TECHNIQUES IN MOTION CAPTURE

Interactive Qualifying Project Report completed in partial fulfillment

of the Bachelor of Science degree at

Worcester Polytechnic Institute, Worcester, MA

Submitted to:
Professor Joseph Farbrook (adviser)

Nicholas Konstantino _____

Antonio Vincentelli _____

Calvin Yoon _____

March 6th, 2012

_____
Advisor Signature

ABSTRACT

This project is designed to figure out and catalog how to use and apply the Motion Capture equipment at WPI so that any student group is able to use it in their own projects with a minimum of difficulty. We have set up a knowledge database, in the form of a wiki, and have added our resources and products to the IMGD repository, but they are ultimately community-driven and will grow along with the growing interest in motion capture technology.

Through independent research and experimentation, we have developed a series of techniques for capturing motion, cleaning the data that comes from it, and applying it to make an animation, with extensive use of Autodesk MotionBuilder.

ACKNOWLEDGEMENTS

Special thanks go to Elliot Borenstein for his guidance and his assistance with the maintenance of the motion capture gear.

AUTHORSHIP

Nicholas Konstantino:

- Project Manager
- Wrote the first drafts of the “Introduction to Motion Capture” and “Hardware Description” sections (now the Introduction & Background) and the first draft of the “Cleaning and Modifying the F-Curve Data” section
- Cleaned the following animations: Anger, Bat Swing, Ball Kick, Crouch, Frisbee, Flex, Jumping Jack, Punching, Roar, Thriller
- Applied all cleaned captures to Space Hulk model

Antonio Vincentelli:

- Was the motion capture model for all captures
- Wiki Founder and Admin
- Responsible for the majority of content posted on the wiki (as of this writing)
- Wrote the first drafts of the “Connecting to the Motion Capture System” and “Collecting Data” sections
- Wrote the final paper, and compiled, edited, and formatted all the sections
- Created the animation RoboSwing

Calvin Yoon:

- Project Founder
- Research into MotionBuilder and manual optical cleaning
- Made a 3D model for testing characterization and Maya to MotionBuilder pipeline
- Wrote the first draft of the Characterization tutorial
- Wrote the first drafts of the “Creating Marker Sets”, “Cleaning the Optical Data”, and “Characterization” sections
- Cleaned approximately 15 total captures, including: Drop Object, Drop Object (simple), Pick Up, Pocket Something, Search In Pockets, three Run animations, one Walk animation, Walk Backwards, Shoulder Roll, several Sit/Stand animations, Turn 180°, Turn Left
- Applied all cleaned captures to Robot model

TABLE OF CONTENTS

LIST OF FIGURES

Introduction & Background, Section I: Introduction to Motion Capture

Section II: Description of WPI's Setup

Methodology, Section II: Capturing Data

Section III: Creating Marker Sets

Section IV: Cleaning the Optical Data

Section V: Cleaning and Modifying the F-Curve Data

Section VI: Characterization and Finalizing the Animation

EXECUTIVE SUMMARY

Motion capture technology is not new, but it is growing in uses and popularity as a process for animators of games and film alike to capture very detailed and very specific movements. WPI, despite having the hardware for it, has made very little use of this technology. In fact, the devices themselves were not even set up until 2010, and many students who could have benefited from them had no idea they even existed on campus. In order to keep the work of IMGD students on the cutting edge, they will need access to this technology and the basic knowledge of how to operate it.

It would be ideal if the knowledge of motion capture operation for WPI students was made standard, understandable, and streamlined. This way, any animator who needs a specifically choreographed sequence done efficiently can just followed the provided instructions to the letter and get the results they are looking for. However, there was practically no reliable source of knowledge or information before this IQP, and the one person who did have intimate knowledge of the system's operations will be graduating soon and cannot be expected to take the time to produce the in-depth database of tutorials and examples necessary for future generations to effectively use the technology.

Through first-hand research, this IQP group has achieved a great understanding of WPI's motion capture technology. We have created a wiki that will conveniently store and display all that we have learned and allow other students to share their experiences as well. The wiki also serves as a communal hub in which to discuss the various facets of the motion capture equipment and techniques. All of this is supplemented by a thorough and varied library of captures and animations for learning and student use (with acknowledgments), which are stored on WPI's

IMGD repository. As of this writing, there are two models, 45 raw captures, 20 edited captures, and 45 fully characterized animations, and these numbers will increase as the community grows.

In brief, the motion capture process – from the very beginning to finalizing the animation – is a four-step process. The first step is simply connecting a laptop to the motion capture system through the Linux desktop computer located on-site. This is very important because, although the Linux box can run the capture software and save the data, there is no way to get the data off of the computer afterward. Therefore, a laptop with the necessary programs installed is essential.

The second step is suiting up the actor and capturing the motion data, which is then stored locally on the laptop. The data generated will naturally be imperfect (although proper procedure mitigates errors), so the third step is cleaning it. Since this group utilized Autodesk MotionBuilder 2012, this involves creating marker sets with which to pin a digital actor to the data points, filtering the data (both manually and using built-in filters), and converting the data to and modifying the f-curves.

The third step, which can actually be done before, during, or after all the previous steps, is creating a character. Although the character can be of many shapes and sizes, it must still generally resemble the shape of the original actor and follow specific, strict guidelines. If one wants to use a character that is not generally a bipedal humanoid, it is better to tailor the original motion capture to the character than to try to adjust either of them after production.

The final step is applying the motion animation to the character, fixing any remaining irregularities, and making sure that everything looks as intended. This step can be much shorter if one took care in the preceding steps.

Motion capture is a great way for getting smoother and more realistic animations. Traditional keyframe animations are extremely time-intensive and, as such, people may be drawn to motion capture as it is a faster and more realistic alternative. However, like all methods, it does come with its own limitations. What follows are the results of this IQP and suggestions on how future students can help to improve our motion capture setup.

This IQP group discovered that motion capture is excellent for making realistic animations that mimic even the subtlest of motions in the body, which a traditional animator might overlook or even exaggerate in compensation. It can accurately capture shifts in weight and, by adding more optical nodes to the suit, can achieve even higher fidelity animations. Once one has learned how to operate the motion capture equipment, it is very easy to take a lot of captures in a single session. The most this group managed to take in a single day was approximately 50 takes of 22 completely different animations. If one were to use traditional 3D animation techniques, accomplishing the same task would take one person an entire term, perhaps even two terms, at the very least. This means that future IMGD project groups that want to have a lot of animations can generate and apply them in a couple of weeks instead of months. In fact, on average, only an hour or two at most is required to effectively clean motion capture data once the operator has achieved proficiency with MotionBuilder.

With the addition of the new IMGD repository for models and animations, students can upload finalized motion capture files, which can then be used by the entire student body. Being able to reduce in this manner the amount of original content required for basic operations permits groups to reallocate time and resources to other aspects of game development, such as 3D

modeling, which is extremely time-intensive, adding features, or even better debugging.

While motion capture is a faster alternative to traditional keyframe animation, there are several limitations that come with using it. First, quick movements, such as punching, jumping, and running, have extremely messy optical data. This is in part due to the currently limited work space mentioned above, but it's also a result of the motion capture suits having less than ideal marker coverage. In particular, the hips and back are sorely missing what would otherwise be extremely useful data points. Messier captures mean that the operator has to spend additional time manually cleaning the motion capture data by manipulating translation waves for every problematic marker. Since most video game animations generally consist of fast actions, this is a significant hindrance. The repository can help with this in that, once one person has properly cleaned and uploaded an animation, it would not be necessary to recapture and clean that particular animation, but no database, no matter how extensive, can possibly cover the extent of human creativity.

The second limitation with the current motion capture setup is the fact that the work area is located in an extremely confined space, which severely limits the type of captures one can take. Professional animation studios have their own gymnasiums that are fitted with hundreds of cameras, whereas WPI imposes a 4x4x4 meter space (the minimum recommended space is 6x6x3) and can only employ eight of the total sixteen cameras. A sufficiently larger room, with an expanded cage, could even mitigate the need for a treadmill, which forces unnatural movement and can block markers, and could possibly allow for captures of multiple people at once.

Despite these limitations, motion capture is still a much better alternative to traditional

methods for realistic animations. Motion capture can even be used in combination with keyframing if so desired. This IQP group recommends experimenting with optical marker placement, rigid body arrays, marker set configurations, and MotionBuilder itself as further research subjects.

INTRODUCTION & BACKGROUND

Section I: Introduction to Motion Capture

Motion capture is the process of recording movements of an actor or actress and transforming that data onto a digital plane. It has had uses in a variety of fields, from military to medicine. It is most notable, however, for its use in the entertainment industry, especially in video games and film. Utilizing motion capture, commonly referred to as “mocap”, can generate loads of data relatively quickly and, after a short clean-up period, deliver stunning, real-to-life results.

In the entertainment industry, animations were typically done by hand, starting by marking down each “key” frame and then filling in all the frames between them. With the exception of motion capture, all the advances in animation technology still require a considerable amount of man-hours to create a lifelike animation of any kind, and the time required only increases with the complexity. This is where mocap steps in; if done properly, the data generated will already have an extreme level of realism and motion fidelity, which needs little modification to be sent out as a finished product. Since collecting and utilizing motion capture data requires a smaller group of people to achieve the same effect as the large teams normally needed for traditional animation, mocap technology allows teams to produce more content or even to reallocate resources to other projects.

In the early days of motion capture, circa 1915, the method of choice was called “rotoscoping”. Using this method, one would have an actual frame of film projected onto a piece of paper for the artist to trace over each frame as necessary to finish the animation. The results

were more lifelike, but the process was neither intuitive nor very practical. However, the notion of capturing animation live carries on into current mocap technology.

Modern motion capture is done mostly optically, with two kinds of technologies. These technologies are considered optical because the subjects are "watched" by cameras that take in their every move. The first method is called "passive" marker technology, in which cameras are set up to track the reflections of light coming off of special surfaces on a suit or the actor's body, usually done with small, highly-reflective balls. Although somewhat cheaper, this method is nowhere near as reliable as the "active" marker technology, which features a suit with LED markers along all major body surfaces, creating a virtual skeleton.

Figure IB1.1: An example of a "passive" suit.

In both optical methods, the actor is suited up and led into a chamber surrounded by cameras to record their movements from all angles. The LEDs are more accurate in their recordings because each LED flashes at an individual, and humanly imperceptible, frequency. Therefore, only direct obstruction will make a camera lose track of a marker, and, if one light

were to pass over another, the computer would be able to tell them apart instantly, massively reducing the amount of errors and time costs.

Figure IB1.2: WPI's motion capture suit when activated.

