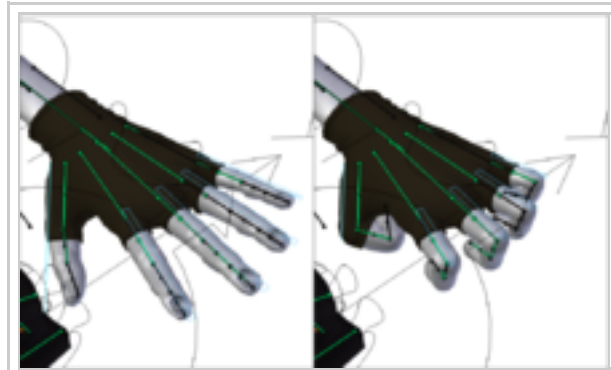


Skeletal animation

From Wikipedia, the free encyclopedia

Skeletal animation is a technique in computer animation in which a character is represented in two parts: a surface representation used to draw the character (called *skin* or *mesh*) and a hierarchical set of interconnected bones (called the *skeleton* or *rig*) used to animate (*pose* and *keyframe*) the mesh.^[1] While this technique is often used to animate humans or more generally for organic modeling, it only serves to make the animation process more intuitive and the same technique can be used to control the deformation of any object — a door, a spoon, a building, or a galaxy. When the animated object is more general than for example a humanoid character the set of bones may not be hierarchical or interconnected, but it just represents a higher level description of the motion of the part of mesh or skin it is influencing.

This technique is used in virtually all animation systems where simplified user interfaces allows animators to control often complex algorithms and a huge amount of geometry; most notably through inverse kinematics and other "goal-oriented" techniques. In principle, however, the intention of the technique is never to imitate real anatomy or physical processes, but only to control the deformation of the mesh data.



'Bones' (in green) used to pose a hand. In practice, the 'bones' themselves are often hidden and replaced by more user-friendly objects. In this example from the open source project Sintel, these 'handles' (in blue) have been scaled down to bend the fingers. The bones are still controlling the deformation, but the animator only sees the 'handles'.

Contents

- 1 Terminology
 - 1.1 Bone
 - 1.2 Transformation
 - 1.3 Bijection
 - 1.4 Translation
 - 1.5 Kinematic pair/Joint/Constraint
 - 1.6 Kinematics
 - 1.7 Inverse kinematics
 - 1.8 Forward kinematics
 - 1.9 Kinematic chain
 - 1.10 Skeleton/Rig
 - 1.11 Rigging
 - 1.12 Mesh/Polygon mesh/Skin
 - 1.13 Skinning
 - 1.14 Vertex
 - 1.15 Vertex weights/Blend weights
 - 1.16 Keyframe/Pose
 - 1.17 Interpolation
 - 1.18 Linear Interpolation
 - 1.19 Quadratic Interpolation

■ 2 Technique
■ 2.1 Benefits and drawbacks
■ 3 Applications
■ 4 See also
■ 5 References

Terminology

Bone

Each bone has a three-dimensional transformation (which includes its position, scale and orientation), and an optional parent bone. The bones therefore form a hierarchy. So moving a thigh-bone will move the lower leg too. As the character is animated, the bones change their transformation over time, under the influence of some animation controller. Skeletal animation is referring to the forward kinematics part of the rig, where a complete set of bones configurations identifies a unique pose.Each bone in the skeleton is associated with some portion of the character's visual representation.In the most common case of a polygonal mesh character, the bone is associated with a group of vertices; for example, in a model of a human being, the 'thigh' bone would be associated with the vertices making up the polygons in the model's thigh. Portions of the character's skin can normally be associated with multiple bones, each one having a scaling factors called vertex weights, or blend weights. The movement of skin near the joints of two bones, can therefore be influenced by both bones. In most state-of-the-art graphical engines, the skinning process is done on the GPU thanks to a shader program. For a polygonal mesh, each vertex can have a blend weight for each bone. To calculate the final position of the vertex, each bone transformation is applied to the vertex position and scaled by its corresponding weight. This algorithm is called matrix palette skinning, because the set of bone transformations (stored as transform matrices) form a palette for the skin vertex to choose from.

