CMPUT 414

Literature review

Team A+
Chris Wang
Di Meng
Ghazal Jangani
Hong Wang
Nsaka Peter

Feb 24th, 2016

Abstract

This paper reviews different methods for integration of hands and body animation, and present another method by using Leap Motion Capture. We use a small USB device called Leap, which is able to track human's hands and fingers motions, and it is able to keep track of the data that has been recorded through Leap motion, and build our own hands models. The paper describes how this technique were applied to build our hand motion into the 3D animation, and compare this method to other results and studies at the same area.

1. Introduction

In the past few decades, 3D animation technologies had brought us significant changes in our daily life. Movie industry, in a very large scale, had benefited a lot from the revolution of 3D animation technologies. From 90s' movie *Toy story* to today's *Transformers* and *Avatar* we can see the 3D animation technologies undergo dramatically improvements in a short period of time. Comparing *Toy story* and *Avatar* we found out the animation of *Avatar* had better facial expression, hand and finger motions which provide audience strong realistic tastes. However, build up 3D model and implement animation onto hand and fingers could consider a challenge task for developers since there are 27 bones on each hand and every bone has its own animation data when hand moves. Because of this difficulty, many human animation models do not implement proper animation details of hand and fingers onto their skeleton. Our goal is to find out a solution to improve the animation of hand and fingers. Moreover, we reviewed different studies and methods, and we found our own solution. In our solution, the main focus is on capturing 3D animation of hand and fingers and convert the recording data into any given hand model. Then we can combine the hand model that generated in first step into the full body skeleton by using a small piece USB device called Leap. This device is able to track human's hands and fingers motions. Therefore, we are able to keep track of the data that has been recorded through Leap motion, and build our own hands models. Also, to be more specific it is important to simplify both hands into bones and joints, so we can use points to represent joints of hand and lines for bones. Then, we can map our data by using Motion Builder to generate all the captured motion that we observed, and modify the data until the whole motions looks natural. Finally, we can rescale hand models and fit into human model.

2. Related work

2.1 Capture data

Data capture is the first step of the project, and it is extremely significant for the entire project. The data need to be precise and reliable, otherwise, it will affect rest of the project. In order to capture accurate data we should consider and follow three steps; first, we need to have a good understanding of the objects that we want to capture, which in this project are hands and fingers. Then, we need to analyze the input devices, different input device use different method to capture data; we will find the best device for our project. Finally, we will explore the application of data capture.

Hands and fingers analyzing

By reviewing Modeling of Human Hand Link Structure from Optical Motion Capture Data, which is about the structure of hands and fingers [1], and as *Figure 1* shown below. We could divide a hand into four different parts. From fingertips to the first bone of fingers, this part is called **distal phalanges**, and from first bone to second bonds is called **middle phalanges** or **intermediate phalanges** and from second bone to third bone is called **proximal phalanges**. And for the palm part is all together called **metacarpal bones**. The method that author used is optical motion capture. They put the marks on these bones, and then use optical cameras to capture the motion of hand, through couple of different experiment and they found that the marker has to put on exact joint, more markers they put on the more accurate result they will get, and the basic joints cannot be ignore, otherwise there will have errors[1].

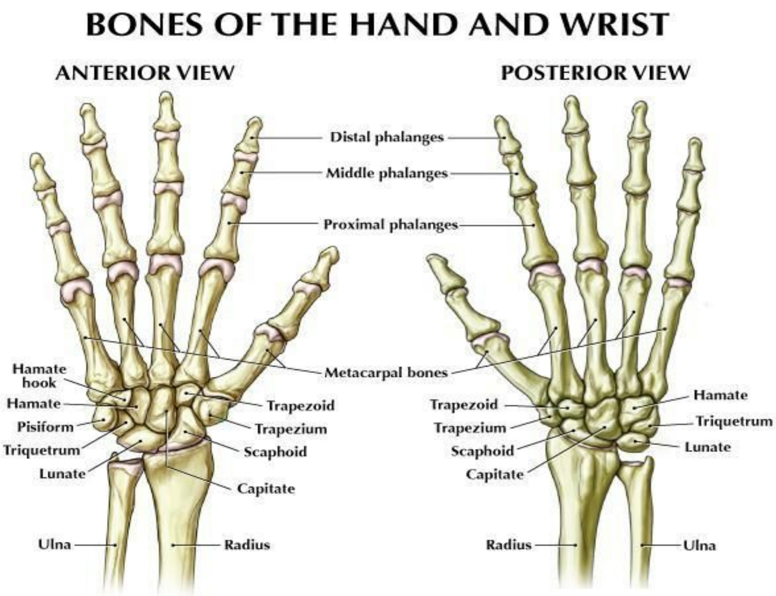


Figure 1 Hand structure.
http://humananatomybody.info/wp-content/uploads/2015/09/bone-anatomy-of-hand-and-wrist-picture-qBVG.jpg

Input devices

There are a plenty of input device could be use for hands and fingers motion capture. Knowing from lecture, there are four types of method to capture data, which are prosthetic, electromagnetic, acoustic, and optical. The most popular input devices are basically use optical method to capture data. The recent optical motion capture technology has great implementation, comparing with last generation technology. With recent technology, users are no longer requiring to put maker on the certain joints the system will automatically recognize every joints from fingers. There are two products we could found on the market, which are Kinect controller, and Leap motion controller.

