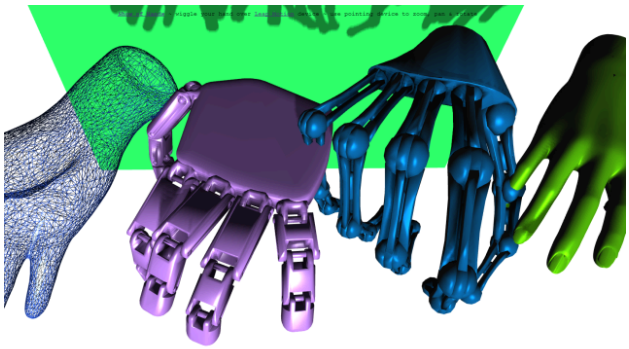
Skeleton API Considerations for Leap Motion Devices R3

Rev 3 ~ 2013-08-08 ~ update of demo software page 5
Rev 2 ~ 2013-08-04
Prepared by Theo Armour ~ 2013-07-31

TL;DR /Abstract / Executive Summary



Screen capture of Leap Motion demo code with seven hands

Are there business and use cases for updating the Leap Motion APIs to add or incorporate procedures that aid character gaming/animation/research relating to skins/meshes and to skeletons/bones/armatures? A brief investigation into skinning has found that most any significant work/need/requirement may be accomplished with the API as is. Adding objects that look like human hand, creature claws and robot appendages is doable, viable and fun.

A brief investigation into the capture of motion data and exporting into industry standard formats has been carried out. Leap Motion data may be saved and exported as BVH files. The files may be replayed as part of character animations and could be used to enhance video games, teach sign language or complex part assembly or enhance/create the next generation of human computer interface/experience.

Both of these aspects - using Leap Motion data for skinning and rigging - would benefit hugely by the addition of coding examples, libraries of sample assets and live demo files.

The development of the Leap Motion API should very much continue in its current direction - exploring a new tools and bringing out previously unthought of outcomes. Of course the API will eventually be enhanced to support even better rigging and meshing. The near term goal, however, should be streamlining and rationalizing the APIs [we are looking at you: JavaScript API].

Contents

Cause For writing

The idea for this paper came via a chance conversation in the Leap Motion offices. It is an FYI/white paper/thought experiment. I have been a 3D person all my rather long life and seek to promote/praise/promulgate 3D thinking wherever I sense it.

Nomenclature

An issue is that there are a lot of variations on the term ‘skeleton’ with regards to motion capture and animation. You could be talking about ‘skeletons’, ‘bones’, ‘armatures’, ‘rigging’ and probably several several other terms. ‘Stick figures’ comes to mind. Please note that while the adult human body has 206-208 bones, the rigs that animators use tend to simplify greatly and incorporate somewhere between twenty to fifty separately movable objects.

The other part to this is what do you put over the ‘bones’? Often these are termed ‘skins’ or ‘meshes’. Meshes can be huge and incorporate vector, raster and programming data. Issues

with representing include bones popping through skin, meshes self intersecting, improper folding or displacement.

This paper - being an overview and a thought paper seeks the broadest least specific meaning.

Business and Uses Cases

What are possible use cases or business cases for a skeleton API for the Leap Motion Control device? Some guesses:

- To record motion capture for use in cinema, animation and game development.
- To aid with medical diagnosis or education use in comparing patient/student gestures with libraries of pre-captured gestures in order to determine things such potential repetitive motion syndrome issues or
- To assist with developing robotic motion libraries.
- To teach humans how to perform complex operations such as watch assembly or surgical techniques

All of these uses might well be in conjunction with existing full body motion capture or for use as attachments to existing motion capture files.

There could be a couple of possible scenarios. The easiest would be to have the recorded JSON file converted and exported into BVH format.

The second and more complicated effort would be to have an app which would display BVH motion capture as in real time.

Current Demand Leap Motion Forums

Having had a quick look on the forums I see only a couple of references to a skeleton API, so there's not that much of a pent up demand for such an API.

There has been a fair amount of chat on sign language - which would really benefit with access to motion capture files in order to build up a corpus of data to be used for pattern recognition.

Nevertheless the development of such an API could have interesting downstream consequences.

Skinning/Meshing

The Leap Motion API documentation frequently displays a 3D mesh representation of a human hand as part of the illustrations used to demonstrate API concepts. Interestingly none of the sample code or demo examples (that I have seen so far) appear to include such a device.

I believe that the lack of displaying such a device has been mentioned in forums and reviews (though I have no specific links at hand). In a recent casual conversation at the Leap Motion offices it was wondered whether the Leap Motion API could and should be revised or updated in order to assist developers with displaying hand-like pointers in their apps.