Here is a brief outline of this process:^[2]

1. Load the object in the default pose.
2. Create the transformation matrices associated with each bone.
3. For each vertex assign blend weights for each bone. (In practice there will be a limit on the number of weights that can be assigned to each vertex.)
4. For each vertex take the weighted sum of all the bone transformation matrices that affect it and apply the resulting transformation.
5. Animate the object by varying the bone transformation matrices on each frame.

This kind of system involves some elements: The object mesh using weighted vertexes. A skeleton for the mesh consisting of a hierarchy of bone joints. Each vertex of the mesh is associated to a bone (or more than one) and has a weight, these weights determine how much bone movement affects the vertex movement. Each bone in the skeleton is connected to two joints, and each joint is connected to at least one bone. When you have to do an animation, just pre-calculate the key-positions of your skeleton (maybe with a 3D editing program, like Blender or MaYa), then create an algorithm that calculate interpolation between movements. Finally, just apply the interpolation of the vertices basing your calculations on the vertex weights and bone hierarchy. ^[3]

Transformation

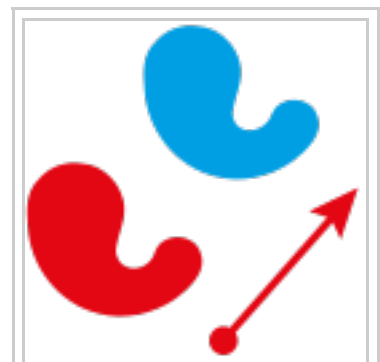
Bijection of a set having some geometric structure to itself or another such set.

Bijection

Function between the elements of two sets, where every element of one set is paired with exactly one element of the other set, and every element of the other set is paired with exactly one element of the first set. There are no unpaired elements. In formal mathematical terms, a bijective function $f: X \rightarrow Y$ is a one to one and onto mapping of a set X to a set Y <https://en.wikipedia.org/wiki/Bijection>

Translation

Function that moves every point a constant distance in a specified direction.[1] A translation can be described as a rigid motion: other rigid motions include rotations and reflections. A translation can also be interpreted as the addition of a constant vector to every point, or as shifting the origin of the coordinate system. If v is a fixed vector, then the translation T_v will work as $T_v(p) = p + v$.
[https://en.wikipedia.org/wiki/Translation_\(geometry\)](https://en.wikipedia.org/wiki/Translation_(geometry))



A translation moves every point of a figure or a space by the same amount in a given direction.