In another paper, Hand Gesture Recognition with Leap Motion and Kinect Devices, researchers compared Leap motion controller and Kinect controller. They first, indicate the purposes of using the two devices [2] that they applied in their project. Leap motion controller, which is designed for hands and fingers motions capture only, and Kinect controller that not only allow the user to detect hands and fingers but also it is able to perform full body motion capture. Second, they talked about the different data types between these two devices, as figure 2 shown below. Leap motion able to capture not only 3D positions of each joints, but also the distance and angles from hand centers. However, Kinect seems have less joints able to capture comparing with Leap motion. Finally, author did an experiment, which is gesture recognition by Leap motion and Kinect. The researchers perform American Sign Language to these two devices at same time; the result shows that Leap motion is about 70% more precise than Kinect.

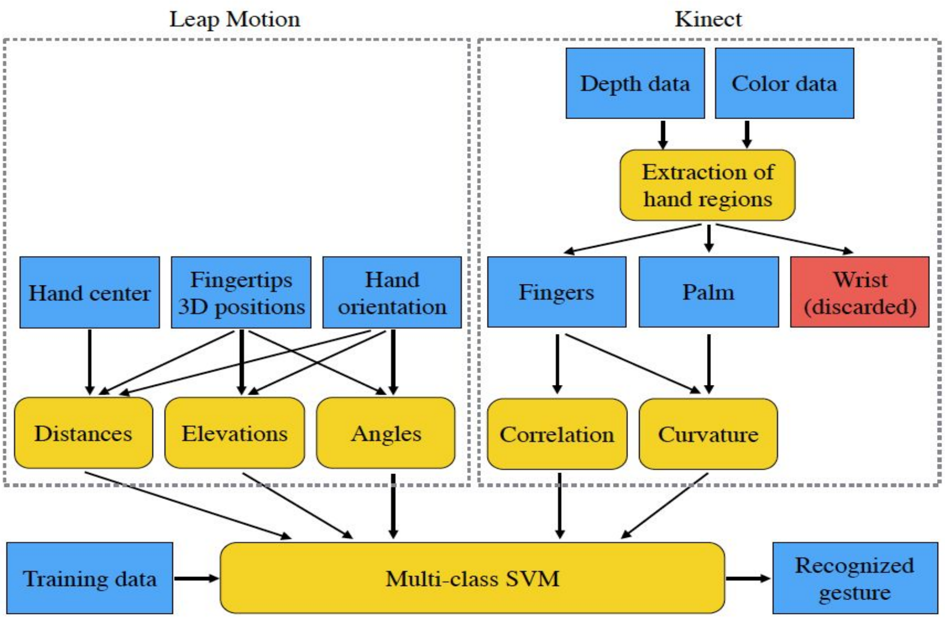


Figure 2 Data type of Leap Motion and Kinect. *Hand Gesture Recognition With Leap Motion And Kinect Devices*

Even though Leap motion controller is more reliable and precise than Kinect controller, put we still not sure how accurate it is. In [3], researchers did experiment from static measurement and dynamic measurement. Static measurement means the hand model is still and then uses Leap motion to record the data. And for dynamic measurement is constantly

moving the stick and then capture the data. There are over thousands sample have been captured and by analyzing the data, researchers conclude that in static measurement, the standard deviation is less than 0.5mm, however, in dynamic measurement for the moving object, the result will be less precise when the object is 250mm above the device and as well as the frequency of the moving object increase.
Beside the Leap motion capture controller and Kinect controller, there is another interesting device called “color glove”. In this article, researchers developed their own device by using glove, which has different color in different part, as figure 3 shown below [4]. We can see the color changes at joint points, which is similar concept with the marker placed on the joint points. This system able to 2D and 3D scenarios and more application like virtual surgery and assembly could be developed based on the “color glover” system.
Figure 3 "Color glove". *Real-Time Hand-Tracking with a Color Glove*

Application of hands and fingers motion captures
After we capture the data, we could use this data to do the animation of hands and fingers same as the method we want to apply in our project. Another approach is gesture control system, where some researchers point to the potential of gesture control device [5]. Like voice control, gesture control is another approach that naturally come from human being. However, unlike voice control there is language barrier, with gesture control, people can develop their own gesture to do the operations. Since everything is try to be smarter than before, for example, smart phone, smart TV, smart lock, and also smart light, with gesture control, we can easily operate these devices. When people interact with machine, we prefer the most natural way to communicate with them, so in the future, there will be a combination of voice control, gesture control, and eye-tracking to make one piece of device to communicate with other machine.

2.2 Convert Data

As long as we obtain the data from the leap motion capture, we need to move our steps forward and consider how to utilize the data and finally generate the animation files.

When we finish first step, we need to combine all the data captured by Leap with the timeline into a whole file. In the last step, we would be able to generate an animation file which would be able to display and modified through Motionbuilder.

Json, as a “text format for the serialization of structured data” [6], would consider as good intermedia format for us to store the animation motion capture data in the first step. “It advocates generally cite its human-readability and the ease of parsing as the main advantages” [7] On the other hand, Json file would be compatible and easily modified by any mainstream programming languages which include python. As the Leap API mentioned, the data we capture would be order by bones and joints with their own orientation and coordination details. Thus using Json would be efficient enough to take the position of intermediate data format since all data stored in hierarchy.

