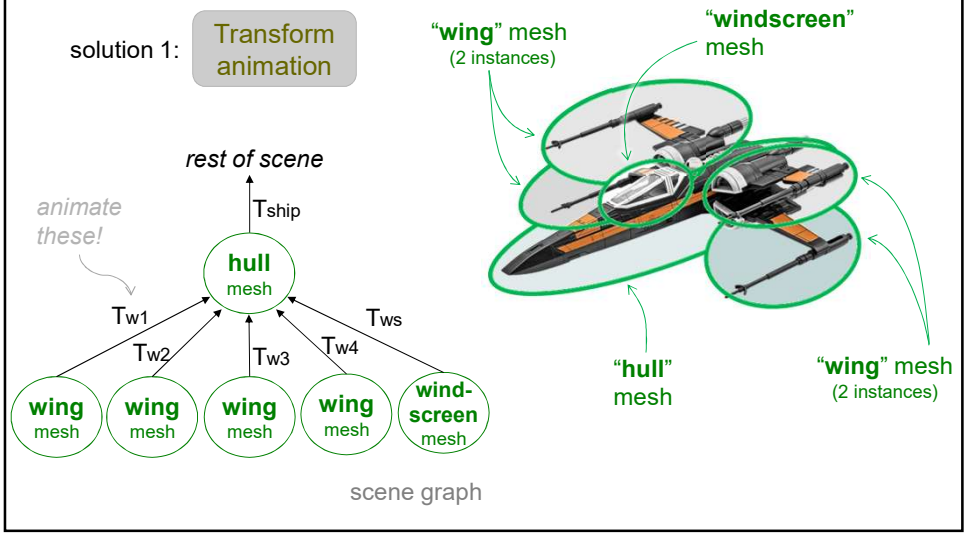
EXAMPLE:
say we want
a model capable of
doing this:



117

Non-procedural Animations: which one to pick?

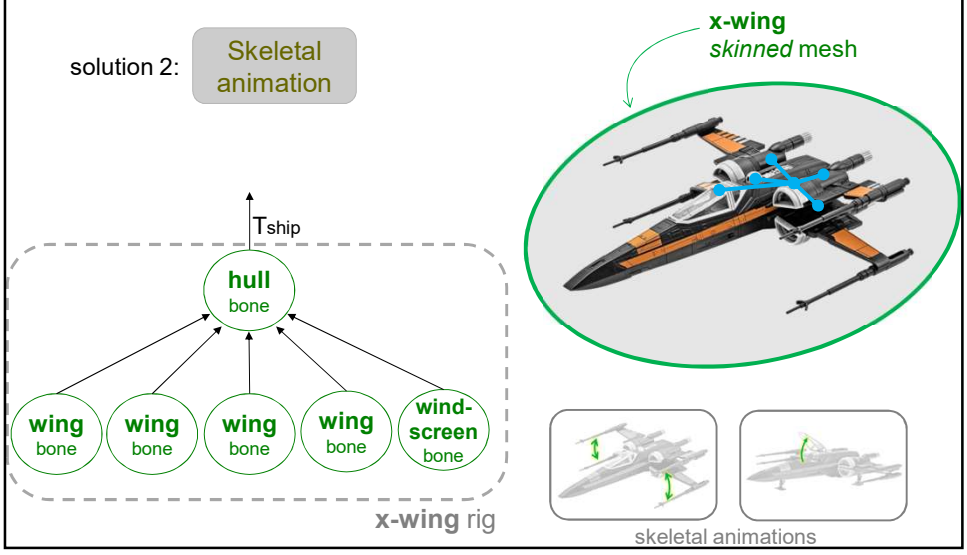
solution 1: Transform animation



118

Non-procedural Animations: which one to pick?

solution 2: Skeletal animation



119

Non-procedural Animations: which one to pick?