Kinematic pair/Joint/Constraint

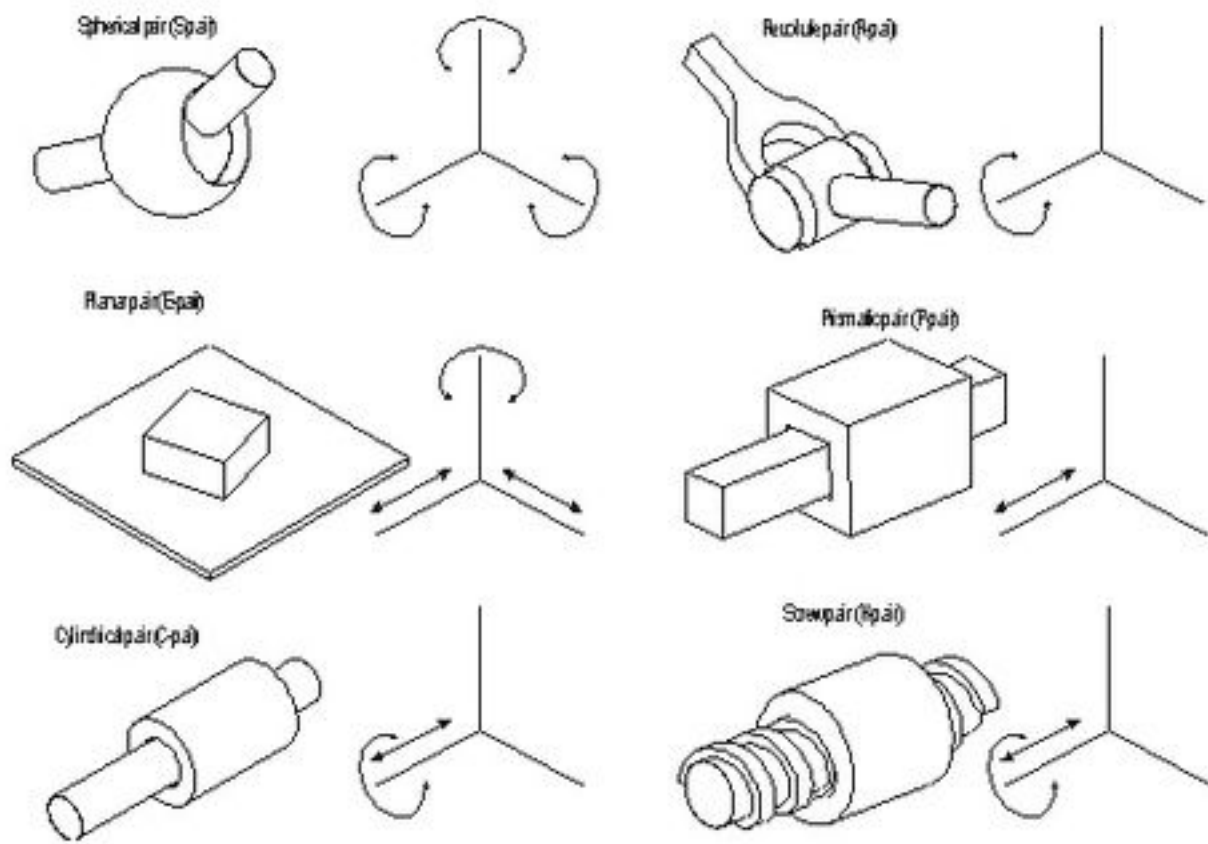
Each bone in the skeleton is connected to two joints, and each joint is connected to at least one bone.
http://content.gpwiki.org/index.php?title=OpenGL:Tutorials:Basic_Bones_System#How_does_a_bone_system_work.3F

Connection between two bodies that imposes constraints on their relative movement.

Lower pair: A lower pair is an ideal joint that constrains contact between a point, line or plane in the moving body to a corresponding point line or plane in the fixed body. A lower pair is one in which there occurs a surface or area contact between two members, e.g. nut and screw, universal joint used to connect two propeller shafts. We have the following cases:

- A revolute pair, or hinged joint, requires a line in the moving body to remain co-linear with a line in the fixed body, and a plane perpendicular to this line in the moving body maintain contact with a similar perpendicular plane in the fixed body. This imposes five constraints on the relative movement of the links, which therefore has one degree of freedom.
- A prismatic joint, or slider, requires that a line in the moving body remain co-linear with a line in the fixed body, and a plane parallel to this line in the moving body maintain contact with a similar parallel plan in the fixed body. This imposes five constraints on the relative movement of the links, which therefore has one degree of freedom.
- A screw pair requires cut threads in two links,so that there is a turning as well as sliding motion between them.This joint has one degree of freedom.
- A cylindrical joint requires that a line in the moving body remain co-linear with a line in the fixed body. It is a combination of a revolute joint and a sliding joint. This joint has two degrees of freedom.
- A spherical joint, or ball joint, requires that a point in the moving body maintain contact with a point in the fixed body. This joint has three degrees of freedom.
- A planar joint requires that a plane in the moving body maintain contact with a plane in fixed body. This joint has three degrees of freedom.

FIGURE 7. There are six lower order kinematic pairs -, spherical, planar, cylindrical, revolute, prismatic and screw can be represented easily in the Physical Markup Language.