However, our ultimate goal of this project is combine the hand animation into human skeleton. Thus we need to have another 3D animation that is able to be read by our software. In “Motion Capture File Formats Explained”[8], the author lists a table of the animation formats which could be used in our final step. There are two commonly used file format which we are going to use are named BVH (BioVision Hierarchical Data) and HTR(Hierarchical Translation-Rotation). BVH is “largely displaced an earlier format Biovision providing skeleton hierarchy information as well as motion data”[9]. BVH file format have 2 separate part of its structure. Firstly it has a section which contains the hierarchy information of every bone as well as the initial position of skeleton. Second section contains the channel data for every frame (small unit of every displayed bone). Later on, HTR was created as an alternative format to the BVH because of BVH’s lack of specification. Comparing, with BVH Contain all necessary details of movement which including “lots of flexibility in data types and ordering and it has a complete basis pose specification where the starting point for both rotations and translations are given” [10]. In addition, it also contains multiple sections: header, segment names hierarchy, based position and motion sections. Either way, both formats can efficiently store the data of skeleton movements and other information.

We also need to pay attention on the fact that we need to consider time as an element of animation. Thus, when we recording the movements and coordinate data we need to add another section named time into Json file. While we generated our Json file, there are many online converting tools for us to transform the Json file into BVH or HTR format. Later on we can combine our hand animation onto our hand model in the next step.

In a summery, we will write some python and parse our data into a structured neat Ison form. After that, we will use the online tools like *Jaanga* to have our final BVH or HTR files.

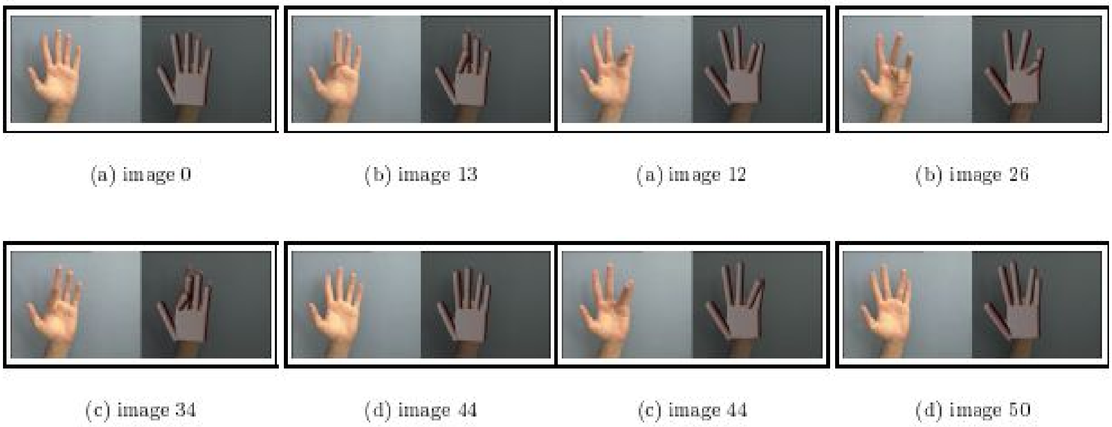
2.3 Improve hand models

Based on current hand models composed of joints and bones, there are two improvements can be conducted. The first is forcing realism and compatibility of hand modelling and animation by applying biomechanical constraints to fingers. The second is wrapping skeleton of hand with skin to make animation more intuitive.

Use biomechanical constraints on human hands to reduce error

Fingers movements are governed by static and dynamic biomechanical constraints [11] and these constraints will help reduce error of mapping data into hand. To be more specific, among all constraints pointed by Kuch and Huang, the most important limitations are interdependence between adduction and extension of a finger and interdependence between adjacent fingers.

To conduct such constraints into hand models, Hocine and Patrick measure angles and lengths of fingers by using non-overlapping method [12] that let experimenters sketch fingers to limitations and keep them in the same plane. Such method helps rescale model according to real hand and also helps to solve the problem that the single view provided by sensor may cause false location on other views.

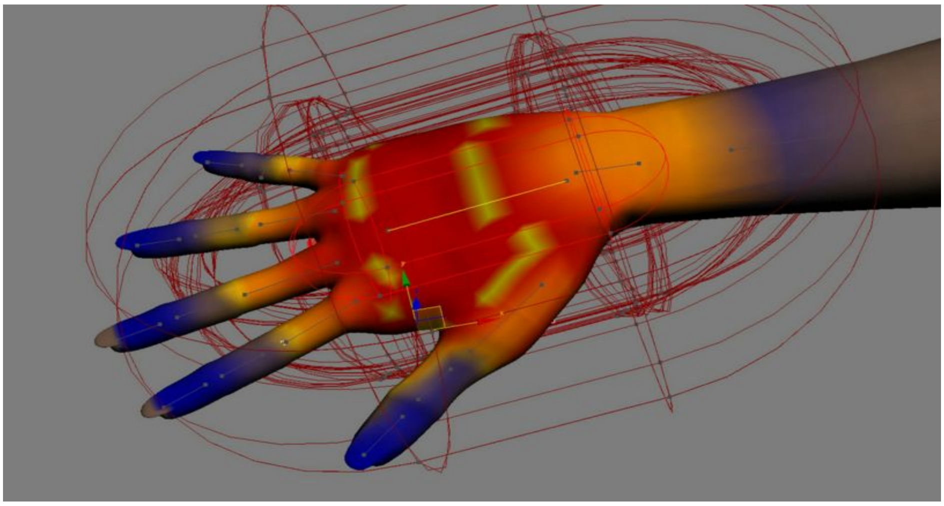


non-overlapping method by Hocine and Patrick

In order to transform static data into dynamic sequence, Franck enables researchers to calculate motions that can take continuous external forces into account and compute location of joints by kinematics [13][14].