There are other forms of motion capture that are not optical, but these are much less widely utilized. One method is using gyroscopes to track changes in movement, which record are recorded either internally or through a sort of exoskeleton. These methods in general are either limiting to the movement of the actor or miss too much data to be reliable.

One example of modern movies making use of motion capture technology is the *Lord of the Rings* trilogy. The persistent antagonist Gollum, although incredibly lifelike, was entirely generated by computers and put into motion by his human actor in a suit designed to capture his every movement, down to his facial expressions. In fact, the same could be said of any big-budget film using any kind of CGI.

Figure IB1.3: Gollum from the recent *Lord of the Rings* trilogy was voiced and performed by Andy Serkis in a motion capture suit.

Video gaming has also made use of motion capture animation for almost 20 years now; the earliest game reported to have used it was released in 1995. Since then, motion capture has evolved gaming just as much as it has revolutionized visuals in cinema. While various movements of any game character can now be made more realistic by any developer utilizing the technology, motion capture has recently started being directed towards the player. The PSMove, Xbox Kinect, and the Wii all use motion capture to control games and perform commands, and this physical involvement creates a more immersive experience. Video gaming has also started down the path of Augmented and Virtual Reality; a sophisticated computer can use motion capture to create the illusion of interaction with a 3D environment.

Figure IB1.4: An example of an "exoskeleton"motion capture suit.

Section II: Description of WPI's Setup

WPI uses the PhaseSpace brand motion capture system, which is operated using the PhaseSpace Improv Program Suite and which utilizes 8 cameras, one of two suits (sizes L and XL), an assortment of LED markers for the suits, one of two configuration-and-power packs, and a calibration wand. Although these were not included in the package, this IQP group also utilized fingerless gloves, athletic slippers, and a headband, each with Velcro tape in key marker locations. The original package did include a pair of full gloves with sown-in markers, but these proved to be more of a hindrance than a convenience because they surpassed the marker limit allowed for each arm. This group has also provided a stripped-down treadmill for use inside the motion capture cage to precisely capture movement cycles.

Figure IB2.1: Some of the equipment types provided by the PhaseSpace bundle.

There exists MotionBuilder plugin available for this hardware, but this group neither set it up or tested it.

HARDWARE SPECIFICATIONS

The IMPROV motion capture system has some of the most sophisticated and advanced hardware available. All of this technology was designed to make motion capture easier and more reliable. PhaseSpace has taken the hard work away from the user and packed it into the hardware. The entire system consists of only a few robust, portable elements.

- **CAMERAS:**

  Dimensions: 108 mm x 92 mm x 57 mm (4.25" x 3.62" x 2.25")
  Weight: 380 grams (13.4 ounces)
  Each camera achieves an Optical Resolution of 3600 x 3600 (12 Megapixel) using two linear detectors with 16-bit dynamic range. Onboard processors produce an impressive Subpixel Resolution of 30,000 x 30,000 at 240 Hz.

- **LED BASE STATION:**

  Dimensions: 127 mm x 76 mm x 30 mm (5" x 3" x 1.2")
  Weight: 136 grams (0.3lbs)
  This 2.4 Ghz Transceiver synchronizes the LED controllers with the server.

- **LED CONTROLLERS:**

  Dimensions: 126 mm x 70 mm x 25 mm (5" x 2.75" x 0.85")
  Weight: 90 grams (0.2lbs)
  The LED Controller, an RF transceiver, utilizes an onboard microprocessor to control up to 42 LEDs. Battery life is 2 to 4 hours of continual use, up to 8 hours of typical use. The IMPROV system can use multiple LED Controllers simultaneously.

- **ACTIVE LED:**

  Dimensions: 20 mm x 14 mm x 3.2 mm (0.8" x 0.55"
  Weight: 4.5 grams (0.01lbs)
  Each LED modulates at a unique frequency resulting in a unique digital ID. LEDs are available in Red visible and Infra-red versions.

- **SERVER COMPUTER:**

  Dimensions: 44.76 x 17.78 x 45.43cm (17.6 x 7 x 17.9in)
  Weight: Starting at 10.7 kg
  Processor: Intel Core i3-540 Processor (3.06 GHz, 4 MB cache, 1333 MHz memory)
  Chipset:Intel 3450 Chipset
  Memory: 4GB
  integrated PhaseSpace HUB in floppy bay drive
  Network: Gigabit

- **CALIBRATION WAND:**

  The calibration wand serves as the principal tool used to accurately calibrate the system. The calibration wand allows a user to calibrate the given capture area in a matter of minutes.

- **CONFIGURATION:**

  The IMPROV system uses standard Cat5 cables to connect between the cameras, basestation, and server.

Figure IB2.2: Hardware specifications from <http://www.phasespace.com/improv_motion_capture.html>

Use of this system greatly favors IMGD students. Not only are these tools used in the games and media industries, they also serve as useful tools for possible student MQPs. IMGD Art students can use the motion capture setup to create intricate animation scenes or cycles for games, and IMGD Tech students can attempt to utilize it for motion controls.

Outside of IMGD students, Biology and Biomedical students can learn how to utilize the optical motion capture system to precisely examine physiology and movement patterns across several test subjects.

One thing that could be done to improve our current system for more accurate results, aside from adding more cameras, is moving the entire setup to a larger room with higher ceilings. Since the current work area has such a low ceiling, the effective capture area is notably smaller than the area of the capture cage itself. A sufficiently larger room, with a similarly expanded cage, could even mitigate the need for a treadmill, which forces unnatural movement, and could possibly allow for captures of multiple people at once. The cages themselves are primarily PVC and are designed for reconfiguration and portability.

As previously mentioned, the computer currently routed to the motion capture system doesn't allow for the recorded files to be taken off of it, so anyone wishing to actually make use of the captures needs to bring and set up their own laptop. Making the connection is generally inconsistent, which slows down the entire process, and could even be avoided altogether with a better setup. This IQP group has not looked into what changes could be made to the computer system and, instead, focused on how to operate with a laptop, but a complete, on-site setup with all the required software (including MotionBuilder) would be preferable.

METHODOLOGY

Section I: Connecting to the Motion Capture System

The motion capture system runs on the PhaseSpace Improv Program Suite and is routed to a desktop computer running a Linux operating system. Since there is no way to offload data from the Linux machine, it is necessary to connect a laptop to it, which must also have the Program Suite installed. The Program Suite does come in a Linux version, but Autodesk MotionBuilder does not, which means that at least one laptop must run on a Windows operating system.

Part 1: Turning on the Linux box

1. Press the power button (the chrome nub to the left of the Power symbol, on the right side of the front panel).
2. At the OS selection screen, Linux should already be selected; just hit Enter. If it is not selected, navigate to it with the arrow keys and hit Enter.
3. One will now be prompted for a user name and password. Type in the user name and hit Enter. Although the user name was displayed as one typed, the password will remain blank (not even asterisks). Regardless, type in the password and hit Enter. Inputting either field incorrectly will restart the prompt.
4. The screen will scroll through lots of text as it loads the desktop. When the text stops, the bottom row will end with " impulse:~% ". Type "startx" (without the quotation marks) and hit Enter.

Part 2: Connecting a Laptop

Since laptops with a Windows operating system are both prevalent and required, these instructions are designed for them. It is potentially possible to collect capture data with a Linux laptop, but this IQP group did not come across that issue.

1. Plug the BLUE Ethernet cable coming out from the back of the Linux box into an Ethernet port on the laptop.
2. Open the Network & Sharing Center. This can easily be done in two different ways:
    - Method 1: On Windows 7/Vista, click on the Internet Connectivity icon (usually a wi-fi signal bar) and click “Open Network and Sharing Center” on the bottom of the pop-up.
    - Method 2: Open Control Panel, then “Network and Internet”, then “Network and Sharing Center”.
3. Click on the Wired Network Connection.
4. Click on Properties.
5. Click “Internet Protocol Version 4” and click Properties.
6. Click “Alternate Configuration” and click the “User configured” bubble.
7. Under the IP Address field, enter “192.168.1.110”
8. Under the Subnet Mask field, enter “255.255.255.0”
9. Save settings and see if it connects. If it does not connect, unplug the cable and try again.

Once the laptop connects, it can control and collect data from the motion capture system.

Section II: Capturing Data

Part 1: Suiting Up

There are two almost-full-body suits that came with the motion capture gear, a Large and an Extra-Large. These suits cover everything except the hands, feet, and head, and they are convenient because they are pre-wired and do not require refitting. Due to the nature of the motion capture, one should strive for a snug fit so that the LED nodes do not move or slide during the capture. That said, the Large suit should fit most people of average height and frame (or smaller). The dressing order is as follows:

1. Put on the footwear, preferably over socks. Put on the suit, feet first, with the zipper in the front.

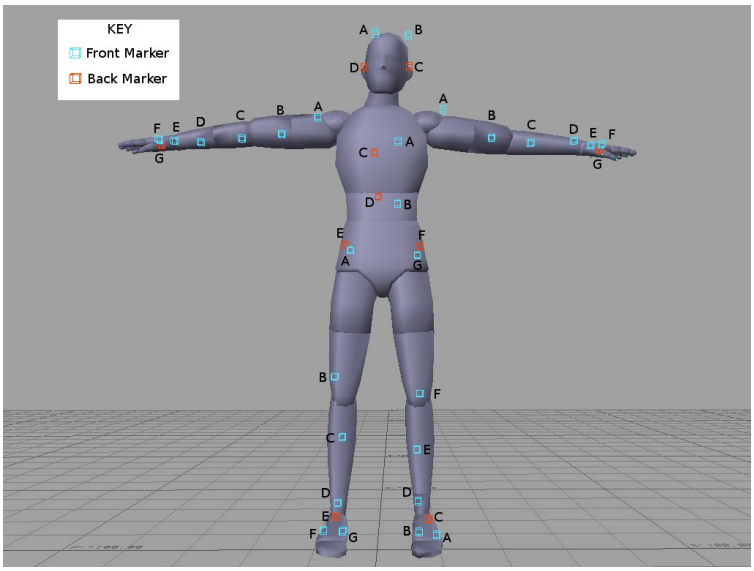

Figure 2.1.1: The marker placement diagram

It is important to note that the Extra-Large suit has ankle marker locations on the inside of the legs instead of the outside

2. After settling into the suit and zipping up, put on the fingerless gloves. The wire at the end of each arm (at the wrists) can go over or under the glove, whichever the wearer finds most convenient or comfortable.
3. Attach the markers to the suit, as per the below diagram.

It is important to understand that the right side of the image is actually the actor's left side. Also, the F and G markers on the hands are in-line with the knuckles and on top of the hand. There should *not* be any markers where there isn't Velcro.