Higher pairs:

- **Cam joint** Generally, a higher pair is a constraint that requires a curve or surface in the moving body to maintain contact with a curve or surface in the fixed body. For example, the contact between a cam and its follower. Similarly, the contact between the involute curves that form the meshing teeth of two gears are cam joints, as is a wheel rolling on surface.

Wrapping pair: A wrapping pair is a constraint that comprises belts, chains, and such other devices. A belt-driven pulley is an example of this pair. https://en.wikipedia.org/wiki/Kinematic_pair

Kinematics

Branch of classical mechanics which describes the motion of points, bodies (objects) and systems of bodies (groups of objects) without consideration of the causes of motion. <https://en.wikipedia.org/wiki/Kinematics>

Inverse kinematics

Use of the kinematics equations of a robot to determine the joint parameters that provide a desired position of the end-effector. Movement of a kinematic chain whether it is a robot or an animated character is modeled by the kinematics equations of the chain. These equations define the configuration of the chain in terms of its joint parameters. Inverse kinematics formulas allow calculation of the joint parameters that position a robot arm to pick up a part. Similar formulas determine the positions of the skeleton of an animated character that is to move in a particular way https://en.wikipedia.org/wiki/Inverse_kinematics

Forward kinematics

Use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters kinematics equations of the robot are used in robotics, computer games, and animation. The reverse process that computes the joint parameters that achieve a specified position of the end-effector is known as inverse kinematics. https://en.wikipedia.org/wiki/Forward_kinematics

Kinematic chain

Rigid bodies connected by joints that is the mathematical model for a mechanical system.Mathematical models of the connections, or joints, between two links are termed kinematic pairs.

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

Skeleton/Rig

Hierarchical set of interconnected bones (called the skeleton or rig) used to animate (pose and keyframe) the mesh. Constructing a series of 'bones,' sometimes referred to as rigging. A rig is generally composed of both forward kinematics and inverse kinematics parts that may interact with each other. Skeletal animation is referring to the forward kinematics part of the rig, where a complete set of bones configurations identifies a unique pose.

Rigging

The following is a typical workflow for rigging:

- Add an armature, which starts out with a single bone.
- Add more bones as needed and link them together to form "limbs".

(you can either extrude an existing bone, which parents it to the extruded bone automatically; or add new bones first and then link them together.)

- Edit the bones to give proper proportions to the skeleton
- Apply constraints to the joints

(e.g. a human elbow can bend through about 170 degrees in one plane only)

- Give a 'rest' (default) position to the skeleton.
- Apply a mesh (body) to the armature (skinning)
- Define how the movement of the armature affects the skin

(folding, flexing, bulging)

- Give poses to the armature.

(There are multiple methods: By arranging each bone of the armature manually, or by copying a template armature, or by arranging the bones to follow a curve, or by making the armature follow externally collected motion-capture data.)

- Check how the armature movement affects the skin, and adjust the parameters.

(adjust the topology of the skin to make it look more natural)

<http://wiki.blender.org/index.php/Doc:2.6/Manual/Rigging>

"Rigging is making our characters able to move. The process of rigging is we take that digital sculpture, and we start building the skeleton, the muscles, and we attach the skin to the character, and we also create a set of animation controls, which our animators use to push and pull the body around."

— Frank Hanner, character CG supervisor of the Walt Disney Animation Studios, provided a basic understanding on the technique of character rigging.^[4]