solution 3: Blend-shape

base shape morph 1 morph 2

"x-wing" blend-shape

120

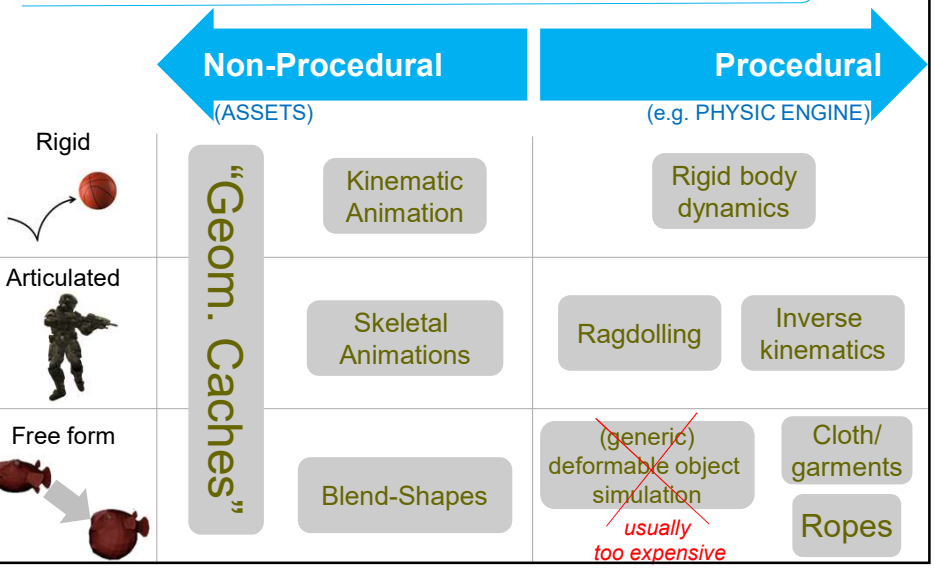
Non-procedural Animations: which one to pick?

In this example:

- **Animation of transforms (of the scenegraph)**:
  - *how:* 3 (rigid) meshes, 5 instances, animate scenegraph transforms
  - can reuse geometry for all wings: most compact on RAM ☺
  - simpler rendering
  - 5 separated draw calls! ☹
- **Skeletal animation**:
  - *how:* one rig + one skinned mesh + few skeletal animations
    - mesh skinning: single bone enough in this case
  - if very low poly mesh (few polys): a waste?
  - more taxing rendering (a bit) ☹
    - real time skinning on vertex any
  - single draw call! ☺
- **Blend shapes**:
  - *how:* blend shape with one base shape + 2 morphs
  - minimal impact
  - worst quality interpolation: linear → straight (non curved) paths
    - vertices on straight paths (unless, more shapes added)
  - heaviest on RAM ☹
    - (a waste of DoF!)
    - not important, if very low res
  - single draw call! ☺
    - but to different buffers each frame / or to a larger buffer

121

Animations in games
(of 3D Solid Objects)



122

Geom. Caches
(for lack of a better name)

- Baked, optimized animations
  - of a mixture of types e.g.
    - blend shapes
    - kinematic animatios
      - (approximated)
    - skinned animations
      - (typically, no scene graph, just final transf)
  - optimized
    - compressed, streamed…
  - Baking of a variety of simulations results

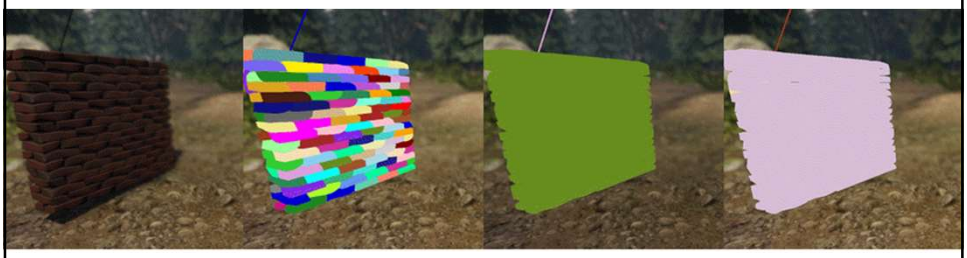
most used file format: .abc

ALEMBIC by imageworks

123

Geom. Caches
(for lack of a better name)

- Baked, optimized animations
  - of the appropriate types including mixtures



Input:
170 Meshes
88400 Verts

as Pre-made Tansforms:
Meshes: 170
Data rate: 0.13 MB/s
Draw calls: 170
(same ones each frame)

as a Blend Shape:
Meshes: 1, with N shapes
Data rate: 4.3 MB/s
Draw calls: 1
(different one each frame)

as a Skeletal Animation
Meshes: 1, w skinning (*)
Data rate: 0.13 MB/s
Draw calls: 1
(same one each frame)
(*) just 1 bone per vertex

Geometry Caches
(a subset of Alembic)

by CRYENGINE

124

Animations
in Unity (+Mecanim) (notes)

- Assets (models, animation, skeletons) imported as formats:
  - fbx, collada
- Animation compression
  - available during import / builds
  - auto reduction of: num of links per vertex, num of keyframes … :
- «Animator Controller» module → deals with:
  - blending between animations: «transitions»
  - compositing animations: «layers»
    - e.g.: a layer overwrites upper body bones
  - and is nicely WYSIWYG (graph visualization)
- Inverse Kinematic: with scripts (`Avatar.SetIKPoistion`)
- Skeletons:
  - way 1: custom (imported as assets)
  - way 2: built-in standard humanoid skeleton provided
    - (~21 ossa)
    - simplified: rigging (predefined constrains), layers (predef. labelling)

125