4. Connect the capture box to the marker wires (at the neck) and place it in the back-sleeve (a yellow mesh on the Large, black spandex on the XL).
5. There is a slight variation in the final step between the Large and the Extra-Large.
    - LARGE: Use the connector (pictured) to connect the plugs to the controller. Note that it only inserts one way (the protrusion on the connector slides into a corresponding hole in the controller). The plugs also only insert one way (wires towards numbered side). When inserting the plugs into the pins, start at the highest numbered pair of pins and continue down.

Figure 2.1.2: The connector

- X-LARGE: This suit is already plugged. Simply insert its connector into the controller. As above, it only inserts one way; the protrusion on the connector slides into the respective hole in the controller.

Part 2: Taking a Capture

To take a capture:

1. Open the Master program. Select File → New Session, select the "..." button (browse), browse for and select a good location to store the data, and input a useful name (such as "captures_yymmdd" or "Jump_Captures_mm-dd").
2. In the top-right, make sure the Server Address field matches one's IP (which should always be a variation of 192.168.1.xxx). The remaining settings are:
   - Mode: Default
   - Frequency: 480
   - Post Processing: On
   - Interpolation: 4

3. In the top-left, click the Connect button. The program should connect to the cameras around the capture cage, which will display the ready actor (as a series of dots or markers) in the view field.

If not all markers are visible, increase the Max Markers, on the bottom-right of the screen, to a big number.

4. To make a capture, click the Take drop-down box and select New Take. Have the actor assume a T pose, click the red Record button, and record the predetermined action. Only record one action at a time. Click Stop to stop recording that take.
5. To view the capture, click Connect again to disconnect; the green box should become hollow. One can now play back the take.

The first take should always start in a T-pose, then gently rotate each joint and limb (shoulder circles, ankle rotations, etc.). This is crucial for accurate alignment in the Cleaning and Characterization stages. All takes after the first are any animations or actions (again, one per take) one wishes to capture.

Section III: Creating Marker Sets

Knowing how to make and use marker sets is an essential skill that one must learn in order to clean the motion capture files. Making a marker set is essentially the process of applying the optical nodes to the default MotionBuilder actor model. This allows one to see in real time an approximation of how the motion will look when applied to a regular model. This will help give an idea of which optical data nodes need to be tweaked, since the body parts on the actor will behave strangely if something is wrong. For example, limbs may twist or bend in unnatural.

The first step is to import the motion data into MotionBuilder by clicking on File → Motion File Import. This will bring up an explorer window that one uses to find and open the desired motion file, which should end in ".c3d". Once the file has been chosen, MotionBuilder will bring up an Import Options window. This window shows various things, like the takes that are associated with the file, a start and end frame, the number of samples in the file, and a sample rate. In general, these options will not need modification and can be safely ignored. Complete the import by pressing the Import button on the bottom right. MotionBuilder will then import the file into the scene, and all the optical points that were used in the capture will be arranged in whatever pose the original actor was in when the capture was begun.

To apply an actor to the nodes, one needs to go to the Asset Browser, open up the Templates folder, and then open the Characters sub-folder. Within this folder should be an object called "Actor", which one can just click and drag into the scene. The blue actor will appear in the middle of the screen in a T-pose. On the bottom of the MotionBuilder window, an Actor Settings window should appear, showing the newly created actor.

The next step is to create a Marker Set for the character. Marker Sets are used to determine which optical markers are used to move each individual body part of the actor. To make a marker set, in the Actor Settings window there should be a button that says "MarkerSet…" After clicking on it, a drop-down menu will appear; just click on Create. This will make a brand new marker set.

There are a total of 19 Marker Groups to which one can apply optical markers. They are as follows:

1. Head	11. Right Hand
2. Torso	12. Left Knee
3. Hips	13. Left Ankle
4. Left Shoulder	14. Left Foot
5. Left Elbow	15. Left Toes
6. Left Wrist	16. Right Knee
7. Left Hand	17. Right Ankle
8. Right Shoulder	18. Right Foot
9. Right Elbow	19. Right Toes
10. Right Wrist	

Figure 3.1: Marker set groups

Each Marker Group refers to a joint on the body. There are also 15 groups for the hands, which relate to the finger joints. No matter how many optical markers are in a relevant area, each marker group can only hold up to a maximum of 5 optical markers. However, this is not a concern with the current marker layout.

To make things easier when cleaning multiple captures, it is generally helpful to take one capture of a T-pose stance at the beginning of a capture session and design the marker set around it. This makes it easier to both align the actor to the optical data and to reuse the marker set for the other captures taken in that same session. However, every time the capture suit or the LED markers are removed or adjusted, it is highly recommended to take a new T-pose capture because the previous marker layout will have changed and the previous marker set will no longer apply properly.

To apply marker nodes to groups, open the Opticals menu in the Navigator and reveal all of the optical markers. Next, select the desired nodes in the scene to highlight them in the Navigator. This will make it much easier to find them and drag them into the correct marker group. For example, using the current setup, there should be four markers for the head, named C3D:M024, C3D:M025, C3D:M026, and C3D:M027. When these are dragged from the Navigator into the Head Marker Group, they will show up in the Objects field and the Head Marker Group will display the number four, indicating it is linked to four markers. Although not necessary, all of the optical markers can be renamed by right-clicking the node in the Navigator and selecting Rename.

The following is an example of a Marker Set:

1. Head	10. Right Wrist
A. M024, M025, M026, M027	1A. M065, M066
2. Torso	11. Right Hand (Empty)
A. M000, M001, M002, M003	12. Left Knee
3. Hips	A. M053
A. M004, M005, M036, M054	13. Left Ankle
4. Left Shoulder	A. M051, M052
A. M012, M013, M014	14. Left Foot
5. Left Elbow	A. M048, M049, M050
A. M015, M016	15. Left Toes (Empty)
6. Left Wrist	16. Right Knee
A. M017, M018	A. M037
7. Left Hand (Empty)	17. Right Ankle
8. Right Shoulder	A. M038, M039
A. M060, M061, M062	18. Right Foot
9. Right Elbow	A. M040, M041, M042
A. M063, M064	19. Right Toes (Empty)

Figure 3.2: Example marker set

As is shown above, not all of the marker groups have to have markers assigned to them. Assigning markers to every group will give more directed movement to the actor and require less compensation. For example, the fingers and toes will not generally move unless they have linked optic markers or are manually keyframed. Having more nodes also means that one will have more nodes in each rigid body.

A rigid body (explained in-depth in the next section) is MotionBuilder's way of linking a number of nodes in space in relation to each other. Having more nodes in a rigid body gives MotionBuilder more to work with when it interpolates the nodes in the rigid body. The benefit to having more nodes in a rigid body is that it gives MotionBuilder more data to better interpret the actor’s movement. This allows the actor to move properly even if one or two nodes are temporarily lost.

Once all of the markers being used are applied, click on the “Activate” checkbox above the Marker Groups window. If done properly, the actor's body will snap to the markers as best as it can and follow them when the capture is played.

Section IV: Cleaning the Optical Data

The motion capture data that one should have immediately after taking the captures will always be very messy unless they are very subtle motions. One of the biggest problems with using this messy data is that the motions will appear extremely jittery or limbs may fly off in random directions. The first step in the cleaning process that needs to be taken to make these captures usable is to clean the optical data manually. This is done by editing the curves which define each optical's motion in XYZ space.

There are several steps to take that streamline the cleaning process, the first of which is working with rigid bodies and characterization. Creating an actor will help to see how the optical data will react when applied to a character, which will determine how it will react with one's own models.

Part 1: Creating Rigid Bodies

Rigid bodies are used to constrain two optical points to each other so that they, ideally, never leave a certain distance from each other. To do this, open the optical graph, select however many points one wants to constrain to each other by ctrl+clicking them, and then press the B key to bind the points into a rigid body. Generally, one will want to make rigid bodies in such a way that they act as the bones for the character, especially for preventing limbs from bending at the wrong spots. For example, two points on the leg can be bound into a line to simulate the femur. It is not necessary for all of the optical points to be bound into rigid bodies, and, in some cases, doing so causes unnatural movements when MotionBuilder tries to interpolate the data while being constrained.

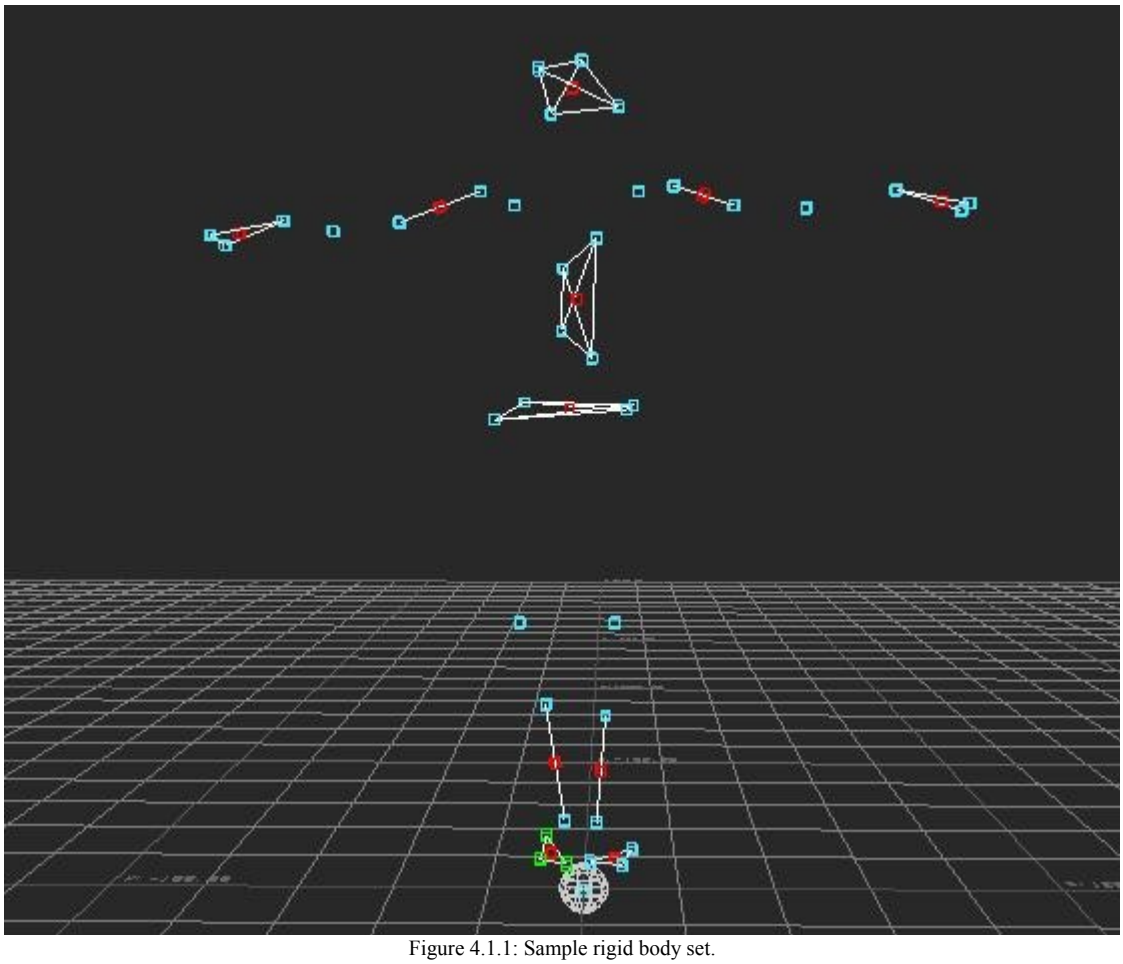

Figure 4.1.1: Sample rigid body set.
Rigid body configurations don't need to follow this one as other configurations may be more efficient.