Mesh/Polygon mesh/Skin

Collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling. The faces usually consist of triangles (triangle mesh), quadrilaterals, or other simple convex polygons, since this simplifies rendering, but may also be composed of more general concave polygons, or polygons with holes. Network representations, "streaming" and "progressive" meshes, are used to transmit polygon meshes over a network. Volumetric meshes are distinct from polygon meshes in that they explicitly represent both the surface and volume of a structure, while polygon meshes only explicitly represent the surface. Polygonal meshes are extensively used in computer graphics, algorithms also exist for ray tracing, collision detection, and rigid-body dynamics of polygon meshes. https://en.wikipedia.org/wiki/Polygon_mesh To begin, let's say that a skin is nothing more than a mesh. This mesh is composed by weighted vertexes: each vertex is associated to a bone, and can be influenced more or less by the movement of this bone. Moving a bone, results in a movement of the associated vertexes. So each vertex don't have only coordinates, but also information about a bone and a weight relative to that bone. http://content.gpwiki.org/index.php?title=OpenGL:Tutorials:Basic_Bones_System#How_does_a_skin_work.3F

Skinning

Each bone in the skeleton is associated with some portion of the character's visual representation. Skinning is the process of creating this association. In the most common case of a polygonal mesh character, the bone is associated with a group of vertices; for example, in a model of a human being, the 'thigh' bone would be associated with the vertices making up the polygons in the model's thigh. Portions of the character's skin can normally be associated with multiple bones, each one having a scaling factors called vertex weights, or blend weights. The movement of skin near the joints of two bones, can therefore be influenced by both bones. https://en.wikipedia.org/wiki/Skeletal_animation

Vertex

In geometry, is a special kind of point that describes the corners or intersections of geometric shapes. In computer graphics, objects are often represented as triangulated polyhedra in which the object vertices are associated not only with three spatial coordinates but also with other graphical information necessary to render the object correctly, such as colors, reflectance properties, textures, and surface normals; [7] these properties are used in rendering by a vertex shader, part of the vertex pipeline. [https://en.wikipedia.org/wiki/Vertex_\(geometry\)](https://en.wikipedia.org/wiki/Vertex_(geometry)) In computer graphics is an object, that has certain attributes, one of which is its position. A vertex is a data structure, that describes certain attributes, among which, the position of a point in 2D or 3D space. Display objects are composed of arrays of flat surfaces (typically triangles) and vertices define the location and other attributes of the corners of the surfaces. [https://en.wikipedia.org/wiki/Vertex_\(computer_graphics\)](https://en.wikipedia.org/wiki/Vertex_(computer_graphics)) Submitting vertex data for rendering requires creating a stream of vertices, and then telling OpenGL how to interpret that stream. In order to render at all, you must be using a shader program. This program has a list of expected Vertex Attributes. This set of attributes determines what attribute values you must send in order to properly render with this shader. https://www.opengl.org/wiki/Vertex_Specification

Vertex weights/Blend weights

Portions of the character's skin can normally be associated with multiple bones, each one having a scaling factors called vertex weights, or blend weights. For a polygonal mesh, each vertex can have a blend weight for each bone. To calculate the final position of the vertex, each bone transformation is applied to the vertex position, scaled by its corresponding weight. This algorithm is called matrix palette skinning, because the set of bone transformations (stored as transform matrices) form a palette for the skin vertex to choose from. https://en.wikipedia.org/wiki/Skeletal_animation

Keyframe/Pose

Used in animation, for creating Keyframe/key poses for characters and then inbetweening them in intermediate frames to make the character appear to move from one pose to the next. Pose-to-pose is used in traditional animation, while the parallel concept in 3D animation is inverse kinematics.

https://en.wikipedia.org/wiki/Pose_to_pose_animation Complete set of bones configurations identifies a unique pose. https://en.wikipedia.org/wiki/Skeletal_animation The term key frame comes from cartoon animation. To produce a cartoon, an artist first quickly sketches a rough sequence of frames for animating a character. Rather than draw every frame required for the final animation, the artist draws only the important, or "key," frames. Later, the artist goes back and fills in the missing frames. These in-between frames are then easier to draw, because the prior and subsequent key frames serve as before-and-after references. Computer animators use a similar technique. A 3D artist makes a key frame for each pose of an animated character. Even a standing character may require a sequence of key frames that show the character shifting weight from one foot to the other. Every key frame for a model must use the exact same number of vertices, and every key frame must share the same vertex connectivity. A vertex used in a given key frame corresponds to the same point on the model in every other key frame of the model. The entire animation sequence maintains this correspondence. However, the position of a particular vertex may change from frame to frame, due to the model's animation.