Beautify model by wrapping skeleton with skin

To approach this goal, motion builder provides several plugin character models and skeleton, which enables users to control model numerically [15]. Also, by using modifiers in 3ds max, users can relates skin and skeleton more tightly and accurately.



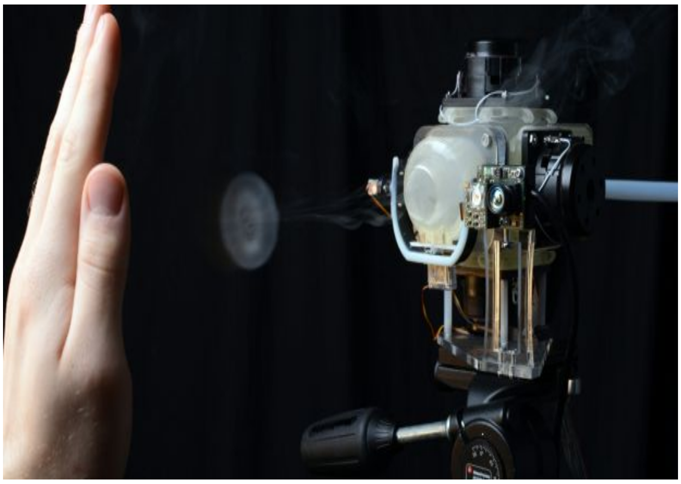
3ds max skin modifiers(envelope)
http://3d-character-animation.blogspot.ca/2010/05/skinning.html

2.4 Mapping data into hand model

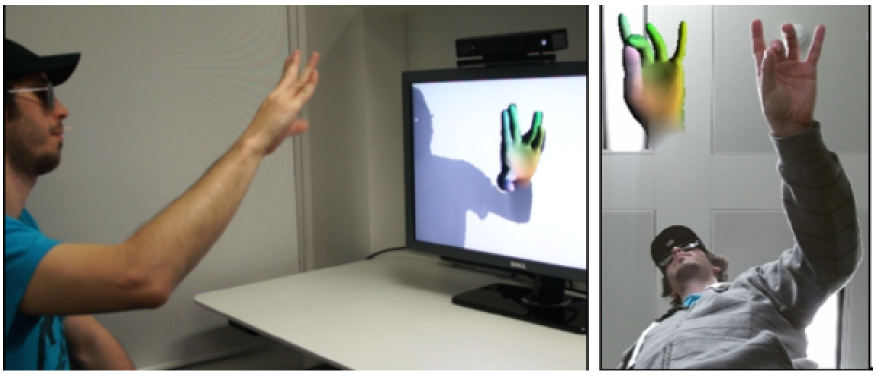
Hand tracking, in general, is a complex and abstract aspect of artificial intelligence that makes use of numerous algorithms and the principles of mathematics and physical sciences to bring real-time interpretation of hand movements, gathered as data and processed into tangible user input. It has been a part of an extensive and lengthy researching efforts of many individuals, especially physicists and scientists, in the past few decades when the computer generation boomed alongside the discovery of artificial intelligence and the birth of virtual reality. Back then, hand tracking is a novel technology only dreamt of by many. Just like virtual reality, it is a futuristic concept that had a specific audience. Therefore, to build and model a hand, and to track all its movement and interactions with objects in real world space, virtual reality, and combining these two into a fluid and seamless experience can be a real challenge. It is also an incredible opportunity to experiment and create in ways that previously existed in the pages of science fiction.

The following products that are in constant development have been introduced publicly and promises to integrate true hands-enabled input with VR.

- *"Aireal is a haptics technology co-developed by Rajinder Sodhi from the University of Illinois and Ivan Poupyrev from Disney Research."*[16] It aims to bring haptic feedback by pulsing out short bursts of air rings – these tiny puffs of air can be felt as textures of any virtual reality content based on the user's interactions. "Aireal is a system made of five modules that can be attached to any source, including computers and virtual reality headsets, and is positioned around the user".[16] As the user moves, the modules track movement and provide haptic feedback by sending out bursts of air that form vortices. These vortices are what rings of air are made of, and these are felt by the user as haptic feedback.



- *"Microsoft Hand Pos, developed by Microsoft, Hand Pose is a cutting-edge hand tracking technology that uses the Kinect motion-sensing camera to detect articulated hand movements in space in real time."* [17] What sets apart Hand Pose from other haptics technology in development is its high degree of robustness in tracking errors, and its fast data processing compensates for these possible errors. Thanks to its utilization of the Kinect camera, "it can track hand movements within the camera's range of view or focus, whether the hand is far or near the camera".[18] The *Hand Pose* is currently in development stage, with the Xbox One's advanced Kinect camera being used as a prototype device for this haptics technology.