Many cases where nodes become inactive and start to fly off can be fixed by constraining the flying opticals to cleaner opticals in a rigid body. This is because, when nodes go inactive, MotionBuilder tries to interpolate where the node is going based on its previous position, trajectory, and velocity. When nodes become inactive due to being obstructed for some reason during the capturing process, MotionBuilder takes the segment at which the node deactivated and turns it into a gap. By constraining the missing optical to a rigid body, MotionBuilder interpolates the gap based on segments from other opticals in the rigid body. Otherwise, it would take the sine of the end of the last segment and the sine of the beginning of the next segment and averages them to make a wave in the gap space.

Part 2: Anatomy of an Optical Graph

After the nodes have been prepared and an actor has been successfully made and activated, the next step is to bring up the Opticals graph. This is done by going to Window → Add Filtered Navigator → Opticals. A window will pop up that has two parts: the Navigator on the left, and the main window on the right. Double-clicking on any optical marker will bring up its graph; do so for a troublesome marker.

Using the default MotionBuilder control set, pressing the A key will center the entire wave in the graph. One can zoom in on certain sections by Ctrl+clicking and dragging over where one wants to zoom in. One can pan the graph by Shift+clicking and dragging.

The opticals graph consists of 3 parts:

1. The Waves
2. Segments
3. Gaps

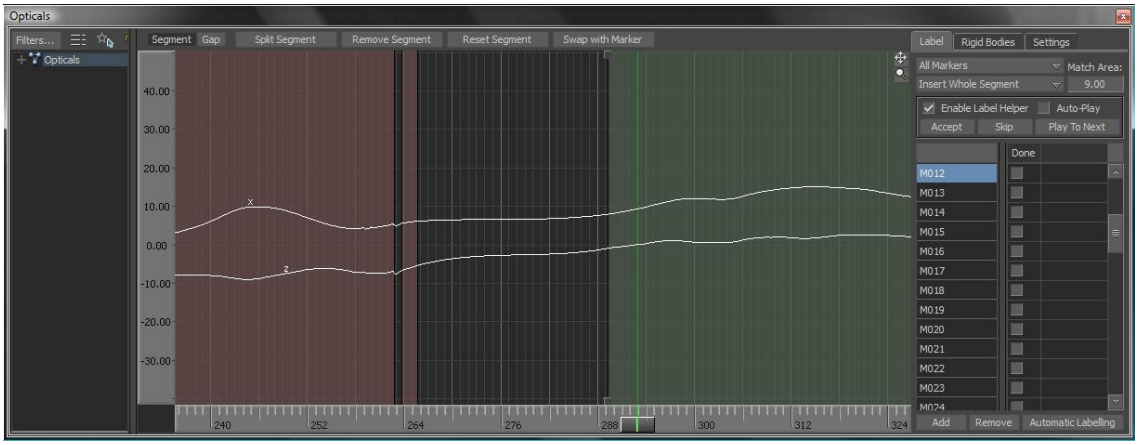

Figure 4.2.1: An optical graph

In order to select a segment or gap for use, scroll the timeline by either holding down the J key and clicking and dragging on the graph or by moving the slider manually.

The waves describe the optical node's movement, and the three lines show how it moves in 3D space. One line shows how far the node moves from 0 in the X-axis, another for the Y-axis, and one last line for the Z-axis. The main problems to fix are unnatural sections in the waveform, such as sudden unintended spikes or plateaus. Spikes will generally translate into jitter or jumps in the animation, depending on the severity of the spike.

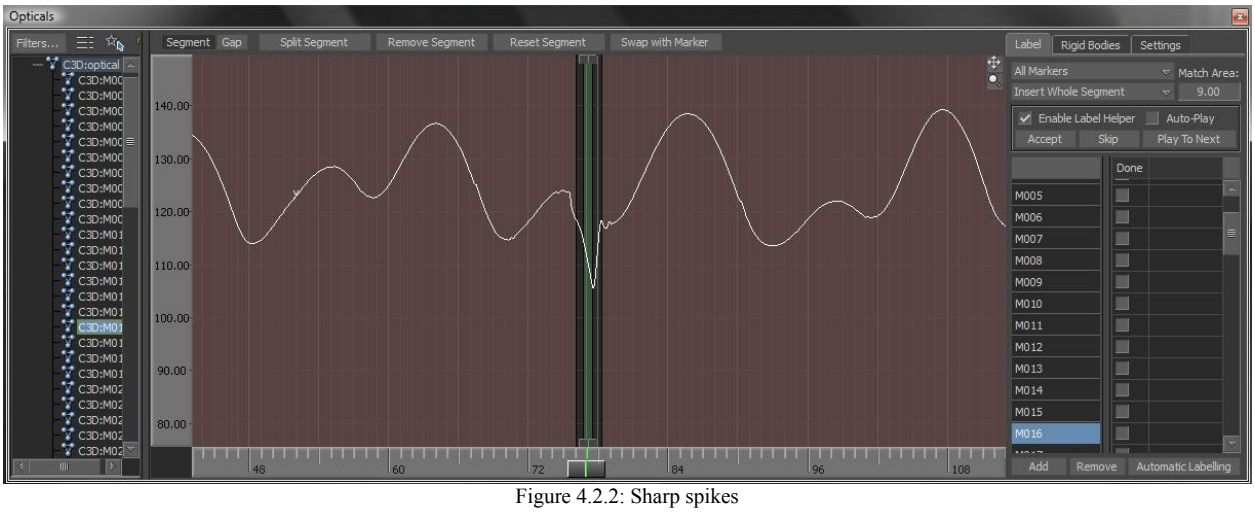

Figure 4.2.2: Sharp spikes

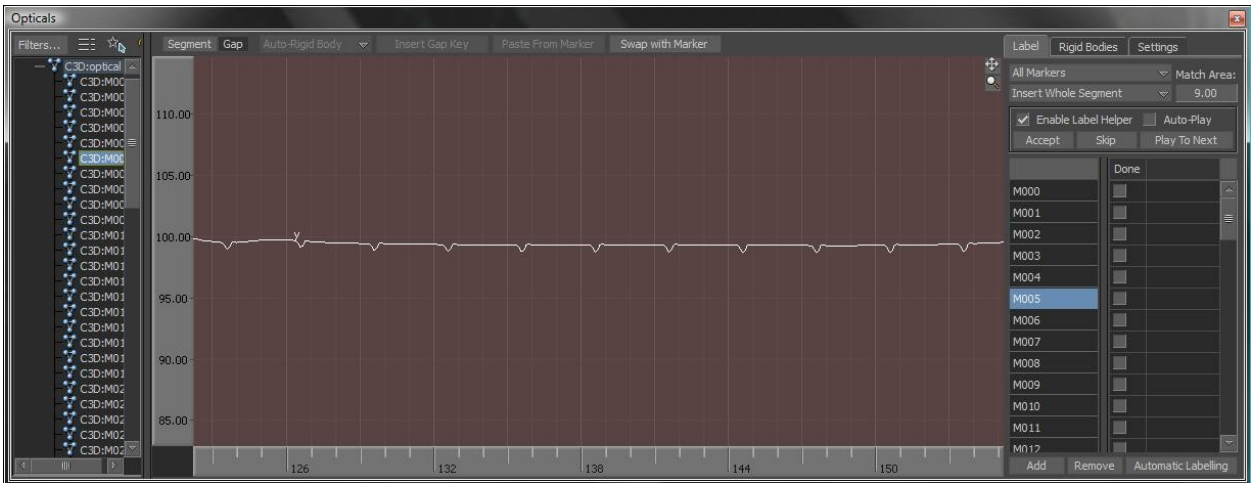

Figure 4.2.3: Shallow spikes

Occasionally, one might get a series of smaller spikes. These don't need to be cut and can be fixed by using the Smooth filter, which is covered later on.

Unnatural plateaus are parts of the wave where, for some reason, the waveform is normal but all the values are raised or lowered a certain amount for a certain duration. Visually, one can tell there might be a plateau in an optical node if the actor's body parts raise in an unnatural way for a short time before dropping back to their normal spot. This mostly happens with the shoulders and, when it does happen, the shoulders will seem to pop out of their sockets for a little while before dropping back down. Certain animations may have natural plateaus for animations, though, so care must be taken before eliminating plateaus on site. Unnatural plateaus differ from natural plateaus in that they usually have sharp, sudden, and spike-like inclines.

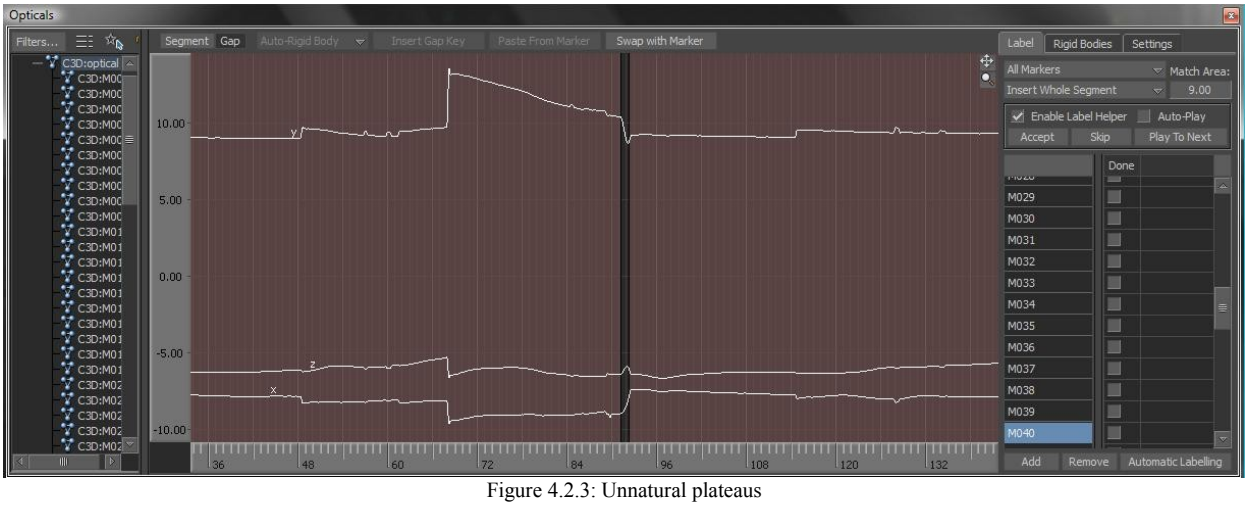

Figure 4.2.3: Unnatural plateaus

Natural plateaus have much smoother inclines, which translates to smoother movement on the actor.