The question therefore is:

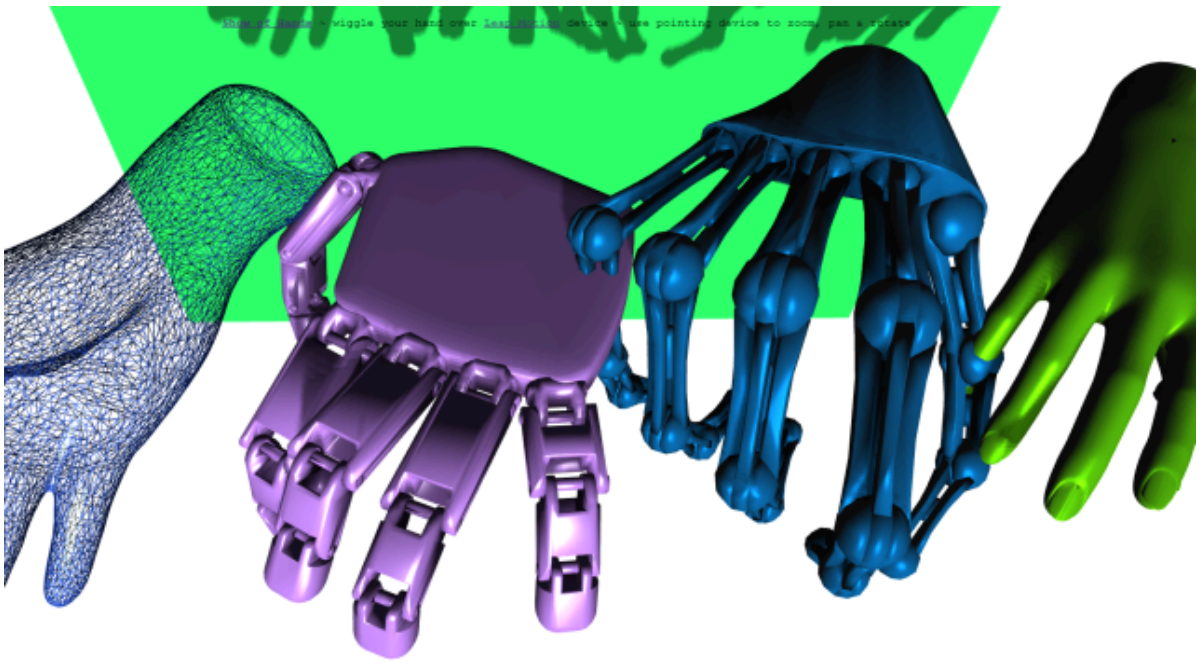
What new tasks could be accomplished or what fresh needs could be met regarding the display of skins and meshes with suitable updates and revisions to the Leap Motion API?

In turn this begs the question:

*What can you *not* do today when using the Leap Motion API to display skins and meshes of hands, claws and appendages?*

The first question seemed as if it could be difficult to answer as it is so open ended. The second question appeared easier for me to try to answer. All I needed to do is figure out: What can I code right now?

Kindly have a look a this JavaScript appt:



http://jaanga.github.io/gestification/show-of-hands

This demo will take a few seconds to load the seven hands

This is a simple coding demo that displays seven hand skins that move as you move your hand over the Leap Motion device. You may also use your mouse or touchpad to zoom, pan and rotate the display. A computer with browser that supports WebGL and has a fast GPU is highly recommended.

As of this writing, the demo only supports one hand and fingers do not move/update. Only the hand updates.

The demo took a few hours to build. Virtually all the time was taken on sourcing the models and converting them into JSON data.

As a result of coding this demo, I have the following thoughts.

- Displaying skins of hands and things that load and update in real-time in Leap Motion apps is easy-peasy. Only 3D has been looked at. 2D should be easier.
- Adding finger movements will not be that much of a challenge.
- Robotic/mechanical devices where each joint is an individual object should pose few issues. (I plan to look into this ASAP)
  Update 2013-08-05
  See http://jaanga.github.io/gestification/work-in-hand/phalanges/
  Most issues dealt with. But the biggie is trying to figure out which Leap Motion pointable data should be assigned to which finger in the model. Made more difficult because there are fewer Leap Motion tools available in JavaScript than in Java.
  Update 2013-08-08
  See: http://jaanga.github.io/gestification/yougest/r4/
  This demo shows support for ten fingers with two joints per finger with first joint being common to the five fingers of each hand.
  It appears to be very similar to the finger functionality displayed in the 'Leap Motion Visualizer' app where there is a common first joint on each hand in the position of the larger yellow globe. The visualizer app does show a second joint that appears to be calculated as the running from the two previous vectors. Too long to explain in detail - code will be in next revision.
- Adding stretchy skins is certainly doable (but may be beyond my pay grade).