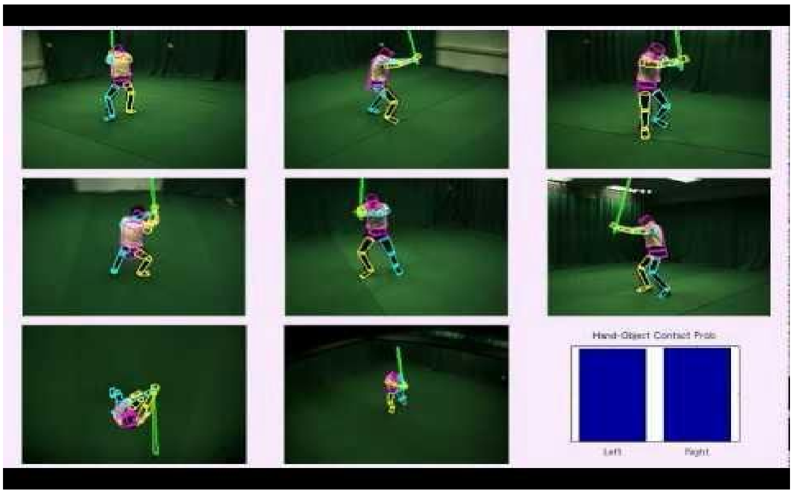


- There are some other companies that present different devices to track hand motion like, *"MindMaze",* Swiss neuro technology company, that unveiled its one-of-a-kind technology that benefits virtual reality [20]. It's called the MindLeap, which is a technology that reads neural signals – it reads the user's mind – to perform certain tasks. Not only the MindLeap is able to use the user's brainpower, it can also track hand movements using advanced tracking sensors. So, this hand and finger tracking device has "high speed tracking up to 120 FPS. High resolution gesture recognition" [20]. However they could not finish their job because the lack of funding.

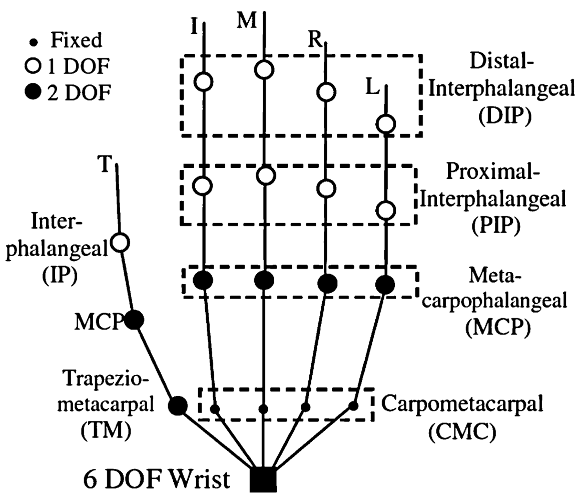
- "*Tracking People Interacting with Objects"* is what students from Brown University has worked on it, and their key idea was to articulated 3D tracking of humans can be enhanced by taking into account knowledge about the pose of objects in the human's hands. Based on what they noted, tracking of rigid objects is an easier task than articulated human tracking; objects in the human's hands can then be tracked independently of the human and then be used as "natural markers," [22]giving cues about hand pose. As described "a standard formulation of human 3D tracking [22],

$$p(\alpha_t \mid D_{1:t}) \propto p(D_t \mid \alpha_t) \int S\alpha \, p(\alpha_t \mid \alpha_{t-1}) p(\alpha_{t-1} \mid D_{1:t-1}) d\alpha_t$$

,so they used an annealed particle filter with a linear temporal update model and a background difference likelihood"[22]. Then, they inferred hand-object contact points by imposing kinematic constraints on the human model that encoded in the temporal update model using a rejection sampling-based approach.



<u>*Skeletal Structure of The Hand*</u>



Human hand consists of 27 bones; consequently there are 27 degrees of freedom (DoF) . Each of the four fingers has four DoF. The distal interphalangeal (DIP) joint and proximal interphalangeal (PIP) joint each has one DoF and the metacarpophalangeal (MCP) joint has two DoF due to flexion (F) and abduction (AA). Flexion refers to bending of fingers with one DoF and abduction refers to lateral movement of fingers with one DoF. The thumb has a different structure which has five degrees of freedom, one for the interphalangeal (IP) joint, and two for each of the thumb MCP joint and trapeziometacarpal (TM) joint both due to flexion and abduction. These all add up to 21 DoF. The remaining 6 degrees of freedom are from the rotational and translational motion of the palm with 3 DoF each. These six parameters can be ignored, since we will only focus on the estimation of the local finger motions rather than the global motion.

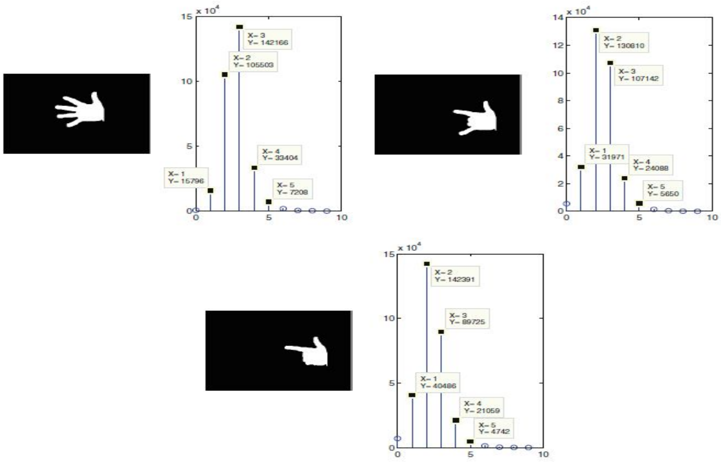
Hand Model Reduction via Constraints

Hand constraints can be roughly divided into three types. Type I constraints are the limits of finger motions as a result of hand anatomy which is usually referred to as static constraints. Type II constraints are the limits imposed on joints during motion, which is usually referred to as dynamic constraints. Type III constraints are applied in performing natural motion, which have not yet been explored. In this paper, we have only considered Type I and Type II constraints.If we impose all these constraints, we can reduce the representation of our hand model to 15 DoF only.