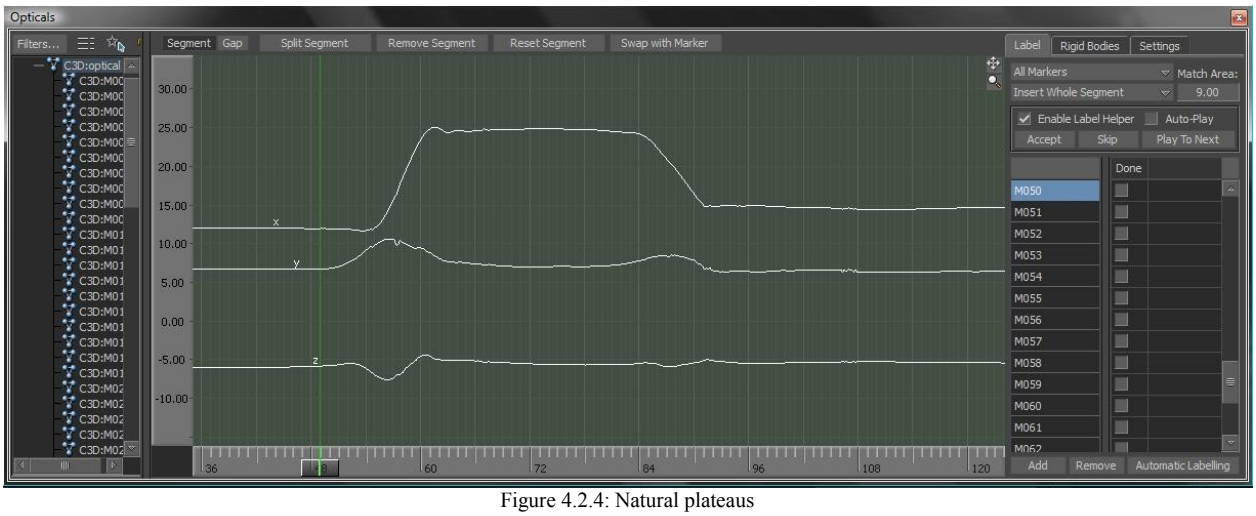

Figure 4.2.4: Natural plateaus

Segments are portions of the motion capture data that MotionBuilder adheres to without any interpolation. MotionBuilder will try to make as much of the graph as possible into segments. The position on the timeline is indicated on the graph by a straight, vertical green line. There are two different kinds of segments: active and inactive segments. Active segments are green and show that the timeline position is inside of that particular segment. All other segments are red and are considered inactive. All segments turn inactive if the position is over a gap between segments.

Part 3: Manually cleaning the Optical Graphs

In order to remove any errors in the wave that are inside segments, one needs to split the segment by scrolling the timeline to the correct position and then pressing the “Split Segment” button. The “Remove Segment” button will then remove whatever is currently the active segment.

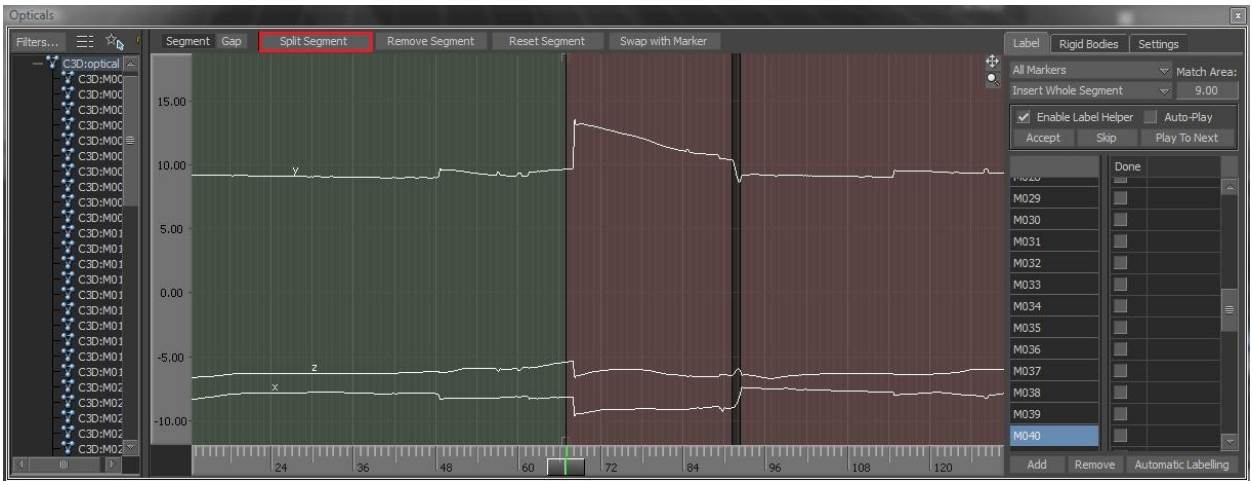

Figure 4.3.1: Before split

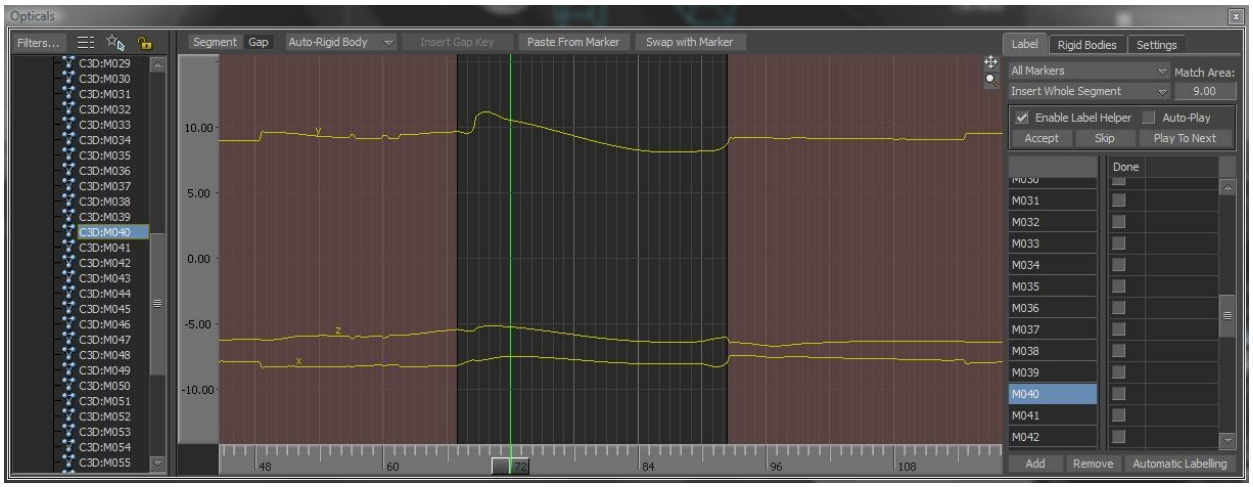

Figure 4.3.2: After split

When an optical node disappears for whatever reason, the space and time for which it was undetected automatically becomes a gap. Gaps are where MotionBuilder disregards the optical point data and tries to interpolate the curve between the segments that come before and after the gap. One can change the waveform inside the gaps by changing the way MotionBuilder interpolates it. There are 6 methods that can be used to interpolate the wave in a gap:

1. Auto-Rigid Body
2. Auto-Constant
3. Auto-Linear

4. Auto-Bezier
5. Control Curve
6. Sample Curve

By default, MotionBuilder will make the wave in the gap using the Auto-Rigid Body method. This method compares the point data with any rigid body constraints and will try to adjust the graph such that the rigid body shape remains the same. While this does keep the shape of the actor intact, it most often leads to a lot of jittering and jumping in the waveform.

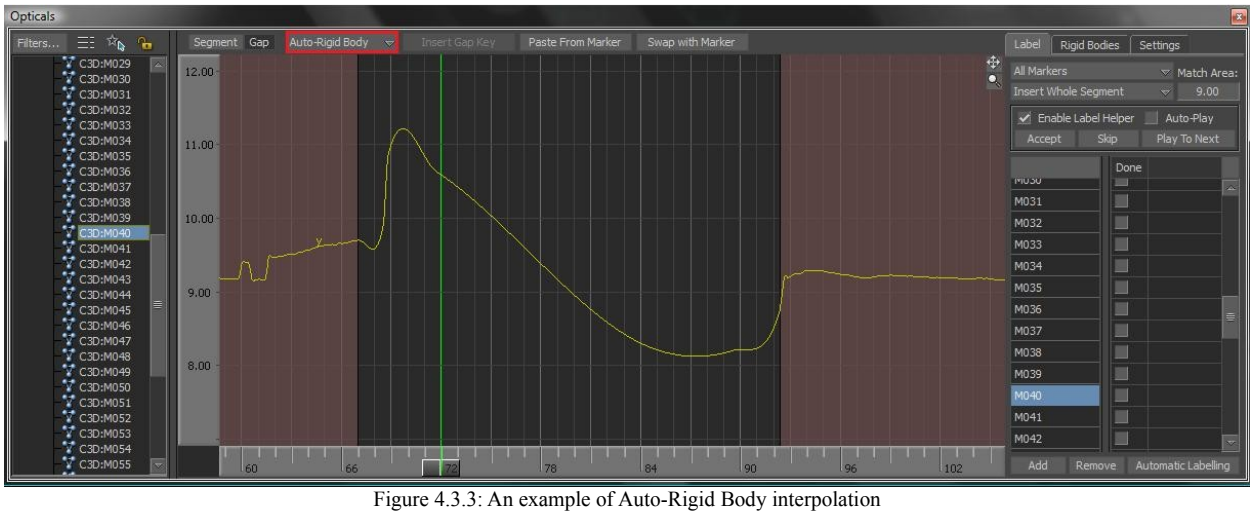

Figure 4.3.3: An example of Auto-Rigid Body interpolation

Auto-Constant will make it so that the wave will remain constant and will not move up or down for the duration of the gap. This is mostly helpful if a node starts out and/or ends dead. If the very beginning or the very end of the line is inside of a gap, it is usually a good idea to use the Auto-Constant method for the start and end gaps.