http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter06.html A keyframe literally means a frame which contains a key. Well, suppose you have a time line for your animation: a keyframe represents a special key where the animation changes in a particular way. For example, you want to animate the weaving action, with your hand open that is moved from left to right. The first keyframe (let's say the frame number 0) is your hand on the left which starts to move toward the right. The second keyframe (frame number 5) is your hand on the right, and the last keyframe (frame number 10) is again your hand on the left. As you can see, these frames are "key" since at these points the movement changes radically: between frames 0 and 1, the body status changes from stopped to in movement position. Between frames n and n+1, with $1 \leq n < 4$ the animation don't change so much, because if you take frame 1 and frame 2, the hand is a little moved to the right. Same with frames 3, 4 and 4, 5. Then, with frames 5 and 6, there is another change in the animation, since your hand stops to move toward right and starts to move to left. And finally at frame 10, the animation changes again because your hand stops. So, a keyframe is a particular status at time T in your animation, which means a particular change in the animation course. In our code, keyframes can be implemented using a Keyframe structure, which keeps track of bone status at a specific time. An animation will be just an array of keyframes http://content.gpwiki.org/index.php?title=OpenGL:Tutorials:Basic_Bones_System#Keyframes

Interpolation

Method of constructing new data points within the range of a discrete set of known data points.

<https://en.wikipedia.org/wiki/Interpolation> Key-frame interpolation assumes that the number and order of vertices are the same in all the key frames for a given model. This assumption ensures that the vertex program is always blending the correct pairs of vertices. http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter06.html This is why nearly all direct animation is done using interpolation. The idea is simple: you define a few Key Frames, which are multiple frames apart. Between these keyframes, the properties' values are computed (interpolated) by Blender and filled in. Thus, the animators' workload is significantly reduced. For example, if you have: a control point of value 0 at frame 0, another one of value 10 at frame 25, linear interpolation, then, at frame 5 we get a value of 2. http://wiki.blender.org/index.php/Doc:2.6/Manual/Animation/Editors/Graph/FCurves#The_concept_of_Interpolation Before I said that with boned animation, you don't have to store all the frames of your animation, but just the keyframes. Alright, now we know what a keyframe is, but the question is: if I've only keyframes, how can I animate the skeleton? The answer is: calculating the other frames using interpolation (or linear interpolation, to be more precise). Keyframes represent a big change in our animation, but other frames don't! This mean that we can guess - knowing some starting conditions - frames between two keyframes. Our guess is done by interpolating the keyframes. Different interpolation algorithms exist, and in some cases you may prefer to use one instead of another. For example, if you know that your animation starts slow to get faster, you may prefer to use a exponential interpolation, or if you know that your animation starts slow, get velocity and then return to get slower, you may want to use a sine interpolation. But in the common case, we prefer to use linear interpolation, that means that our animation don't change speed http://content.gpwiki.org/index.php?title=OpenGL:Tutorials:Basic_Bones_System#Keyframes

Linear Interpolation

With linear interpolation, the transition between positions happens at a constant rate.

http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter06.html

Quadratic Interpolation

Linear interpolation is good for many situations, but sometimes you want the rate of transition to change over time.

For example, you might want the transition from positionA to positionB to start out slowly and get faster as time passes. For this, you might use quadratic interpolation

http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter06.html

Technique

This technique is used by constructing a series of 'bones,' sometimes referred to as *rigging*. Each bone has a three-dimensional transformation (which includes its position, scale and orientation), and an optional parent bone. The bones therefore form a hierarchy. The full transform of a child node is the product of its parent transform and its own transform. So moving a thigh-bone will move the lower leg too. As the character is animated, the bones change their transformation over time, under the influence of some animation controller. A rig is generally composed of both forward kinematics and inverse kinematics parts that may interact with each other. Skeletal animation is referring to the forward kinematics part of the rig, where a complete set of bones configurations identifies a unique pose.