For the purpose of this project, we have only considered the range of motion of each finger that can be achieved without applying external forces such as bending of fingers backward using the other hand.

Extracting the Hand Parameters & Locating the MP Joints

A three-dimensional hand model is constructed using truncated quadrics as building blocks, approximating the anatomy of a real human hand. Once the important frames of interest are extracted, we now proceed with deriving hand parameters from these frames. The hand parameters include information about different angles of different fingers, thus describing the hand pose and shape in each key frame. Using these parameters we can estimate the pose of hand. As a next step towards hand feature extraction, MP joints of the hand should be identified. To perform this task, we use a corner detection technique. To start with, the distance transform of the palm region is found out. Subsequently, we obtain an image by thresholding, where there are some sharp points which correspond to MP joints of the fingers. So, these corner points are to be detected to find the approximate locations of the MP joints.

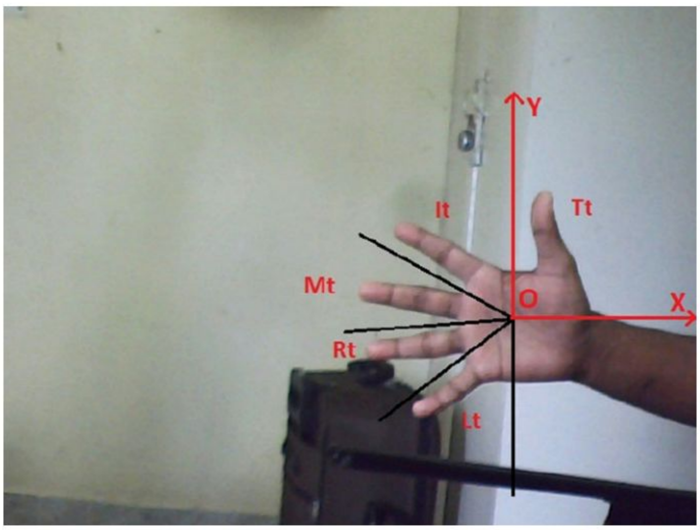


Locating the Finger Tips

To find the finger tips, the fingers from the palm are separated out. Next, the contours of the separated fingers are found out. For each contour, centroid (center of mass) is estimated and the nearest MP joint from the corresponding centroid for all the contours is found out. Finally, we can find the point on the contour, which is farthest from the MP joint. All these points are the tips of the fingers. Obviously, the tip obtained from a contour and the nearest MP joint belong to a particular finger. So, the distance between these two points gives the finger pose length.

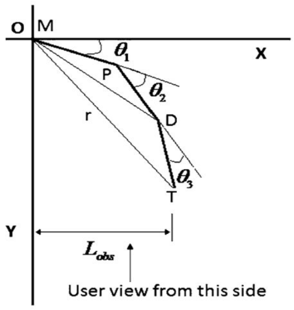
Ambiguity of Locating only the MP Joints and the Finger Tips

Either MP joints or finger tips individually or jointly cannot give exact information about which finger they corresponds to. In order to solve this ambiguity, some radial range for each MP joint is considered. Let us consider the simple case of horizontal orientation of the hand. Angular ranges are defined to identify the fingers as shown below.



Let us assume that, the center of the largest circle inscribed coincides with the origin of the reference coordinate system as shown in the figure above. Subsequently, the angle made by the position vector of all the MP joints with X-axis is determined. MP joint of thumb (if present) falls in the first quadrant of the coordinate system. Hence, we can fix the range for MP joint of thumb as 0deg:90deg similarly, the angular range for MP joints of index, middle, ring and pinky fingers are 90deg.:150deg, 150deg.: 185deg., 185deg.: 215deg. and 215deg.: 270deg. respectively.

<u>*Hand posture estimationI just m*</u>



By applying all of the techniques above, we have will now be able to estimate the posture of a hand. The joint angles of the fingers are calculated only using the information of finger posture length i.e., the distance between MP joint and the corresponding fingertip. For this, the hand modelling constraints discussed earlier is used. Let, ‘‘L’’ be the original length of the finger. Similarly, ‘‘theta 1’’ is the joint angle corresponds to flexion of the MP joint, ‘theta 2’ is the joint angle corresponds to PIP joint and ‘‘theta 3’ is the angle corresponds to DIP joint of finger. And let ‘‘Lobs’’ be the observed length, i.e., distance between MP joint and the corresponding fingertip. In this, the known parameters are, length of the finger (L) and observed Length (Lobs). Apparently, M, P, D and T represents MP joint, PIP joint, DIP joint and Tip of the finger, which are shown in the above picture. It is assumed that the lengths of all the joints of a finger are of equal length and MP, PIP, DIP and fingertip (T) lie in the same plane. The finger plane is aligned with x–y plane, such that MP joint coincides with the origin of the reference coordinate system. Here, finger plane is the plane passing through M, P, D and T. Let us assume that, all phalangeal lengths are equal i.e., each phalangeal is of length equal to one third of the total length of the finger. The coordinates of M are (0, 0), and the coordinates of the tip, T are given by

$$T_x = \frac{L}{3}(\cos(\theta_1) + \cos(\theta_1 + \theta_2) + \cos(\theta_1 + \theta_2 + \theta_3))$$
$$T_y = \frac{L}{3}(\sin(\theta_1) + \sin(\theta_1 + \theta_2) + \sin(\theta_1 + \theta_2 + \theta_3))$$