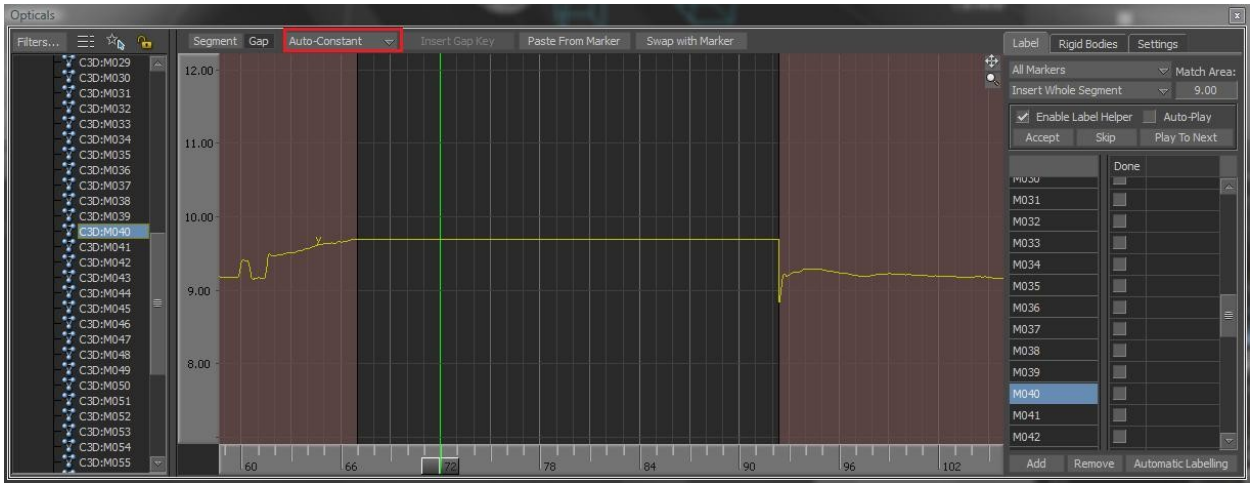

Figure 4.3.4: An example of Auto-Constant interpolation

Auto-Linear will take the start and end positions of the gap and make the wave into a straight line from start to end. This is mostly used to fix instances where an already near-straight section of the wave has plateaus or spikes.

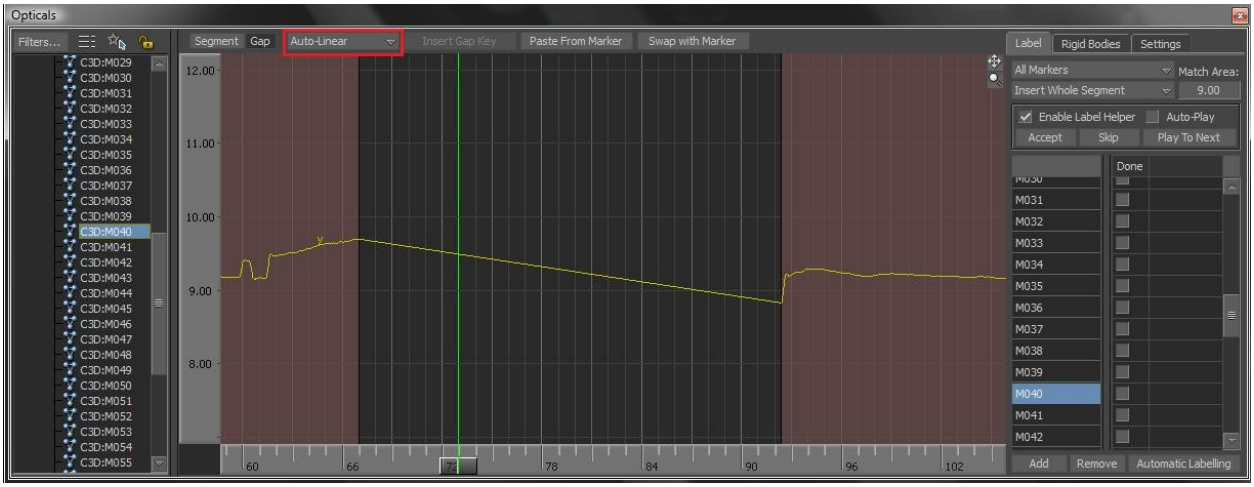

Figure 4.3.5: An example of Auto-Linear interpolation

Auto-Bezier is probably one of the most useful methods in terms of cutting out errors in the wave. The Auto-Bezier method takes the the sine of the wave as it enters the gap and as it leaves, and averages them to make a smooth wave. The Auto-Bezier method can be used to clean most errors that show up and is usually a better option to use than everything else because it makes a clean and smooth wave. However, overuse of the Auto-Bezier method may make the

animation look “floaty” and less crisp. This is especially true for faster animations, such as punches or jumps. If the Auto-Bezier line is radically different than everything else, check the start and end for spikes, split the segment, and then remove them.

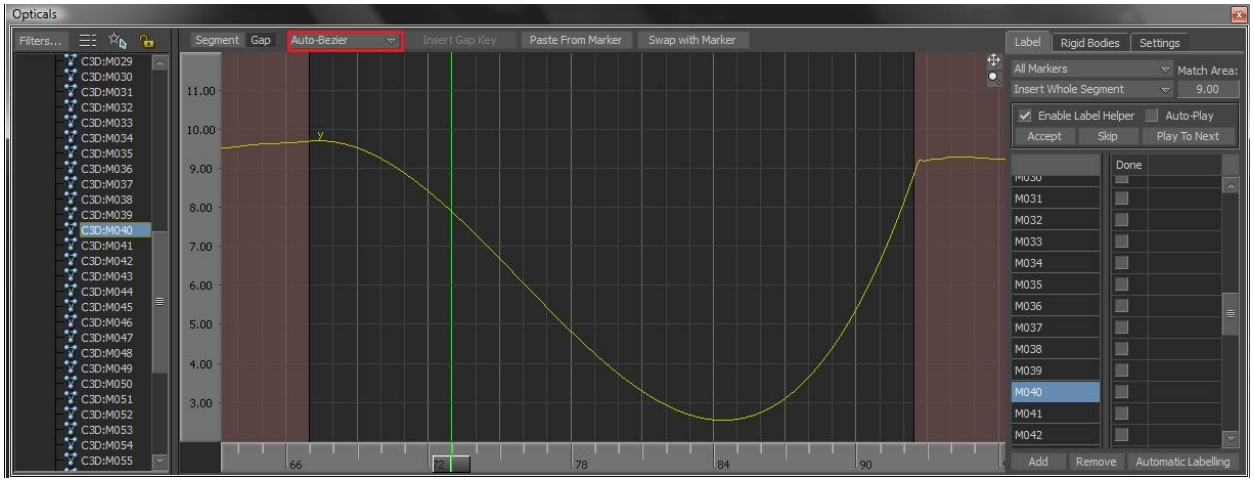

Figure 4.3.6: An example of Auto-Bezier interpolation

The Control Curve will take the last used wave and lets one manipulate the start and end positions of the wave by inserting a keyframe at each end. It also unlocks the “Insert Gap Key” button, which enables one to place a key in the wave that can then be manipulated. It works by making a wave out of the keyframes and then averaging the newly made one with the last used wave. This is particularly useful for manually editing a wave to any shape desired.

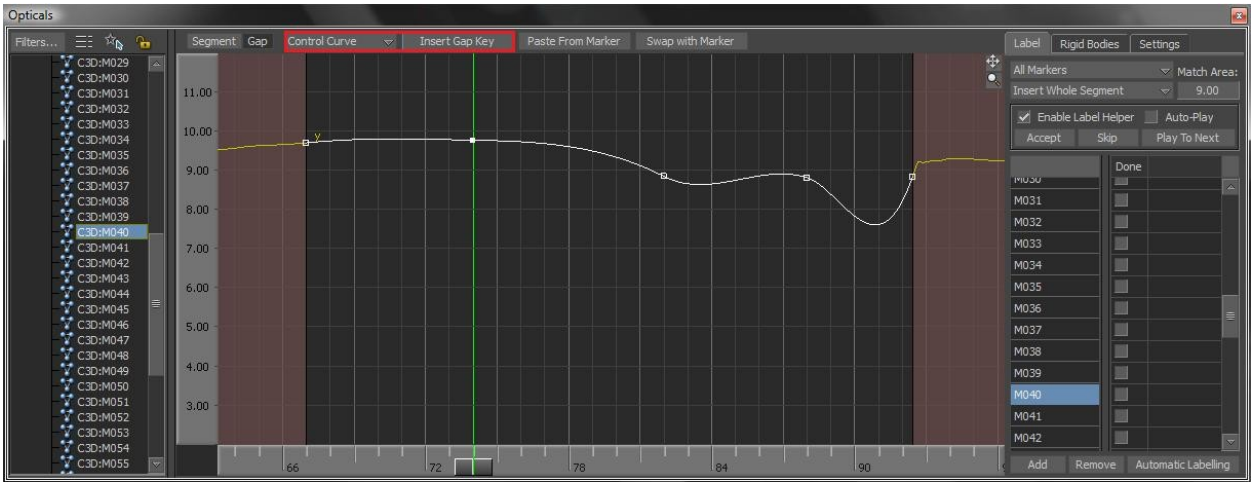

Figure 4.3.7: An example of a Control Curve

The final method is Sample Curve. This method enables one to manually manipulate all of the nodes in the gap, if one is zoomed in far enough. It is best used for fixing unnatural plateaus since one can select all of the plateau keys and manually move them down. An alternate use for the Sample Curve method is for rebuilding large gaps by first setting a gap to the Sample Curve and then setting it to Control Curve. By setting the gap to Sample Curve, it tells MotionBuilder to make a wave without any interpolation at all. When one switches it to Control Curve, it will Bezier the line. One can then add gap keys to create a new wave from scratch to use in the gap. This is useful if large portions of the graph (i.e. one or more wavelengths) are gaps and the waveforms in the gaps have a lot of errors.