All of this could be done via enhancing the Leap Motion SDK with coding examples, sample models and demo apps.

I can see three stages of SDK improvement in the near term:

Release 1

- A dozen or so 2D/3D models added to the SDK in, say, STL, FBX, Collada formats with basic colors and materials.

- Sample code that uses the hands to carry out several tasks
- Working demos with source code.
- Fingers move but each joint is a separate object

Release 2

- More models with greater choice of materials - including shaders
- Unified skins and meshes that stretch and twist with human-like tendencies

Release 3

- Demo code that allows you to design, build save your own appendages

Skinning Considerations

So far, I can see a ton of things that I can do or get to work with today's API.

There may well be some beautiful and useful things that could done with an enhanced API, but you will know much more about that once you have had some time building on what you have already.

Skeletons/Bones/Armatures

In a process similar to the skinning questions, I asked myself:

What can you do with the current Leap Motion API in terms of capturing motion data, saving and exporting data to industry file formats?

And, again, the best response was to see what I could code.

Currently I have two apps in the works that are unfinished. The first captures, records and saves Leap Motion frame data. The second is a start at reading and writing BVH files (see following section). These apps were easy to write and here is a fair amount of support out there already for this kind of thing. I plan to complete the app in the near future.

Again there appear to be few if any limits to this process that are in any way due to a limitation of the current Leap Motion API.

In principle it should be possible to build apps that output Leap Motion data translated into BVH data in real-time or as data to be saved in text files.

Using this files it will be possible to build up a corpora of gestures and signs that could be in turn used to vastly improve gesture recognition and assist with many other cases.

Skeletons/Bones/Armatures Conclusions

As I mentioned in the previous section, I think there is a lot already availavle in the Leap Motion API that has not yet been utilized in apps and demo code that I have seen so far.

There may well be awesome things that could be added to the API that would significantly enhance motion capture - but itś hard to say what those things should be until the limits of the existing API are tested.

Motion Capture File Formats

When thinking about skeletons in the structure of animations, it is usually the sake of formats such as Collada, FBX and some of the other fairly standard file formats such as 3DS, or OBJ. None of these formats, however, or specifically tailored for motion capture.

There are a number of file formats that are specifically tailored for the storage of motion capture.

Probably the two most popular or BVH and C3D. C3D is digital and quite complicated. BVH is ASCII and quite straightforward. I would very much recommend the usage of BVH format in a skeleton API.

http://en.wikipedia.org/wiki/List_of_motion_and_gesture_file_formats

http://www.c3d.org/

http://en.wikipedia.org/wiki/Biovision_Hierarchy

Further justification that BVH and C3D are the two most popular motion capture file formats. These are the two formats mentioned in a serious Blender tutorial. See: http://wiki.blender.org/index.php/Doc:2.4/Tutorials/Animation/Advanced/MoCap/3:Practical_example

The handy thing about BVH is that you can easily use it with apps like Blender and then with Blender you can translate into many other proprietary formats.

glTF as a BVH Alternative

As of this writing the Khronos Group - overseers of the Collada format - is working on another format tailored to online transmission.

See: https://github.com/KhronosGroup/glTF/blob/master/specification/README.md

In due course I will try to ascertain whether glTF might be a more modern more attractive motion capture solution.

References of Possible Interest

Plenty more to be added if anybody is interested...

Recent NY Times Article on the 250 Italian Gestures
http://www.nytimes.com/2013/07/01/world/europe/when-italians-chat-hands-and-fingers-do-the-talking.html

Recurring Structures in Gestures
http://www.academia.edu/294806/New_insights_into_the_medium_hand_-_Discovering_recurrent_structures_in_gestures

Human hand bones
http://en.wikipedia.org/wiki/File:Scheme_human_hand_bones-en.svg

BVH & Motion Capture

http://www.cgspeed.com/
Much info. See link to BVH player

http://www.bvhacker.com/
Even better BVH player

http://www.akjava.com/demo/bvhplayer/
Play BVH files in your browser

http://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html
Good overview - as identified by Gabriel

http://www.mmeredith.staff.shef.ac.uk/fileformats/mocapff.pdf
Motion capture formats explained - also sourcecode

https://code.google.com/p/papervision3d/source/browse/trunk/as3/trunk/src/org/papervision3d/objects/parsers/mocap/BVH.as?spec=svn938&r=938
Source for reading BVH files for Papervision

http://www.blendernation.com/2011/05/27/motion-capture-data-bvh-tools-and-files/
Good portal as to what is out there

BVH File Sources

http://www.mixamo.com/motions
Excellent source for motion files

http://www.animstreet.com/
Another great source - search on hands

http://www.turbosquid.com/Search/Motion-Capture/finger/bvh