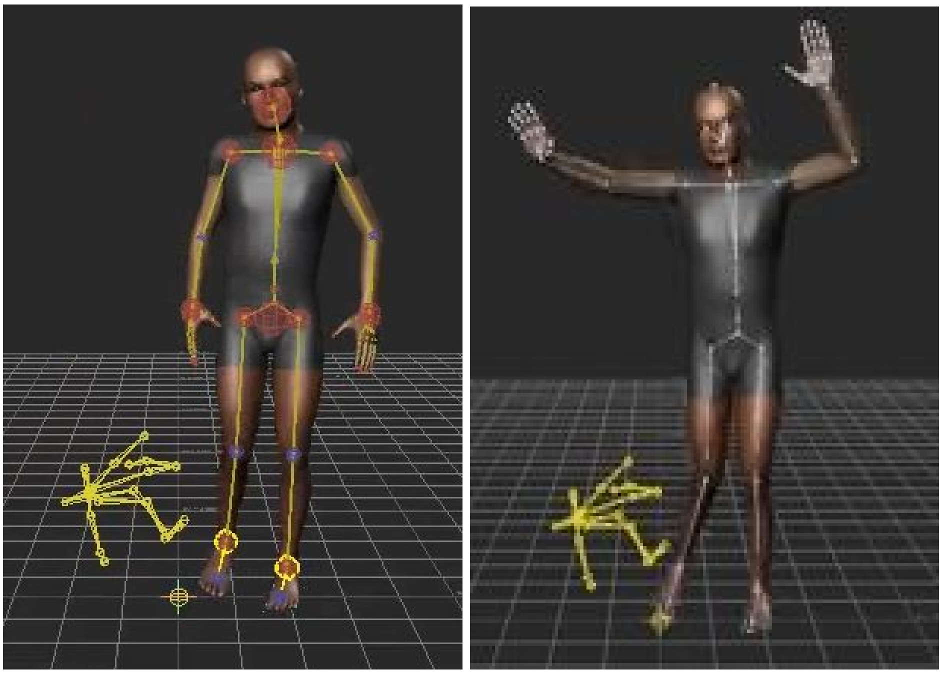
Therefore, we can express L_{obs} as,

$$L_{obs} = \frac{L}{3}(\cos(\theta_1) + \cos(\theta_1 + \theta_2) + \cos(\theta_1 + \theta_2 + \theta_3))$$

2.5 Synchronization of hand data and whole-body animation

Consistency and compatibility are very important when attaching detailed hand data to existing whole-body animation. However, consistency may be violated by imperfect process of timeline and dislocation of joints for connected body part (e.g. joints that connect arm and wrist).

As for consistency of timeline, since that leap motion sensor could not provide storable data or consistent video data, the better way to insert image data into animation data is using keyframing [28] which involves dragging each joint of the character's hand to create a pose. In this approach, it would be much easier to map gesture data to hand in animation.



left(raw body animation)VS right(after combination of hand through timeline)

As for consistency of joints, Szilard mentioned motion interpolation technique [29] which enables to combine data with height changes of fingers into walking movement sequences. This strategy will help change location of hand while keep other part of body unchanged. Also, Szilard removes redundant motions by using weights-based combination [29] of conflicting/duplicated data, which assigns actions with scope and priority before combination. By this method, duplicated parts of hand data and body animation can be correctly linked. In addition, for error of data caused by recording location in different views (e.g. front view and left view), Dong evaluates a strategy to minimize the error. He suggests stabilizing position tracking of each axis while causing differential position errors between each axis and the other two axes converge to zero [30]. Inspired by this idea, 2D data provided by leap sensor could be evaluated one by one according to x,y,z axis and finally summarized to 3D data.

3. Conclusion

After reviewing the complexity of tracking 3D hand motion, we can state that the most approaches from different studies have assumed the person is isolated and not interacting with the environment. In many cases, tracking rigid environmental objects is simpler than tracking high-dimensional human motion. When a human is in contact with objects in the world, their poses constrain the pose of body, essentially removing degrees of freedom. Thus what would appear to be a harder problem, combining object and human tracking, is actually simpler. We aim to study and focus on different angles of hands and fingers to capture, convert, and map all these data to animate the best movements from the hands and its integration with body. Since, it is hard to conclude a unique result from all these different studies and methods, and present a best method, since integration of hand and body animation is complex. As most of studies are based on finding the best result by using different devices and methods. Our proposed solution is to track hand and finger interaction with objects in the environment with the help of Leap Motion. Our study is based on what a previous group from University of Alberta had done, and cover up some points that are missing in their studies, for example, to capture data, there are 2 points we need to be aware of, first when we record the motions of hands, we need to keep our hands within 250mm from Leap motion controller, and the movement need to be constant. Second, when we record the motion, we need to present all the fingers to Leap motion controller; otherwise there will be errors while we animate the movement of the whole body. We hope to make improvement for hands movements on body animation in terms of accuracy and efficiency.