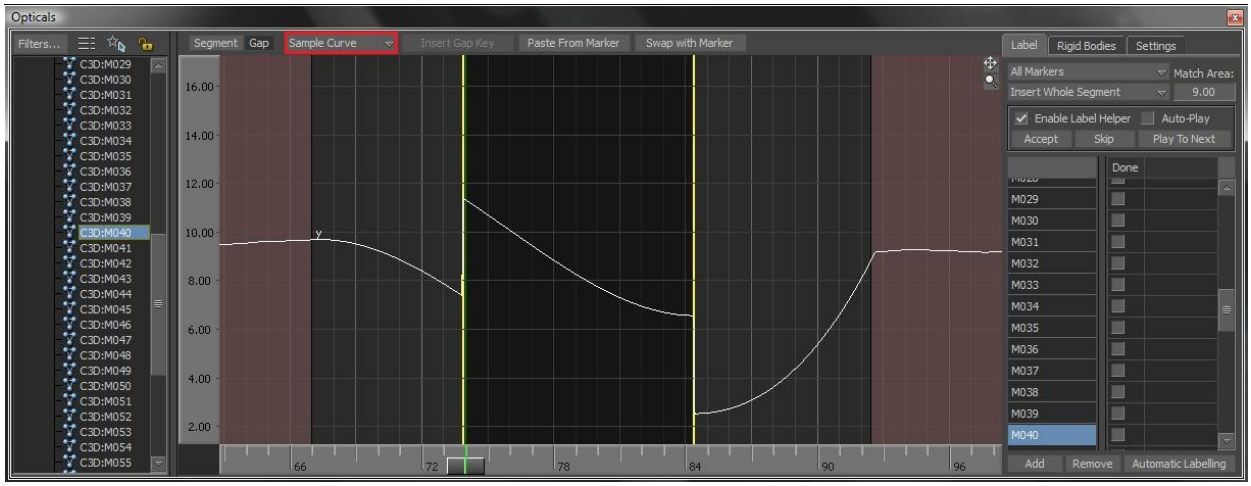

Figure 4.3.8: An example of a Sample Curve

Only the spikes that deviate from the wave by at least 4 units really need to be removed since the smaller problems can be removed by using filters.

Section V: Cleaning and Modifying the F-Curve Data

Even after cleaning the optical data, the animation will probably be far from perfect. At this point, the data just needs some general cleanup before one can apply it to a character and begin layered animation in order to tailor the data to the character.

Part 1: Converting the Optical Data into F-Curves

To the right of the Opticals graph is a window with three tabs called Label, Rigid Bodies, and Settings. First, open the Rigid Bodies tab, click Snap, then click All in the pop-up. Next, in the Settings tab, deselect Automatic Regenerate, which will now allow modification of the f-curves while active.

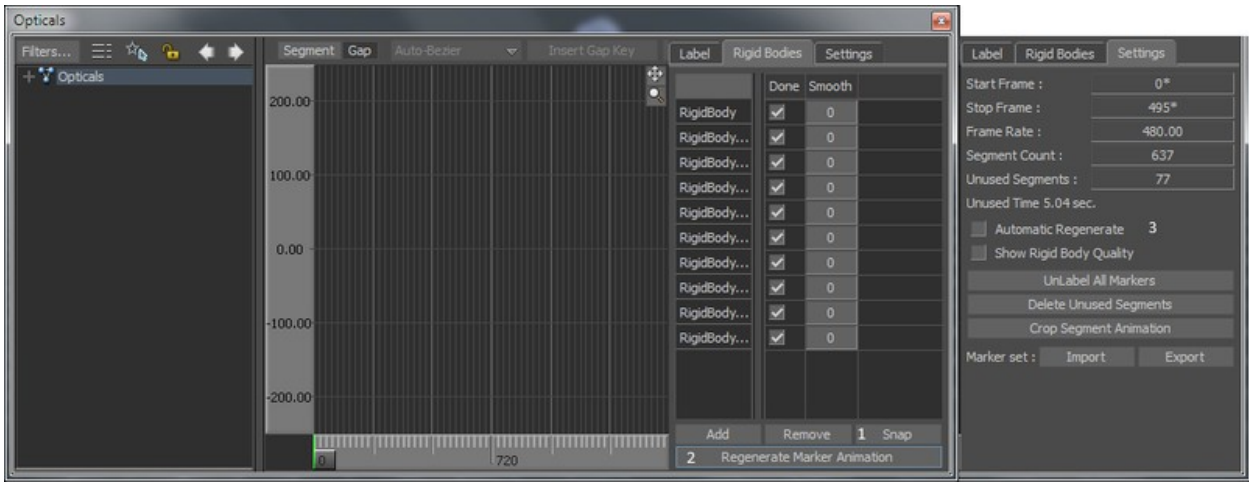

Figure 5.1.1: The three tabs

To finally convert the optical points into f-curves, in the Label tab, simply click on the "Done" box next to all of the opticals and rigid bodies that have motion point data on them. All the data will be gone from the Opticals graph, but that's because it's now f-curve data. Open the f-curve window by selecting Window → F-Curves.

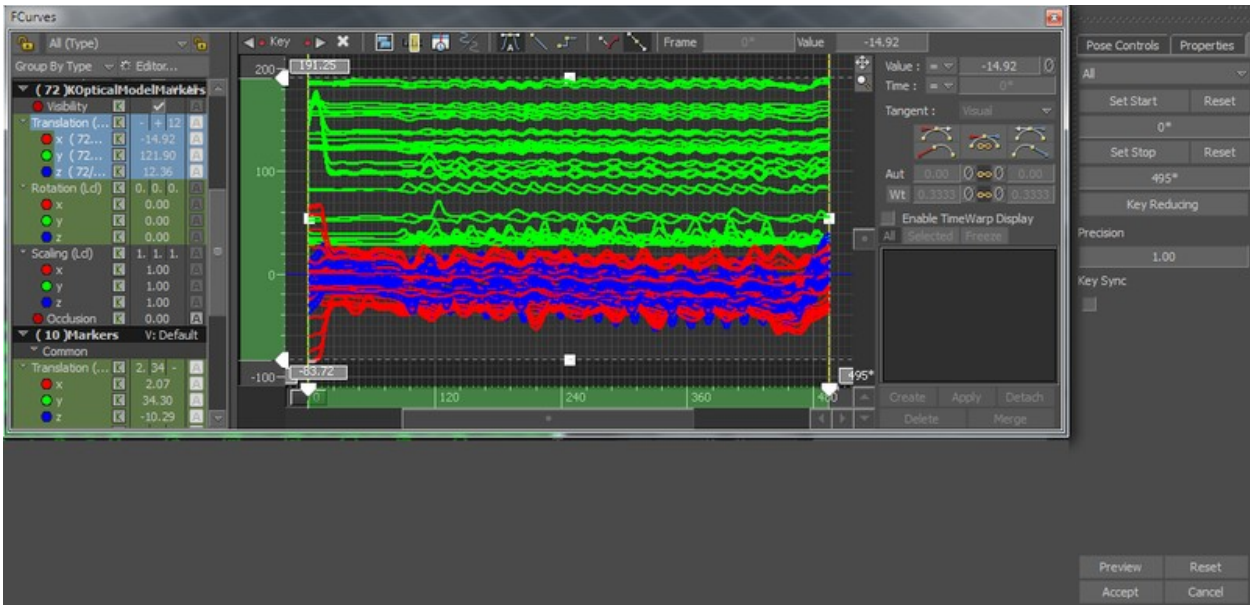

Figure 5.1.2: The F-Curves window

Part 2: Applying Preliminary Filters to the F-Curves

Once everything has been converted into f-curves, one should go over the animation with a few filters, such as the Peak Removal, Smooth, or Buttersworth filter. This is done by selecting all of the optical points, clicking on a filter, entering any parameters, clicking on the Preview button, and then clicking Accept if one is satisfied with the result.

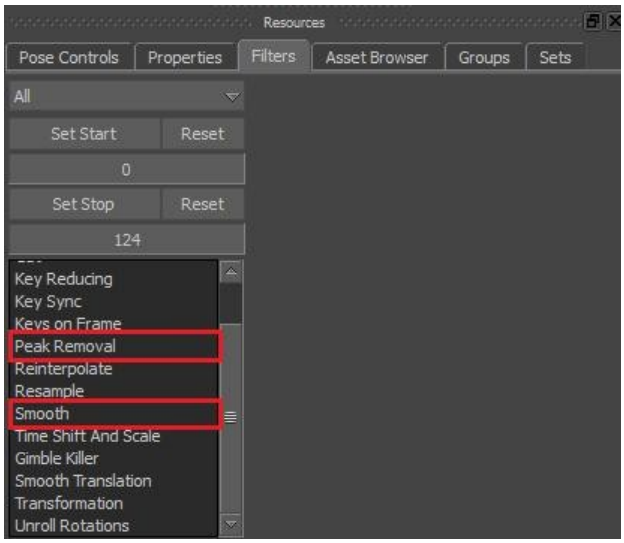

Figure 5.2.1: Some filters

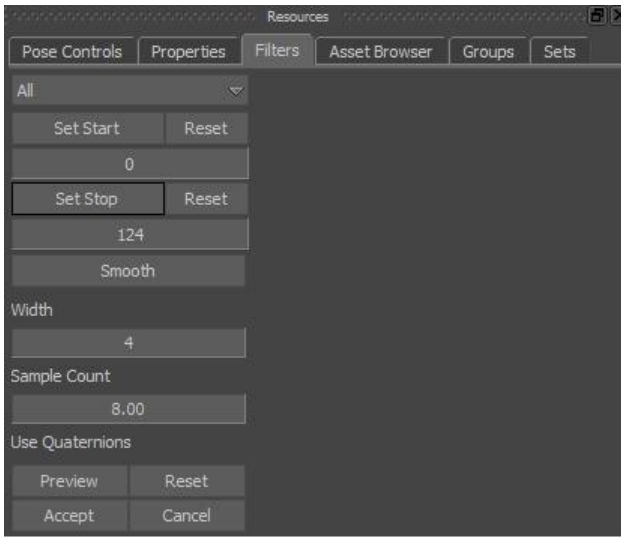

Figure 5.2.2: Some filter parameters

It is usually a good idea to give the animation a few passes with the Smooth filter, once with the width set to 4 and then again after that with the width set to 2. Make sure to press Accept after previewing the filter before giving it the second pass. WARNING: Selecting all of the f-curves is taxing on the computer, as there is a lot of data. Only do a few parts at a time if the computer doesn't have a very strong processor.

Section VI: Characterization and Finalizing the Animation

Characterization is the process of taking a model from Maya and prepping it for use in MotionBuilder. The process involves applying the joint nodes from the model's rig onto MotionBuilder's character template.