Each bone in the skeleton is associated with some portion of the character's visual representation. *Skinning* is the process of creating this association. In the most common case of a polygonal mesh character, the bone is associated with a group of vertices; for example, in a model of a human being, the 'thigh' bone would be associated with the vertices making up the polygons in the model's thigh. Portions of the character's skin can normally be associated with multiple bones, each one having a scaling factors called vertex weights, or blend weights. The movement of skin near the joints of two bones, can therefore be influenced by both bones. In most state-of-the-art graphical engines, the skinning process is done on the GPU thanks to a shader program.

For a polygonal mesh, each vertex can have a blend weight for each bone. To calculate the final position of the vertex, each bone transformation is applied to the vertex position, scaled by its corresponding weight. This algorithm is called matrix palette skinning, because the set of bone transformations (stored as transform matrices) form a palette for the skin vertex to choose from.

Benefits and drawbacks

Strengths

- Bone represent set of vertices (or some other objects, which represent for example a leg).
 - Animator controls fewer characteristics of the model
 - Animator can focus on the large scale motion.
 - Bones are independently movable.

An animation can be defined by simple movements of the bones, instead of vertex by vertex (in the case of a polygonal mesh).

Weaknesses

- Bone represents set of vertices (or some other object).
 - Does not provide realistic muscle movement and skin motion
 - Possible solutions to this problem:

- Special muscle controllers attached to the bones
- Consultation with physiology experts (increase accuracy of musculoskeletal realism with more thorough virtual anatomy simulations)

Applications

Skeletal animation is the standard way to animate characters or mechanical objects for a prolonged period of time (usually over 100 frames). It is commonly used by video game artists and in the movie industry, and can also be applied to mechanical objects and any other object made up of rigid elements and joints.

Performance capture (or motion capture) can speed up development time of skeletal animation, as well as increasing the level of realism.

For motion that is too dangerous for performance capture, there are computer simulations that automatically calculate physics of motion and resistance with skeletal frames. Virtual anatomy properties such as weight of limbs, muscle reaction, bone strength and joint constraints may be added for realistic bouncing, buckling, fracture and tumbling effects known as virtual stunts. However, there are other applications of virtual anatomy simulations such as military^[5] and emergency response. Virtual soldiers, rescue workers, patients, passengers and pedestrians can be used for training, virtual engineering and virtual testing of equipment. Virtual anatomy technology may be combined with artificial intelligence for further enhancement of animation and simulation technology.

See also

- Computer animation
- 3D computer graphics
- Forward kinematic animation
- Inverse kinematics
- Morph target animation
- Interactive skeleton-driven simulation
- Cutout animation

References

- ↑ Soriano, Marc. "Skeletal Animation" (http://alumni.cs.ucr.edu/~sorianom/cs134_09win/lab5.htm). Bourns College of Engineering. Retrieved January 2011.
- ↑ https://www.opengl.org/wiki/Skeletal_Animation
- ↑ http://content.gpwiki.org/index.php?title=OpenGL:Tutorials:Basic_Bones_System#How_does_a_bone_system_work.3F
- ↑ Wloszczyna, Susan (November 26, 2013). "With Frozen, Director Jennifer Lee Breaks the Ice for Women Directors" (<http://blogs.indiewire.com/womenandhollywood/with-frozen-director-jennifer-lee-breaks-the-ice-for-women-directors>). *Indiewire*. Retrieved 8 January 2014.
- ↑ "Defense" (<http://www.santoshumaninc.com/industries/defense/>). Santos Human Inc. Retrieved January 2011.

Retrieved from "http://en.wikipedia.org/w/index.php?title=Skeletal_animation&oldid=635354642"

Categories: Computer animation | Computer graphic techniques | 3D computer graphics | Animation techniques | Anatomical simulation

- This page was last modified on 25 November 2014 at 08:20.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply.
By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.