References

[1] Miyata, N., M. Kouchi, T. Kmhara, and M. Mochimaru. "Modeling of Human Hand Link Structure from Optical Motion Capture Data." *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Web.
[2] Marin, Giulio, Fabio Dominio, and Pietro Zanuttigh. "Hand Gesture Recognition with Leap Motion and Kinect Devices." *2014 IEEE International Conference on Image Processing (ICIP)* (2014). Web.
[3] Guna, Jože, Grega Jakus, Matevž Pogačnik, Sašo Tomažič, and Jaka Sodnik. "An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking." *Sensors* 14.2 (2014): 3702-720. Web.
[4]Garber, Lee. "Gestural Technology: Moving Interfaces in a New Direction [Technology News]." *Computer* 46.10 (2013): 22-25. Web.
[5] Wang, Robert Y., and Jovan Popović. "Real-time Hand-tracking with a Color Glove." *TOG ACM Trans. Graph. ACM Transactions on Graphics* 28.3 (2009): 1. Web.
[6] "RFC 4627 - The Application/json Media Type for JavaScript Object Notation (JSON)." *RFC 4627 - The Application/json Media Type for JavaScript Object Notation (JSON)*. Web. 22 Feb. 2016
[7] Alexander, Keith. "RDF/JSON: A Specification for Serialising RDF in JSON." 1. Web.
[8] M, Meredith, and Maddock S. "Motion Capture File Formats Explained." Web.
[9] "Biovision Hierarchy"*Wikipedia*. Wikimedia Foundation. Web. 22 Feb. 2016.
[10] "Motion Analysis' HTR." *Motion Analysis' HTR*. Web. 22 Feb. 2016.
[11] J. J. Kuch and T. S. Huang, "Vision Based Modeling and Tracking for Virtual Telecon ferencing and Telecollaboration", proc. of ICCV, June 1995, pp. 666-671.
[12] "3D Hand Gesture Tracking by Model Registration", by Hocine OUHADDI, Patrick HORAIN
[13] "From Motion Capture to Real-Time Character Animation", Franck Multon1,2, Richard Kulpa1, Ludovic Hoyet2, Taku Komura3
[14] "Character-Prop interaction in Motion Builder using constraints", by Ricardo Tobon
[15] "How to characterize your character in motion builder", by mocappy
[16]"Rajinder Sodhi." *Rajinder Sodhi*. N.p., n.d. Web. 22 Feb. 2016.
Sodhi, Rajinder. *AIREAL: Interactive Tactile Experiences in Free Air*. Disney Research, Pittsburgh, University of Illinois. Web. 22 Feb. 2016.
<http://www.disneyresearch.com/wp-content/uploads/Aireal_FNL.pdf>.
[17]Sharp, Toby, David Kim, Daniel Freedman, Cem Keskin, and Christoph Rhemann. *Accurate, Robust, and Flexible Real-time Hand Tracking*. Microsoft Research. Web. 22 Feb. 2016. <http://research.microsoft.com/pubs/238453/pn362-sharp.pdf>.
[18]Sharp, Toby, David Kim, Daniel Freedman, Cem Keskin, and Christoph Rhemann. *Accurate, Robust, and Flexible Real-time Hand Tracking*. Microsoft Research. Web. 22 Feb. 2016. <http://research.microsoft.com/pubs/238453/pn362-sharp-supplementary.pdf>. Supplementary Material

[19]Hinckley, Ken, Randy Pausch, John C. Goble, and Neal F. Kassell. "Passive Real-World." *Neurosurgical Visualization*. University of Virginia, n.d. Web. 22 Feb. 2016. <http://research.microsoft.com/en-us/um/people/kenh/papers/props.htm>.

[20]Kjellstrom, Hedvig, Danica Kragic, and Michael J. Black. *Tracking People Interacting with Objects*. Brown University, Providence, RI 02912, USA. Web. 22 Feb. 2016. <http://cs.brown.edu/~black/Papers/hedvigCVPR10.pdf>.

[21]Rautaray, Siddharth S., and Anupam Agrawal. *REAL TIME HAND GESTURE RECOGNITION SYSTEM FOR DYNAMIC APPLICATIONS*. Indian Institute of Information Technology Allahabad, India, Jan. 2012. Web. 22 Feb. 2016. <http://airccse.org/journal/iju/papers/3112iju03.pdf>.

[22]"Virtual Reality Gaming & Motion Capture Software." *MindMaze*. N.p., n.d. Web. 22 Feb. 2016. <http://www.mindmaze.ch/vr/>.

[23]Pavlovic VI, Sharma R, Huang TS (1997) Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans Pattern Anal Mach Intell 19(7):677–695

[24] Zhang S, McCullagh P, Nugent C, Zheng H, Baumgarten M (2011) Optimal model selection for posture recognition in home- based healthcare. Int J Mach Learn Cybern 2(1):1–14

[25] Wu Y, Huang TS (2001) Hand modelling, analysis, and recognition for vision based human computer interaction. IEEE Signal Process Mag 18:51–60

[26]Ong SCW, Ranganath S (1997) Automatic sign language analysis: a survey and the future beyond lexical meaning. IEEE Trans Pattern Anal Mach Intell 27(6):873–891

[27]Horace HSIp, Chan Sam CS, Lam Maria SW (2000) Hand gesture animation from static postures using an anatomy-based model. In Proceedings of computer graphics international, pp 29–36

[28] http://www.alliantech.com/pdf/capture

[29] “ Computer Animation for Articulated 3D Characters”, by Szilard Kiss

[30] “Position synchroniztion of multiple motion axes with adaptive coupling control”, by Dong Sun