Part 1: Preparing the Model

Before characterizing a model in MotionBuilder, one needs to make a skeleton in Maya and rig the model. MotionBuilder requires one's rig to have, at the very least, the following joints:

- a Hip
- Left- and RightUpperLeg
- Left- and RightLeg (actually the knee)
- Left- and RightFoot (ankle)
- 1 Spine joint (i.e. one vertebrae), though preferably at least three
- Left- and RightArm (actually the shoulder joint)
- Left- and RightForearm (elbow)
- Left- and RightHand (wrist)
- Head

Recommended additional joints are:

- Left- and RightToeBase (where the toes attach to the rest of the foot)

- Left- and RightShoulder (similar to the clavicle)
- two Neck joints (one at the base of the neck, one where it connects to the head
- Left- and RightFingerBase

Part 2: Setup

Before starting on the characterization process, make sure that, in MotionBuilder, the Layout is set to the Editing Layout (press Ctrl+Shift+1 or select Layout > Editing) and Producer Perspective View (Ctrl+E or View > Perspective > Producer Perspective).

The first step in preparation is to export the rigged Maya file as a .fbx file. By default, Maya will not load the .fbx export plug-in so one will need to turn it on by going to the plug-in manager (Window → Settings/Preferences → Plug-in Manager) and clicking on the two checkboxes to the right of the plug-in called "fbxmaya.mll". Maya should come with the plug-in installed but, if not, Autodesk has a page on their website to download it. Once the plug-in has been activated, one can export the model by going to File → Export All and selecting .fbx under the export file type.

Once the model has been exported as an .fbx, the next step is to import it into MotionBuilder. To import the file, either drag-and-drop the file from a folder into the Viewer or select File → Open and select the file. Either way, a pop-up titled "Open Options" will appear. Click Open to continue.

Part 3: Characterization Procedure

1. Make a new character. There are at least two ways of doing so (below). Either way, it will be called "Character" by default. (To rename the character, click the blue box on the top-left of the tab, select Character, and select Rename.)
    - In the Resources box (bottom-right of the screen), expand the Templates folder and select Characters. In the contents view (on the right), select the Character and drag-and-drop it into the Viewer.
    - In the Character Tools box (top-right of the screen), select the "Characterization Tools" tab. Click the blue box (on the top-left of the tab), select Character, and select Create.
2. In the Navigator, expand the Characters asset and double-click on Character.
3. Click on the "Character Definition" tab, towards the top-left of the of the Navigator but to the right of the asset list.
4. Expand the Scene asset (in the asset list) and find the model asset. Drag it into the "Mapping List" slot next to the "Reference" section (in Character Definition). Most engines will require a reference object, which is the model that's going to be animated by the rig and that the player will see.
5. In the Character Definition tab, open the "Base (required)" node and drag and drop the corresponding joints from the rig onto each of the Base Character nodes. There can only be one joint per character node. Then, if one is using the recommended additional joints, open the Auxiliary, Spine, and Neck nodes, and fill them out as appropriate (Spine and Neck numeration is from bottom-up). Technically, one could make a rig complex enough as to use every single Character node, but this is by no means necessary.

6. Once one has applied as many joints as possible, click on the Characterize checkbox under the Character Definition tab that one should still be in. In the pop-up, select Biped if the Character has two feet and Quadruped if it has four.
7. Now the character needs a Control Rig. On the right end of the Character Definitions tab, there should be a Control Rig section. Click on Create and then click on FK/IK (which stands for Forward Kinematic and Inverse Kinematic).

The model should now be characterized. One can move the model about by playing with the Character Controls. It's at this point that the model is fully functional in MotionBuilder.

Part 4: Applying Mocap Data

One can now apply whatever motion capture data one has to the model by importing an Actor that has motion capture data mapped to it. NOTE: The scale of the model doesn't matter; the model's relatively tiny skeleton will take whatever motions the imported actor has. This is because MotionBuilder isn't working on the model itself but on the character template, which acts as a pointer telling the assigned parts what to do.

1. Merge a scene with an actor that has a motion file imported and mapped to it. The method to prepping an actor for merging is covered in the Cleaning of Opticals section. The actor can be merged by going to [File → Merge…] and then opening the the scene that one wants to merge.

2. Double click on the character object in the Navigator. This will open up the Character Controls and Characterization Tools window on the right.
3. Click on the blue square in the top-left of the Character Controls.
4. Scroll down to Source, then select Actor. The model should now be moving along with the actor.

Part 5 (optional): Returning an animated character to its initial form

1. Click on the blue square in the top-left of the Character Controls.
2. Scroll down to Source, then select Stance or Control Rig. This will return the characterized model to its original t-pose stance.

Part 6: Using Layers to Modify the Animation

Go to Window → Key Controls. In the second drop-down menu on the left, there should be something that reads “BaseAnimation” or “AnimLayer1”. Select (New Layer) or choose an empty one. This field is to select the layer on which one is currently animating. One can enable, disable, merge, delete, or create new layers in the Animation Layers menu.

Select all of the FK/IK Controls and click “Key” (or press the S key if using Maya controls) on the starting frame. Then, click the button with the picture of a key on it to enable auto-key framing.

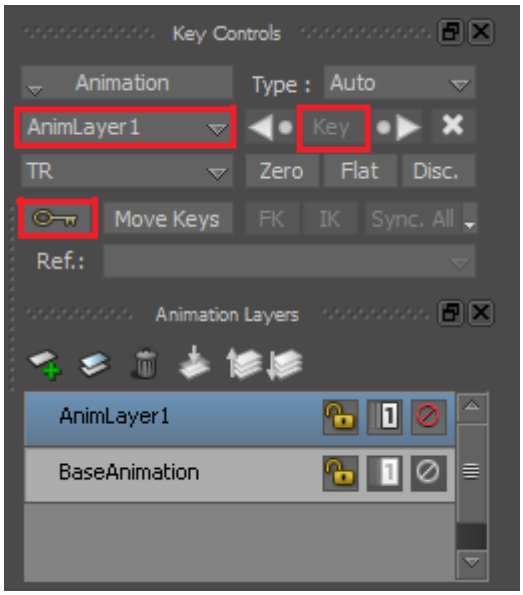

Fig. 6.6.1: Key controls

If one gets stuck, merge all of the layers, create a new layer on top, and begin cleaning it all up again. It may take some time, but this will garner the best results. When merging layers, this IQP group recommends just using the default settings; Smart Plot doesn't appear to work well yet.

One may notice that any key frames made on the top layer carry through the entire animation until it meets another key frame. If one wants to quickly revert a frame back to the way it was on the base layer, press the Zero button on the key controls. On the other hand, one can still run filters on the f-curve data that the character controller uses, but this usually isn't necessary.

Part 7: Animation Suggestions

- When adjusting and capturing foot positions, this group found it to be helpful to use a triangular setup for the feet. One can then easily manipulate the foot by selecting and moving any amount of the nodes as necessary to get the foot into position without much

hassle. It is very intuitive and really helps get a feel for what kind of movements one needs to make to fix inaccurate data.

- Animating the character's hands can more easily be done on the model itself.
- If there are some obvious single frame errors, it is easier to go into the curves themselves and fix them than attempting to create layers to tone it down. Alternatively, those single frame errors are generally regarded by MotionBuilder as a peak, so one can just as easily run a Peak Removal or Smooth Transformations filter on the curve to fix it.

Part 8: Finalizing the Animation

After creating a character set and plotting it to a character, one is now in the final stages of the general motion capture pipeline. If there are any remaining errors in the animations when being played back and attached to the model, one can clear those up manually using the layer animation techniques and moving the control rig nodes.

After one is perfectly satisfied with the animation, one can save it in the native .fbx format, such as for directly importing it into Unity as an in-game animation, or export it to 3DS Max or Maya from the file menu.

RESULTS AND CONCLUSIONS

Motion capture is a great way for getting smoother and more realistic animations. Traditional keyframe animations are extremely time-intensive and, as such, people may be drawn to motion capture as it is a faster and more realistic alternative. However, like all methods, it does come with its own limitations. What follows are the results of this IQP and suggestions on how future students can help to improve our motion capture setup.

This IQP group discovered that motion capture is excellent for making realistic animations that mimic even the subtlest of motions in the body, which a traditional animator might overlook or even exaggerate in compensation. It can accurately capture shifts in weight and, by adding more optical nodes to the suit, can achieve even higher fidelity animations. Once one has learned how to operate the motion capture equipment, it is very easy to take a lot of captures in a single session. The most this group managed to take in a single day was approximately 50 takes of 22 completely different animations. If one were to use traditional 3D animation techniques, accomplishing the same task would take one person an entire term, perhaps even two terms, at the very least. This means that future IMGD project groups that want to have a lot of animations can generate and apply them in a couple of weeks instead of months. In fact, on average, only an hour or two at most is required to effectively clean motion capture data once the operator has achieved proficiency with MotionBuilder.

With the addition of the new IMGD repository for models and animations, students can upload finalized motion capture files, which can then be used by the entire student body. Being able to reduce in this manner the amount of original content required for basic operations permits

groups to reallocate time and resources to other aspects of game development, such as 3D modeling, which is extremely time-intensive, adding features, or even better debugging.

While motion capture is a faster alternative to traditional keyframe animation, there are several limitations that come with using it. First, quick movements, such as punching, jumping, and running, have extremely messy optical data. This is in part due to the currently limited work space mentioned above, but it's also a result of the motion capture suits having less than ideal marker coverage. In particular, the hips and back are sorely missing what would otherwise be extremely useful data points. Messier captures mean that the operator has to spend additional time manually cleaning the motion capture data by manipulating translation waves for every problematic marker. Since most video game animations generally consist of fast actions, this is a significant hindrance. The repository can help with this in that, once one person has properly cleaned and uploaded an animation, it would not be necessary to recapture and clean that particular animation, but no database, no matter how extensive, can possibly cover the extent of human creativity.

The second limitation with the current motion capture setup is the fact that the work area is located in an extremely confined space, which severely limits the type of captures one can take. Professional animation studios have their own gymnasiums that are fitted with hundreds of cameras, whereas WPI imposes a 4x4x4 meter space (the minimum recommended space is 6x6x3) and can only employ eight of the total sixteen cameras. A sufficiently larger room, with an expanded cage, could even mitigate the need for a treadmill, which forces unnatural movement and can block markers, and could possibly allow for captures of multiple people at once.

Despite these limitations, motion capture is still a much better alternative to traditional methods for realistic animations. Motion capture can even be used in combination with keyframing if so desired. This IQP group recommends experimenting with optical marker placement, rigid body arrays, marker set configurations, and MotionBuilder itself as further research subjects.

EXTERNAL REFERENCES

WPI motion capture wiki. (2011). Retrieved from http://wpi-mocap.wikispaces.com/

IMGD repository. (2012). Retrieved from https://sharepoint.wpi.edu/projects/imgdmodels

Guindon, M. (Performer) (2009). *Siggraph 2009 masterclasses* [Web]. Retrieved from http://area.autodesk.com/masterclasses/class09_mbuilder

Kitagawa, M., & Windsor, B. (2008). *Mocap for artists: Workflow and techniques for motion capture*. USA: Elsevier Inc.